Dr. Joseph Teguh Santoso, S.Kom, M.Kom.

TEKNOLOGI AI
(Artificial Intelligence)

YAYASAN PRIMA AGUS TEKNIK

TEKNOLOGI AI
(Artificial Intelligence)

Dr. Joseph Teguh Santoso, S.Kom, M.Kom.

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7227-55-3 (PDF)
9 786347 227553

TEKNOLOGI AI (Artificial Intelligence)

Penulis :
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

ISBN : 978-634-7227-55-3 (PDF)

Editor :
Dr. Agus Wibowo, M.Kom, M.Si, MM.

Penyunting :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds., M.Kom

Penebit :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa karena dengan rahmat dan karunia-Nya, buku berjudul "*Teknologi AI (Artificial Intellegence)*" ini dapat diselesaikan. Buku ini hadir sebagai salah satu upaya untuk memberikan wawasan mendalam mengenai kecerdasan buatan (Artificial Intelligence) dan berbagai aspek teknologi terkait yang saat ini sangat berkembang dan memberikan dampak luas pada berbagai bidang kehidupan.

Dalam buku ini, pembaca akan diajak menelusuri konsep dasar kecerdasan buatan, teknologi pendukung, serta aplikasi-aplikasi yang sudah dan sedang dikembangkan, termasuk strategi pengembangan AI dari perusahaan teknologi terkemuka seperti Huawei. Selain itu, berbagai topik penting seperti pembelajaran mesin, pembelajaran mendalam, serta framework Frameworks pembelajaran mendalam dan pengembangan AI juga dijelaskan secara rinci.

Bab 1 menghadirkan pengantar umum mengenai kecerdasan buatan, mulai dari konsep dasar, teknologi terkait, aplikasi yang berkembang, hingga strategi pengembangan AI oleh Huawei serta pembahasan kontroversi dan tren terkini. Dengan bab ini, pembaca diberi fondasi yang kuat untuk memahami perkembangan AI secara luas. Bab 2 membahas pembelajaran mesin sebagai salah satu metode penting dalam kecerdasan buatan. Penjelasan jenis-jenis pembelajaran mesin, proses keseluruhan, hingga parameter model dan algoritma umum disajikan secara sistematis, dilengkapi dengan studi kasus yang memudahkan pemahaman praktis.

Pada Bab 3, fokus beralih ke pembelajaran mendalam (deep learning) dengan pembahasan menyeluruh mulai dari aturan pelatihan, fungsi aktivasi, regularisasi, hingga jenis jaringan saraf tiruan dan tantangan yang sering dihadapi. Bab ini memberikan wawasan teknis esensial untuk pengembangan model AI canggih. Bab 4 dan Bab 5 menyajikan kerangka kerja penting dalam pengembangan pembelajaran mendalam dan AI, terutama menggunakan TensorFlow 2.0 dan Huawei MindSpore. Kedua bab ini menekankan aspek praktis dalam pemrograman dan pengembangan solusi AI modern.

Bab 6 menguraikan solusi komputasi AI Huawei Atlas yang mencakup arsitektur perangkat keras dan lunak prosesor AI, serta implementasinya dalam berbagai industri. Penjelasan ini menunjukkan bagaimana AI diaplikasikan secara nyata melalui infrastruktur komputasi canggih. Bab 7 dan Bab 8 membahas platform terbuka Huawei AI dan layanan Huawei Cloud Enterprise Intelligence yang mendukung pengembangan aplikasi dan solusi berbasis AI dengan skala dan fleksibilitas tinggi. Topik ini penting bagi pembaca yang ingin mengeksplorasi ekosistem AI secara menyeluruh.

Lebih lanjut, buku ini membahas solusi komputasi AI Huawei Atlas, platform terbuka Huawei AI, hingga layanan Huawei Cloud Enterprise Intelligence yang menjadi fondasi utama dalam implementasi teknologi AI untuk industri dan bisnis. Studi kasus dan pembahasan teknis yang disajikan diharapkan dapat mendukung pemahaman sekaligus menjadi referensi yang bermanfaat bagi praktisi, peneliti, mahasiswa, dan semua pihak yang tertarik pada bidang kecerdasan buatan.

Kami menyadari bahwa pembahasan yang luas dan mendalam ini tidak akan terwujud tanpa dukungan berbagai pihak. Oleh karena itu, kami mengucapkan terima kasih kepada semua kontributor, pakar, dan institusi yang telah berperan aktif dalam penyusunan buku ini. Semoga buku ini dapat memberikan manfaat serta mendorong pengembangan teknologi AI yang semakin canggih dan bermanfaat bagi bangsa dan dunia.

Akhir kata, kami berharap buku ini dapat menjadi sumber inspirasi dan pengetahuan yang komprehensif dalam perjalanan memahami dan mengembangkan teknologi kecerdasan buatan.

Semarang, Oktober 2025
Penulis

Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

DAFTAR ISI

BAB 1
PENGANTAR UMUM KECERDASAN BUATAN (ARTIFICIAL INTELLEGENCE)

Kemunculan dan kebangkitan kecerdasan buatan tidak diragukan lagi memainkan peran penting dalam perkembangan internet. Selama dekade terakhir, dengan aplikasi yang luas di masyarakat, kecerdasan buatan menjadi semakin relevan dengan kehidupan sehari-hari. Bab ini memperkenalkan konsep kecerdasan buatan, teknologi terkait, dan kontroversi yang ada seputar topik ini.

1.1 KONSEP KECERDASAN BUATAN

Apa Itu Kecerdasan Buatan?

Ketika mendengar istilah *kecerdasan buatan* atau *artificial intelligence* (AI), sebagian besar orang mungkin langsung teringat pada gambaran yang sering muncul di berita, film, atau berbagai aplikasi teknologi yang kini meresap dalam kehidupan sehari-hari. Mulai dari asisten virtual di ponsel, rekomendasi film di platform digital, hingga mobil tanpa pengemudi semuanya adalah contoh konkret bagaimana AI bekerja dalam keseharian manusia. Akan tetapi, untuk benar-benar memahami apa itu kecerdasan buatan, kita perlu melampaui bayangan populer yang terbentuk dari hiburan dan media, lalu menelusuri akar konseptual serta landasan ilmiah dari bidang ini.

Salah satu definisi awal yang cukup berpengaruh mengenai kecerdasan buatan diajukan oleh John McCarthy, seorang ilmuwan komputer yang kerap disebut sebagai "bapak AI." Definisi tersebut dikemukakan pada Konferensi Dartmouth tahun 1956, sebuah forum penting yang menandai kelahiran resmi disiplin AI sebagai bidang penelitian. McCarthy mendeskripsikan kecerdasan buatan sebagai upaya untuk membuat mesin dapat meniru perilaku cerdas manusia sedekat mungkin. Dengan kata lain, AI pada awalnya dipahami sebagai simulasi kecerdasan manusia oleh mesin. Definisi ini memberi pijakan awal yang sangat berarti, karena berhasil meletakkan dasar untuk pengembangan teknologi AI di dekade-dekade berikutnya.

Namun, definisi McCarthy juga memiliki keterbatasan. Ia menitikberatkan pada *peniruan* perilaku manusia, sehingga seolah-olah kecerdasan buatan hanyalah refleksi atau tiruan dari kemampuan manusia. Pandangan semacam ini tidak sepenuhnya mengakomodasi kemungkinan adanya apa yang disebut sebagai kecerdasan buatan kuat (*strong AI*). Istilah ini merujuk pada jenis kecerdasan buatan yang tidak hanya meniru perilaku cerdas, tetapi juga memiliki kapasitas untuk benar-benar *memahami* serta *menalar*, bahkan menyelesaikan masalah secara mandiri tanpa selalu mengandalkan instruksi manusia. Gagasan ini membawa kita pada diskusi filosofis yang lebih mendalam: apakah mesin suatu hari benar-benar dapat berpikir layaknya manusia, ataukah ia hanya mampu meniru permukaan perilaku kita tanpa kesadaran sejati?

Sebelum lebih jauh membahas konsep AI, ada baiknya kita terlebih dahulu memahami makna dari kata “kecerdasan” itu sendiri. Secara umum, kecerdasan dapat dipahami sebagai kemampuan untuk memperoleh pengetahuan, memahami informasi, serta menggunakannya dalam menyelesaikan masalah. Akan tetapi, dalam kajian psikologi modern, kecerdasan bukanlah kemampuan tunggal yang seragam. Salah satu teori yang cukup berpengaruh adalah teori kecerdasan majemuk (*multiple intelligences theory*) yang dikemukakan oleh Howard Gardner. Menurut teori ini, manusia memiliki setidaknya tujuh jenis kecerdasan yang berbeda, masing-masing mencerminkan aspek unik dari potensi berpikir dan bertindak.

Pertama, ada kecerdasan linguistik, yakni kemampuan menggunakan bahasa secara efektif, baik dalam berbicara maupun menulis. Orang yang memiliki kecerdasan ini biasanya piawai dalam mengolah kata, seperti penulis, penyair, atau orator. Kedua, kecerdasan logika-matematika, yang berkaitan dengan kemampuan berpikir rasional, mengenali pola, serta menyelesaikan persoalan kuantitatif. Inilah kecerdasan yang sering diasosiasikan dengan ilmuwan atau matematikawan. Ketiga, kecerdasan spasial, yakni kemampuan membayangkan dan memanipulasi ruang secara mental. Seorang arsitek atau pelukis, misalnya, sangat bergantung pada jenis kecerdasan ini.

Selain itu, Gardner juga menyebutkan kecerdasan kinestetik-jasmani, yakni keterampilan menggunakan tubuh untuk mengekspresikan ide atau menyelesaikan tugas. Atlet, penari, atau aktor sering kali menonjol dalam jenis kecerdasan ini. Lalu ada kecerdasan musikal, yang berkaitan dengan kepekaan terhadap nada, irama, dan melodi. Orang dengan kecerdasan ini biasanya memiliki bakat alami dalam mencipta atau menafsirkan musik. Selanjutnya, terdapat kecerdasan interpersonal, yaitu kemampuan memahami orang lain, membaca emosi mereka, serta berinteraksi secara efektif. Kecerdasan ini sangat penting bagi pemimpin, konselor, atau guru. Terakhir, kecerdasan intrapersonal, yakni kesadaran mendalam akan diri sendiri, termasuk kekuatan, kelemahan, dan motivasi pribadi. Individu dengan kecerdasan ini sering mampu melakukan refleksi diri yang kuat dan memiliki arah hidup yang jelas.

Jika kita menghubungkan teori kecerdasan majemuk dengan perkembangan AI, muncul pertanyaan menarik: sejauh mana mesin dapat meniru atau bahkan menggantikan jenis-jenis kecerdasan manusia tersebut? Dalam beberapa bidang, seperti logika-matematika, komputer dan algoritma sudah menunjukkan keunggulan luar biasa, bahkan melampaui kemampuan manusia. Sistem AI dapat menghitung jutaan kemungkinan dalam hitungan detik, sesuatu yang mustahil dilakukan oleh otak manusia. Pada bidang linguistik, kita juga menyaksikan kemajuan signifikan: mesin penerjemah otomatis, chatbot, dan sistem pengenalan suara sudah menjadi bagian dari keseharian kita. Namun, pada aspek-aspek lain, seperti kecerdasan intrapersonal atau musikal dalam tingkat kedalaman emosional, AI masih menghadapi tantangan besar.

Pemahaman ini penting agar kita tidak terjebak pada pandangan yang terlalu sederhana tentang kecerdasan buatan. AI bukanlah satu paket kemampuan yang serba bisa. Ia berkembang secara bertahap, dengan keunggulan di bidang tertentu, namun juga dengan keterbatasan pada bidang lainnya. Justru melalui pemetaan jenis-jenis kecerdasan manusia, kita dapat melihat dengan lebih jelas arah perkembangan AI: apakah ia hanya akan menjadi

alat bantu yang sangat cerdas, atau suatu saat benar-benar berkembang menjadi entitas dengan kecerdasan mandiri yang sebanding dengan manusia.

Dengan demikian, memahami apa itu kecerdasan buatan tidak bisa dilepaskan dari pemahaman tentang kecerdasan manusia itu sendiri. AI adalah refleksi, sekaligus tantangan, bagi kita dalam mendefinisikan kembali batas-batas kecerdasan. Ia lahir dari upaya meniru kemampuan manusia, tetapi juga berpotensi melampaui kita dalam beberapa aspek. Maka, mengenali landasan konseptual AI, baik dari sisi sejarah maupun teori psikologi, merupakan langkah awal yang penting untuk memahami potensi dan risiko teknologi yang kini semakin membentuk dunia kita.

A. Kecerdasan Linguistik

Kecerdasan linguistik merupakan jenis kecerdasan yang berkaitan dengan kemampuan seseorang dalam menggunakan bahasa, baik secara lisan maupun tulisan, untuk mengekspresikan ide, gagasan, dan perasaan secara efektif. Individu dengan kecerdasan ini umumnya mampu memahami dan menafsirkan makna kata, kalimat, maupun teks dengan baik, serta menguasai berbagai aspek bahasa seperti fonologi (ilmu bunyi), semantik (makna), dan tata bahasa.

Kekuatan utama mereka terletak pada keluwesan dalam mengolah bahasa, baik untuk mengelola pikiran verbal maupun menyampaikan pesan dengan nuansa tertentu sesuai konteks. Karena itu, orang yang memiliki kecerdasan linguistik tinggi sering kali menonjol dalam bidang-bidang yang menuntut keterampilan komunikasi. Profesi seperti politisi, aktivis, pengacara, pembawa acara, penulis, jurnalis, editor, guru, hingga pembicara publik merupakan jalur karier yang sangat cocok bagi mereka, karena semuanya menuntut kemampuan menyampaikan ide dengan jelas, persuasif, dan penuh makna.

B. Kecerdasan Logika-Matematika

Kecerdasan logika-matematika berkaitan dengan kemampuan seseorang dalam berpikir secara rasional, sistematis, dan analitis. Individu dengan kecerdasan ini biasanya terampil dalam melakukan perhitungan, mengukur, menalar, serta mengklasifikasikan informasi secara efektif. Mereka juga mampu menyelesaikan operasi matematika yang kompleks dan memiliki kepekaan terhadap konsep-konsep abstrak, seperti pola, hubungan logis, pernyataan, serta fungsi. Kemampuan ini membuat mereka unggul dalam menghubungkan data yang tampak tidak berhubungan menjadi sebuah kesimpulan yang logis. Tidak mengherankan, orang yang memiliki kecerdasan logika-matematis tinggi umumnya lebih cocok berkarier di bidang-bidang yang menuntut ketelitian dan penalaran abstrak, seperti ilmuwan, akuntan, ahli statistik, insinyur, hingga pengembang perangkat lunak komputer.

Berita (News)	Film (Movies)	Aplikasi Sehari-hari (Daily Application)
Taman Haidian, Taman AI Pertama di Dunia! Program AI mengalahkan pemain manusia terbaik di *StarCraft II*, AlphaStar meraih	*The Terminator 2001: A Space Odyssey The Matrix I, Robot Blade*	Pemeriksaan keamanan swalayan Penilaian keterampilan berbicara Rekomendasi film dan

ketenaran! Potret oleh Program AI *Portrait of Edmond Belamy* terjual seharga Rp 7,009,000,000 Permintaan terhadap programmer AI melonjak 35 kali lipat! Gaji menduduki peringkat pertama! 50% pekerjaan akan digantikan oleh AI di masa depan "Musim Dingin Akan Datang?" Tantangan besar yang dihadapi AI ……	*Runner Her Bicentennial Man* ……	musik Pengeras suara pintar Robot penyedot debu Terminal swalayan bank Layanan pintar Siri ……
Aplikasi AI Tren industri dan prospek AI Tantangan AI ……	**AI mengendalikan manusia Jatuh cinta dengan AI Kesadaran diri AI** ……	**Keamanan & perlindungan Hiburan Rumah pintar Keuangan** ……

Gambar 1.1 Kognisi sosial AI

C. Kecerdasan Spasial

Kecerdasan spasial merujuk pada kemampuan seseorang untuk memahami, mengenali, dan memanipulasi ruang visual serta objek-objek di sekitarnya dengan akurat. Individu dengan kecerdasan ini biasanya mampu membayangkan bentuk atau posisi suatu benda dalam ruang, serta menuangkannya kembali dalam bentuk representasi visual, seperti gambar, sketsa, atau grafik. Mereka memiliki kepekaan yang tinggi terhadap unsur-unsur visual, termasuk warna, garis, bentuk, maupun komposisi secara keseluruhan. Kecerdasan ini sangat penting dalam bidang yang menuntut kreativitas visual dan presisi ruang, sehingga profesi yang sering dikaitkan dengannya antara lain arsitek, desainer interior, fotografer, pelukis, hingga pilot yang membutuhkan keterampilan orientasi ruang yang baik.

D. Kecerdasan Kinestetik-Jasmani

Kecerdasan kinestetik-jasmani menggambarkan kemampuan seseorang dalam memanfaatkan tubuhnya secara optimal untuk mengekspresikan gagasan maupun emosi, sekaligus menguasai keterampilan motorik dalam menciptakan atau memanipulasi objek. Kecerdasan ini tercermin melalui penguasaan gerakan tubuh yang melibatkan aspek keseimbangan, koordinasi, kelincahan, kekuatan, kelenturan, kecepatan, serta kepekaan taktil atau kemampuan merasakan melalui sentuhan. Individu dengan kecerdasan ini biasanya unggul dalam bidang yang menuntut keterampilan fisik, baik dalam seni maupun aktivitas teknis. Oleh karena itu, jalur karier yang sesuai bagi mereka antara lain atlet, penari, aktor, ahli bedah, mekanik, hingga pengrajin perhiasan yang membutuhkan ketelitian dan keterampilan tangan tingkat tinggi.

E. Kecerdasan Musikal

Kecerdasan musikal merujuk pada kemampuan seseorang untuk mengenali, membedakan, dan mengolah unsur-unsur bunyi seperti nada, melodi, ritme, serta warna suara. Individu dengan kecerdasan ini biasanya memiliki kepekaan tinggi terhadap struktur musik dan mampu meresponsnya dengan baik, baik dalam bentuk penampilan, penciptaan,

maupun refleksi musikal. Keunggulan ini membuat mereka lebih mudah mengembangkan kreativitas di bidang musik, sekaligus lebih kompetitif ketika berkecimpung dalam dunia seni suara. Profesi yang sangat sesuai bagi mereka antara lain penyanyi, komposer, konduktor, kritikus musik, penyetem piano, atau pekerjaan lain yang menuntut sensitivitas serta keterampilan dalam mengolah musik.

F. Kecerdasan Interpersonal

Kecerdasan interpersonal mencerminkan kemampuan seseorang dalam memahami, merasakan, dan membangun hubungan yang efektif dengan orang lain. Individu dengan kecerdasan ini biasanya peka terhadap suasana hati, temperamen, serta emosi orang di sekitarnya. Mereka mampu membaca isyarat nonverbal, memahami pesan tersirat dalam interaksi sosial, dan menanggapi situasi interpersonal dengan cara yang tepat. Kemampuan untuk berempati dan menyesuaikan diri inilah yang menjadikan mereka unggul dalam menjaga komunikasi dan menjalin hubungan yang harmonis. Oleh karena itu, profesi yang sesuai bagi mereka antara lain politisi, diplomat, pemimpin organisasi, psikolog, tenaga hubungan masyarakat (humas), hingga bidang penjualan yang menuntut keterampilan komunikasi dan pengelolaan relasi yang baik.

G. Kecerdasan Intrapersonal

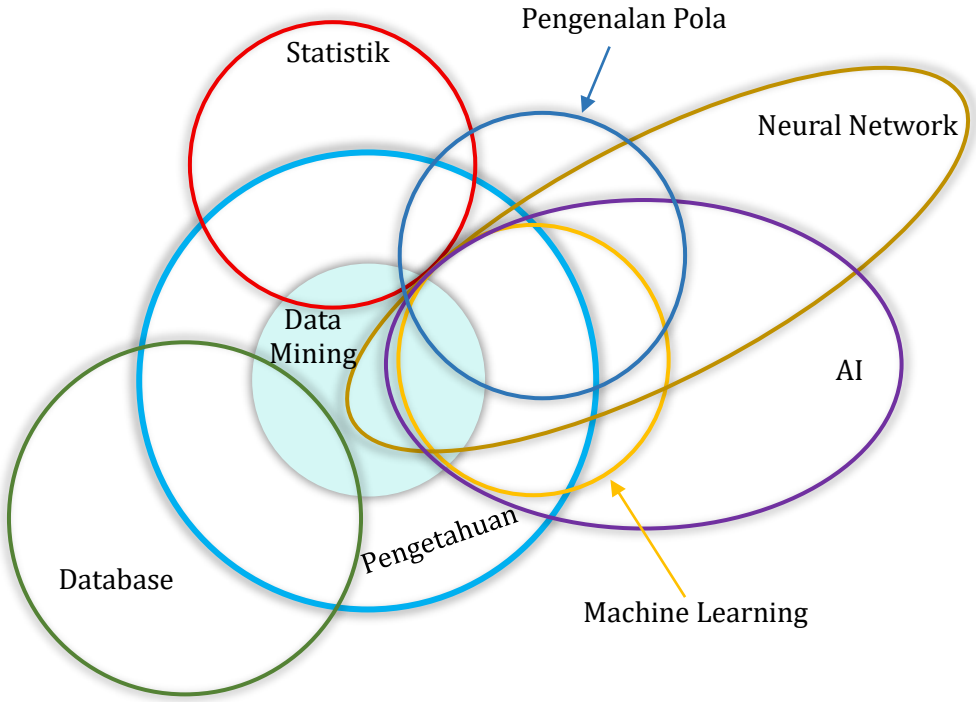
Kecerdasan intrapersonal berhubungan erat dengan kemampuan seseorang untuk mengenali, memahami, dan mengelola dirinya sendiri. Individu dengan kecerdasan ini biasanya memiliki kesadaran diri yang tinggi, mampu membedakan kekuatan dan kelemahan pribadi, serta mengenali suasana hati, motivasi, temperamen, hingga harga diri mereka. Mereka cenderung gemar melakukan refleksi diri, berpikir secara mandiri, dan menggunakan pemahaman tersebut untuk mengarahkan keputusan serta tindakan sehari-hari. Dengan keterampilan ini, mereka dapat menjaga konsistensi antara apa yang mereka pikirkan, rasakan, dan lakukan. Karier yang sesuai bagi orang dengan kecerdasan intrapersonal yang menonjol mencakup filsuf, politisi, pemikir, psikolog, serta profesi lain yang menuntut kedalaman refleksi dan pemahaman diri.

H. Kecerdasan Naturalis

Kecerdasan naturalis adalah kemampuan seseorang untuk peka terhadap dunia alam, termasuk mengamati, mengenali, serta mengklasifikasikan berbagai bentuk kehidupan dan fenomena alam. Individu dengan kecerdasan ini biasanya mampu membedakan pola, mengelompokkan objek, serta memahami hubungan antara sistem alami dan sistem buatan. Contoh profesi yang cocok bagi mereka antara lain ahli biologi, ahli botani, petani, peneliti lingkungan, hingga konservasionis.

Namun, berbeda dengan kecerdasan alami manusia, kecerdasan buatan (AI) hadir sebagai cabang ilmu pengetahuan dan teknologi modern yang dirancang untuk meniru, memperluas, bahkan memodernisasi cara berpikir manusia. AI berfokus pada pengembangan teori, metode, serta aplikasi yang memungkinkan mesin meniru proses penalaran dan pengambilan keputusan layaknya manusia. Seiring perkembangannya, definisi AI tidak lagi hanya sebatas simulasi perilaku cerdas, melainkan telah melebar menjadi bidang

interdisipliner yang memadukan ilmu komputer, matematika, kognisi, linguistik, hingga ilmu saraf, sebagaimana digambarkan dalam Gambar 1.2.



Gambar 1.2 Bidang yang dicakup oleh kecerdasan buatan

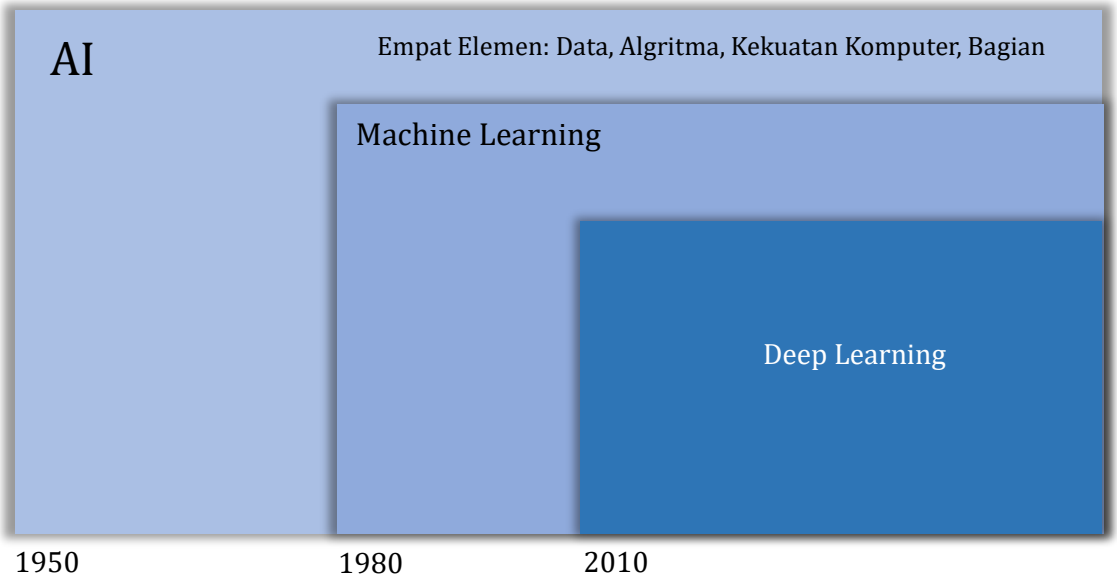
Pembelajaran mesin (*Machine Learning*/ML) saat ini dipandang sebagai salah satu fokus utama dalam pengembangan kecerdasan buatan (AI) yang bersifat interdisipliner. Tom Mitchell, yang dikenal luas sebagai *bapak ML global*, memberikan salah satu definisi klasik mengenai konsep ini. Ia menyatakan bahwa jika sebuah program komputer mampu meningkatkan kinerjanya pada suatu tugas tertentu (T) dengan pengalaman (E), yang diukur menggunakan indikator kinerja (P), maka program tersebut dapat dikatakan "belajar" dari pengalaman tersebut. Definisi ini terdengar sederhana dan cukup abstrak, namun sebenarnya menyimpan pemahaman yang mendalam tentang bagaimana sebuah mesin dapat meniru proses belajar manusia.

Seiring waktu, pemahaman kita mengenai pembelajaran mesin terus berkembang. Konsepnya tidak lagi bisa dijelaskan secara tuntas hanya dengan satu atau dua kalimat, karena cakupan ML begitu luas baik dari sisi teori maupun aplikasinya. Lebih dari itu, ML juga terus bertransformasi dengan sangat cepat, sehingga definisi dan lingkupnya senantiasa mengalami perluasan. Secara umum, pembelajaran mesin bekerja dengan mengolah data dalam jumlah besar menggunakan sistem pemrosesan dan algoritma tertentu. Dari proses tersebut, mesin berusaha menemukan pola-pola tersembunyi yang tidak mudah ditangkap manusia. Pola inilah yang kemudian digunakan untuk membuat prediksi atau pengambilan keputusan secara otomatis. Dengan demikian, ML menjadi salah satu sub-bidang paling penting dari kecerdasan buatan modern.

Hubungan ML dengan bidang lain juga sangat erat, terutama dengan data mining (penambangan data/DM) dan knowledge discovery in databases (penemuan pengetahuan dalam basis data/KDD). Dalam arti yang lebih luas, ketiganya saling melengkapi: data mining fokus menggali pola dari data mentah, KDD berorientasi pada transformasi data menjadi pengetahuan yang bermanfaat, sementara ML berperan sebagai jembatan algoritmik yang memungkinkan sistem komputer belajar, beradaptasi, dan membuat prediksi berbasis pola yang ditemukan.

Hubungan Antara AI, Pembelajaran Mesin, dan Pembelajaran Mendalam

Studi pembelajaran mesin bertujuan untuk memungkinkan komputer mensimulasikan atau melakukan kemampuan belajar manusia dan memperoleh pengetahuan serta keterampilan baru. Pembelajaran mendalam (DL) berasal dari studi jaringan saraf tiruan (JST). Sebagai subbidang baru pembelajaran mesin, pembelajaran mendalam berfokus pada peniruan mekanisme otak manusia dalam menginterpretasikan data seperti gambar, suara, dan teks. Hubungan antara AI, pembelajaran mesin, dan pembelajaran mendalam ditunjukkan pada Gambar 1.3.



Gambar 1.3 Hubungan antara kecerdasan buatan, pembelajaran mesin, dan pembelajaran mendalam

Di antara ketiga konsep tersebut, pembelajaran mesin merupakan pendekatan atau bagian dari AI, dan pembelajaran mendalam merupakan salah satu bentuk khusus ML. Jika kita menganggap AI sebagai otak, maka pembelajaran mesin adalah proses perolehan kemampuan kognitif, dan pembelajaran mendalam adalah sistem pelatihan mandiri yang sangat efisien yang mendominasi proses ini. Kecerdasan buatan adalah target dan hasil, sementara pembelajaran mendalam dan pembelajaran mesin adalah metode dan alat.

Jenis-jenis AI

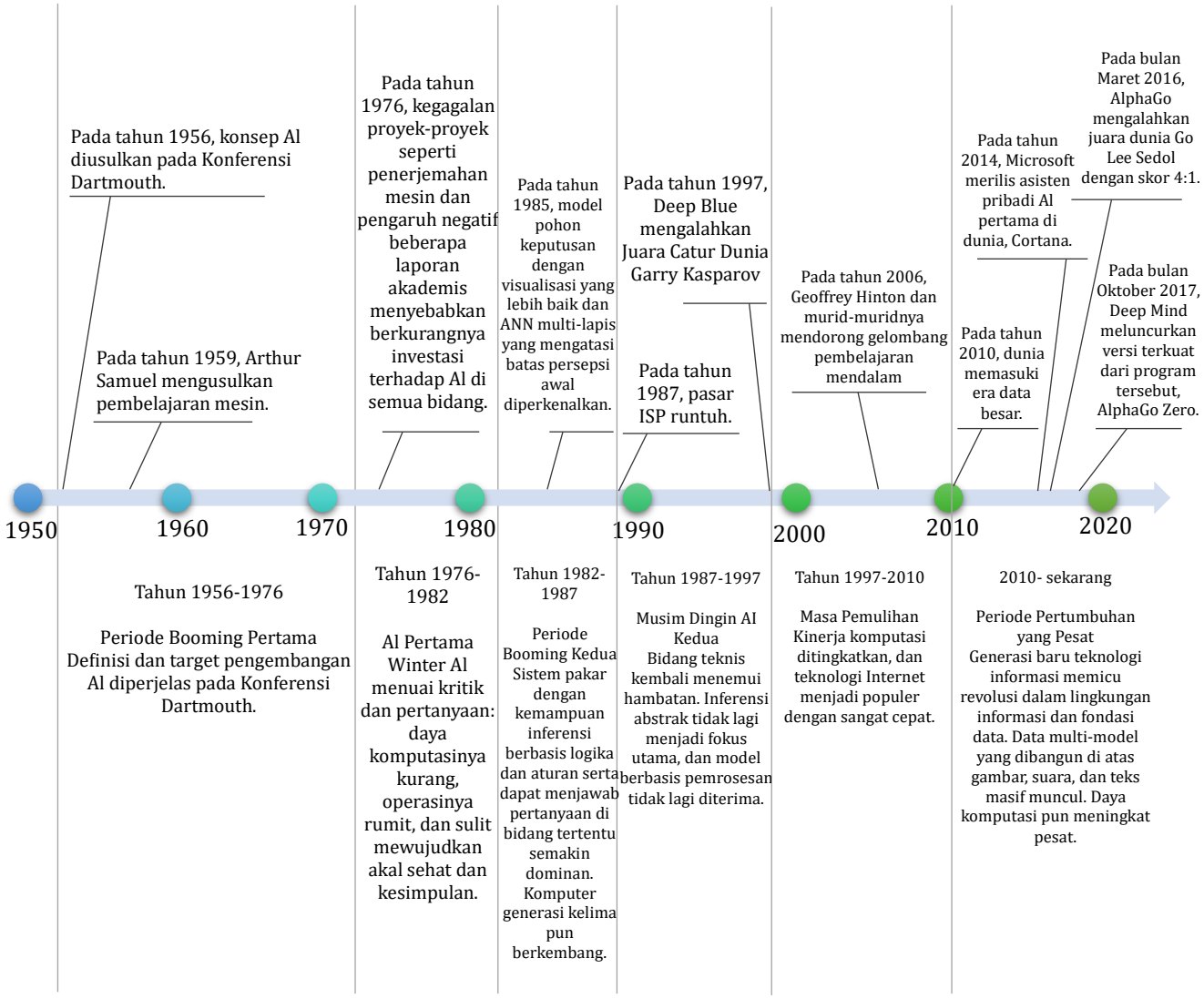
AI dapat dibagi menjadi dua jenis: kecerdasan buatan kuat dan kecerdasan buatan lemah. Kecerdasan buatan kuat adalah tentang kemungkinan untuk menciptakan mesin cerdas yang dapat menyelesaikan tugas-tugas penalaran dan pemecahan masalah. Mesin-mesin jenis ini diyakini memiliki kesadaran dan kesadaran diri serta mampu berpikir secara mandiri dan menghasilkan solusi terbaik untuk masalah tersebut. AI kuat juga memiliki nilai-nilai dan pandangan dunia yang khas, dan diberkahi dengan naluri, seperti kebutuhan untuk bertahan hidup dan rasa aman, layaknya semua makhluk hidup. Dalam arti tertentu, AI kuat adalah sebuah peradaban baru. Kecerdasan buatan yang lemah menggambarkan keadaan ketika ia tidak mampu menciptakan mesin yang benar-benar mampu melakukan penalaran dan pemecahan masalah. Mesin-mesin ini mungkin terlihat pintar, tetapi sebenarnya tidak memiliki kecerdasan atau kesadaran diri.

Kita saat ini berada di era kecerdasan buatan yang lemah. Pengenalan kecerdasan buatan yang lemah mengurangi beban kerja intelektual dengan cara kerja yang mirip dengan bionik canggih. Baik itu AlphaGo, maupun robot yang menulis berita dan novel, semuanya tergolong kecerdasan buatan yang lemah dan hanya mengungguli manusia dalam bidang-bidang tertentu. Di era kecerdasan buatan yang lemah, tidak dapat disangkal bahwa data dan daya komputasi sama-sama krusial, karena keduanya dapat memfasilitasi komersialisasi AI. Di era kecerdasan buatan yang kuat yang akan datang, kedua faktor ini akan tetap menjadi dua elemen penentu. Sementara itu, eksplorasi komputasi kuantum oleh perusahaan-perusahaan seperti Google dan IBM juga meletakkan dasar bagi munculnya era kecerdasan buatan yang kuat.

Sejarah AI

Gambar 1.4 menyajikan sejarah singkat AI. Asal usul resmi AI modern dapat ditelusuri kembali ke Tes Turing yang diajukan oleh Alan M. Turing, yang dikenal sebagai "Bapak Kecerdasan Buatan", pada tahun 1950. Menurut asumsinya, jika komputer dapat berdialog dengan manusia tanpa terdeteksi sebagai komputer, maka komputer tersebut dianggap memiliki kecerdasan. Pada tahun yang sama ketika ia mengajukan asumsi ini, Turing dengan berani meramalkan bahwa menciptakan mesin dengan kecerdasan manusia yang sesungguhnya adalah mungkin di masa depan.

Namun hingga saat ini, belum ada komputer yang pernah lulus Tes Turing. Meskipun AI adalah sebuah konsep dengan sejarah hanya beberapa dekade, fondasi teoretis dan teknologi pendukungnya telah berkembang selama kurun waktu yang panjang. Kemakmuran AI saat ini merupakan hasil dari kemajuan semua disiplin ilmu terkait dan upaya kolektif para ilmuwan dari semua generasi.



Gambar 1.4 Sejarah singkat AI

Pendahulu dan Periode Inisiasi (sebelum 1956)

Landasan teoretis bagi lahirnya AI dapat ditelusuri kembali ke abad keempat SM, ketika filsuf dan ilmuwan Yunani kuno yang terkenal, Aristoteles, menemukan konsep logika formal. Bahkan, teori silogismenya masih berfungsi sebagai landasan yang tak terpisahkan dan menentukan bagi penalaran deduktif hingga saat ini. Pada abad ke-17, matematikawan Jerman Gottfried Leibniz mengembangkan notasi universal dan beberapa gagasan revolusioner tentang penalaran dan kalkulasi, yang meletakkan dasar bagi pembentukan dan pengembangan logika matematika.

Pada abad ke-19, matematikawan Inggris George Boole mengembangkan aljabar Boolean, yang merupakan landasan pengoperasian komputer modern, dan pengenalannya memungkinkan penemuan komputer. Pada periode yang sama, penemu Inggris Charles Babbage menciptakan Difference Engine, komputer pertama di dunia yang mampu memecahkan persamaan polinomial kuadrat. Meskipun fungsinya terbatas, komputer ini untuk pertama kalinya mengurangi beban otak manusia dalam kalkulasi. Sejak saat itu, mesin-mesin telah diberkahi dengan kecerdasan komputasional.

Pada tahun 1945, John Mauchly dan J. Presper Eckert dari tim di Moore School merancang Electronic Numerical Integrator and Calculator (ENIAC), komputer elektronik digital serbaguna pertama di dunia. Sebagai sebuah pencapaian yang luar biasa, ENIAC masih memiliki kekurangan yang fatal, seperti ukurannya yang sangat besar, konsumsi daya yang berlebihan, dan ketergantungan pada operasi manual untuk memasukkan dan menyesuaikan perintah. Pada tahun 1947, John von Neumann, bapak komputer modern, memodifikasi dan meningkatkan basis ENIAC dan menciptakan komputer elektronik modern dalam arti sebenarnya: Mathematical Analyzer Numerical Integrator and Automatic Computer (MANIAC).

Pada tahun 1946, ahli fisiologi Amerika Warren McCulloch menetapkan model jaringan saraf pertama. Penelitiannya tentang kecerdasan buatan pada tingkat mikroskopis meletakkan fondasi penting bagi pengembangan jaringan saraf. Pada tahun 1949, Donald O. Hebb mengusulkan teori Hebbian, sebuah paradigma pembelajaran neuropsikologis, yang menyatakan prinsip-prinsip dasar plastisitas sinaptik, yaitu, efikasi transmisi sinaptik akan meningkat pesat dengan stimulasi yang berulang dan persisten dari neuron presinaptik ke neuron postsinaptik.

Teori ini fundamental bagi pemodelan jaringan saraf. Pada tahun 1948, Claude E. Shannon, pendiri teori informasi, memperkenalkan konsep entropi informasi dengan meminjam istilah tersebut dari termodinamika, dan mendefinisikan entropi informasi sebagai jumlah rata-rata informasi setelah redundansi dihilangkan. Dampak teori ini cukup luas karena memainkan peran penting dalam bidang-bidang seperti inferensi non-deterministik dan pembelajaran mesin.

Periode Booming Pertama (1956–1976)

Akhirnya, pada tahun 1956, John McCarthy secara resmi memperkenalkan AI sebagai disiplin ilmu baru pada Konferensi Dartmouth yang berlangsung selama 2 bulan, yang menandai lahirnya AI. Sejumlah kelompok riset AI dibentuk di Amerika Serikat sejak saat itu, seperti kelompok Carnegie-RAND yang dibentuk oleh Allen Newell dan Herbert Alexander Simon, kelompok riset Massachusetts Institute of Technology (MIT) oleh Marvin Lee Minsky dan John McCarthy, serta kelompok riset teknik IBM milik Arthur Samuel, dan lain-lain. Dalam dua dekade berikutnya, AI berkembang pesat di berbagai bidang, dan berkat antusiasme para peneliti yang tinggi, teknologi dan aplikasi AI terus berkembang.

(a) Pembelajaran Mesin

Pada tahun 1956, Arthur Samuel dari IBM menulis program permainan catur yang terkenal, yang mampu mempelajari model implisit dengan mengamati posisi pada papan catur untuk menginstruksikan langkah-langkah pada kasus-kasus selanjutnya. Setelah bermain melawan program tersebut selama beberapa putaran, Arthur Samuel menyimpulkan bahwa program tersebut dapat mencapai tingkat kinerja yang sangat tinggi selama proses pembelajaran. Dengan program ini, Samuel membantah anggapan bahwa komputer tidak dapat melampaui kode tertulis dan mempelajari pola seperti manusia. Sejak saat itu, ia menciptakan dan mendefinisikan istilah baru pembelajaran mesin.

(b) Pengenalan Pola

Pada tahun 1957, C.K. Chow mengusulkan penerapan teori keputusan statistik untuk menangani pengenalan pola, yang mendorong perkembangan pesat penelitian pengenalan pola sejak akhir 1950-an. Pada tahun yang sama, Frank Rosenblatt mengusulkan model matematika sederhana yang meniru pola pengenalan otak manusia perceptron, mesin pertama yang memungkinkan pelatihan sistem pengenalan berdasarkan sampel dari setiap kategori yang diberikan, sehingga sistem tersebut mampu mengklasifikasikan pola dari kategori lain yang tidak diketahui dengan tepat setelah pembelajaran.

(c) Pencocokan Pola

Pada tahun 1966, ELIZA, program percakapan pertama di dunia, diciptakan, yang ditulis oleh Laboratorium Kecerdasan Buatan MIT. Program ini mampu melakukan pencocokan pola berdasarkan aturan yang ditetapkan dan pertanyaan pengguna, sehingga dapat memberikan jawaban yang tepat dengan memilih dari arsip jawaban yang telah ditulis sebelumnya. Ini juga merupakan program pertama yang berhasil lolos Uji Turing. ELIZA pernah menyamar sebagai psikoterapis untuk berbicara dengan pasien, dan banyak dari mereka gagal mengenalinya sebagai robot ketika pertama kali diterapkan. "Percakapan adalah pencocokan pola", sehingga hal ini mengungkap teknologi percakapan bahasa alami komputer.

Selain itu, selama periode pengembangan pertama AI, John McCarthy mengembangkan LISP, yang menjadi bahasa pemrograman dominan untuk AI pada dekade-dekade berikutnya. Marvin Minsky meluncurkan studi yang lebih mendalam tentang jaringan saraf dan menemukan kelemahan jaringan saraf sederhana. Untuk mengatasi keterbatasan ini, para ilmuwan mulai memperkenalkan jaringan saraf tiruan berlapis dan algoritma Propagasi Balik (BP). Sementara itu, sistem pakar (ES) juga muncul.

Selama periode ini, robot industri pertama diterapkan pada lini produksi General Motors, dan dunia juga menyaksikan kelahiran robot bergerak pertama yang mampu beroperasi secara otonom. Kemajuan disiplin ilmu terkait juga berkontribusi pada kemajuan pesat AI. Kemunculan bionik pada tahun 1950-an memicu antusiasme penelitian para ilmuwan, yang kemudian mengarah pada penemuan algoritma simulasi anil. Algoritma ini merupakan jenis algoritma heuristik, dan merupakan fondasi bagi algoritma pencarian, seperti algoritma koloni semut.

Musim Dingin AI Pertama (1976–1982)

Namun, kegilaan AI tidak berlangsung lama, karena proyeksi yang terlalu optimistis gagal terpenuhi seperti yang dijanjikan, sehingga menimbulkan keraguan dan kecurigaan terhadap teknologi AI secara global. Perceptron, yang pernah menjadi sensasi di dunia akademis, mengalami kesulitan pada tahun 1969 ketika Marvin Minsky dan ilmuwan lainnya mengembangkan operasi logika terkenal, Exclusive OR (XOR), yang menunjukkan keterbatasan perceptron dalam hal data linear tak terpisahkan yang serupa dengan masalah XOR. Bagi dunia akademis, masalah XOR menjadi tantangan yang hampir tak terkalahkan.

Pada tahun 1973, AI dipertanyakan secara ketat oleh komunitas ilmiah. Banyak ilmuwan percaya bahwa tujuan AI yang tampaknya ambisius hanyalah ilusi yang belum

terpenuhi, dan penelitian yang relevan telah terbukti gagal total. Karena meningkatnya kecurigaan dan keraguan, AI menuai kritik tajam, dan nilai sebenarnya pun dipertanyakan. Akibatnya, pemerintah dan lembaga penelitian di seluruh dunia menarik atau mengurangi pendanaan untuk AI, dan industri ini mengalami musim dingin pertama perkembangannya pada tahun 1970-an.

Kemunduran pada tahun 1970-an bukanlah suatu kebetulan. Karena keterbatasan daya komputasi pada saat itu, meskipun banyak masalah dapat dipecahkan secara teoritis, masalah tersebut tidak dapat dipraktikkan sama sekali. Sementara itu, terdapat banyak kendala lain, seperti kesulitan bagi sistem pakar untuk memperoleh pengetahuan, sehingga banyak proyek berakhir dengan kegagalan. Penelitian tentang visi mesin mulai berkembang pada tahun 1960-an. Metode deteksi tepi dan komposisi kontur yang diusulkan oleh ilmuwan Amerika L.R. Roberts tidak hanya teruji oleh waktu, tetapi juga masih banyak digunakan hingga saat ini. Namun, memiliki dasar teoritis tidak menjamin hasil yang nyata. Pada akhir 1970-an, para ilmuwan menyimpulkan bahwa agar komputer dapat meniru penglihatan retina manusia, ia perlu mengeksekusi setidaknya satu miliar instruksi.

Namun, kecepatan kalkulasi superkomputer tercepat di dunia Cray-1 pada tahun 1976 (yang menghabiskan biaya jutaan dolar AS untuk membuatnya) hanya dapat mencatat tidak lebih dari 100 juta kali per detik, dan kecepatan komputer biasa bahkan tidak dapat mencapai lebih dari satu juta kali per detik. Perangkat keras membatasi perkembangan AI. Selain itu, basis data menjadi dasar utama kemajuan AI. Pada saat itu, komputer dan internet belum sepopuler saat ini, sehingga para pengembang tidak memiliki tempat untuk mengumpulkan data dalam jumlah besar.

Selama periode ini, kecerdasan buatan berkembang lambat. Meskipun konsep BP telah diusulkan oleh Juhani Linnainmaa dalam "mode flip diferensial otomatis" pada tahun 1970-an, konsep tersebut baru diterapkan pada multilayer perceptron oleh Paul J. Werbos pada tahun 1981. Penemuan multilayer perceptron dan algoritma BP berkontribusi pada lompatan kedua jaringan saraf tiruan. Pada tahun 1986, David E. Rumelhart dan peneliti lainnya mengembangkan algoritma BP yang efektif untuk melatih multilayer perceptron, yang memberikan dampak yang sangat besar.

Periode Booming Kedua (1982–1987)

Pada tahun 1980, XCON, sebuah sistem pakar lengkap yang dikembangkan oleh Carnegie Mello University (CMU), resmi digunakan. Sistem ini berisi lebih dari 2.500 aturan yang ditetapkan, dan memproses lebih dari 80.000 perintah dengan akurasi lebih dari 95% pada tahun-tahun berikutnya. Hal ini dianggap sebagai tonggak sejarah yang menandai era baru, ketika sistem pakar mulai menunjukkan potensinya di bidang-bidang tertentu, yang mengangkat teknologi AI ke tingkat perkembangan yang benar-benar baru dan pesat. Sistem pakar biasanya berfokus pada satu bidang profesional tertentu.

Dengan meniru pemikiran para pakar manusia, sistem ini mencoba menjawab pertanyaan atau memberikan pengetahuan untuk membantu para praktisi dalam pengambilan keputusan. Dengan fokus yang sempit, sistem pakar ini menghindari tantangan terkait kecerdasan umum buatan (AGI) dan mampu memanfaatkan sepenuhnya pengetahuan

serta pengalaman para pakar yang ada untuk memecahkan masalah di domain tertentu. Kesuksesan komersial XCON yang besar mendorong 60% perusahaan Fortune 500 untuk mengembangkan dan menerapkan sistem pakar di bidangnya masing-masing pada tahun 1980-an. Menurut statistik, dari tahun 1980 hingga 1985, lebih dari 1 miliar dolar AS diinvestasikan dalam AI, dengan mayoritas dialokasikan untuk departemen AI internal perusahaan-perusahaan tersebut, dan pasar menyaksikan lonjakan perusahaan perangkat lunak dan perangkat keras AI.

Pada tahun 1986, Universitas Bundeswehr di Munich melengkapi sebuah van Mercedes-Benz dengan komputer dan beberapa sensor, yang memungkinkan kontrol otomatis roda kemudi, akselerator, dan rem. Instalasi ini disebut VaMoRs, yang terbukti menjadi mobil self-driving pertama di dunia dalam arti sebenarnya.

LISP merupakan bahasa pemrograman utama yang digunakan untuk pengembangan AI pada masa itu. Untuk meningkatkan efisiensi operasional program LISP, banyak lembaga beralih mengembangkan chip komputer dan perangkat penyimpanan yang dirancang khusus untuk program LISP eksekutif. Meskipun mesin LISP telah mengalami beberapa kemajuan, komputer pribadi (PC) juga mengalami peningkatan. IBM dan Apple dengan cepat memperluas kehadiran pasar di seluruh pasar komputer. Dengan peningkatan frekuensi dan kecepatan CPU yang stabil, PC menjadi jauh lebih canggih daripada mesin LISP yang mahal.

Musim Dingin AI Kedua (1987–1997)

Pada tahun 1987, bersamaan dengan runtuhnya pasar perangkat keras mesin LISP—yang sebelumnya dianggap sebagai tulang punggung komputasi AI—industri kecerdasan buatan kembali memasuki periode stagnasi yang dikenal sebagai *musim dingin AI* kedua. Fase ini ditandai dengan berkurangnya minat pasar, runtuhnya industri perangkat keras khusus AI, serta menurunnya dukungan pendanaan dari pemerintah maupun lembaga penelitian di seluruh dunia. Akibatnya, perkembangan teknologi AI sempat terhenti dalam skala besar dan banyak proyek ambisius yang tidak lagi berlanjut.

Meskipun demikian, dekade yang tampak suram ini tetap menghasilkan sejumlah pencapaian penting. Pada tahun 1988, misalnya, ilmuwan Amerika Judea Pearl memperkenalkan pendekatan probabilistik dalam inferensi AI, yang kemudian dikenal melalui *Bayesian Networks*. Pendekatan ini menjadi tonggak penting karena memungkinkan sistem AI untuk menangani ketidakpastian dan membuat keputusan berbasis probabilitas, sehingga kelak menjadi fondasi utama dalam pengembangan pembelajaran mesin dan aplikasi modern seperti pengenalan pola, diagnosis medis, hingga sistem rekomendasi.

Dalam hampir dua dekade setelah musim dingin AI kedua, perkembangan teknologi AI tidak sepenuhnya mandek. Justru, AI secara perlahan terintegrasi dengan teknologi komputer dan perangkat lunak yang terus berevolusi. Namun, kemajuan dalam teori algoritma AI berjalan lambat. Sebagian besar penelitian hanya memperluas atau memodifikasi teori-teori yang sudah ada, sementara lompatan besar dalam inovasi belum muncul. Kemajuan yang terlihat pada masa ini lebih banyak ditopang oleh hadirnya perangkat keras komputer yang lebih cepat dan canggih, sehingga meskipun teori baru terbatas, aplikasi praktis AI tetap dapat berkembang secara bertahap.

Periode Pemulihan (1997–2010)

Pada tahun 1995, Richard S. Wallace terinspirasi oleh ELIZA dan mengembangkan program chatbot baru bernama A.L.I.C.E. (Entitas Komputer Internet Linguistik Buatan). Robot ini mampu mengoptimalkan konten dan memperkaya kumpulan datanya secara otomatis melalui internet. Pada tahun 1996, superkomputer IBM, Deep Blue, bermain catur melawan juara catur dunia Gary Kasparov dan kalah. Gary Kasparov percaya bahwa mustahil bagi komputer untuk mengalahkan manusia dalam permainan catur. Setelah pertandingan tersebut, IBM meningkatkan Deep Blue.

Deep Blue yang baru ini disempurnakan dengan 480 CPU khusus dan kecepatan kalkulasi dua kali lipat hingga 200 juta kali per detik, yang memungkinkannya memprediksi 8 langkah atau lebih berikutnya di papan catur. Dalam pertandingan ulang berikutnya, komputer berhasil mengalahkan Gary Kasparov. Namun, peristiwa penting ini sebenarnya hanya menandai kemenangan komputer atas manusia dalam permainan dengan aturan yang jelas dengan mengandalkan kecepatan kalkulasi dan enumerasinya. Ini bukanlah AI yang sebenarnya. Pada tahun 2006, ketika Geoffrey Hinton menerbitkan sebuah makalah di Science Magazine, industri AI memasuki era pembelajaran mendalam.

Periode Pertumbuhan Pesat (2010–sekarang)

Pada tahun 2011, sistem Watson, juga sebuah program dari IBM, berpartisipasi dalam acara kuis Jeopardy, bersaing dengan pemain manusia. Sistem Watson mengalahkan dua juara manusia dengan kemampuan pemrosesan bahasa alami yang luar biasa dan basis data pengetahuan yang kuat. Kali ini, komputer sudah dapat memahami bahasa manusia, yang merupakan kemajuan besar dalam AI.

Pada abad ke-21, dengan meluasnya penggunaan PC dan pesatnya perkembangan internet seluler serta teknologi komputasi awan, lembaga-lembaga mampu menangkap dan mengakumulasi data dalam jumlah yang tak terbayangkan besarnya, menyediakan materi dan dorongan yang memadai untuk pengembangan AI yang berkelanjutan. Pembelajaran mendalam menjadi arus utama teknologi AI, dicontohkan oleh proyek Google Brain yang terkenal, yang meningkatkan tingkat pengenalan dataset ImageNet hingga 84% dengan selisih yang besar.

Pada tahun 2011, konsep jaringan semantik diusulkan. Konsep ini berasal dari World Wide Web. Pada dasarnya, AI merupakan basis data terdistribusi berskala besar yang berpusat pada data web dan menghubungkan data web tersebut dengan metode pemahaman dan pemrosesan mesin. Kemunculan jaringan semantik sangat mendorong kemajuan teknologi representasi pengetahuan. Setahun kemudian, Google pertama kali mengumumkan konsep grafik pengetahuan dan meluncurkan layanan pencarian berbasis grafik pengetahuan.

Pada tahun 2016 dan 2017, Google meluncurkan dua kompetisi Go antara pemain manusia dan mekanik yang menggemparkan dunia. Program AI-nya, AlphaGo, mengalahkan dua juara dunia Go, pertama Lee Sedol dari Korea Selatan dan kemudian Ke Jie dari Tiongkok. Saat ini, AI dapat ditemukan di hampir semua aspek kehidupan manusia. Misalnya, asisten suara, seperti Siri yang paling umum di Apple, didasarkan pada teknologi pemrosesan bahasa alami (NLP).

Dengan dukungan NLP, komputer dapat memproses bahasa manusia dan mencocokkannya dengan perintah dan respons yang sesuai dengan harapan manusia secara lebih alami. Saat pengguna menjelajahi situs web e-commerce, mereka mungkin menerima umpan rekomendasi produk yang dihasilkan oleh algoritma rekomendasi. Algoritma rekomendasi dapat memprediksi produk yang mungkin ingin dibeli pengguna dengan meninjau dan menganalisis data historis pembelian dan preferensi terkini pengguna.

Tiga Aliran Utama AI

Saat ini, simbolisme, koneksionisme, dan behaviorisme merupakan tiga aliran utama AI. Bagian-bagian berikut akan memperkenalkan mereka secara rinci.

1. Simbolisme

Teori dasar simbolisme meyakini bahwa proses kognitif manusia terdiri dari inferensi dan pemrosesan simbol. Manusia adalah contoh sistem simbol fisik, begitu pula komputer. Oleh karena itu, komputer harus mampu mensimulasikan aktivitas kecerdasan manusia. Representasi pengetahuan, penalaran pengetahuan, dan aplikasi pengetahuan merupakan tiga hal krusial bagi kecerdasan buatan. Simbolisme berpendapat bahwa pengetahuan dan konsep dapat direpresentasikan oleh simbol, sehingga kognisi adalah proses pemrosesan simbol, dan penalaran adalah proses pemecahan masalah dengan pengetahuan heuristik. Inti dari simbolisme terletak pada penalaran, yaitu penalaran simbolik dan penalaran mesin.

2. Koneksionisme

Fondasi koneksionisme adalah bahwa sifat berpikir logis manusia adalah neuron, bukan proses pemrosesan simbol. Koneksionisme meyakini bahwa otak manusia berbeda dari komputer, dan mengajukan model koneksionis yang meniru kerja otak untuk menggantikan model kerja komputer yang dioperasikan oleh simbol. Koneksionisme diyakini berasal dari bionik, terutama dalam studi model otak manusia. Dalam koneksionisme, suatu konsep direpresentasikan oleh serangkaian angka, vektor, matriks, atau tensor, yaitu, oleh mode aktivasi spesifik dari keseluruhan jaringan.

Setiap simpul (neuron) dalam jaringan tidak memiliki makna spesifik, tetapi setiap simpul berpartisipasi dalam ekspresi konsep secara keseluruhan. Misalnya, dalam simbolisme, konsep kucing dapat direpresentasikan oleh "simpul kucing" atau sekelompok simpul yang menampilkan atribut kucing (misalnya, yang memiliki "dua mata", "empat kaki", atau "berbulu"). Namun, koneksionisme meyakini bahwa setiap simpul tidak memiliki makna spesifik, sehingga mustahil untuk mencari "simpul kucing" atau "neuron mata". Inti koneksionisme terletak pada jaringan neuron dan pembelajaran mendalam.

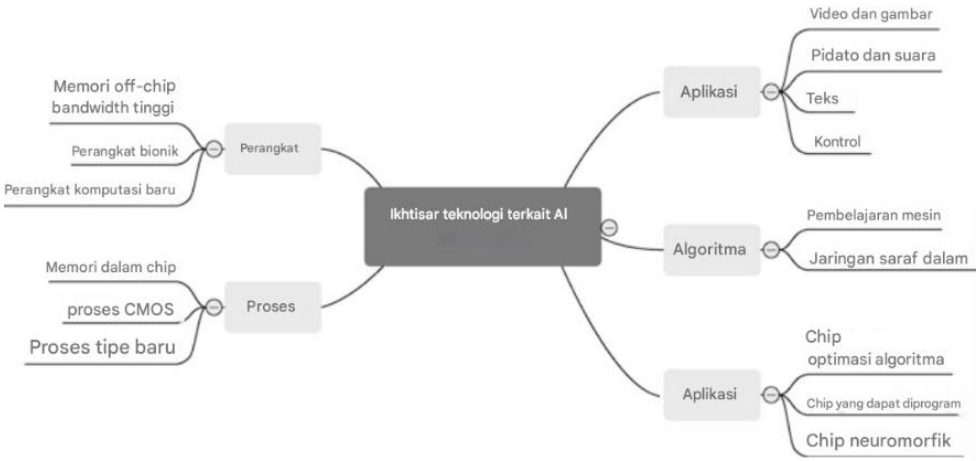
3. Behaviorisme

Teori fundamental behaviorisme meyakini bahwa kecerdasan bergantung pada persepsi dan perilaku. Behaviorisme memperkenalkan model "persepsi-tindakan" untuk aktivitas cerdas. Behaviorisme percaya bahwa kecerdasan tidak ada hubungannya dengan pengetahuan, representasi, atau penalaran. AI dapat berevolusi secara bertahap seperti kecerdasan manusia, dan aktivitas cerdas hanya dapat terwujud melalui interaksi berkelanjutan manusia dengan lingkungan sekitarnya di dunia nyata. Behaviorisme menekankan penerapan dan praktik serta pembelajaran berkelanjutan dari lingkungan untuk

memodifikasi aktivitas. Inti behaviorisme terletak pada pengendalian perilaku, adaptasi, dan komputasi evolusioner.

1.2 TEKNOLOGI TERKAIT AI

Teknologi AI berlapis-lapis, berjalan melalui berbagai tingkat teknis seperti aplikasi, algoritma, chip, perangkat, dan proses, seperti yang ditunjukkan pada Gambar 1.5. Teknologi AI telah mencapai perkembangan berikut di semua tingkat teknis.



Gambar 1.5 Tinjauan umum teknologi terkait AI

1. Tingkat Aplikasi

- ✓ Video dan gambar: pengenalan wajah, deteksi target, pembuatan gambar, retouching gambar, pencarian gambar per gambar, analisis video, tinjauan video, dan augmented reality (AR).
- ✓ Ucapan dan suara: pengenalan suara, sintesis suara, pengaktifan suara, pengenalan sidik suara, dan pembuatan musik.
- ✓ Teks: analisis teks, penerjemahan mesin, dialog manusia-mesin, pemahaman bacaan, dan sistem rekomendasi.
- ✓ Kontrol: pengemudian otonom, drone, robot, otomasi industri.

2. Tingkat Algoritma

- ✓ Algoritma pembelajaran mesin: jaringan saraf tiruan, mesin vektor pendukung (SVM), algoritma K-tetangga terdekat (KNN), algoritma Bayesian, pohon keputusan, model Markov tersembunyi (HMM), pembelajaran ansambel, dll.
- ✓ Algoritma optimasi umum untuk pembelajaran mesin: penurunan gradien, metode Newton, metode kuasi-Newton, gradien konjugat, plastisitas bergantung waktu spiking (STDP), dll.

Pembelajaran mendalam (deep learning) merupakan salah satu teknologi terpenting untuk pembelajaran mesin. Jaringan saraf tiruan dalam (DNN) merupakan pusat penelitian di bidang ini dalam beberapa tahun terakhir, yang terdiri dari multilayer perceptron (MLP) dan jaringan saraf tiruan konvolusional (CNN), jaringan saraf tiruan berulang (RNN), jaringan saraf tiruan spiking (SNN), dan jenis lainnya. CNN yang relatif populer antara lain AlexNet, ResNet, dan

VGGNet, sedangkan RNN yang populer antara lain jaringan memori jangka panjang (LSTM) dan Neural Turing Machine (NTM). Misalnya, BERT (Bidirectional Encoder Representation from Transformers) Google adalah teknologi pra-pelatihan pemrosesan bahasa alami yang dikembangkan berdasarkan jaringan saraf tiruan.

Selain pembelajaran mendalam, pembelajaran transfer, pembelajaran penguatan, pembelajaran sekali pakai, dan pembelajaran mesin adversarial juga merupakan teknologi penting untuk mewujudkan pembelajaran mesin, dan solusi untuk beberapa kesulitan yang dihadapi oleh pembelajaran mendalam.

3. Chip Level

- ✓ Chip optimasi algoritma: optimasi kinerja, optimasi konsumsi daya rendah, optimasi kecepatan tinggi, dan optimasi fleksibilitas—seperti akselerator pembelajaran mendalam dan chip pengenalan wajah.
- ✓ Chip neuromorfik: otak bionik, kecerdasan yang terinspirasi oleh otak biologis, imitasi mekanisme otak.
- ✓ Chip yang dapat diprogram: mempertimbangkan fleksibilitas, kemampuan pemrograman, kompatibilitas algoritma, dan kompatibilitas perangkat lunak umum, seperti chip pemrosesan sinyal digital (DSP), unit pemrosesan grafis (GPU), dan field programmable gates array (FPGA).
- ✓ Struktur sistem pada chip: multi-inti, banyak-inti, instruksi tunggal, banyak data (SIMD), struktur operasi array, arsitektur memori, struktur jaringan pada chip, struktur interkoneksi multi-chip, antarmuka memori, struktur komunikasi, cache multi-level.
- ✓ Rantai alat pengembangan: koneksi antara kerangka kerja pembelajaran mendalam (TensorFlow, Caffe, MindSpore), kompiler, simulator, pengoptimal (kuantisasi dan kliping), pustaka operasi atom (jaringan).

4. Tingkat Perangkat

- ✓ Memori off-chip bandwidth tinggi: memori bandwidth tinggi (HBM), memori akses acak dinamis (DRAM), memori laju data ganda grafis berkecepatan tinggi (GDDR), laju data ganda daya rendah (LPDDR SDRAN), memori akses acak magnetik torsi transfer-spin (STT-MRAM).
- ✓ Perangkat interkoneksi berkecepatan tinggi: serializer/deserializer (SerDes), komunikasi interkoneksi optik.
- ✓ Perangkat bionik (sinapsis buatan, neuron buatan): memristor.
- ✓ Perangkat komputasi tipe baru: komputasi analog, komputasi dalam memori (IMC).

5. Tingkat Proses

- ✓ Memori on-chip (array sinaptik): memori akses acak statis terdistribusi (SRAM), memori akses acak resistif (ReRAM), dan memori akses acak perubahan fase (PCRAM).
- ✓ Proses CMOS: simpul teknologi (16 nm, 7 nm).
- ✓ Integrasi multi-lapis CMOS: teknologi IC/SiP 2.5D, teknologi 3D-Stack, dan 3D Monolitik.
- ✓ Proses tipe baru: 3D NAND, Flash Tunneling FET, FeFET, FinFET.

Kerangka Kerja Pembelajaran Mendalam

Pengenalan kerangka kerja pembelajaran mendalam telah membuat pembelajaran mendalam lebih mudah dibangun. Dengan kerangka kerja pembelajaran mendalam ini, kita tidak perlu mengodekan jaringan saraf kompleks terlebih dahulu dengan algoritma backpropagation, tetapi cukup mengonfigurasi hiperparameter model sesuai kebutuhan kita, dan parameter model dapat dipelajari secara otomatis dari proses pelatihan.

Kita juga dapat menambahkan lapisan khusus untuk model yang sudah ada, atau memilih pengklasifikasi dan algoritma optimasi yang kita butuhkan di bagian atas. Kita dapat menganggap kerangka kerja pembelajaran mendalam sebagai sekumpulan blok penyusun. Setiap blok, atau komponen dari kumpulan tersebut, merupakan model atau algoritma, dan kita dapat menyusun komponen-komponen tersebut menjadi arsitektur yang memenuhi kebutuhan. Kerangka kerja pembelajaran mendalam arus utama saat ini meliputi: TensorFlow, Caffe, PyTorch, dan sebagainya.

Tinjauan Umum Prosesor AI

Dalam empat elemen kunci teknologi AI (data, algoritma, daya komputasi, dan skenario), daya komputasi adalah yang paling bergantung pada prosesor AI. Juga dikenal sebagai akselerator AI, prosesor AI adalah modul fungsional khusus untuk menangani tugas komputasi skala besar dalam aplikasi AI.

1. Jenis-jenis Prosesor AI

Prosesor AI dapat diklasifikasikan ke dalam berbagai kategori dari berbagai perspektif, dan di sini kita akan membahas perspektif arsitektur teknis dan fungsinya. Berdasarkan arsitektur teknisnya, prosesor AI secara garis besar dapat diklasifikasikan menjadi empat jenis.

(a) CPU

Unit pemrosesan pusat (CPU) adalah sirkuit integrasi skala besar, yang merupakan inti dari komputasi dan kendali komputer. Fungsi utama CPU adalah untuk menginterpretasikan instruksi program dan memproses data dalam perangkat lunak yang diterimanya dari komputer.

(b) GPU

Unit pemrosesan grafis (GPU), juga dikenal sebagai inti tampilan (DC), unit pemrosesan visual (VPU), dan chip tampilan, adalah mikroprosesor khusus yang menangani pemrosesan gambar di komputer pribadi, stasiun kerja, konsol gim, dan beberapa perangkat seluler (seperti tablet dan ponsel pintar).

(c) ASIC

Sirkuit terpadu khusus aplikasi (ASIC) dirancang untuk produk sirkuit terpadu yang disesuaikan untuk penggunaan tertentu.

(d) FPGA

Field Programmable Gate Array (FPGA) dirancang untuk membangun chip semi-kustom yang dapat dikonfigurasi ulang, yang berarti struktur perangkat kerasnya dapat disesuaikan dan dikonfigurasi ulang secara fleksibel secara real-time sesuai kebutuhan.

Berdasarkan fungsinya, prosesor AI dapat diklasifikasikan menjadi dua jenis: prosesor pelatihan dan prosesor inferensi.

(a) Untuk melatih model jaringan saraf tiruan dalam yang kompleks, pelatihan AI biasanya memerlukan input data dalam jumlah besar dan metode pembelajaran seperti pembelajaran penguatan. Pelatihan merupakan proses komputasi yang intensif. Data pelatihan berskala besar dan struktur jaringan saraf tiruan dalam yang kompleks yang melibatkan pelatihan memberikan tantangan besar bagi kecepatan, akurasi, dan skalabilitas prosesor. Prosesor pelatihan yang populer meliputi GPU NVIDIA, unit pemrosesan tensor (TPU) Google, dan unit pemrosesan jaringan saraf tiruan (NPU) Huawei.

(b) Inferensi di sini berarti menyimpulkan berbagai kesimpulan dengan data baru yang diperoleh berdasarkan model yang telah dilatih. Misalnya, monitor video dapat membedakan apakah wajah yang ditangkap merupakan target spesifik dengan memanfaatkan model jaringan saraf tiruan dalam (deep neural network) backend. Meskipun inferensi memerlukan komputasi yang jauh lebih sedikit daripada pelatihan, inferensi tetap melibatkan banyak operasi matriks. GPU, FPGA, dan NPU umumnya digunakan dalam prosesor inferensi.

2. Status Prosesor AI Saat Ini

(a) CPU

Peningkatan kinerja CPU pada awalnya terutama bergantung pada kemajuan yang dicapai oleh teknologi perangkat keras yang mendasarinya sejalan dengan Hukum Moore. Dalam beberapa tahun terakhir, seiring Hukum Moore yang tampaknya secara bertahap kehilangan efektivitasnya, pengembangan sirkuit terpadu melambat, dan teknologi perangkat keras menghadapi hambatan fisik. Keterbatasan pembuangan panas dan konsumsi daya membatasi kinerja CPU dan efisiensi program serial pada arsitektur tradisional untuk mencapai banyak kemajuan.

Status quo industri mendorong para peneliti untuk terus mencari arsitektur CPU dan kerangka kerja perangkat lunak yang relevan yang dapat beradaptasi lebih baik dengan Era pasca-Moore. Hasilnya, prosesor multi-inti muncul, yang memungkinkan kinerja CPU yang lebih tinggi dengan lebih banyak inti. Prosesor multi-inti dapat memenuhi tuntutan perangkat lunak pada perangkat keras dengan lebih baik. Misalnya, prosesor Intel Core i7 mengadopsi prosesor paralel tingkat instruksi dengan beberapa kernel independen pada set instruksi x86, yang meningkatkan kinerja secara signifikan, tetapi juga menyebabkan konsumsi daya dan biaya yang lebih tinggi. Karena jumlah inti tidak dapat ditingkatkan tanpa batas, dan sebagian besar program tradisional ditulis dalam pemrograman serial, pendekatan ini memiliki peningkatan yang terbatas pada kinerja CPU dan efisiensi program.

Selain itu, kinerja AI dapat ditingkatkan dengan menambahkan set instruksi. Misalnya, menambahkan set instruksi seperti AVX512 ke arsitektur komputer set instruksi kompleks (CISC) x86, menambahkan set instruksi fused-multiply-add (FMA) ke modul unit logika aritmatika (ALU), dan menambahkan set instruksi ke arsitektur komputer set instruksi tereduksi ARM (RISC). Performa CPU juga dapat ditingkatkan dengan meningkatkan frekuensi, tetapi ada batasannya, dan frekuensi yang tinggi akan menyebabkan konsumsi daya yang berlebihan dan suhu yang tinggi.

(b) GPU

GPU sangat kompetitif dalam komputasi matriks dan komputasi paralel dan berfungsi sebagai mesin komputasi heterogen. Pertama kali diperkenalkan ke bidang AI sebagai akselerator untuk memfasilitasi pembelajaran mendalam dan kini telah membentuk ekologi yang mapan. Terkait GPU di bidang pembelajaran mendalam, NVIDIA berupaya terutama dalam tiga aspek berikut:

- ✓ Memperkaya ekologi: NVIDIA meluncurkan NVIDIA CUDA deep neural network horary (CUDNN), pustaka akselerasi GPU yang dikustomisasi untuk pembelajaran mendalam, yang mengoptimalkan arsitektur dasar GPU dan memastikan penerapan GPU yang lebih mudah dalam pembelajaran mendalam.
- ✓ Meningkatkan kustomisasi: merangkul berbagai tipe data (tidak lagi terpaku pada float32, dan mengadopsi int8, dll.).
- ✓ Menambahkan modul khusus untuk pembelajaran mendalam (misalnya, GPU NVIDIA V100 Tensor Core mengadopsi arsitektur Volta yang disempurnakan, memperkenalkan dan melengkapinya dengan inti tensor).

Tantangan utama GPU saat ini adalah biaya tinggi, rasio konsumsi energi rendah, serta latensi input dan output yang tinggi.

(c) TPU

Sejak 2016, Google telah berkomitmen untuk menerapkan konsep sirkuit terpadu khusus aplikasi (ASIC) dalam studi jaringan saraf tiruan. Pada tahun 2016, Google meluncurkan prosesor AI yang dikembangkan khusus, TPU, yang mendukung kerangka kerja pembelajaran mendalam sumber terbuka TensorFlow. Dengan menggabungkan larik sistolik skala besar dan memori on-chip berkapasitas tinggi, TPU berhasil mempercepat operasi konvolusional yang paling umum dalam jaringan saraf tiruan dalam secara efisien: larik sistolik dapat mengoptimalkan perkalian matriks dan operasi konvolusional, sehingga dapat meningkatkan daya komputasi dan mengurangi konsumsi energi.

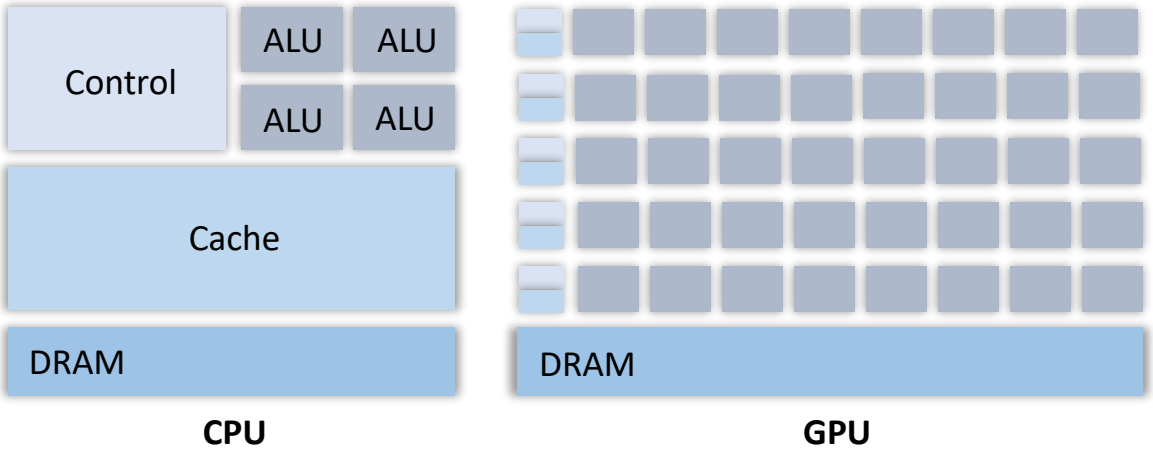
(d) FPGA

FPGA menggunakan bahasa deskripsi perangkat keras (HDL) yang dapat diprogram, yang fleksibel, dapat dikonfigurasi ulang, dan dapat dikustomisasi secara mendalam. FPGA dapat memuat model DNN pada chip untuk melakukan operasi latensi rendah dengan menggabungkan beberapa FPGA, yang berkontribusi pada kinerja komputasi yang lebih tinggi daripada GPU. Namun, karena harus memperhitungkan proses penghapusan yang konstan, kinerja FPGA tidak dapat mencapai optimal. Karena FPGA dapat dikonfigurasi ulang, risiko pasokan dan litbangnya relatif rendah. Biaya perangkat keras ditentukan oleh jumlah perangkat keras yang dibeli, sehingga mudah untuk mengontrol biaya. Namun, desain FPGA dan proses tape-out dipisahkan, sehingga siklus pengembangannya panjang, yang biasanya memakan waktu setengah tahun, dan memiliki standar yang tinggi.

3. Perbandingan Antara Desain GPU dan CPU

GPU umumnya dirancang untuk menangani data berskala besar yang sangat terpadu jenisnya dan independen satu sama lain, serta menangani lingkungan komputasi murni tanpa interupsi. CPU dirancang lebih umum, sehingga dapat memproses berbagai jenis data, dan

melakukan keputusan logis secara bersamaan, dan juga perlu memperkenalkan sejumlah besar instruksi lompat cabang dan pemrosesan interupsi. Perbandingan antara arsitektur CPU dan GPU ditunjukkan pada Gambar 1.6.



Gambar 1.6 Arsitektur CPU dan GPU

GPU memiliki banyak arsitektur komputasi paralel masif yang terdiri dari ribuan inti yang jauh lebih kecil (dirancang untuk pemrosesan simultan beberapa tugas). CPU terdiri dari beberapa inti yang dioptimalkan untuk pemrosesan serial.

(a) GPU bekerja dengan banyak ALU dan sedikit memori cache. Tidak seperti CPU, cache GPU hanya berfungsi untuk thread dan berperan sebagai penerusan data. Ketika beberapa thread perlu mengakses data yang sama, cache akan menggabungkan akses-akses ini, kemudian mengakses DRAM, dan meneruskan data ke setiap thread setelah mendapatkannya, yang akan menyebabkan latensi. Namun, karena banyaknya ALU memastikan thread berjalan secara paralel, latensi pun berkurang. Selain itu, unit kontrol GPU dapat menggabungkan akses.

(b) CPU memiliki ALU yang kuat, yang dapat menyelesaikan komputasi dalam siklus clock yang sangat singkat. CPU memiliki banyak cache untuk mengurangi latensi, dan unit kontrol yang rumit yang dapat melakukan prediksi cabang dan penerusan data: ketika sebuah program memiliki banyak cabang, unit kontrol akan mengurangi latensi melalui prediksi cabang; untuk instruksi yang bergantung pada hasil instruksi sebelumnya, unit kontrol harus menentukan posisi instruksi ini dalam alur proses dan meneruskan hasil instruksi sebelumnya secepat mungkin.

GPU unggul dalam menangani operasi intensif dan mudah dijalankan secara paralel, sementara CPU unggul dalam kontrol logika dan operasi serial. Perbedaan arsitektur antara GPU dan CPU terletak pada penekanannya. GPU memiliki keunggulan luar biasa dalam memproses komputasi paralel data intensif berskala besar, sementara CPU lebih menekankan kontrol logika saat mengeksekusi instruksi. Untuk mengoptimalkan program, CPU dan CPU seringkali perlu mengoordinasikan CPU dan GPU secara bersamaan untuk memaksimalkan kemampuannya.

4. Prosesor AI Huawei Ascend

NPU mengacu pada prosesor yang menjalankan desain optimasi khusus untuk komputasi jaringan saraf tiruan, yang kinerjanya dalam memproses tugas jaringan saraf tiruan jauh lebih tinggi daripada CPU dan GPU. NPU meniru neuron dan sinapsis manusia pada sirkuit, dan secara langsung memproses neuron dan sinapsis berskala besar melalui rangkaian instruksi prosesor pembelajaran mendalam. Dalam NPU, pemrosesan sekelompok neuron hanya membutuhkan satu instruksi. Saat ini, contoh umum NPU meliputi prosesor AI Huawei Ascend, chip Cambrian, dan chip TrueNorth IBM. Ada dua jenis prosesor AI Huawei Ascend:

- Ascend 310 dan Ascend 910. Ascend 910 terutama diterapkan pada skenario pelatihan, sebagian besar diterapkan di pusat data. Sementara Ascend 310 terutama dirancang untuk skenario penalaran, yang penerapannya mencakup skenario perangkat, edge, dan cloud.
- Ascend 910 saat ini merupakan prosesor AI dengan daya komputasi terkuat dan kecepatan pelatihan tercepat di dunia, daya komputasinya dua kali lipat dari prosesor AI terkemuka internasional, setara dengan 50 CPU terbaru dan terkuat saat ini.

Parameter Ascend 310 dan Ascend 910 yang relevan ditunjukkan pada Tabel 1.1.

Tabel 1.1 Parameter yang relevan dari Ascend 310 dan Ascend 910

Ascend 310	Ascend 910
Chip: Ascend Mini	Chip: Ascend-Max
Arsitektur: Da Vinci	Arsitektur: Da Vinci
Performa presisi titik mengambang 16-bit (FP16): 8 TFLOPS	Performa presisi titik mengambang 16-bit (FP16): 256 TFLOPS
Kinerja presisi integer 8-bit (INT8): 16 TOPS	Kinerja presisi integer 8-bit (INT8):512 TOPS
Dekoder video HD penuh 16 saluran	Dekoder video HD penuh 128 saluran
H.264/265	H.264/265
Encoder video HD penuh 1 saluran	Konsumsi daya maksimum: 350 W
H.264/265	Proses: 7nm
Konsumsi daya maksimum: 8 W	
Proses: FFC 12 nm	

Ekosistem Industri AI

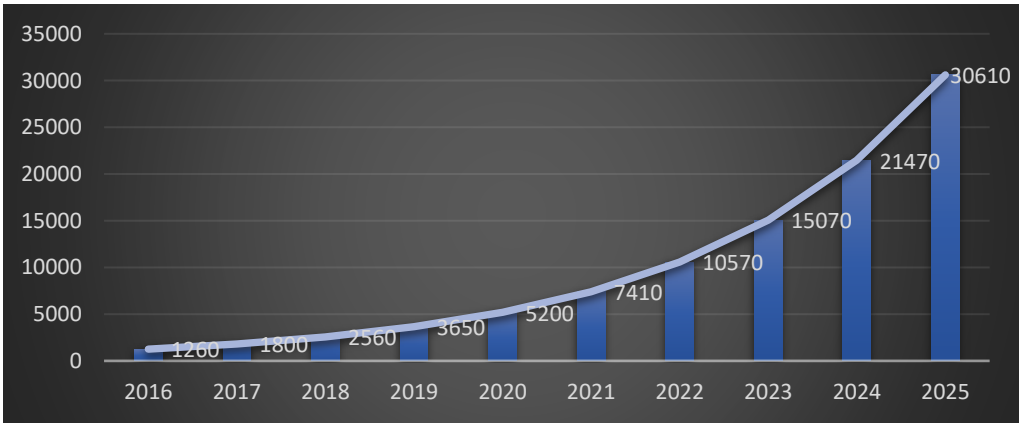
Selama setengah abad terakhir, dunia telah menyaksikan tiga gelombang AI. Ketiga gelombang ini dicontohkan dan diungkap oleh pertandingan manusia-komputer. Gelombang pertama terjadi pada tahun 1962, ketika program permainan catur yang dikembangkan oleh Arthur Samuel dari IBM mengalahkan pemain catur terbaik dunia, Amerika Serikat. Gelombang kedua terjadi pada tahun 1997, ketika superkomputer IBM, Deep Blue, mengalahkan juara dunia catur manusia, Garry Kasparov, dengan skor 3,5:2,5. Gelombang ketiga AI terjadi pada tahun 2016 ketika AI Go, AlphaGo, yang dikembangkan oleh DeepMind, anak perusahaan Google, mengalahkan juara dunia Go dan pemain sembilan-dan dari Korea Selatan, Lee Sedol. Di masa depan, AI akan tertanam di setiap aspek kehidupan, mulai dari otomotif, keuangan, barang konsumsi hingga ritel, layanan kesehatan, pendidikan,

manufaktur, komunikasi, energi, pariwisata, budaya dan hiburan, transportasi, logistik, real estat, perlindungan lingkungan, dan lain-lain.

Misalnya, dalam industri otomotif, teknologi mengemudi cerdas seperti mengemudi dengan bantuan, pengambilan keputusan dengan bantuan, dan mengemudi otomatis penuh semuanya diwujudkan dengan bantuan AI. Sebagai pasar yang besar, mengemudi cerdas juga dapat mendukung penelitian teknis di bidang AI, sehingga membentuk siklus positif dan menjadi fondasi yang kokoh bagi pengembangan AI. Sedangkan untuk industri keuangan, dengan akumulasi data yang sangat besar, AI dapat membantu manajemen aset cerdas, robo-advisor, dan pengambilan keputusan keuangan yang lebih bijaksana.

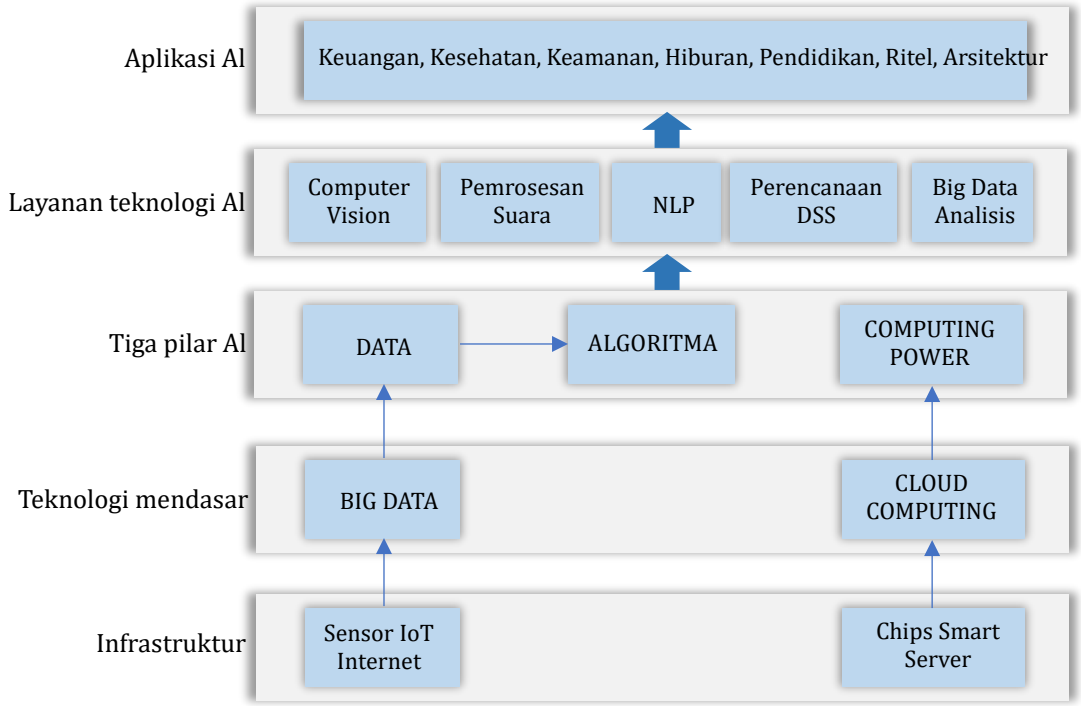
AI juga dapat berperan dalam memerangi penipuan keuangan dan diadopsi dalam kampanye anti-penipuan dan anti-pencucian uang, karena program AI terkait dapat menyimpulkan keandalan suatu transaksi dengan menganalisis data dan materi dari berbagai jenis, untuk memprediksi ke mana dana akan mengalir, dan mengidentifikasi siklus pasar keuangan. AI juga dapat digunakan secara luas dalam industri perawatan kesehatan. Misalnya, AI dapat membantu dokter mendiagnosis dan mengobati penyakit dengan mengidentifikasi masalah yang tercermin dalam citra sinar-X, setelah dilatih untuk menginterpretasikan citra pada tingkat geometri. AI dapat membedakan sel kanker dari sel normal setelah pelatihan tugas klasifikasi.

Data penelitian terkait menunjukkan bahwa pada tahun 2025, ukuran pasar AI akan melebihi 3 triliun dolar AS, seperti yang ditunjukkan pada Gambar 1.7. Sebagaimana dapat disimpulkan dari Gambar 1.7, pasar AI memiliki potensi yang sangat besar. Diketahui bahwa AI memiliki tiga pilar, yaitu data, algoritma, dan daya komputasi.



Gambar 1.7 Perkiraan ukuran pasar AI *Dalam Miliar Rupiah*

Namun, untuk menerapkan AI dalam kehidupan nyata, ketiga pilar ini masih jauh dari cukup, karena kita juga perlu mempertimbangkan skenario. Data, algoritma, dan daya komputasi dapat mendorong perkembangan AI secara teknis, tetapi tanpa skenario aplikasi, perkembangan teknologi hanya bergantung pada data. Kita perlu mengintegrasikan AI dengan komputasi awan, big data, dan Internet of Things (IoT) agar penerapan AI di dunia nyata dapat terwujud. Hal ini merupakan fondasi arsitektur platform untuk aplikasi AI, seperti yang ditunjukkan pada Gambar 1.8.



Gambar 1.8 Arsitektur platform untuk aplikasi AI

Infrastruktur tersebut mencakup sensor pintar dan cip pintar, yang memperkuat daya komputasi industri AI dan menjamin perkembangannya. Layanan teknologi AI utamanya berfokus pada pembangunan platform teknologi AI dan penyediaan solusi serta layanan kepada pengguna eksternal. Produsen teknologi AI ini berperan penting dalam rantai industri AI, karena mereka menyediakan platform, solusi, dan layanan teknologi utama untuk semua jenis aplikasi AI berkat infrastruktur yang kuat dan data masif yang mereka peroleh.

Dengan percepatan kampanye membangun Tiongkok yang kompetitif melalui pengembangan manufaktur, internet, dan industri digital, permintaan AI di bidang manufaktur, peralatan rumah tangga, keuangan, pendidikan, transportasi, keamanan, perawatan medis, logistik, dan bidang lainnya akan semakin meningkat, dan produk AI akan semakin beragam bentuk dan kategori. Hanya ketika infrastruktur, empat elemen utama AI dan layanan teknis AI bertemu, arsitektur dapat sepenuhnya menopang aplikasi lapisan atas ekosistem industri AI.

Meskipun teknologi AI dapat diterapkan di berbagai bidang, pengembangan dan penerapannya juga menghadapi tantangan: ketidakseimbangan antara keterbatasan pengembangan AI dan permintaan pasar yang besar. Saat ini, pengembangan dan penerapan AI perlu mengatasi tiga masalah berikut.

1. Standar pekerjaan yang tinggi: Untuk terlibat dalam industri AI, seseorang harus memiliki pengetahuan yang memadai dalam pembelajaran mesin, pembelajaran mendalam, statistika, aljabar linear, dan kalkulus.

2. Efisiensi rendah. Pelatihan model akan membutuhkan siklus kerja yang panjang, yang terdiri dari pengumpulan data, pembersihan data, pelatihan dan penyetelan model, serta optimalisasi pengalaman visualisasi.
3. Kemampuan dan pengalaman yang terfragmentasi: untuk menerapkan model AI yang sama dalam skenario lain, diperlukan pengumpulan data, pembersihan data, pelatihan dan penyetelan model, serta optimalisasi pengalaman yang berulang, dan kemampuan model AI tidak dapat langsung ditularkan ke skenario berikutnya.
4. Peningkatan dan perbaikan kapasitas yang sulit: peningkatan model dan pengambilan data yang efektif merupakan tugas yang sulit.

Saat ini, AI pada perangkat yang berpusat pada ponsel pintar telah menjadi konsensus industri. Semakin banyak ponsel pintar yang akan memiliki kemampuan AI. Seperti yang diperkirakan oleh beberapa lembaga konsultan di Inggris dan AS, sekitar 80% ponsel pintar dunia akan memiliki kemampuan AI pada tahun 2022 atau 2023. Untuk memenuhi prospek pasar dan mengatasi tantangan AI, HUAWEI meluncurkan platform kemampuan AI terbuka untuk perangkat pintar, yaitu HUAWEI HiAI. Dengan misi "memberikan kemudahan kepada pengembang sekaligus menghubungkan kemungkinan tak terbatas melalui AI", HUAWEI HiAI memungkinkan pengembang untuk memberikan pengalaman aplikasi pintar yang lebih baik kepada pengguna dengan memanfaatkan kemampuan pemrosesan AI Huawei yang canggih secara cepat.

Platform Aplikasi Huawei CLOUD Enterprise Intelligence

1. Tinjauan Umum Platform Aplikasi Huawei CLOUD Enterprise Intelligence.

Platform aplikasi Huawei CLOUD Enterprise Intelligence (EI) merupakan penggerak kecerdasan perusahaan yang bertujuan menyediakan platform yang terbuka, kredibel, dan cerdas berbasis teknologi AI dan big data, serta dalam bentuk layanan cloud (termasuk cloud publik dan cloud khusus, dll.). Dengan menggabungkan skenario industri, sistem aplikasi perusahaan yang dibuat dengan Huawei CLOUD dapat divisualisasikan, didengar, dan mengekspresikan diri, serta dilengkapi kemampuan untuk menganalisis dan menginterpretasikan gambar, video, bahasa, dan teks, serta akses yang lebih mudah ke layanan AI dan big data. Huawei CLOUD dapat membantu perusahaan mempercepat pengembangan bisnis dan memberikan manfaat bagi masyarakat.

2. Fitur Huawei CLOUD EI

Huawei CLOUD EI memiliki empat fitur unggulan.

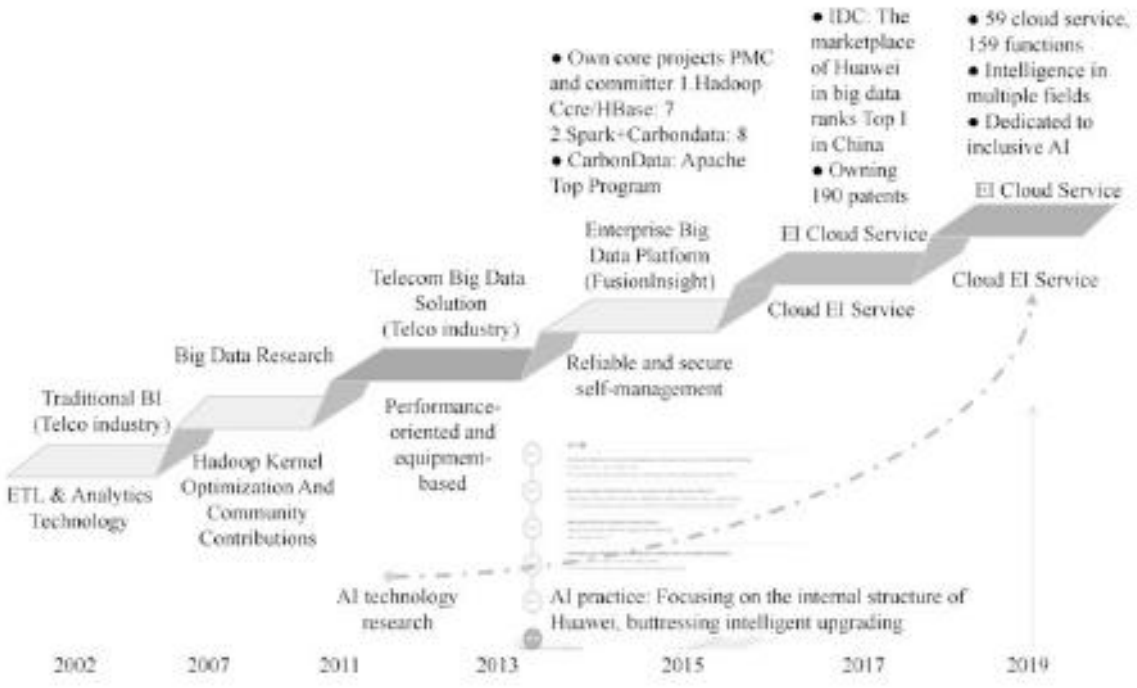
(a) *Kearifan industri*: Huawei CLOUD memiliki pemahaman mendalam tentang industri, pengetahuan industri, dan kekurangan utama industri. Ia mencari solusi untuk permasalahan dalam teknologi AI dan menavigasi implementasi AI.

(b) *Data industri*: Memungkinkan perusahaan memanfaatkan data mereka sendiri untuk menciptakan nilai besar melalui pemrosesan data dan penambangan data.

(c) *Algoritma*: Menyediakan perusahaan dengan pustaka algoritma dan pustaka model yang luas, serta solusi untuk permasalahan perusahaan melalui layanan AI umum dan platform pengembangan terpadu.

(d) *Daya komputasi*: Berdasarkan pengalaman Huawei selama 30 tahun di bidang TIK, platform pengembangan AI tumpukan penuh dapat menyediakan daya komputasi AI terkuat dan paling ekonomis bagi perusahaan untuk fusi dan perubahan.

3. Sejarah Huawei CLOUD EI



Gambar 1.9 Sejarah Huawei CLOUD EI

Perkembangan Huawei CLOUD EI ditunjukkan pada Gambar 1.9. Perkembangan Huawei CLOUD EI adalah sebagai berikut.

(a) Pada tahun 2002, Huawei mulai mengembangkan produk tata kelola dan analisis data yang menargetkan operasi Kecerdasan Bisnis (BI) tradisional di bidang telekomunikasi.

(b) Pada tahun 2007, Huawei memulai proyek penelitian teknologi Hadoop, memetakan strategi terkait data besar, dan membangun kumpulan profesional dan paten teknologi yang relevan.

(c) Pada tahun 2011, Huawei mencoba menerapkan teknologi big data dalam solusi big data telekomunikasi untuk menangani diagnosis dan analisis jaringan, perencanaan jaringan, dan penyetelan jaringan.

(d) Pada tahun 2013, beberapa perusahaan besar seperti China Merchants Bank dan Industrial and Commercial Bank bertukar pandangan dengan Huawei terkait kebutuhan mereka terkait big data dan memulai kerja sama teknis. Pada bulan September di tahun yang sama, Huawei meluncurkan platform analisis big data kelas perusahaan, FusionInsight, di Huawei Cloud Congress (HCC), yang telah diadopsi oleh berbagai industri.

(e) Pada tahun 2012, Huawei secara resmi memasuki industri AI dan memproduksi hasil penelitiannya secara berturut-turut sejak tahun 2014. Pada akhir tahun 2015, produk

yang dikembangkan untuk keuangan, rantai pasokan, penerimaan pekerjaan teknik, dan e-commerce mulai digunakan secara internal, dengan pencapaian berikut.

- ✓ Pengenalan karakter optik (OCR) untuk pengenalan dokumen deklarasi pabean: Efisiensi impor meningkat 10 kali lipat.
- ✓ Perencanaan rute pengiriman: Biaya tambahan berkurang 30%.
- ✓ Audit cerdas: Efisiensi meningkat 6 kali lipat.
- ✓ Rekomendasi cerdas untuk pengguna e-commerce: Tingkat konversi aplikasi meningkat 71%.

(f) Pada tahun 2017, Huawei resmi terjun ke layanan EI dalam bentuk layanan cloud dan bekerja sama dengan lebih banyak mitra untuk menyediakan layanan AI yang lebih beragam kepada pengguna eksternal.

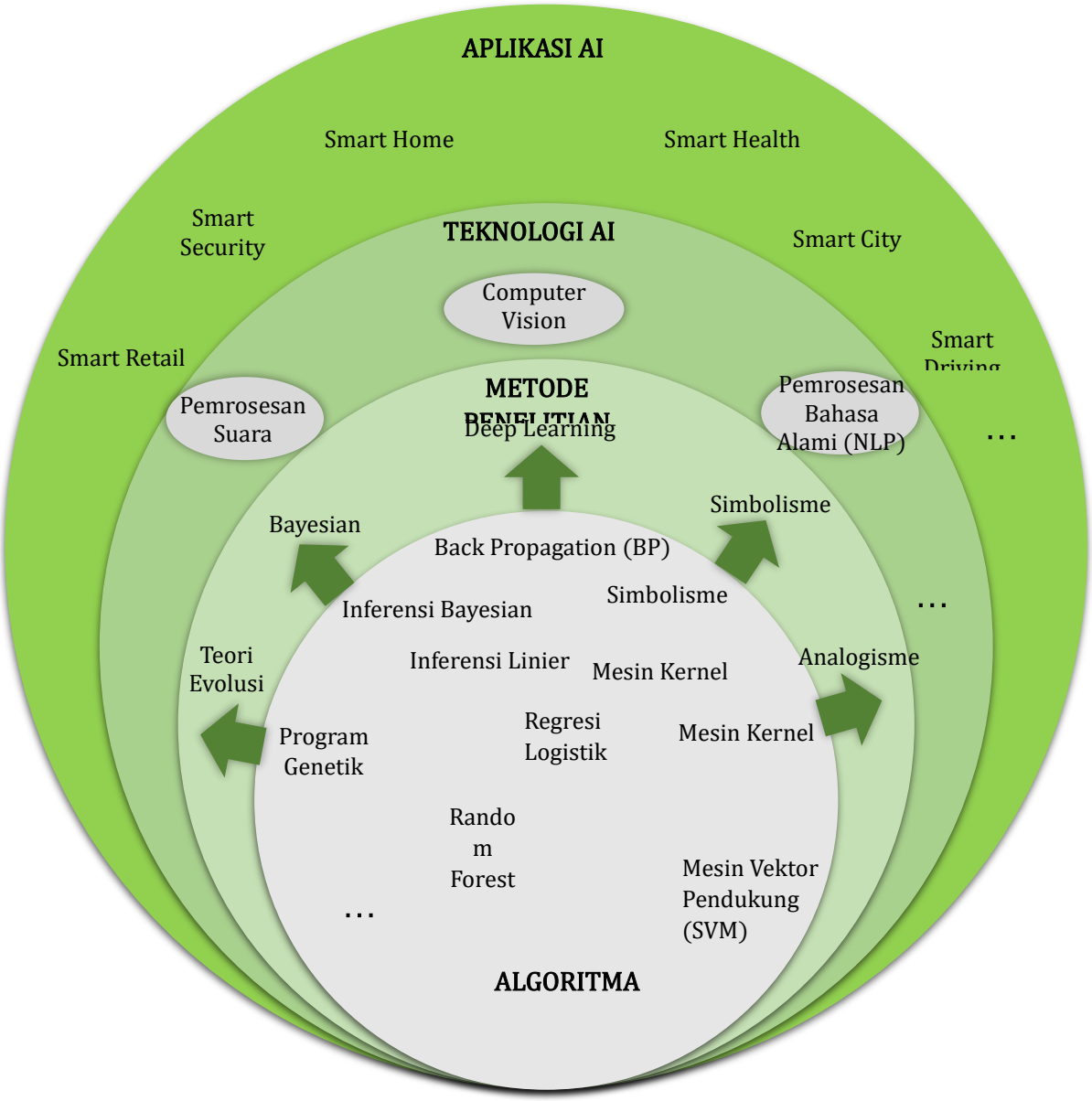
(g) Pada tahun 2019, Huawei CLOUD EI mulai menekankan AI yang inklusif, dengan harapan dapat menjadikan AI terjangkau, mudah diakses, dan aman digunakan. Berdasarkan chip Ascend yang dikembangkan sendiri, Huawei menyediakan 59 layanan cloud (21 layanan platform, 22 layanan vision, 12 layanan bahasa, dan 4 layanan pengambilan keputusan) dan mengembangkan 159 fungsi (52 fungsi platform, 99 fungsi antarmuka pemrograman aplikasi [API], dan 8 solusi pra-integrasi).

Ribuan pengembang Huawei terlibat dalam proyek-proyek R&D teknologi yang disebutkan di atas (termasuk penelitian dan pengembangan teknologi produk, serta teknologi mutakhir seperti algoritma analisis, algoritma pembelajaran mesin, dan pemrosesan bahasa alami), sementara Huawei juga secara aktif berbagi hasil penelitian dengan komunitas riset AI Huawei.

1.3 TEKNOLOGI DAN APLIKASI AI

Teknologi AI

Seperti yang ditunjukkan pada Gambar 1.10, teknologi AI terutama mencakup tiga jenis teknologi aplikasi: visi komputer, pemrosesan ucapan, dan pemrosesan bahasa alami.

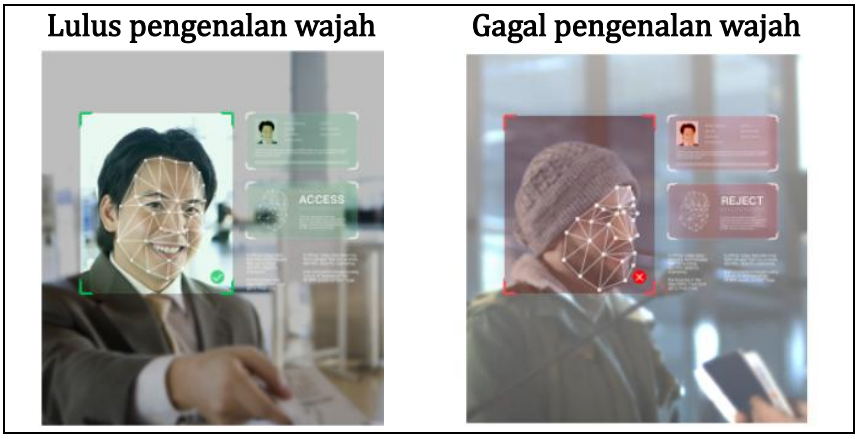


Gambar 1.10 Teknologi AI

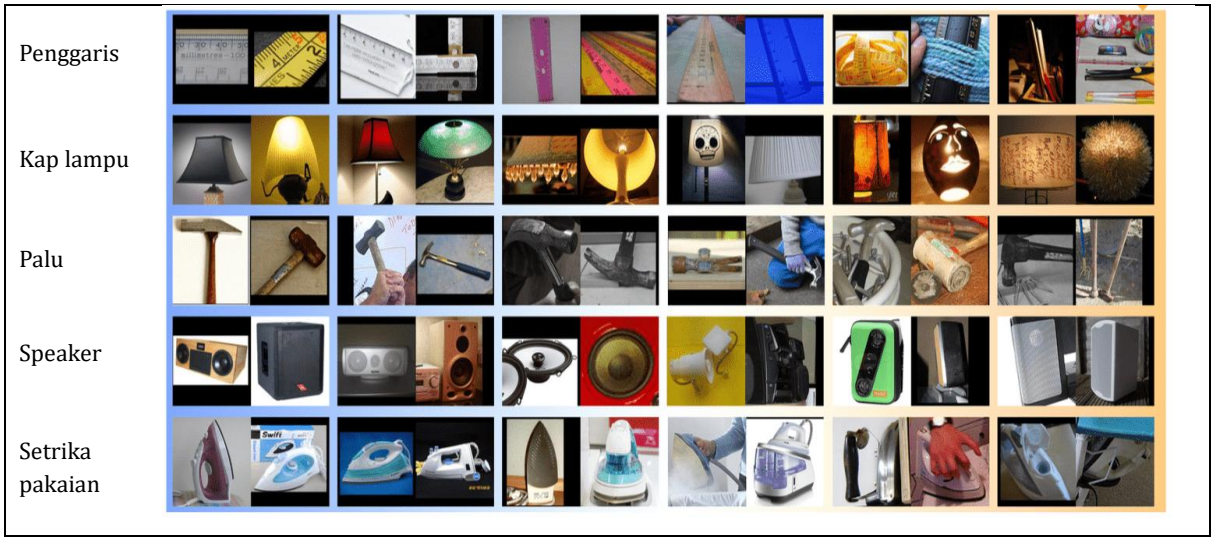
1. Visi Komputer

Visi komputer adalah ilmu yang mengeksplorasi bagaimana komputer dapat "melihat" sesuatu, dan merupakan teknologi yang paling mapan di antara tiga jenis teknologi aplikasi AI. Subjek yang terutama ditangani oleh visi komputer meliputi klasifikasi gambar, deteksi objek, segmentasi gambar, pelacakan visual, pengenalan teks, dan pengenalan wajah. Saat ini, visi komputer umumnya digunakan dalam pelacakan kehadiran elektronik, verifikasi identitas, pengenalan gambar, dan pencarian gambar, seperti yang ditunjukkan pada Gambar 1.11, 1.12, 1.13, dan 1.14. Di masa depan, visi komputer akan ditingkatkan ke tingkat yang lebih maju, yaitu mampu menafsirkan, menganalisis gambar, dan membuat keputusan secara otonom. Dengan demikian, mesin benar-benar memiliki kemampuan untuk "melihat", dan memainkan peran yang lebih besar dalam skenario seperti kendaraan tanpa awak dan rumah pintar.

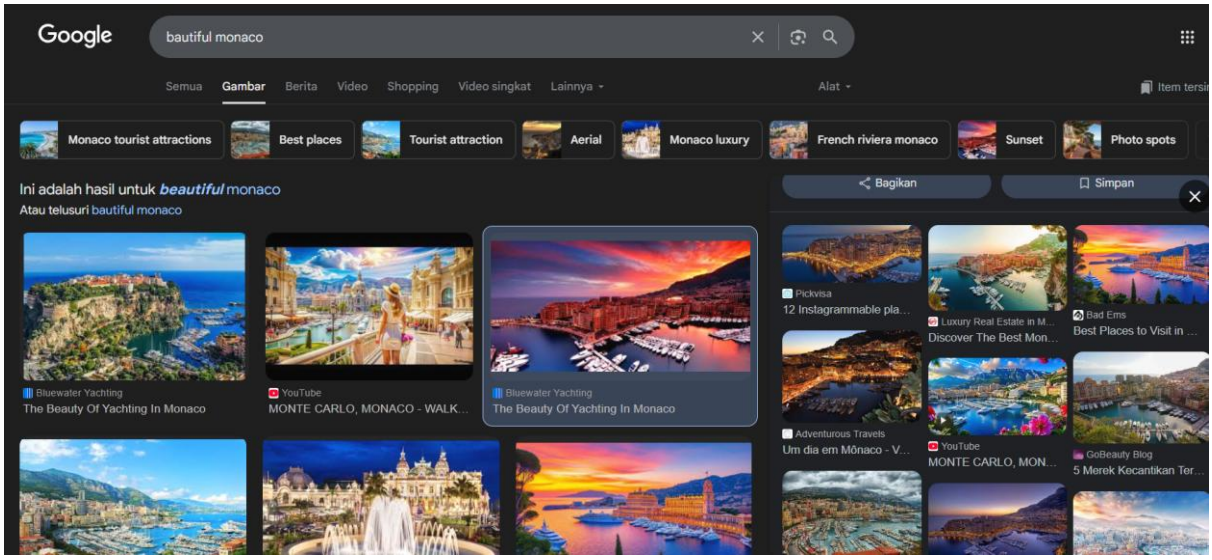
Gambar 1.11 Absensi Elektronik



Gambar 1.12 Verifikasi identitas



Gambar 1.13 Pengenalan gambar



Gambar 1.14 Pencarian gambar

2. Pemrosesan Ucapan

Pemrosesan ucapan adalah studi tentang karakteristik statistik sinyal ucapan dan produksi suara. Teknologi pemrosesan seperti pengenalan ucapan, sintesis ucapan, dan pembangkitan suara secara kolektif dapat disebut sebagai "pemrosesan ucapan". Sub-domain penelitian pemrosesan ucapan sebagian besar mencakup pengenalan ucapan, sintesis ucapan, pembangkitan suara, pengenalan sidik suara, dan deteksi peristiwa suara. Sub-domain yang paling matang adalah pengenalan ucapan, yang dapat mencapai tingkat akurasi 96% berdasarkan lingkungan dalam ruangan yang tenang dan pengenalan medan dekat. Saat ini, teknologi pengenalan ucapan terutama digunakan dalam tanya jawab cerdas dan navigasi cerdas, seperti yang ditunjukkan pada Gambar 1.15 dan 1.16.

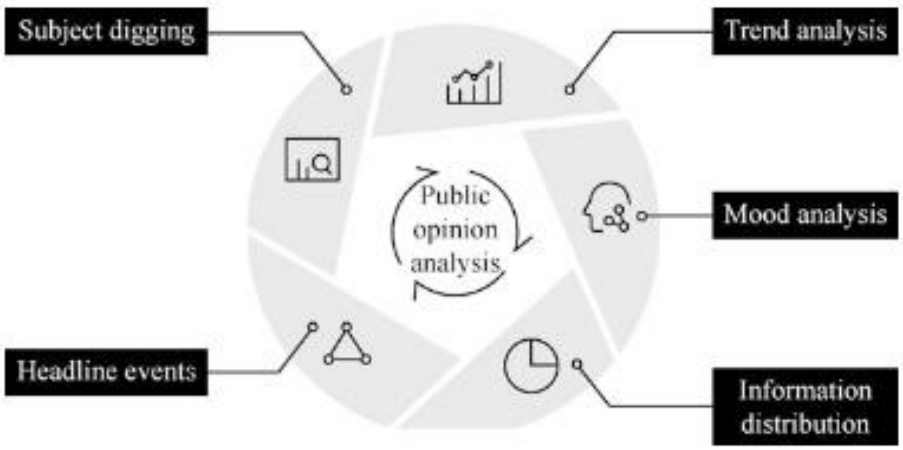
Gambar 1.15 Tanya Jawab Cerdas

Gambar 1.16 Navigasi Cerdas

3. Pemrosesan Bahasa Alami (NLP)

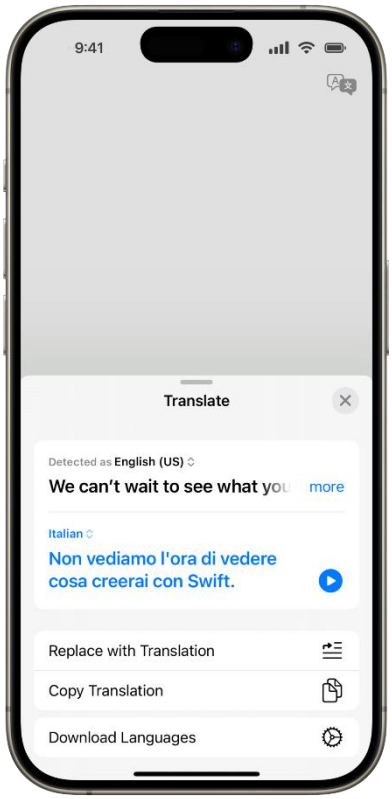
Pemrosesan bahasa alami adalah teknologi yang bertujuan untuk menafsirkan dan memanfaatkan bahasa alami melalui teknologi komputer. Subjek NLP meliputi penerjemahan mesin, penambangan teks, dan analisis sentimen. Dihadapkan dengan sejumlah tantangan teknis, NLP belum menjadi teknologi yang sangat matang saat ini. Karena kompleksitas semantik yang tinggi, mustahil bagi AI untuk menyaingi manusia dalam memahami semantik hanya dengan pembelajaran mendalam berbasis data besar dan komputasi paralel.

Di masa depan, AI diharapkan dapat berkembang ke tahap yang dapat secara otomatis mengekstrak fitur dan memahami semantik mendalam dari status saat ini yang hanya dapat memahami semantik dangkal, dan untuk meningkatkan dari kecerdasan tunggal (pembelajaran mesin) menjadi kecerdasan hibrida (pembelajaran mesin, pembelajaran mendalam, dan pembelajaran penguatan). Teknologi NLP sekarang banyak diterapkan di bidang-bidang seperti analisis opini publik, analisis komentar, dan penerjemahan mesin, seperti yang ditunjukkan pada Gambar 1.17, 1.18, dan 1.19.



Gambar 1.17 Analisis Opini Publik

Gambar 1.18 Analisis Komentar



Gambar 1.19 Penerjemahan Mesin

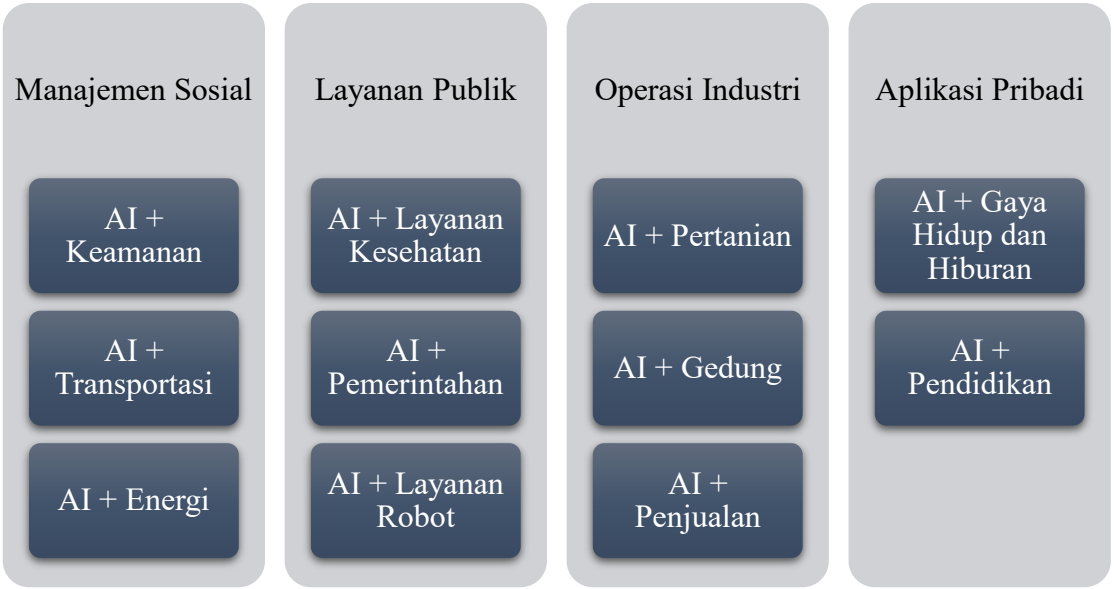
Diterapkan dalam bidang-bidang seperti analisis opini publik, analisis komentar, dan penerjemahan mesin, seperti yang ditunjukkan pada Gambar 1.17, 1.18, dan 1.19.

Aplikasi AI

Aplikasi AI adalah sebagai berikut.

1. Kota Pintar

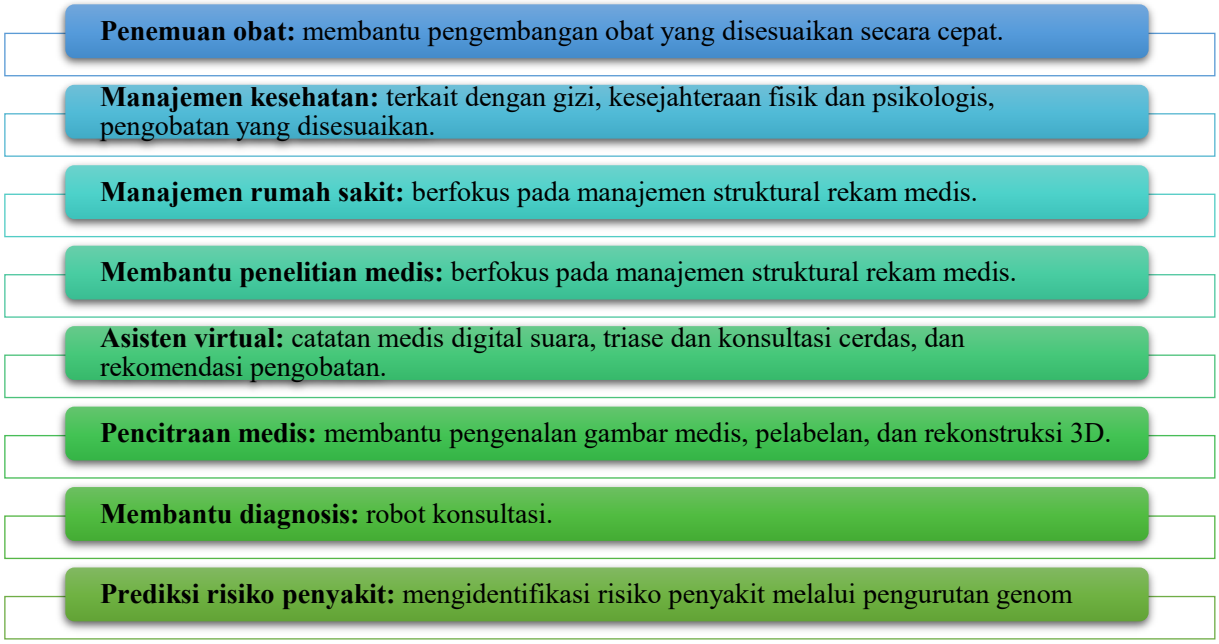
Kota pintar adalah kota yang memanfaatkan teknologi TIK untuk merasakan, menganalisis, dan mengintegrasikan informasi kunci dari sistem operasi inti perkotaan, sehingga dapat secara cerdas merespons kebutuhan kota dalam hal penghidupan masyarakat, perlindungan lingkungan, keselamatan publik, layanan perkotaan, serta kegiatan industri dan komersial. Hakikat kota pintar adalah mewujudkan manajemen dan operasi kota yang cerdas melalui teknologi informasi mutakhir, sehingga meningkatkan standar hidup warga, dan mendorong pembangunan kota yang harmonis dan berkelanjutan. Dalam kota pintar, AI terutama dicontohkan sebagai lingkungan cerdas, ekonomi cerdas, kehidupan cerdas, informasi cerdas, rantai pasokan cerdas, dan pemerintahan cerdas. Lebih spesifiknya, teknologi AI diadopsi oleh logistik pemantauan lalu lintas, dan pengenalan wajah untuk keamanan dan perlindungan. Gambar 1.20 menunjukkan struktur kota pintar.



Gambar 1.20 Kota Pintar

2. Layanan Kesehatan Cerdas

Kita dapat memungkinkan AI untuk "mempelajari" pengetahuan medis profesional, "menghafal" banyak rekam medis, dan menganalisis citra medis dengan visi komputer, sehingga dapat memberikan bantuan yang andal dan efisien kepada dokter, seperti yang ditunjukkan pada Gambar 1.21. Misalnya, untuk pencitraan medis yang banyak digunakan saat ini, AI dapat membangun model berdasarkan data historis untuk menganalisis citra medis dan mendeteksi lesi dengan cepat, sehingga meningkatkan efisiensi konsultasi.



Gambar 1.21 Layanan Kesehatan Cerdas

3. Ritel Cerdas

AI juga akan merevolusi industri ritel. Contoh kasusnya adalah supermarket tanpa awak. Supermarket tanpa awak Amazon, Amazon Go, mengadopsi sensor, kamera, visi komputer, dan algoritma pembelajaran mendalam serta membatalkan proses pembayaran tradisional, sehingga pelanggan cukup masuk ke toko, mengambil produk yang mereka butuhkan, dan pergi. Salah satu tantangan utama yang dihadapi oleh supermarket tanpa awak adalah bagaimana menagih pelanggan dengan benar. Hingga saat ini, Amazon Go adalah satu-satunya kasus yang berhasil, tetapi hal ini juga dicapai dengan banyak prasyarat. Misalnya, Amazon Go hanya terbuka untuk anggota Prime Amazon. Perusahaan lain perlu membangun sistem keanggotaan mereka sendiri terlebih dahulu jika ingin mengikuti model Amazon.

4. Keamanan Cerdas

AI lebih mudah diimplementasikan di bidang keamanan, dan perkembangannya relatif matang. Data gambar dan video terkait keamanan yang melimpah telah memberikan fondasi yang baik untuk pelatihan algoritma dan model AI. Dalam domain keamanan, penerapan teknologi AI dapat diklasifikasikan sebagai untuk penggunaan sipil dan untuk penggunaan kepolisian. Untuk penggunaan sipil: pengenalan wajah, peringatan dini potensi bahaya, pertahanan rumah, dan sebagainya. Untuk penggunaan kepolisian: identifikasi target yang mencurigakan, analisis kendaraan, pelacakan tersangka, pencarian dan perbandingan tersangka kriminal, penjaga pintu masuk area pengawasan utama, dll.

5. Rumah Pintar

Rumah pintar mengacu pada ekosistem rumah berbasis teknologi IoT yang terdiri dari perangkat keras, perangkat lunak, dan platform cloud, yang menyediakan layanan kehidupan yang disesuaikan bagi pengguna dan lingkungan tinggal yang lebih nyaman, aman, dan praktis. Peralatan rumah pintar ini dirancang untuk dikendalikan oleh teknologi pemrosesan suara, seperti mengatur suhu AC, membuka tirai, dan mengendalikan sistem pencahayaan.

Keamanan rumah bergantung pada teknologi visi komputer, seperti membuka kunci melalui pengenalan wajah atau sidik jari, pemantauan kamera pintar secara real-time, dan deteksi penyusupan ilegal ke dalam rumah. Dengan bantuan pembelajaran mesin dan pembelajaran mendalam, rumah pintar dapat membuat potret pengguna dan memberikan rekomendasi berdasarkan catatan historis yang tersimpan di speaker pintar dan TV pintar.

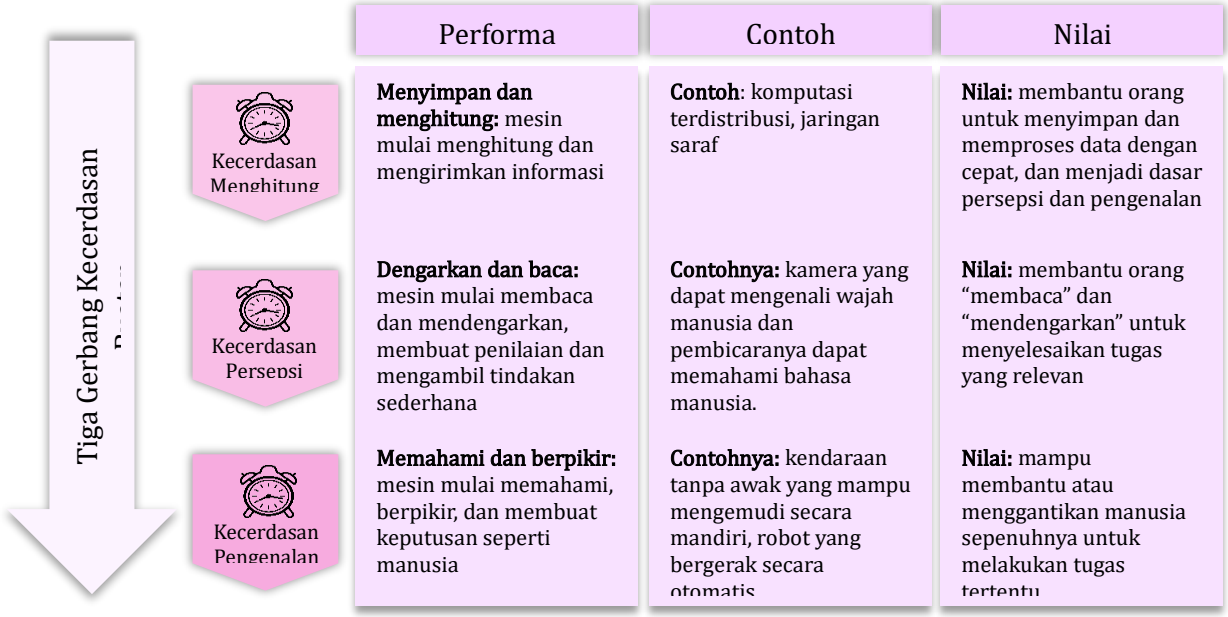
6. Mengemudi Cerdas

Society of Automotive Engineers (SAE) mendefinisikan enam tingkat untuk mengemudi otonom dari L0 hingga L5 berdasarkan tingkat ketergantungan kendaraan pada sistem pengemudian. Kendaraan level L0 harus sepenuhnya bergantung pada pengoperasian pengemudi, dan kendaraan pada level L3 ke atas memungkinkan pengemudian tanpa pengemudi dalam kondisi tertentu, sementara kendaraan level L5 sepenuhnya dioperasikan oleh sistem pengemudian tanpa pengemudi dalam semua skenario.

Saat ini, hanya segelintir model produsen kendaraan penumpang komersial seperti Audi A8, Tesla, dan Cadillac yang dilengkapi dengan Sistem Bantuan Mengemudi Canggih (ADAS) L2 dan L3. Dengan peningkatan lebih lanjut pada sensor dan prosesor on-board, tahun 2020 menyaksikan kemunculan lebih banyak model L3. Kendaraan dengan sistem mengemudi otonom L4 dan L5 diharapkan pertama kali digunakan pada platform kendaraan komersial di kawasan industri tertutup. Namun, untuk mengemudi otonom tingkat tinggi pada platform kendaraan penumpang, diperlukan optimalisasi lebih lanjut dalam teknologi, kebijakan yang relevan, dan pembangunan infrastruktur. Diperkirakan kendaraan penumpang semacam itu tidak akan digunakan di jalan umum hingga tahun 2025.

Status AI Saat Ini

Seperti yang ditunjukkan pada Gambar 1.22, perkembangan AI telah melalui tiga tahap dan saat ini AI masih dalam tahap kecerdasan perseptual.



Gambar 1.22 Tiga tahap kecerdasan buatan

1.4 STRATEGI PENGEMBANGAN AI HUAWEI

Solusi AI Full-Stack untuk Semua Skenario

Pada kuartal pertama tahun 2020, kerangka kerja komputasi AI semua skenario Huawei, MindSpore, dirilis untuk komunitas sumber terbuka. Kemudian, Huawei merilis basis data mandiri GaussDB OLTP untuk komunitas sumber terbuka pada Juni 2020, dan merilis sistem operasi server untuk komunitas sumber terbuka pada 31 Desember 2020. Full-stack mengacu pada solusi full-stack yang mencakup chip, chip enable, kerangka kerja pelatihan dan penalaran, serta application enable.

All-scenario mengacu pada lingkungan penerapan semua skenario yang mencakup cloud publik, cloud privat, semua jenis komputasi tepi, terminal IoT, dan terminal konsumen. Sebagai fondasi solusi AI semua skenario tumpukan penuh Huawei, solusi komputasi kecerdasan buatan Atlas, yang berbasis pada prosesor AI Ascend, menyediakan produk dalam berbagai bentuk, termasuk modul, papan sirkuit, dan server untuk memenuhi permintaan daya komputasi semua skenario oleh pelanggan.

Arah AI Full-Stack Huawei

1. Platform pengembangan AI terpadu Huawei; ModelArts

ModelArts adalah platform pengembangan terpadu yang dirancang Huawei untuk para pengembang AI. Platform ini mendukung prapemrosesan data skala besar, pelabelan semi-otomatis, pelatihan terdistribusi, pembangunan model otomatis, dan penerapan model sesuai permintaan di end-to-end, edge, dan cloud, untuk membantu para pengembang membangun dan menerapkan model dengan cepat serta mengelola siklus hidup pengembangan AI secara menyeluruh. ModelArts memiliki karakteristik sebagai berikut.

(a) ***Pembelajaran otomatis***: Dengan fungsi pembelajaran otomatis, ModelArts dapat secara otomatis merancang model, menyesuaikan parameter, melatih, mengompresi, dan menerapkan model berdasarkan data berlabel, sehingga para pengembang tidak perlu memiliki pengalaman dalam pengodean atau pengembangan model. Pembelajaran otomatis ModelArts terutama diwujudkan melalui ModelArts Pro, sebuah kit pengembangan profesional yang dirancang untuk AI tingkat perusahaan. Berdasarkan algoritma canggih dan kemampuan pelatihan cepat HUAWEI CLOUD, ModelArts Pro menyediakan alur kerja dan model pra-instal untuk meningkatkan efisiensi dan mengurangi kesulitan pengembangan aplikasi AI oleh perusahaan. ModelArts Pro mendukung pengguna untuk menciptakan kembali alur kerja secara mandiri dan pengembangan, berbagi, serta peluncuran aplikasi secara real-time, yang kondusif untuk membangun ekosistem terbuka melalui upaya bersama, dan implementasi AI di industri yang bermanfaat bagi masyarakat umum. Toolkit ModelArts Pro mencakup perangkat pemrosesan bahasa alami, pengenalan teks, visi komputer, dll., yang memungkinkannya untuk merespons dengan cepat tuntutan berbagai industri dan skenario implementasi AI.

(b) ***Device-edge-cloud***: Device, edge, dan cloud masing-masing merujuk pada end-device, perangkat edge cerdas Huawei, dan Huawei CLOUD.

(c) ***Mendukung inferensi daring***: Inferensi daring adalah layanan daring (layanan web) yang menghasilkan prediksi real-time atas setiap permintaan.

(d) ***Mendukung inferensi batch***: inferensi batch bertujuan untuk menghasilkan sejumlah prediksi pada sejumlah data.

(e) ***Prosesor AI Ascend***: Prosesor AI Ascend adalah chip AI yang dirancang oleh Huawei dengan daya komputasi tinggi dan konsumsi daya rendah.

(f) ***Efisiensi persiapan data yang tinggi***: ModelArts memiliki kerangka kerja data AI bawaan, yang dapat meningkatkan efisiensi persiapan data melalui konvergensi pra-pelabelan otomatis dan pelabelan set data contoh keras.

(g) ***Waktu pelatihan yang lebih singkat***: ModelArts diinstal dengan kerangka kerja terdistribusi berkinerja tinggi yang dikembangkan sendiri oleh Huawei, MoXing, menggunakan teknologi inti termasuk paralelisme hibrida bertingkat, kompresi gradien, dan akselerasi konvolusi untuk mempercepat pelatihan model secara signifikan.

(h) ***ModelArts mendukung penerapan model sekali klik***: ModelArts mendukung penerapan model ke perangkat dan skenario end, edge, dan cloud hanya dengan sekali klik, yang dapat memenuhi berbagai persyaratan seperti konkurensi tinggi dan perangkat ringan secara bersamaan. (i) Manajemen proses penuh: ModelArts menyediakan manajemen alur kerja visual untuk data, pelatihan, model, dan inferensi (mencakup seluruh siklus pengembangan AI), serta memungkinkan pelatihan dimulai ulang otomatis setelah pemadaman listrik, perbandingan hasil pelatihan, dan manajemen model yang dapat dilacak.

(i) ***Pasar AI aktif***: ModelArts mendukung berbagi data dan model, yang dapat membantu perusahaan meningkatkan efisiensi aktivitas pengembangan AI internal dan juga memungkinkan pengembang mengubah pengetahuan mereka menjadi nilai tambah.

2. MindSpore, Kerangka Kerja Komputasi AI untuk Semua Skenario

Meskipun penerapan layanan AI pada perangkat, edge, dan skenario cloud berkembang pesat di era cerdas ini, teknologi AI masih menghadapi tantangan besar, termasuk standar teknologi yang tinggi, melonjaknya biaya pengembangan, dan siklus penerapan yang panjang. Tantangan-tantangan ini menjadi penghambat perkembangan ekosistem AI bagi pengembang di semua industri. Oleh karena itu, kerangka kerja komputasi AI untuk semua skenario, MindSpore, diperkenalkan.

Kerangka kerja ini dirancang berdasarkan tiga prinsip: ramah pengembangan, eksekusi yang efisien, dan penerapan yang fleksibel. Di dunia kerangka kerja pembelajaran mendalam saat ini, jika kita menyebut TensorFlow milik Google, MXNet milik Amazon, PyTorch milik Facebook, dan CNTK milik Microsoft sebagai "empat raksasa", maka MindSpore milik Huawei adalah pesaing terkuatnya. Berkat paralelisasi otomatis yang disediakan oleh MindSpore, para ilmuwan data senior dan insinyur algoritma yang berdedikasi pada pemodelan data dan pemecahan masalah dapat mengirimkan algoritma untuk mengunjungi puluhan atau bahkan ribuan node pemrosesan AI hanya dengan beberapa baris kode. MindSpore mendukung arsitektur dengan berbagai ukuran dan tipe, yang dapat diadaptasi untuk penerapan independen di semua skenario, prosesor AI Ascend, dan prosesor lain seperti GPU dan CPU.

3. CANN

Compute Architecture for Neural Networks (CANN) adalah lapisan pengaktifan chip yang dibangun Huawei untuk jaringan neural dalam dan prosesor AI Ascend. CANN terdiri dari empat modul fungsi utama berikut.

(a) Fusion Engine: Mesin fusi tingkat operator terutama digunakan untuk melakukan fusi operator guna mengurangi perpindahan memori antar operator dan meningkatkan kinerja hingga 50%.

(b) Pustaka operator CCE: Ini adalah pustaka operator umum Huawei yang dioptimalkan secara mendalam dan dapat memenuhi sebagian besar kebutuhan jaringan neural visi komputer dan NLP arus utama. Tentu saja, beberapa klien dan mitra mungkin meminta operator khusus karena alasan ketepatan waktu, privasi, atau riset. Hal ini memerlukan modul fungsi ketiga dari CANN.

(c) Tensor Boost Engine (TBE). Ini adalah alat pengembangan operator kustom yang efisien dan berkinerja tinggi, yang mengubah sumber daya perangkat keras menjadi antarmuka pemrograman aplikasi (API). Klien dapat dengan cepat membangun operator yang mereka butuhkan.

(d) Modul terakhir adalah kompiler di bagian bawah. Modul ini memberikan optimalisasi kinerja terbaik untuk mendukung prosesor Ascend AI di semua skenario.

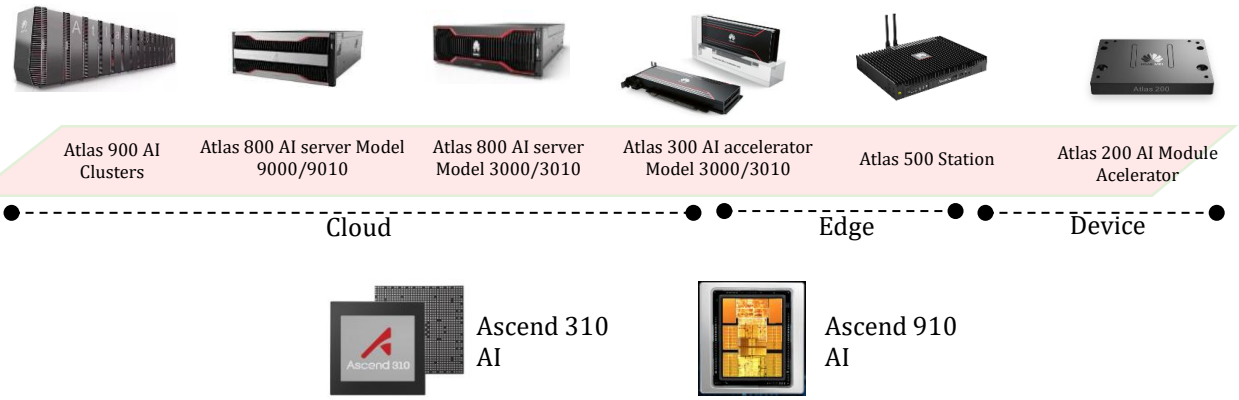
4. Prosesor AI Ascend

Mengingat meningkatnya permintaan AI, pasar prosesor AI saat ini dimonopoli oleh beberapa perusahaan, yang menyebabkan harga tinggi, siklus pasokan yang panjang, dan dukungan layanan lokal yang lemah. Permintaan AI di banyak industri belum terpenuhi secara efektif. Pada konferensi HUAWEI CONNECT di bulan Oktober 2018, Huawei merilis prosesor Ascend 310 dan Ascend 910 yang dikhususkan untuk skenario inferensi dan pelatihan AI. Arsitektur Da Vinci 3D Cube yang unik pada prosesor AI Ascend menjadikan seri ini cukup kompetitif dalam hal daya komputasi, efisiensi energi, dan skalabilitas.

Ascend 310 adalah sistem-pada-chip (SoC) AI yang sangat efisien yang dirancang untuk skenario inferensi cerdas tepi. SoC ini menggunakan chip 12 nm dan menghasilkan daya komputasi hingga 16 TOPS (operasi tera per detik) dengan konsumsi daya hanya 8 W, sangat cocok untuk skenario kecerdasan tepi yang membutuhkan konsumsi daya rendah. Ascend 910 saat ini merupakan chip tunggal dengan kepadatan komputasi tertinggi, yang cocok untuk pelatihan AI. Chip ini mengadopsi chip 7 nm dan menyediakan daya komputer hingga 512 TOPS dengan konsumsi daya maksimum 350 W.

5. Solusi komputasi kecerdasan buatan Atlas

Solusi komputasi kecerdasan buatan Huawei Atlas didasarkan pada prosesor AI Huawei Ascend untuk membangun solusi infrastruktur AI semua skenario untuk skenario perangkat, edge, dan cloud melalui beragam produk termasuk modul, papan sirkuit, stasiun edge, server dan kluster, dll., seperti yang ditunjukkan pada Gambar 1.23.



Gambar 1.23 Panorama platform komputasi kecerdasan buatan atlas

Sebagai bagian penting dari solusi AI semua skenario tumpukan penuh Huawei, Atlas meluncurkan produk inferensinya pada tahun 2019 dan menghadirkan solusi komputasi AI yang lengkap bagi industri dengan melengkapi produk pelatihan pada tahun 2020. Sementara itu, Huawei menciptakan platform kolaborasi perangkat-edge-cloud melalui penerapan semua skenario, yang memungkinkan AI untuk memberdayakan setiap tautan di industri.

1.5 KONTROVERSI AI

Bias Algoritmik

Bias algoritmik terutama disebabkan oleh data yang bias. Saat kita membuat keputusan dengan bantuan AI, algoritma dapat belajar mendiskriminasi sekelompok individu tertentu yang dilatih berdasarkan data yang dikumpulkan. Misalnya, algoritma dapat membuat keputusan yang rentan diskriminatif berdasarkan ras, jenis kelamin, atau faktor lainnya. Bahkan jika kita mengecualikan faktor-faktor seperti ras atau jenis kelamin dari data, algoritma tetap dapat membuat keputusan diskriminatif berdasarkan informasi pribadi seperti nama atau alamat seseorang. Berikut contohnya. Jika Anda mencari dengan nama yang terdengar seperti orang Afrika-Amerika, Anda mungkin mendapatkan iklan untuk alat penyelidikan catatan kriminal, yang kemungkinan besar tidak akan terjadi jika Anda mencari dengan gaya nama lain. Pengiklan daring cenderung menayangkan iklan produk dengan harga lebih rendah kepada pemirsa perempuan. Aplikasi gambar Google pernah keliru menandai foto orang kulit hitam sebagai "gorila".

Masalah Privasi

Saat ini, algoritma AI yang ada sepenuhnya berbasis data, karena pelatihan model membutuhkan data yang sangat besar. Meskipun menikmati kemudahan yang dibawa oleh AI, manusia juga terancam oleh risiko kebocoran privasi. Misalnya, sejumlah besar data pengguna yang dikumpulkan oleh beberapa perusahaan teknologi dapat menempatkan kita pada risiko terekspos nya kehidupan sehari-hari kita secara penuh jika data ini bocor. Saat pengguna daring, secara teknis perusahaan teknologi dapat merekam setiap klik, setiap pengguliran halaman, waktu menonton konten apa pun, dan riwayat penelusuran pengguna. Perusahaan teknologi ini juga dapat mengetahui lokasi pengguna, tempat yang telah mereka kunjungi, apa

yang telah mereka lakukan, serta latar belakang pendidikan, daya beli, preferensi, dan privasi pribadi lainnya berdasarkan catatan perjalanan dan pembelian pengguna setiap hari.

Kontradiksi Antara Teknologi dan Etika

Seiring perkembangan visi komputer, semakin sulit untuk menilai kredibilitas gambar. Orang dapat menghasilkan gambar palsu atau hasil manipulasi melalui prosesor gambar (misalnya, Photoshop, PS), jaringan adversarial generatif (GAN), dan teknik lainnya, sehingga sangat sulit untuk membedakan apakah gambar tersebut palsu atau asli. Mari kita ambil GAN sebagai contoh. Konsep ini diperkenalkan oleh peneliti pembelajaran mesin, Ian Goodfellow, pada tahun 2014. Dalam namanya, "G" adalah singkatan dari "generative", yang dikutip di sini untuk menunjukkan bahwa model tersebut menghasilkan informasi seperti gambar, alih-alih nilai prediksi yang terkait dengan data masukan. Dan "AN" adalah singkatan dari "adversarial network", karena model tersebut menggunakan dua kelompok jaringan saraf yang saling bersaing seperti dalam permainan kucing-kucingan, atau seperti kasir yang melawan pemalsu uang kertas: pemalsu mencoba menipu kasir agar percaya bahwa ia memegang uang asli, dan kasir mencoba mengidentifikasi keasliannya.

Akankah Semua Orang Menganggur?

Sepanjang perjalanan perkembangan manusia, manusia selalu mencari cara untuk meningkatkan efisiensi, yaitu, memanen lebih banyak dengan sumber daya yang lebih sedikit. Kita menggunakan batu tajam untuk berburu dan mengumpulkan makanan dengan lebih efisien, dan menciptakan mesin uap untuk mengurangi ketergantungan pada kuda. Di era AI, AI akan menggantikan pekerjaan dengan tingkat repetisi tinggi, kreativitas rendah, dan interaksi sosial yang jarang, sementara pekerjaan dengan kreativitas tinggi tidak akan mudah tergantikan.

1.6 TREN PENGEMBANGAN AI

1. Kerangka Kerja Pengembangan yang Lebih Mudah

Semua kerangka kerja pengembangan AI berevolusi menjadi lebih sederhana dalam pengoperasian namun memiliki fungsi yang sangat beragam. Ambang batas untuk pengembangan AI terus diturunkan.

2. Algoritma dan Model dengan Performa yang Lebih Baik

Dalam visi komputer, GAN mampu menghasilkan gambar berkualitas tinggi yang tidak dapat dibedakan oleh mata manusia. Algoritma terkait GAN telah mulai diterapkan pada tugas-tugas terkait visi lainnya, seperti segmentasi semantik, pengenalan wajah, sintesis video, dan pengelompokan tanpa pengawasan. Dalam pemrosesan bahasa alami, terobosan besar telah dicapai dalam model pra-pelatihan berbasis Transformer. Model-model relevan seperti BERT, GPT, dan XLNet telah mulai diterapkan secara luas pada skenario industri. Dalam pembelajaran penguatan, AlphaStar dari DeepMind mengalahkan pemain manusia teratas dalam permainan StarCraft II.

3. Model Dalam yang Lebih Kecil

Model dengan kinerja yang lebih baik seringkali disertai dengan parameter yang lebih besar, dan model yang lebih besar harus menghadapi masalah efisiensi operasional selama

implementasi industri. Oleh karena itu, semakin banyak teknik kompresi model telah diusulkan untuk lebih mengurangi ukuran dan parameter model, mempercepat kecepatan inferensi, dan memenuhi persyaratan aplikasi industri sekaligus memastikan kinerjanya.

4. Pengembangan daya komputasi menyeluruh di perangkat, edge, dan cloud

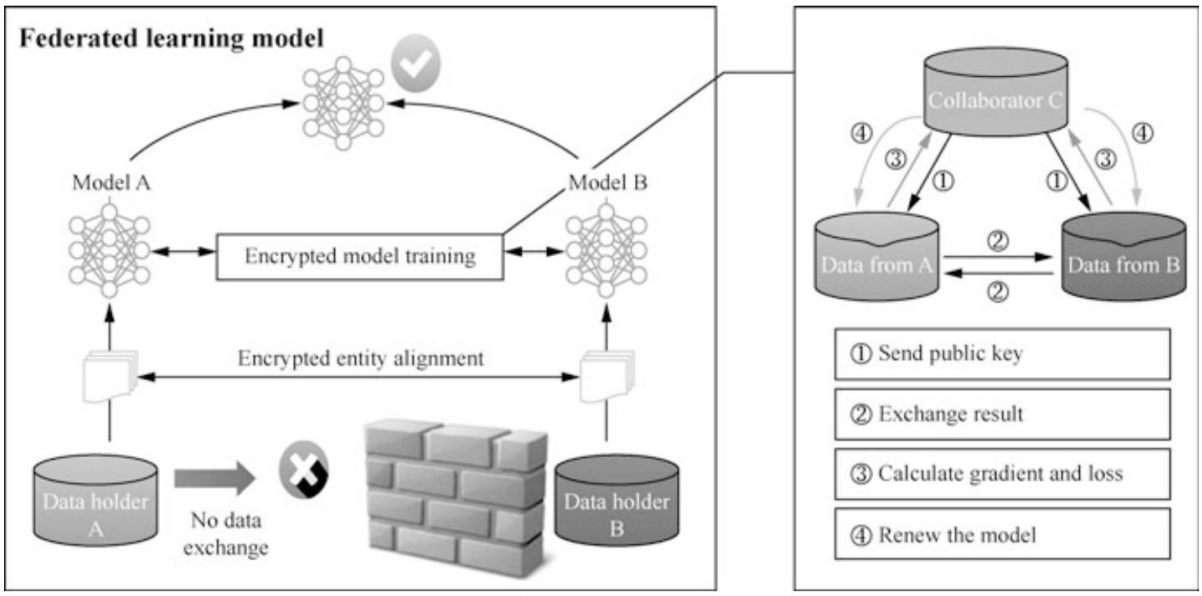
Penerapan chip kecerdasan buatan pada cloud, perangkat edge, dan terminal seluler semakin meluas, semakin memecahkan masalah daya komputasi untuk AI.

5. Layanan Data Dasar AI yang Lebih Canggih

Seiring dengan semakin matangnya layanan data dasar AI, kita akan melihat semakin banyak platform dan alat pelabelan data terkait yang diperkenalkan ke pasar.

6. Berbagi Data yang Lebih Aman

Dengan premis memastikan privasi dan keamanan data, pembelajaran terfederasi memanfaatkan berbagai sumber data untuk melatih model secara kolaboratif, sehingga dapat mengatasi hambatan data seperti yang ditunjukkan pada Gambar 1.24.



Gambar 1.24 Pembelajaran Terfederasi

Laporan prospek industri global Huawei, GIV 2020 (disingkat GIV 2025), mencantumkan 10 tren utama perkembangan teknologi cerdas di masa depan.

1. Popularitas robot cerdas: Huawei memprediksi bahwa pada tahun 2025, 14% keluarga di seluruh dunia akan memiliki robot pintar, yang akan memainkan peran penting dalam kehidupan sehari-hari masyarakat.
2. Popularitas AR/VR: Laporan tersebut memprediksi bahwa persentase perusahaan yang menggunakan teknologi VR/AR akan mencapai 10% di masa mendatang. Penerapan teknologi termasuk realitas virtual akan membawa semangat dan peluang bagi industri seperti presentasi komersial dan hiburan audio-visual.
3. Penerapan AI di berbagai bidang: Diprediksi bahwa 97% perusahaan besar akan mengadopsi teknologi AI, terutama dicontohkan oleh penggunaan kecerdasan bicara, pengenalan gambar, pengenalan wajah, interaksi manusia-komputer, dan sebagainya.

4. Mempopulerkan aplikasi big data: Perusahaan akan memanfaatkan 86% data yang mereka hasilkan secara efisien. Analisis dan pemrosesan big data akan menghemat waktu dan meningkatkan efisiensi perusahaan.
5. Melemahnya mesin pencari: Di masa depan, 90% orang akan memiliki asisten pribadi pintar, yang berarti kemungkinan Anda mencari sesuatu dari portal pencarian akan sangat berkurang.
6. Mempopulerkan Internet Kendaraan: Teknologi seluler vehicle-to-everything (C-V2X) akan dipasang di 15% kendaraan di dunia. Kendaraan dan mobil pintar berbasis internet akan dipopulerkan secara substansial, memberikan pengalaman berkendara yang lebih aman dan andal.
7. Mempopulerkan robot industri: Robot industri akan bekerja berdampingan dengan manusia di bidang manufaktur, dengan 103 robot untuk setiap 10.000 karyawan. Tugas-tugas berbahaya, berpresisi tinggi, dan berintensitas tinggi akan dibantu atau diselesaikan oleh robot industri secara mandiri.
8. Popularisasi teknologi dan aplikasi cloud: Tingkat penggunaan aplikasi berbasis cloud akan mencapai 85%. Mayoritas aplikasi dan kolaborasi program akan dilakukan di cloud.
9. Popularisasi 5G: Lima puluh delapan persen populasi dunia akan menikmati layanan 5G. Kita mungkin mengantisipasi revolusi industri komunikasi di masa depan, ketika teknologi dan kecepatan komunikasi akan sangat maju.
10. Popularisasi ekonomi digital dan big data: Jumlah data penyimpanan global yang dihasilkan setiap tahun akan mencapai 180 ZB. Ekonomi digital dan teknologi blockchain akan dipadukan secara luas dengan internet.

1.7 KESIMPULAN

Bab ini memperkenalkan konsep dasar, sejarah perkembangan, dan latar belakang penerapan AI. Dengan membaca bab ini, pembaca dapat memahami bahwa, sebagai ilmu interdisipliner, penerapan dan pengembangan kecerdasan buatan tidak akan tercapai tanpa dukungan disiplin ilmu lain. Implementasi fisiknya bergantung pada perangkat keras berskala besar, dan penerapan lapisan atasnya bergantung pada desain perangkat lunak dan metode implementasi. Sebagai pembelajar, pembaca diharapkan memahami batasan penerapan kecerdasan buatan sehingga dapat memperbaiki dan meningkatkan diri atas dasar ini.

Latihan Soal

1. Terdapat berbagai interpretasi kecerdasan buatan dalam berbagai konteks. Mohon jelaskan lebih lanjut tentang kecerdasan buatan menurut Anda.
2. Kecerdasan buatan, pembelajaran mesin, dan pembelajaran mendalam adalah tiga konsep yang sering disebutkan bersamaan. Apa hubungan di antara ketiganya? Apa persamaan dan perbedaan antara ketiga istilah tersebut?

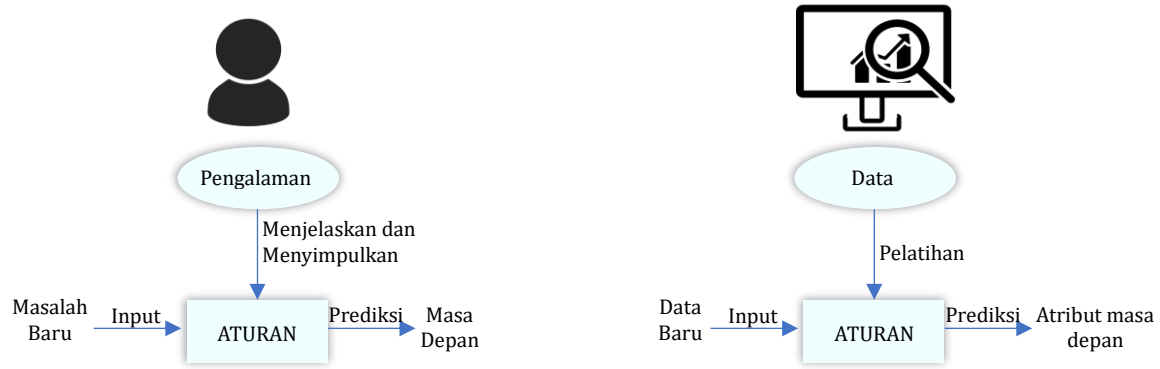
3. Setelah membaca skenario penerapan kecerdasan buatan dalam bab ini, mohon jelaskan secara rinci bidang penerapan AI dan skenarionya dalam kehidupan nyata berdasarkan pengalaman hidup Anda sendiri.
4. CANN adalah lapisan pengaktifan chip yang diperkenalkan Huawei untuk jaringan saraf dalam dan prosesor AI Ascend. Mohon jelaskan secara singkat empat modul utama CANN.
5. Berdasarkan pengetahuan dan pemahaman Anda saat ini, mohon jelaskan lebih lanjut tentang tren perkembangan kecerdasan buatan di masa depan menurut pandangan Anda.

BAB 2
PEMBELAJARAN MESIN (MACHINE LEARNING)

Pembelajaran mesin saat ini menjadi pusat penelitian utama dalam industri AI, yang mencakup berbagai disiplin ilmu seperti teori probabilitas, statistika, dan optimasi konveks. Bab ini pertama-tama memperkenalkan definisi "pembelajaran" dalam algoritma pembelajaran dan proses pembelajaran mesin. Berdasarkan hal ini, bab ini menawarkan beberapa algoritma pembelajaran mesin yang umum digunakan. Pembaca kami akan mempelajari beberapa konsep kunci seperti hiperparameter, penurunan gradien, dan validasi silang.

2.1 PENGANTAR PEMBELAJARAN MESIN

Pembelajaran mesin (termasuk cabangnya, pembelajaran mendalam) adalah studi tentang "algoritma pembelajaran". Yang disebut "pembelajaran" di sini mengacu pada situasi di mana kinerja program komputer yang diukur dengan metrik kinerja P pada tugas tertentu T meningkat dengan sendirinya seiring pengalaman E, maka kita menyebut program komputer ini belajar dari pengalaman E. Misalnya, mengidentifikasi email sampah adalah tugas T. Kita dapat menyelesaikan tugas-tugas tersebut dengan mudah, karena kita memiliki banyak pengalaman dalam melakukannya dalam kehidupan sehari-hari. Pengalaman-pengalaman ini dapat berasal dari email harian, pesan spam, atau bahkan iklan di TV.



(a) Pembelajaran manusia (a) Pembelajaran Mesin

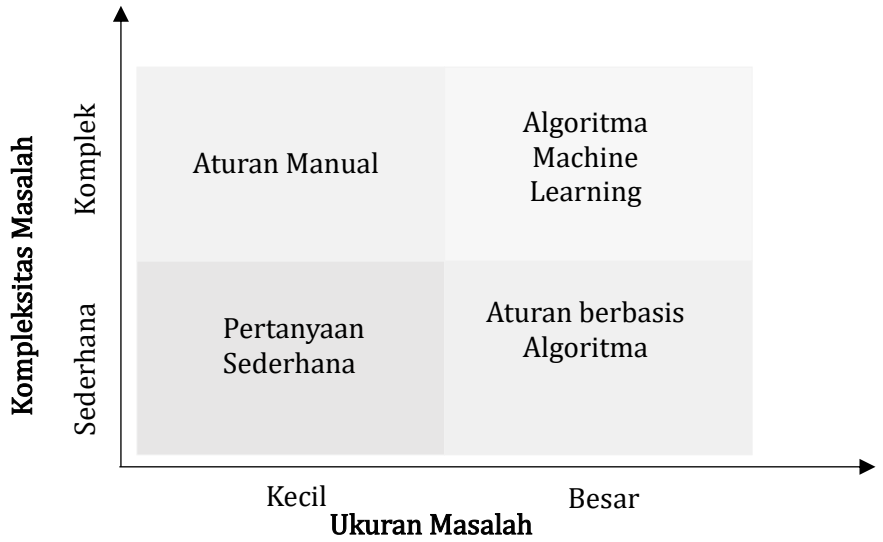
Gambar 2.1 Mode Pembelajaran

Kita dapat meringkas dan menyimpulkan dari pengalaman-pengalaman ini bahwa surel dari pengguna tak dikenal yang berisi kata-kata seperti "diskon" dan "risiko nol" lebih mungkin menjadi spam. Merujuk pada pengetahuan yang dipelajari, kita dapat membedakan apakah surel yang belum pernah dibaca sebelumnya adalah spam, seperti yang ditunjukkan pada Gambar 2.1a. Jadi, dapatkah kita menulis program komputer untuk mensimulasikan proses di atas? Seperti yang ditunjukkan pada Gambar 2.1b, kita dapat menyiapkan sejumlah surel, dan memilih surel sampah dengan tangan, sebagai pengalaman E untuk program ini. Namun,

program tidak dapat secara otomatis meringkas pengalaman-pengalaman ini. Pada saat ini, perlu untuk melatih program melalui algoritma pembelajaran mesin. Program komputer yang telah dilatih disebut model, dan secara umum semakin besar jumlah surel yang digunakan untuk pelatihan, semakin baik model tersebut dapat dilatih, dan semakin besar nilai metrik kinerja P.

Identifikasi spam sangat sulit dicapai melalui metode pemrograman tradisional. Secara teoritis, kita seharusnya dapat menemukan serangkaian aturan yang sesuai dengan fitur semua email sampah, alih-alih email biasa. Metode penggunaan pemrograman eksplisit untuk memecahkan masalah ini dikenal sebagai pendekatan berbasis aturan. Namun, dalam praktiknya, hampir mustahil untuk menemukan serangkaian aturan semacam itu. Oleh karena itu, untuk memecahkan masalah ini, pembelajaran mesin mengadopsi metode berbasis statistik. Pada dasarnya, kita dapat mengklaim bahwa pembelajaran mesin adalah metode yang memungkinkan mesin mempelajari aturan secara otomatis melalui sampel. Dibandingkan dengan metode berbasis aturan, pembelajaran mesin dapat mempelajari aturan yang lebih kompleks dan aturan yang sulit dijelaskan, sehingga dapat menangani tugas yang lebih rumit.

Pembelajaran mesin sangat adaptif dan dapat memecahkan banyak masalah di bidang AI, tetapi ini tidak berarti bahwa pembelajaran mesin akan selalu digunakan sebagai pendahuluan dalam semua kasus. Seperti yang ditunjukkan pada Gambar 2.2, pembelajaran mesin cocok untuk masalah yang memerlukan solusi kompleks atau melibatkan sejumlah besar data, tetapi distribusi probabilitas datanya tidak diketahui.



Gambar 2.2 Skenario aplikasi pembelajaran mesin

Pembelajaran mesin tentu dapat memecahkan masalah dalam kasus lain, tetapi biayanya seringkali lebih tinggi daripada metode tradisional. Ambil kuadran kedua yang ditunjukkan pada Gambar 2.2 sebagai contoh. Jika ukuran masalah cukup kecil untuk diselesaikan dengan

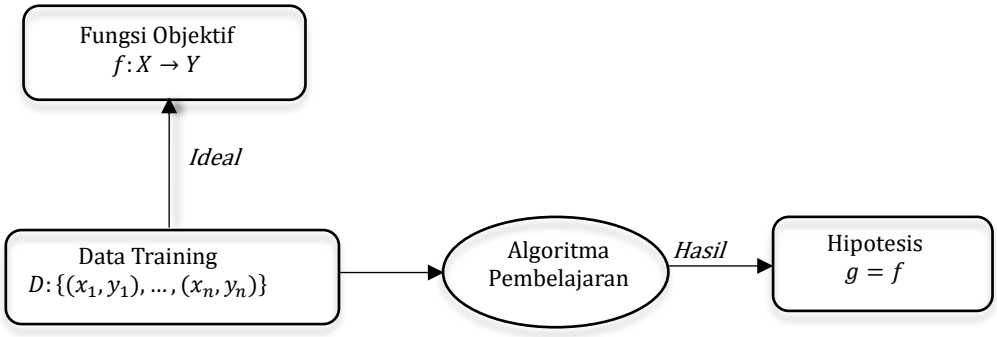
aturan buatan, maka tidak perlu menggunakan algoritma pembelajaran mesin. Secara umum, terdapat dua skenario aplikasi utama untuk pembelajaran mesin.

1. Aturannya cukup rumit atau tidak dapat dijelaskan, seperti pengenalan wajah dan pengenalan suara.
2. Distribusi data itu sendiri berubah seiring waktu, dan program perlu terus-menerus diadaptasi ulang, seperti memprediksi tren penjualan komoditas.

Pemahaman Rasional Algoritma Pembelajaran Mesin

Sifat algoritma pembelajaran mesin adalah pencocokan fungsi. Misalkan f adalah fungsi objektif, tujuan algoritma pembelajaran mesin adalah untuk memberikan fungsi hipotetis g yang membuat $g(x)$ dan $f(x)$ sedekat mungkin dengan input x dalam domain apa pun yang ditentukan. Contoh sederhananya adalah estimasi kepadatan probabilitas dalam statistik. Berdasarkan hukum bilangan besar, tinggi badan semua penduduk Tiongkok harus mengikuti distribusi normal. Meskipun fungsi kepadatan probabilitas f dari distribusi normal ini tidak diketahui, kita dapat menaksir rerata dan varians distribusi dengan mengambil sampel, lalu menaksir f.

Hubungan antara fungsi hipotesis dan fungsi objektif ditunjukkan pada Gambar 2.3. Untuk suatu tugas tertentu, kita dapat mengumpulkan sejumlah besar data pelatihan. Data ini harus sesuai dengan fungsi objektif f yang diberikan; jika tidak, mempelajari tugas semacam itu akan sia-sia. Berikutnya, algoritma pembelajaran dapat menganalisis data pelatihan ini dan memberikan fungsi hipotetis g yang semirip mungkin dengan fungsi objektif f.



Gambar 2.3 Hubungan antara fungsi hipotesis dan fungsi objektif

Oleh karena itu, keluaran algoritma pembelajaran tidak selalu sempurna dan tidak dapat sepenuhnya konsisten dengan fungsi objektif. Namun, dengan perluasan data pelatihan, tingkat aproksimasi fungsi hipotesis g terhadap fungsi objektif f secara bertahap ditingkatkan, dan akhirnya tingkat akurasi yang memuaskan dapat dicapai dalam pembelajaran mesin.

Perlu disebutkan bahwa fungsi objektif f terkadang sangat abstrak. Untuk tugas klasifikasi gambar klasik, fungsi objektif berarti pemetaan dari himpunan gambar ke himpunan kategori. Agar program dapat memproses informasi logis seperti gambar dan kelas, perlu menggunakan metode pengkodean tertentu untuk memetakan gambar atau kategori satu per satu ke dalam skalar, vektor, atau matriks. Misalnya, Anda dapat memberi nomor setiap

kategori dari 0, sehingga memetakan kelas ke skalar. Anda juga dapat menggunakan vektor one-hot yang berbeda untuk merepresentasikan kelas yang berbeda, yang disebut pengkodean one-hot. Pengkodean gambar sedikit lebih rumit, dan umumnya direpresentasikan oleh matriks tiga dimensi. Dengan metode pengkodean ini, kita dapat melihat domain definisi fungsi objektif f sebagai kumpulan matriks tiga dimensi dan rentangnya sebagai kumpulan deret nomor seri untuk kelas. Meskipun proses pengkodean bukan bagian dari algoritma pembelajaran, dalam beberapa kasus, pilihan metode pengkodean juga akan memengaruhi efisiensi algoritma pembelajaran.

Masalah Utama yang Diselesaikan oleh Pembelajaran Mesin

Pembelajaran mesin dapat menangani berbagai jenis masalah, termasuk yang paling umum seperti klasifikasi, regresi, dan pengelompokan. Klasifikasi dan regresi adalah dua jenis utama masalah prediksi, yang mencakup 80–90% dari semua masalah. Perbedaan utamanya adalah keluaran klasifikasi berupa nomor seri diskrit kelas (umumnya disebut "label" dalam pembelajaran mesin), sedangkan keluaran regresi berupa nilai kontinu. Masalah klasifikasi mengharuskan program untuk menunjukkan kelas mana dari k kelas yang menjadi milik masukan. Untuk menyelesaikan masalah ini, algoritma pembelajaran mesin biasanya menghasilkan pemetaan dari domain D ke label kategori $\{1, 2, \ldots, k\}$. Tugas klasifikasi citra merupakan masalah klasifikasi yang umum.

Dalam masalah regresi, program perlu memprediksi nilai keluaran untuk masukan yang diberikan. Keluaran algoritma pembelajaran mesin biasanya berupa pemetaan dari domain D ke domain bilangan riil R. Misalnya, memprediksi jumlah klaim tertanggung (digunakan untuk menetapkan premi asuransi), atau memprediksi harga sekuritas di masa mendatang, semuanya merupakan kasus yang relevan. Faktanya, masalah klasifikasi juga dapat disederhanakan menjadi masalah regresi. Dengan memprediksi probabilitas citra yang termasuk dalam setiap kelas, pembelajaran mesin dapat memperoleh hasil klasifikasi.

Masalah pengelompokan perlu membagi data ke dalam beberapa kategori berdasarkan kesamaan inheren data. Tidak seperti masalah klasifikasi, kumpulan data dari masalah pengelompokan tidak berisi label yang diberi label secara manual. Algoritma pengelompokan membuat data sedekat mungkin satu sama lain dalam kelas, sementara kesamaan data antar kelas relatif kecil, sehingga memungkinkan penerapan klasifikasi. Algoritma pengelompokan dapat digunakan dalam skenario seperti pengambilan citra, pembuatan potret pengguna, dan lain-lain.

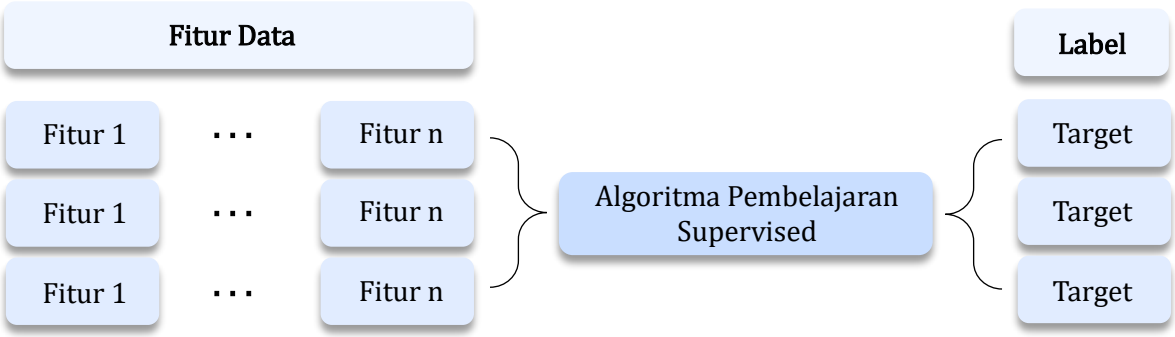
2.2 JENIS-JENIS PEMBELAJARAN MESIN

Berdasarkan apakah set data pelatihan berisi label yang diberi tag manual, pembelajaran mesin secara umum dapat dibagi menjadi dua jenis pembelajaran terawasi dan pembelajaran tanpa pengawasan. Terkadang, untuk membedakannya dari pembelajaran tanpa pengawasan, pembelajaran terawasi juga disebut pembelajaran di bawah pengawasan. Jika beberapa data dalam set data berisi label dan sebagian besar data tidak, maka algoritma pembelajaran ini disebut pembelajaran semi-terawasi. Pembelajaran penguatan terutama

berfokus pada masalah pengambilan keputusan multi-langkah, dan secara otomatis mengumpulkan data untuk pembelajaran dalam interaksi dengan lingkungan.

Pembelajaran Terawasi

Secara umum, pembelajaran terawasi memungkinkan komputer untuk membandingkan jawaban standar ketika dilatih untuk menjawab pertanyaan pilihan ganda. Komputer berusaha sebaik mungkin untuk menyesuaikan parameter modelnya, berharap bahwa jawaban yang disimpulkan sekonsisten mungkin dengan jawaban standar, dan akhirnya mempelajari cara menyelesaikan pertanyaan tersebut. Dengan menggunakan sampel label yang diketahui, pembelajaran terawasi dapat melatih model optimal yang memenuhi kinerja yang dibutuhkan. Dengan menggunakan model ini, setiap masukan dapat dipetakan ke keluaran yang sesuai, sehingga dapat memprediksi data yang tidak diketahui.



Gambar 2.4 Algoritma pembelajaran terbimbing

Gambar 2.4 menunjukkan algoritma pembelajaran terawasi yang sangat disederhanakan. Fitur dalam grafik dapat dipahami sebagai item data. Meskipun interpretasi ini tidak memadai sampai batas tertentu, hal itu tidak akan memengaruhi deskripsi kita tentang algoritma pembelajaran terawasi ini. Algoritma pembelajaran terawasi mengambil fitur sebagai masukan dan nilai target yang diprediksi sebagai keluaran. Gambar 2.5 menunjukkan contoh praktis. Dalam contoh ini, kami berharap dapat membuat prediksi menyeluruh tentang apakah pengguna menikmati latihan dengan mempertimbangkan kondisi cuaca. Setiap baris dalam tabel merupakan serangkaian contoh pelatihan yang mencatat karakteristik cuaca pada hari tertentu dan tingkat kenikmatan pengguna terhadap latihan. Algoritma serupa dapat diterapkan pada skenario lain seperti rekomendasi produk.

Fitur			Target
Cuaca	Temperatur	Kecepatan Angin	Menikmati Olahraga
Cerah	Hangat	Kuat	Ya
Hujan	Dingin	Sedang	Tidak
Cerah	Dingin	lemah	Ya

Gambar 2.5 Data sampel

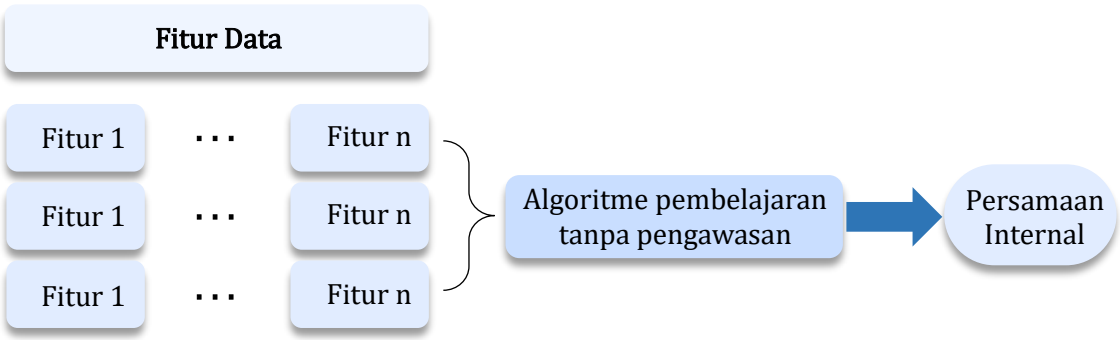
Masukan (fitur) dan keluaran (target) dari algoritma pembelajaran terawasi dapat berupa kontinu atau diskrit. Ketika nilai variabel target kontinu, keluaran dari algoritma pembelajaran terawasi disebut model regresi. Model regresi mencerminkan fitur nilai atribut dalam kumpulan data sampel, dan menyatakan hubungan pemetaan sampel melalui fungsi untuk menunjukkan ketergantungan antara nilai atribut. Nilai atribut yang disebutkan di sini mencakup fitur dan target.

Model regresi banyak digunakan dalam peramalan deret waktu. Misalnya, berapa banyak uang yang dapat Anda hasilkan dari saham minggu depan? Berapa suhu besok? Dan seterusnya. Dengan demikian, ketika variabel target mengambil nilai diskrit, keluaran dari algoritma pembelajaran terawasi disebut model klasifikasi. Melalui model klasifikasi, sampel dalam kumpulan data sampel dapat dipetakan ke kelas yang diberikan. Seperti apakah akan ada kemacetan lalu lintas di jalan raya selama jam sibuk pagi hari besok? Mana yang lebih menarik bagi pelanggan, voucher lima yuan atau diskon 25%? Dll.

Meskipun rentang model regresi dapat berupa himpunan tak terbatas, keluaran model klasifikasi seringkali terbatas. Hal ini karena ukuran dataset tidak dapat bertambah tanpa batas, dan jumlah kelas dalam dataset harus semaksimal mungkin sama dengan jumlah contoh pelatihan. Oleh karena itu, jumlah kelas tidaklah tak terbatas. Saat melatih model klasifikasi, himpunan kelas L yang ditentukan secara artifisial seringkali diperlukan agar model dapat memilih keluaran kategori. Ukuran himpunan L umumnya dicatat sebagai K, yang merupakan jumlah kelas yang memungkinkan.

Pembelajaran Tanpa Supervisi

Dibandingkan dengan pembelajaran tersupervisi, pembelajaran tanpa supervisi seperti membiarkan komputer mengerjakan soal pilihan ganda tanpa memberi tahu jawaban yang benar. Dalam hal ini, sulit bagi komputer untuk memastikan jawaban yang benar. Namun, dengan menganalisis hubungan antara pertanyaan-pertanyaan ini, komputer dapat mengklasifikasikan pertanyaan-pertanyaan tersebut sehingga soal pilihan ganda di setiap kelas memiliki jawaban yang sama.

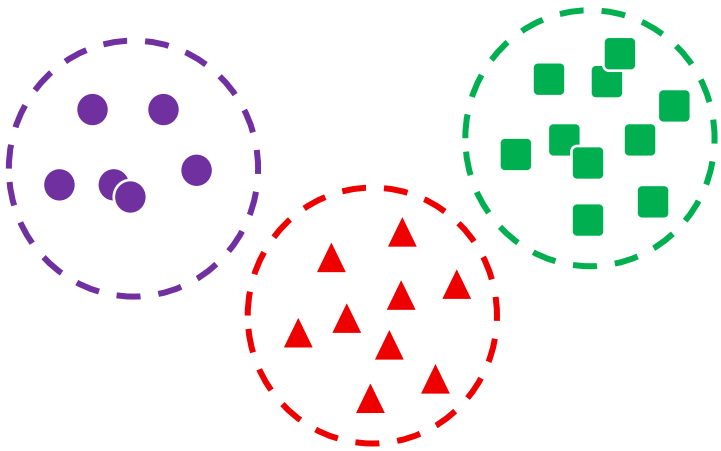


Gambar 2.6 Algoritma pembelajaran tanpa pengawasan

Algoritma pembelajaran tanpa pengawasan tidak memerlukan pelabelan sampel, melainkan pemodelan langsung dari dataset masukan, seperti yang ditunjukkan pada Gambar 2.6. Algoritma pengelompokan merupakan algoritma pembelajaran tanpa pengawasan yang

umum, yang dapat diringkas dengan pepatah: "burung yang sejenis akan berkumpul bersama". Algoritma ini hanya menggabungkan sampel-sampel dengan tingkat kemiripan yang tinggi. Untuk sampel masukan baru, algoritma ini hanya perlu menghitung kemiripannya dengan sampel yang sudah ada, lalu mengklasifikasikannya berdasarkan tingkat kemiripannya.

Para ahli biologi telah lama menggunakan konsep pengelompokan untuk mempelajari hubungan antarspesies. Seperti yang ditunjukkan pada Gambar 2.7, setelah mengklasifikasikan bunga iris berdasarkan ukuran sepal dan petal, iris telah dibagi menjadi tiga kategori. Melalui model pengelompokan, sampel-sampel dalam dataset sampel dapat dibagi menjadi beberapa kategori, sehingga tingkat kemiripan sampel dalam kategori yang sama relatif lebih tinggi. Skenario penerapan model pengelompokan meliputi jenis audiens yang suka menonton film dengan subjek yang sama, komponen mana yang rusak serupa, dan sebagainya.



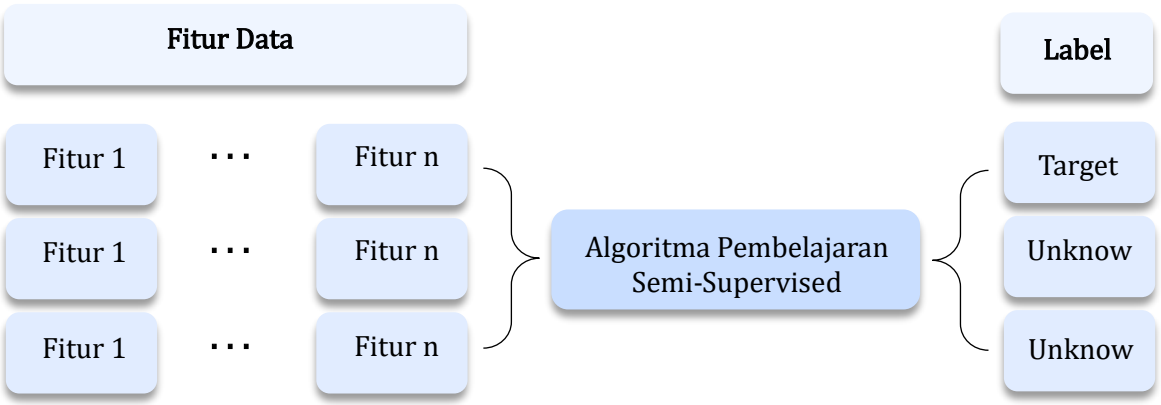
Gambar 2.7 Contoh algoritma pengelompokan

Pembelajaran Semi-Supervised

Pembelajaran semi-supervised merupakan kombinasi dari pembelajaran tersupervised dan pembelajaran tak tersupervised. Algoritma ini berupaya memungkinkan pembelajar untuk secara otomatis memanfaatkan sejumlah besar data tak berlabel untuk membantu pembelajaran sejumlah kecil data berlabel. Algoritma pembelajaran tersupervised tradisional perlu belajar dari sejumlah besar sampel pelatihan berlabel untuk membangun model guna memprediksi label sampel baru. Misalnya, dalam tugas klasifikasi, label menunjukkan kelas sampel. Dan dalam tugas regresi, label menunjukkan keluaran sampel yang bernilai riil.

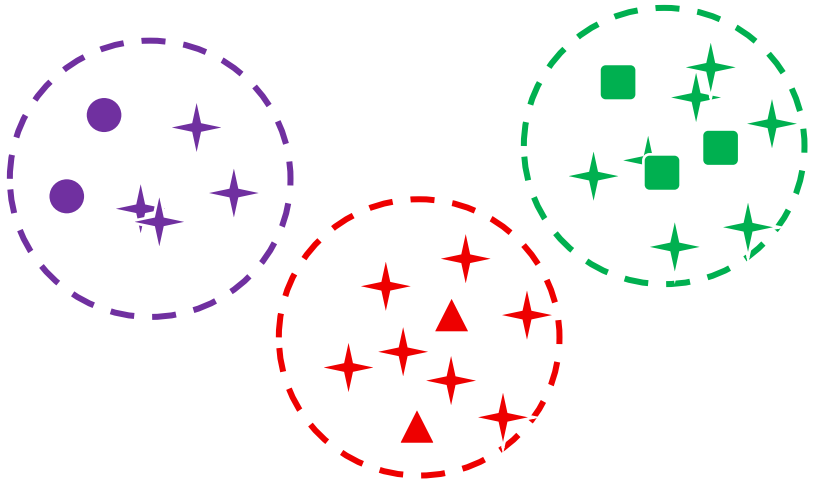
Dengan pesatnya perkembangan kemampuan manusia untuk mengumpulkan dan menyimpan data, dalam banyak tugas praktis, sangat mudah untuk memperoleh sejumlah besar data tak berlabel, dan pelabelannya seringkali membutuhkan banyak upaya dan sumber daya. Misalnya, untuk rekomendasi halaman web, pengguna diharuskan menandai halaman web yang mereka minati. Namun, hanya sedikit pengguna yang bersedia meluangkan banyak waktu untuk menandai, sehingga halaman web dengan informasi yang ditandai menjadi

terbatas. Namun, ada banyak sekali halaman web yang tidak ditandai, atau ditandai, yang dapat digunakan sebagai data tak bertanda.



Gambar 2.8 Algoritma pembelajaran semi-supervised

Seperti ditunjukkan pada Gambar 2.8, pembelajaran semi-supervised tidak memerlukan pelabelan manual pada semua sampel seperti pembelajaran tersupervised, juga tidak sepenuhnya independen dari target seperti pembelajaran tak tersupervised. Dalam kumpulan data pembelajaran semi-supervised, secara umum, hanya ada beberapa sampel yang diberi label. Mengambil masalah klasifikasi iris yang disajikan pada Gambar 2.7 sebagai contoh, sejumlah kecil informasi tersupervised ditambahkan ke kumpulan data, seperti yang ditunjukkan pada Gambar 2.9.



Gambar 2.9 Dataset bunga iris dengan informasi yang diawasi

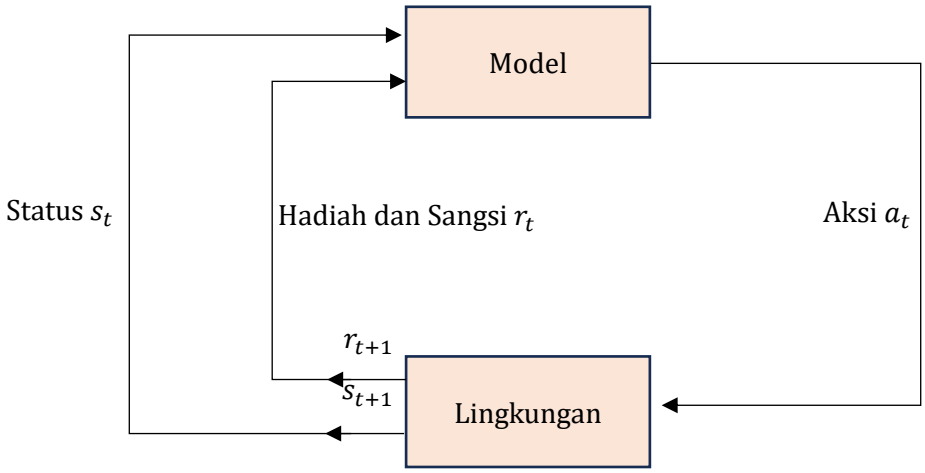
Mari kita asumsikan lingkaran mewakili sampel Setosa, segitiga mewakili sampel Versicolor, persegi mewakili sampel Virginica, dan bintang mewakili sampel yang tidak diketahui. Algoritma pengelompokan telah diperkenalkan dalam pembelajaran tak tersupervised, misalkan keluarannya ditunjukkan oleh lingkaran putus-putus pada Gambar 2.9. Menghitung lingkaran termasuk jumlah sampel tertinggi yang diketahui dan kemudian kelas ini dapat digunakan sebagai kelas untuk klaster ini. Lebih spesifiknya, klaster kiri atas milik Setosa, sedangkan klaster kanan atas jelas milik Virginica. Dengan menggabungkan algoritma tanpa

pengawasan dan informasi yang diawasi, algoritma semi-supervised dapat mencapai akurasi yang lebih tinggi dengan biaya tenaga kerja yang lebih rendah.

Pembelajaran Penguatan

Pembelajaran penguatan terutama digunakan untuk memecahkan masalah pengambilan keputusan multi-langkah, seperti permainan Go, permainan video, dan navigasi visual. Tidak seperti masalah yang dipelajari oleh pembelajaran terawasi dan pembelajaran tanpa pengawasan, masalah-masalah ini seringkali sulit untuk menemukan jawaban yang akurat. Mengambil Go sebagai contoh, dibutuhkan sekitar 10.170 operasi untuk menyelesaikan hasil permainan (hanya ada 1080 atom di alam semesta). Jadi, untuk situasi tertentu dan umum, sulit untuk menemukan langkah yang sempurna.

Karakteristik lain dari masalah keputusan multi-langkah adalah mudahnya mendefinisikan fungsi imbalan untuk mengevaluasi apakah tugas telah selesai. Fungsi imbalan Go dapat didefinisikan sebagai apakah akan memenangkan permainan; fungsi imbalan permainan elektronik dapat didefinisikan sebagai skor. Tujuan pembelajaran penguatan adalah untuk menemukan strategi tindakan guna memaksimalkan nilai fungsi imbalan. Seperti ditunjukkan pada Gambar 2.10, dua bagian terpenting dari algoritma pembelajaran penguatan adalah model dan lingkungan. Dalam lingkungan yang berbeda, model dapat menentukan tindakannya sendiri, dan tindakan yang berbeda dapat memiliki efek yang berbeda pula terhadap lingkungan.



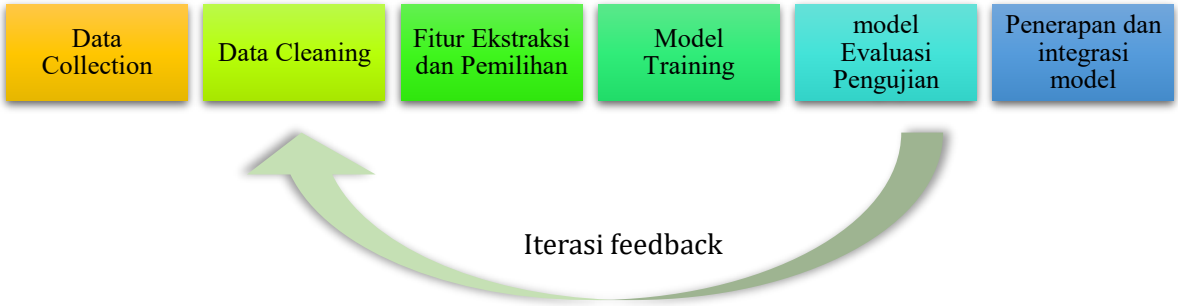
Gambar 2.10 Algoritma Pembelajaran Penguatan

Namun, dalam kasus penyelesaian soal ujian, komputer dapat memberikan jawaban secara acak, dan guru akan memberikan skor berdasarkan jawaban yang diberikan. Namun, jika situasinya hanya terbatas pada kasus ini, mustahil bagi komputer untuk mempelajari cara menyelesaikan soal, karena penilaian guru tidak berkontribusi pada proses pelatihan. Dalam hal ini, pentingnya status serta imbalan dan hukuman disorot. Skor ujian yang lebih tinggi dapat membuat guru puas dan kemudian memberikan komputer hadiah tertentu. Sebaliknya, skor ujian yang lebih rendah dapat menimbulkan penalti.

Sebagai komputer yang "termotivasi", komputer tentu berharap bahwa dengan menyesuaikan parameter modelnya sendiri, ia dapat memperoleh lebih banyak hadiah karena jawabannya. Dalam proses ini, tidak ada yang menyediakan data pelatihan untuk algoritma pembelajaran atau memberi tahu sistem pembelajaran penguatan cara melakukan langkah yang benar. Semua data dan sinyal reward dihasilkan secara dinamis selama interaksi antara model dan lingkungan, serta dipelajari secara otomatis dan dinamis. Baik perilaku baik maupun buruk, hal ini dapat membantu model untuk belajar.

2.3 PROSES KESELURUHAN PEMBELAJARAN MESIN

Sebuah proyek pembelajaran mesin yang lengkap seringkali melibatkan pengumpulan data, pembersihan data, ekstraksi dan pemilihan fitur, pelatihan model, evaluasi dan pengujian model, penerapan dan integrasi model, seperti yang ditunjukkan pada Gambar 2.11. Bagian ini memperkenalkan konsep-konsep terkait pengumpulan data dan pembersihan data, yang merupakan hal mendasar untuk memahami apa itu pemilihan fitur. Setelah memilih fitur yang sesuai, penting untuk melatih dan mengevaluasi model berdasarkan fitur-fitur tersebut. Proses ini bukanlah proses sekali jalan, tetapi membutuhkan umpan balik dan iterasi yang konstan untuk mendapatkan hasil yang memuaskan. Terakhir, model perlu diterapkan pada skenario aplikasi spesifik untuk menerapkan teori-teori tersebut.



Gambar 2.11 Proses Pembelajaran Mesin secara Keseluruhan

Pengumpulan Data

Kumpulan data adalah sekumpulan data yang digunakan dalam proyek pembelajaran mesin, dan setiap data disebut sampel. Item atau atribut yang mencerminkan kinerja atau sifat sampel dalam aspek tertentu disebut fitur. Kumpulan data yang digunakan dalam proses pelatihan disebut set pelatihan, dan setiap sampel disebut sampel pelatihan. Pembelajaran (pelatihan) adalah proses mempelajari model dari data. Proses penggunaan model untuk membuat prediksi disebut pengujian, dan kumpulan data yang digunakan untuk pengujian dikenal sebagai set uji. Setiap sampel dalam set uji disebut sampel uji.

		Fitur 1	Fitur 2	Fitur 3	Label
	Serial Number	Area Lantai	Diskrit Sekolah	Orientasi	Harga Rumah (dalam Juta Rupiah)
Training Set	1	100	8	Selatan	1000
	2	120	9	Barat Daya	1300
	3	60	6	Utara	700

	4	80	9	Tenggara	1100
Test Set	5	95	3	Selatan	850

Gambar 2.12 Contoh data

Gambar 2.12 menunjukkan kumpulan data tipikal. Dalam kumpulan data ini, setiap baris menunjukkan sampel, dan setiap kolom mengacu pada fitur atau label. Ketika tugas ditentukan, seperti memprediksi harga rumah berdasarkan luas lantai, distrik sekolah, dan orientasi rumah, fitur dan label juga ditentukan. Oleh karena itu, tajuk baris dan kolom dari kumpulan data harus tetap tidak berubah selama proyek pembelajaran mesin berlangsung. Set pelatihan dan set pengujian relatif bebas untuk dipisahkan, dan peneliti dapat menentukan sampel mana yang termasuk dalam set pelatihan berdasarkan pengalaman.

Proporsi set pengujian yang terlalu rendah dapat mengakibatkan pengujian model yang acak, sehingga tidak dapat mengevaluasi kinerja model dengan tepat. Sementara itu, proporsi set pelatihan yang tinggi dapat mengakibatkan pemanfaatan sampel yang rendah dan model tidak dapat belajar secara menyeluruh. Oleh karena itu, rasio umum antara set pelatihan dan set data adalah bahwa set pelatihan mencakup 80% dari total jumlah sampel, dan set pengujian mencakup 20%. Dalam contoh ini, terdapat empat sampel dalam set pelatihan dan satu sampel dalam set pengujian.

Pembersihan Data

Data sangat penting bagi model dan menentukan batas kemampuan model. Tanpa data yang baik, tidak akan ada model yang baik. Namun, kualitas data merupakan masalah umum yang mengganggu data nyata, seperti yang ditunjukkan pada Gambar 2.13.

#	Id	Nama	Tanggal Lahir	Gender	Guru?	Jumlah Siswa	Negara	Kota	Catatan Kesalahan
1	11	John	31/12/1990	M	0	0	Irlandia	Dublin	-
2	222	Mery	15/10/1978	F	1	15	Islandia	*(Kosong)*	Missing value (kota kosong)
3	333	Alice	19/04/2000	F	0	0	Spanyol	Madrid	-
4	444	Mark	01/11/1997	M	0	0	Prancis	Paris	-
5	555	Alex	15/03/2000	A	1	23	Jerman	Berlin	Invalid value (Gender seharusnya M/F, bukan A)
6	555	Peter	1983-12-01	M	1	10	Italia	Roma	Invalid repetition (Id ganda: 555), Format error (tanggal lahir)
7	777	Calvin	05/05/1995	M	0	0	Italia	Italia	Value in the other column (Negara & Kota sama)
8	888	Roxane	03/08/1948	F	0	0	Portugal	Lisbon	-
9	999	Anne	05/09/1992	F	0	5	Swiss	Jenewa	-
10	101010	Paul	14/11/1992	M	1	26	Ytali	Roma	Attribute dependency (Guru tidak bisa punya murid langsung),

									Wrong spelling (Ytali → Italia)

Gambar 2.13 Data "Kotor"

Berikut adalah beberapa masalah umum terkait kualitas data.

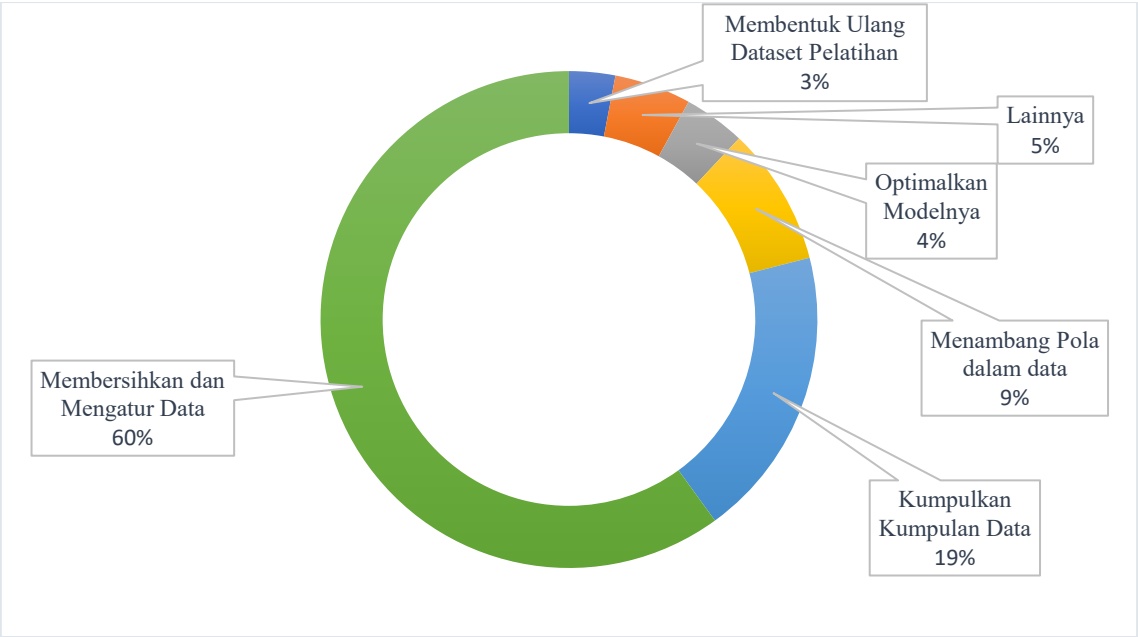
1. Tidak lengkap: Data kekurangan atribut atau mengandung nilai yang hilang.
2. Berisik: Data mengandung catatan yang salah atau outlier.
3. Tidak konsisten: Data mengandung catatan yang bertentangan atau ketidaksesuaian.

Data semacam ini disebut data "kotor". Proses mengisi nilai yang hilang, menemukan, dan menghilangkan ketidaknormalan data disebut pembersihan data. Selain itu, prapemrosesan data sering kali melibatkan reduksi dimensionalitas data dan standarisasi data. Tujuan reduksi dimensionalitas data adalah untuk menyederhanakan atribut data dan menghindari ledakan dimensi; sementara tujuan standarisasi data adalah untuk menyatukan dimensi setiap fitur, sehingga mengurangi kesulitan pelatihan. Materi reduksi dimensionalitas data dan standarisasi data akan dijelaskan secara rinci di bagian selanjutnya, dan bagian ini hanya membahas pembersihan data.

Semua fitur ditangani oleh model pembelajaran mesin. Fitur yang disebut adalah representasi numerik dari variabel input yang dapat digunakan dalam model. Dalam kebanyakan kasus, data yang dikumpulkan dapat digunakan oleh algoritma setelah prapemrosesan. Operasi prapemrosesan terutama mencakup prosedur-prosedur berikut.

1. Pemfilteran data.
2. Penanganan data yang hilang.
3. Menangani kemungkinan kesalahan atau outlier.
4. Menggabungkan data dari berbagai sumber.
5. Agregasi data.

Beban kerja pembersihan data seringkali cukup berat. Penelitian menunjukkan bahwa ilmuwan data menghabiskan 60% waktunya untuk membersihkan dan mengorganisasikan data dalam penelitian pembelajaran mesin, seperti yang ditunjukkan pada Gambar 2.14.



Gambar 2.14 Pentingnya pembersihan data

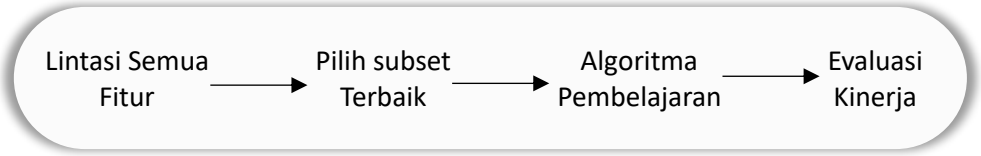
Di satu sisi, hal ini menunjukkan betapa sulitnya pembersihan data, dan bahwa pengumpulan data yang menampilkan metode dan konten yang berbeda akan memerlukan metode pembersihan data yang berbeda pula. Lebih lanjut, hal ini juga menunjukkan bahwa pembersihan data memainkan peran krusial dalam pelatihan dan optimasi model selanjutnya. Peran penting lainnya adalah semakin menyeluruh data dibersihkan, semakin kecil kemungkinan model terpengaruh oleh data abnormal, sehingga memastikan performa pelatihan model.

Pemilihan Fitur

Biasanya, terdapat banyak fitur berbeda dalam sebuah dataset, beberapa di antaranya mungkin redundan atau tidak terkait dengan target. Misalnya, ketika memprediksi harga rumah berdasarkan luas lantai, distrik sekolah, dan suhu harian, suhu harian tampaknya merupakan fitur yang tidak relevan. Melalui pemilihan fitur, fitur-fitur yang redundan atau tidak relevan ini dapat dihilangkan, sehingga model disederhanakan dan lebih mudah diinterpretasikan oleh pengguna. Pada saat yang sama, pemilihan fitur juga dapat secara efektif mengurangi waktu pelatihan model, menghindari ledakan dimensi, meningkatkan kinerja generalisasi model, dan menghindari overfitting. Metode umum untuk pemilihan fitur meliputi metode filter, metode wrapper, dan metode tertanam, yang akan diperkenalkan secara berurutan dalam bagian-bagian berikut.

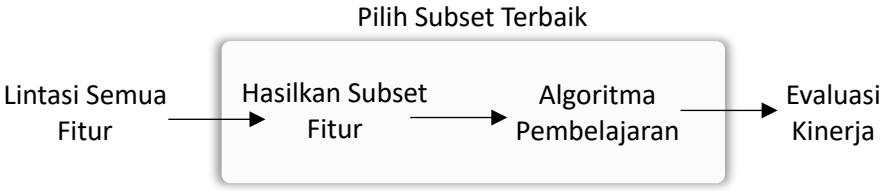
Metode filter bersifat independen ketika memilih fitur dan tidak ada hubungannya dengan model itu sendiri. Dengan mengukur korelasi antara setiap fitur dan atribut target, metode filter menerapkan pengukuran statistik untuk menilai setiap fitur. Dengan mengurutkan fitur-fitur ini berdasarkan skor, Anda dapat memutuskan untuk mempertahankan atau menghilangkan fitur-fitur tertentu. Gambar 2.15 menunjukkan proses pembelajaran mesin menggunakan metode filter. Ukuran statistik yang umum digunakan dalam pemfilteran meliputi koefisien korelasi Pearson, koefisien Chi-Square, dan informasi

mutual. Karena filter tidak mempertimbangkan hubungan antar fitur, filter hanya cenderung memilih variabel yang redundan.



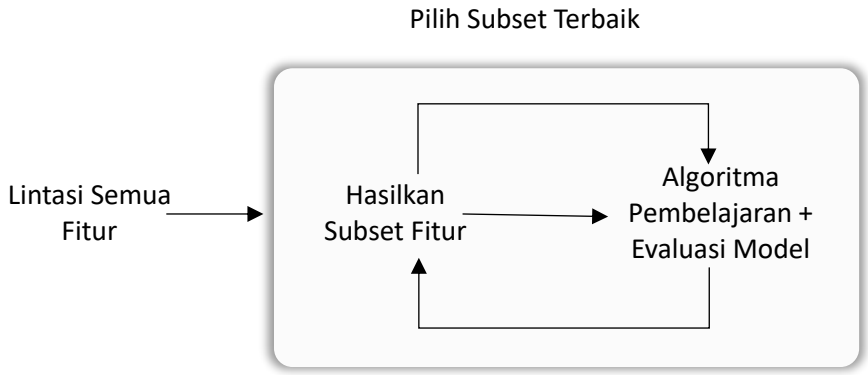
Gambar 2.15 Proses pembelajaran mesin menggunakan penyaringan

Metode wrapper menggunakan model prediktif untuk menilai subset fitur, dan menganggap masalah pemilihan fitur sebagai masalah pencarian, di mana wrapper akan mengevaluasi dan membandingkan berbagai kombinasi fitur, dan model prediktif akan digunakan sebagai alat untuk mengevaluasi kombinasi fitur. Semakin tinggi akurasi model prediksi, semakin banyak kombinasi fitur yang harus dipertahankan. Gambar 2.16 menampilkan pembelajaran mesin menggunakan metode wrapper. Salah satu metode wrapper yang populer adalah eliminasi fitur rekursif (RFE). Metode wrapper biasanya menyediakan set fitur berkinerja terbaik untuk kelas model tertentu, tetapi perlu melatih model baru untuk setiap subset fitur, sehingga jumlah operasinya sangat banyak.



Gambar 2.16 Proses pembelajaran mesin menggunakan wrapper

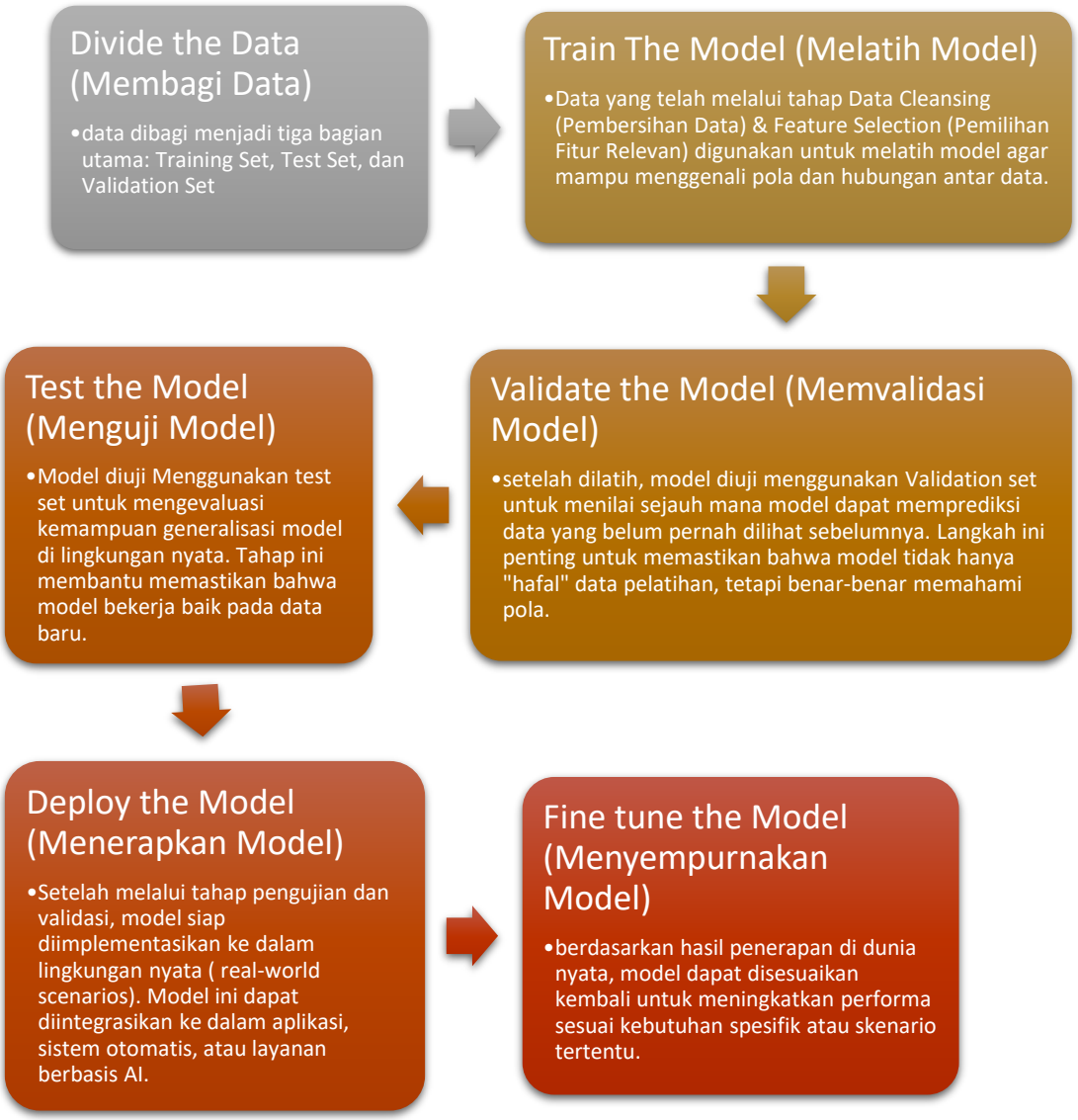
Metode embedded menggunakan pemilihan fitur sebagai bagian dari pembangunan model, seperti yang ditunjukkan pada Gambar 2.17. Berbeda dengan metode filter dan wrapper, model yang menggunakan metode tertanam secara aktif mempelajari cara melakukan seleksi fitur selama pelatihan. Metode seleksi fitur tertanam yang paling umum adalah regularisasi. Regularisasi juga disebut metode penalti. Dengan memperkenalkan batasan tambahan saat mengoptimalkan algoritma prediksi, kompleksitas model berkurang, yaitu jumlah fitur berkurang. Metode regularisasi yang umum meliputi regresi ridge dan regresi Lasso.



Gambar 2.17 Proses pembelajaran mesin menggunakan metode tertanam

Konstruksi Model Pembelajaran Mesin

Setelah pembersihan data dan ekstraksi fitur selesai, saatnya membangun model. Mengambil pembelajaran terawasi sebagai contoh, konstruksi model umumnya mengikuti langkah-langkah yang ditunjukkan pada Gambar 2.18.



Gambar 2.18 Proses keseluruhan konstruksi model

Inti dari konstruksi model adalah pelatihan, verifikasi, dan pengujian model. Bagian ini menjelaskan secara singkat proses pelatihan dan prediksi menggunakan satu contoh sederhana. Detail lebih lanjut akan dijelaskan pada bab-bab berikutnya. Dalam contoh di bagian ini, kita perlu menggunakan model klasifikasi untuk menentukan apakah seseorang perlu mengganti pemasok berdasarkan fitur tertentu. Dengan asumsi Gambar 2.19 menunjukkan dataset yang telah dibersihkan, tugas model adalah memprediksi target seakurat mungkin berdasarkan fitur yang diketahui.

Nama	Kota	Usia	Perubahan (Label)	Keterangan
Mike	Miami	42	ya	Training set
Jerry	New York	32	tidak	Training set
Bryan	Orlando	18	tidak	Training set
Patricia	Miami	45	ya	Training set
Elodie	Phoenix	35	tidak	Test set
Remy	Chicago	72	ya	Test set
John	New York	48	ya	Test set

Gambar 2.19 Set pelatihan dan set pengujian

Selama proses pelatihan, model dapat mempelajari hubungan pemetaan antara fitur dan target berdasarkan sampel dalam set pelatihan. Setelah pelatihan, kita dapat memperoleh model berikut:

```
def model(city, age):
  if city == "Miami": return 0.7
  if city == "Orlando": return 0.2
  if age > 42: return 0.05 * age + 0.06
  else: return 0.01 * age + 0.02
```

Keluaran model adalah probabilitas nilai kebenaran target. Kita tahu bahwa seiring bertambahnya data pelatihan, akurasi model juga akan meningkat. Jadi, mengapa tidak menggunakan semua data untuk pelatihan, alih-alih hanya mengambil sebagiannya sebagai set pengujian? Hal ini karena kinerja model dalam menghadapi data yang tidak diketahui, bukan data yang diketahui, yang harus kita perhatikan. Set pelatihan ibarat bank soal yang dibaca siswa saat mempersiapkan diri menghadapi ujian. Hal ini bukanlah hal yang mengejutkan, betapapun tingginya tingkat akurasi bagi siswa, karena bank soal selalu memiliki keterbatasan. Selama mereka memiliki daya ingat yang baik, siswa bahkan dapat menghafal semua jawaban.

Hanya ujian formal yang benar-benar dapat menguji perolehan pengetahuan siswa, karena soal-soal dalam ujian resmi mungkin merupakan sesuatu yang belum pernah mereka lihat sebelumnya. Set pengujian setara dengan ujian yang disiapkan oleh peneliti untuk model tersebut. Dengan kata lain, di seluruh set data (termasuk set pelatihan dan set pengujian), model hanya berhak untuk merujuk pada fitur-fitur dari set pelatihan dan set pengujian. Target set pengujian hanya dapat digunakan oleh para peneliti ketika mengevaluasi kinerja model.

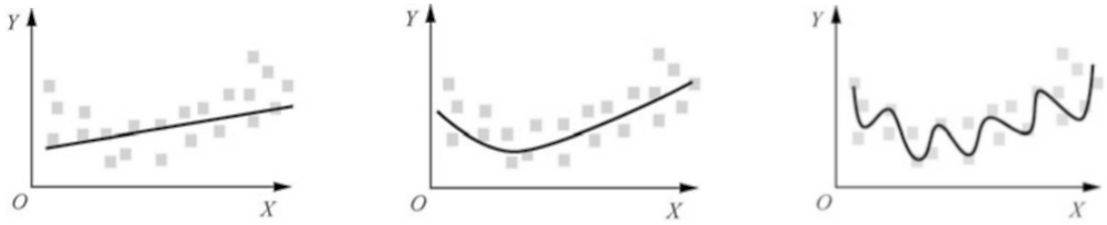
Evaluasi Model

Apa itu model yang baik? Indikator evaluasi yang paling penting adalah kemampuan generalisasi model, yang juga dikenal sebagai akurasi prediksi model terhadap data bisnis aktual. Ada juga beberapa indikator rekayasa yang dapat digunakan untuk mengevaluasi model: interpretabilitas, yang menggambarkan tingkat kelurusan hasil prediksi model; laju prediksi, yang mengacu pada waktu rata-rata yang dibutuhkan model untuk memprediksi setiap sampel; plastisitas, yang mengacu pada tingkat penerimaan laju prediksi model dalam proses bisnis aktual seiring dengan peningkatan volume bisnis.

Tujuan pembelajaran mesin adalah membuat model yang dipelajari dapat diterapkan pada sampel baru, bukan hanya pada sampel pelatihan. Kemampuan model yang dipelajari untuk diterapkan pada sampel baru disebut kemampuan generalisasi, yang juga disebut sebagai kekokohan. Selisih antara hasil prediksi model yang dipelajari pada sampel dan hasil sebenarnya dari sampel tersebut disebut galat. Kesalahan pelatihan mengacu pada kesalahan model pada set pelatihan, dan kesalahan generalisasi mengacu pada kesalahan model pada sampel baru (set uji). Tentu saja, kita ingin memiliki model dengan kesalahan generalisasi yang lebih kecil.

Setelah model terbentuk dan diperbaiki, semua fungsi yang mungkin akan membentuk ruang, yang disebut ruang hipotesis. Algoritma pembelajaran mesin dapat dilihat sebagai algoritma yang mencari fungsi fitting yang sesuai dalam ruang hipotesis. Model matematika yang terlalu sederhana, atau waktu pelatihan yang terlalu singkat, akan menyebabkan peningkatan kesalahan pelatihan pada model. Fenomena ini disebut underfitting. Untuk underfitting, model harus menggunakan model yang lebih kompleks untuk pelatihan ulang; untuk underfitting, hanya perlu memperpanjang waktu untuk menghilangkan underfitting secara efektif. Namun, untuk menentukan penyebab underfitting secara akurat seringkali membutuhkan pengalaman dan metode tertentu.

Sebaliknya, jika model terlalu kompleks, hal ini dapat menyebabkan kesalahan pelatihan yang kecil, tetapi kemampuan generalisasinya lebih lemah, yang berarti kesalahan generalisasi yang lebih besar yang dikenal sebagai overfitting. Ada banyak metode untuk mengurangi overfitting. Metode yang umum digunakan antara lain menyederhanakan model dengan tepat, mengakhiri pelatihan sebelum overfitting terjadi, dan menggunakan metode dropout dan peluruhan bobot. Gambar 2.20 menunjukkan hasil underfitting, good fitting, dan overfitting untuk dataset yang sama.



(a) under-fitting, gagal mempelajari fitur

(b) cocok

(c) over-fitting, noise dipelajari

Gambar 2.20 Underfitting, good fitting, dan overfitting

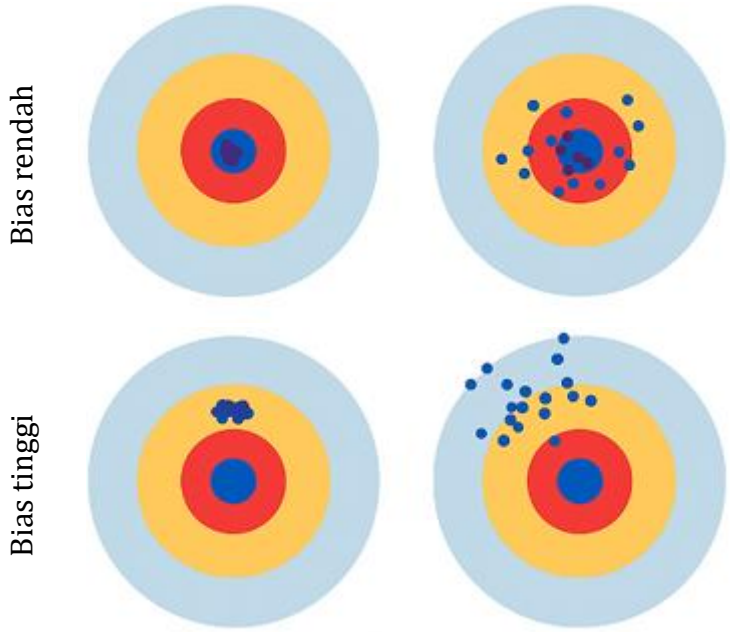
Kapasitas suatu model mengacu pada kemampuannya untuk menyesuaikan berbagai fungsi, yang juga dikenal sebagai kompleksitas suatu model. Ketika kapasitasnya sesuai dengan kompleksitas tugas dan jumlah data pelatihan yang disediakan, algoritma biasanya akan memiliki kinerja terbaik. Model dengan kapasitas yang tidak memadai tidak dapat menangani tugas-tugas yang kompleks, sehingga underfitting dapat terjadi. Seperti ditunjukkan pada Gambar 2.20a, distribusi data berbentuk kait, tetapi modelnya linear dan tidak dapat menggambarkan distribusi data dengan baik.

Model dengan kapasitas tinggi dapat menangani tugas-tugas kompleks, tetapi ketika kapasitasnya melampaui tingkat yang dibutuhkan tugas tersebut, overfitting dapat terjadi. Seperti ditunjukkan pada Gambar 2.20c, model mencoba menyesuaikan data dengan fungsi yang sangat kompleks. Meskipun kesalahan pelatihan berkurang, dapat disimpulkan bahwa model tersebut tidak dapat memprediksi nilai target sampel baru dengan baik. Kapasitas efektif model dibatasi oleh algoritma, parameter, dan metode regularisasi. Secara umum, kesalahan generalisasi dapat diartikan sebagai:

$$Total\ error\ =\ Bias\ 2\ +\ Variance\ +\ Unresolvable\ error$$

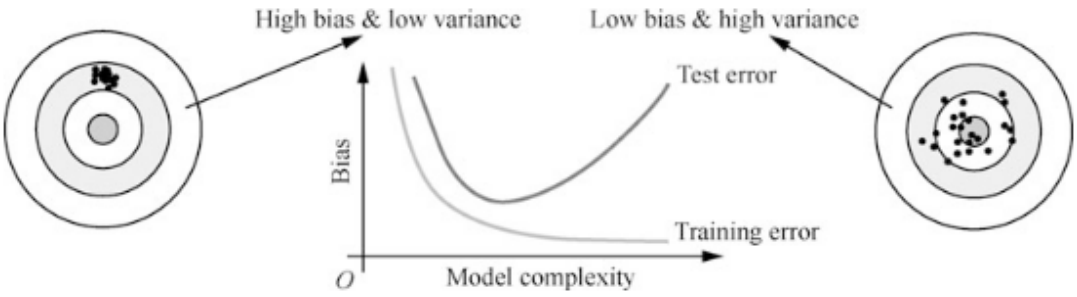
Di antara keduanya, bias dan varians adalah dua sub-bentuk yang perlu kita perhatikan. Seperti ditunjukkan pada Gambar 2.21, Varians adalah derajat deviasi hasil prediksi model di dekat rata-rata, yang merupakan kesalahan yang berasal dari sensitivitas model terhadap fluktuasi kecil pada set pelatihan. Bias adalah selisih antara nilai rata-rata hasil prediksi model dan nilai benar yang ingin kita prediksi. Kesalahan yang tidak dapat diselesaikan mengacu pada kesalahan yang disebabkan oleh ketidaksempurnaan model dan keterbatasan data. Secara teori, jika terdapat jumlah data yang tak terbatas dan model yang sempurna, kesalahan yang tidak dapat diselesaikan dapat diselesaikan. Namun pada kenyataannya, hal ini mustahil untuk diwujudkan, sehingga kesalahan generalisasi tidak akan pernah dapat dihilangkan.

Varians rendah Varians tinggi



Gambar 2.21 Varians dan Bias

Idealnya, kita ingin memilih model yang dapat secara akurat menangkap hukum dalam data pelatihan dan juga dapat meringkas data yang tidak terlihat (yang disebut data baru). Namun, secara umum, mustahil bagi kita untuk mencapai kedua hal ini secara bersamaan. Seperti yang ditunjukkan pada Gambar 2.22, seiring meningkatnya kompleksitas model, kesalahan pelatihan secara bertahap menurun. Pada saat yang sama, kesalahan pengujian akan menurun hingga titik tertentu seiring meningkatnya kompleksitas, dan kemudian meningkat ke arah yang berlawanan, membentuk kurva cekung. Titik terendah kurva kesalahan pengujian menunjukkan tingkat kompleksitas model yang ideal.



Gambar 2.22 Hubungan antara Kompleksitas Model dan Kesalahan

Saat mengukur kinerja model regresi, indikator yang umum digunakan meliputi galat absolut rata-rata (MAE), galat kuadrat rata-rata (MSE), dan koefisien korelasi $R2$. Dengan asumsi bahwa nilai target sebenarnya dari contoh uji adalah $y1, y2, \ldots, ym$, dan nilai prediksi yang sesuai adalah $\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_m$, maka definisi indikator di atas adalah sebagai berikut:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{m}|y_i - \hat{y}_i|}{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

Di mana, TSS merepresentasikan selisih antara nilai sampel, dan RSS merepresentasikan selisih antara nilai prediksi dan nilai sampel. Nilai indikator MAE dan MSE keduanya non-negatif, dan semakin mendekati 0, semakin baik kinerja model. Nilai $R2$ tidak lebih besar dari 1, dan semakin mendekati 1, semakin baik kinerja model.

Perdiksi / Aktual	Ya	Tidak	Total
Ya	TP	FN	P
Tidak	FP	TN	N
Total	P'	N'	$P + N$

Gambar 2.23 Matriks kebingungan untuk pengklasifikasi biner

Saat mengevaluasi kinerja model klasifikasi, metode yang disebut matriks konfusi sering digunakan, seperti yang ditunjukkan pada Gambar 2.23. Matriks konfusi adalah matriks persegi berdimensi-k, dengan k merepresentasikan jumlah semua kategori. Nilai pada baris ke$-i$ dan kolom ke-j pada Gambar 2.23 merepresentasikan jumlah sampel yang sebenarnya bertipe ke-i tetapi dinilai bertipe ke-j oleh model. Idealnya, untuk pengklasifikasi dengan akurasi yang lebih tinggi, sebagian besar contoh harus direpresentasikan oleh diagonal matriks konfusi, sementara nilai lainnya adalah 0 atau mendekati 0. Untuk matriks konfusi dua-pengklasifikasi yang ditunjukkan pada Gambar 2.23, definisi setiap simbol adalah sebagai berikut.

1. Tupel positif P: tupel dari kelas-kelas utama yang diminati.
2. Tupel negatif N: tupel lain kecuali P.
3. Contoh positif sejati TP: tupel positif yang diklasifikasikan dengan benar oleh pengklasifikasi.
4. Contoh negatif sejati TN: tupel negatif yang diklasifikasikan dengan benar oleh pengklasifikasi.

Pengukuran	Persamaan
Tingkat Akurasi, Tingkat Pengenalan	$\frac{TP + TN}{P + N}$
Tingkat Kesalahan, tingkat kesalahan klasifikasi	$\frac{FP + FN}{P + N}$

Tingkat Positif sebenarnya dan tingkat penarikan kembali	$\frac{TP}{P}$
Tingkat Negatif sebenarnya	$\frac{TN}{P}$
Tingkat Presisi	$\frac{TP}{TP+TN}$
Nilai $\boldsymbol{F_1}$ (Rata-rata harmonik presisi dan perolehan kembali)	$\frac{2 \times Presisi \times Recall}{Presisi \times Recall}$
Nilai $\boldsymbol{F_\beta}$ ($\boldsymbol{\beta}$ Adalah bilangan riil non-negatif)	$\frac{(1+\beta^2) \times Presisi \times Recall}{\beta^2 \times Presisi \times Recall}$

Gambar 2.24 Konsep lain dalam matriks kebingungan untuk pengklasifikasi biner

5. Contoh positif palsu FP: tupel negatif yang salah ditandai sebagai tupel positif.
6. Contoh negatif palsu FN: Tupel positif yang salah ditandai sebagai tupel negatif.

Di sini, mari kita kutip contoh temu kembali dokumen untuk memperjelas konsep presisi dan penarikan kembali. Presisi menggambarkan proporsi dokumen yang benar-benar terkait dengan subjek di antara semua dokumen yang diambil. Penarikan kembali menggambarkan dokumen yang diambil terkait dengan subjek pencarian, dan proporsi semua dokumen terkait di pustaka.

Di akhir bagian ini, mari kita ambil contoh untuk mengilustrasikan perhitungan matriks kebingungan pengklasifikasi biner. Dengan asumsi bahwa pengklasifikasi dapat mengidentifikasi apakah ada kucing dalam suatu gambar dan 200 gambar sekarang digunakan untuk memverifikasi kinerja model ini. Di antara gambar-gambar tersebut, 170 gambar berlabel kucing dan 30 gambar tidak berlabel. Performa model ditunjukkan pada Gambar 2.25. Terlihat bahwa hasil pengenalan model menunjukkan 160 gambar berlabel kucing dan 40 gambar tidak berlabel. Presisi model dapat dihitung sebesar 140/160 = 87,5%, recall sebesar 140/170 = 82,4%, dan akurasinya (140 + 10) / 200 = 75%.

Perdiksi / Aktual	Ya	Tidak	Total
Ya	140	30	170
Tidak	20	10	30
Total	160	40	200

Gambar 2.25 Kasus matriks kebingungan

2.4 PARAMETER MODEL DAN HIPERPARAMETER

Parameter, sebagai bagian dari apa yang telah dipelajari model dari data pelatihan historis, merupakan kunci bagi algoritma pembelajaran mesin. Secara umum, parameter model tidak ditetapkan secara manual oleh peneliti, melainkan diperoleh melalui estimasi

data atau pembelajaran data. Mengidentifikasi nilai parameter model sama dengan mendefinisikan fungsi model, sehingga parameter model biasanya disimpan sebagai bagian dari model pembelajaran. Saat mengimplementasikan prediksi model, parameter juga merupakan komponen yang sangat penting. Contoh parameter model meliputi bobot dalam jaringan saraf tiruan, vektor pendukung dalam mesin vektor pendukung, dan koefisien dalam regresi linier atau regresi logistik.

Tidak hanya terdapat parameter, tetapi juga hiperparameter dalam model. Berbeda dengan parameter, hiperparameter merupakan konfigurasi eksternal model dan sering digunakan dalam proses estimasi parameter model. Perbedaan paling mendasar antara keduanya adalah parameter dipelajari secara otomatis oleh model, sementara hiperparameter direkayasa secara manual. Saat menangani berbagai masalah pemodelan prediksi, biasanya perlu menyesuaikan hiperparameter model.

Selain ditentukan langsung oleh peneliti, hiperparameter model juga dapat diatur menggunakan metode heuristik. Hiperparameter model yang umum meliputi koefisien penalti dalam regresi Lasso atau ridge, laju pembelajaran, jumlah iterasi, ukuran batch, fungsi aktivasi, dan jumlah neuron dalam jaringan saraf tiruan pelatihan, C dan σ dari mesin vektor pendukung, dan K dalam KNN, jumlah model pohon keputusan dalam hutan acak, dll.

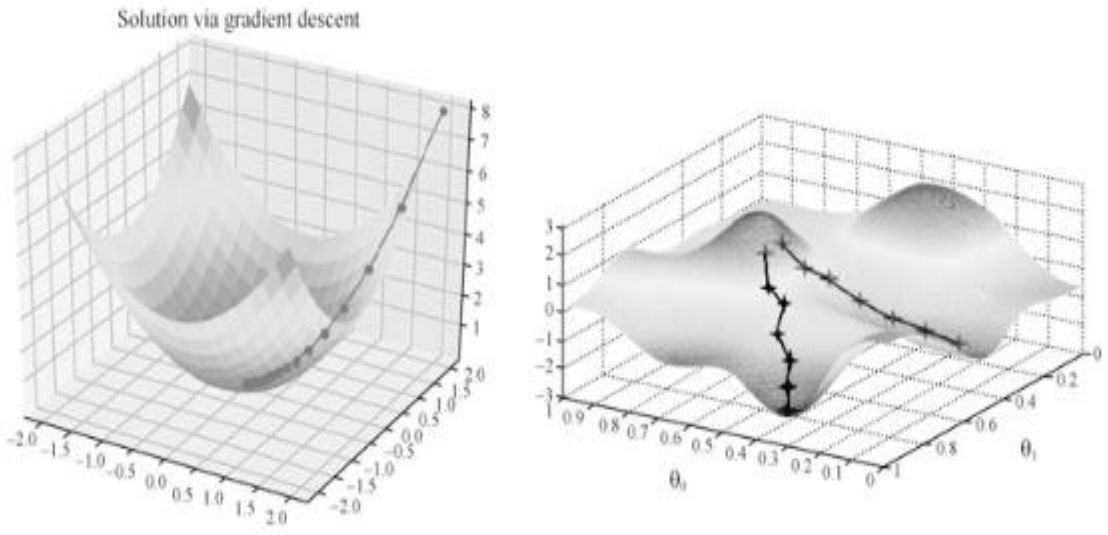
Pelatihan model umumnya mengacu pada pengoptimalan parameter model, dan proses ini diselesaikan oleh algoritma penurunan gradien. Berdasarkan efek pelatihan model, serangkaian algoritma pencarian hiperparameter dapat digunakan untuk mengoptimalkan hiperparameter model. Bagian ini pertama-tama memperkenalkan algoritma penurunan gradien, kemudian konsep set validasi, dan kemudian menguraikan algoritma pencarian hiperparameter dan validasi silang.

Penurunan Gradien

Ide optimasi algoritma penurunan gradien adalah menggunakan arah gradien negatif dari posisi saat ini sebagai arah pencarian, yang merupakan arah penurunan tercepat dari posisi saat ini, seperti yang ditunjukkan pada Gambar 2.26a. Rumus untuk penurunan gradien adalah sebagai berikut:

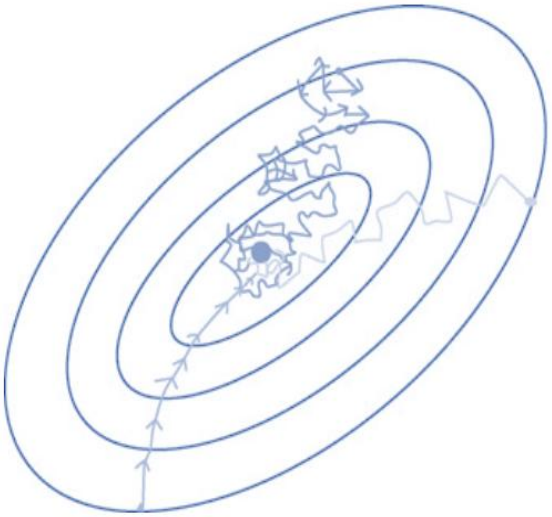
$$w_{k+1} = w_k - \eta \nabla f_{wk}(x)$$

Di mana, η disebut laju pembelajaran, dan w merepresentasikan parameter model. Seiring w mendekati nilai target, jumlah perubahan w secara bertahap berkurang. Ketika nilai fungsi objektif hampir tidak berubah atau mencapai jumlah maksimum iterasi penurunan gradien, maka algoritma mencapai konvergensi. Perlu dicatat bahwa ketika menggunakan algoritma penurunan gradien untuk menemukan nilai minimum fungsi non-konveks, nilai awal yang berbeda dapat menghasilkan hasil yang berbeda, seperti yang ditunjukkan pada Gambar 2.26b. Saat menerapkan penurunan gradien pada pelatihan model, beberapa varian dapat digunakan.



Gambar 2.26 Algoritma penurunan gradien

Penurunan Gradien Batch (BGD) menggunakan rata-rata gradien sampel di semua dataset pada titik saat ini untuk memperbarui parameter bobot. Penurunan Gradien Stokastik (SGD) secara acak memilih sampel dalam dataset, dan memperbarui parameter bobot melalui gradien sampel ini. Penurunan Gradien Mini-batch (MBGD) menggabungkan karakteristik BGD dan SGD, dan memilih rata-rata gradien n sampel dalam dataset untuk memperbarui parameter bobot setiap kali. Gambar 2.27 menunjukkan perbedaan kinerja ketiga varian penurunan gradien. Di antaranya, kurva bottom-up berkorespondensi dengan BGD, kurva top-down berkorespondensi dengan SGD, dan kurva kanan-ke-kiri berkorespondensi dengan MBGD. BGD paling stabil saat runtime, tetapi karena setiap pembaruan harus melintasi semua sampel, ia menghabiskan banyak sumber daya komputasi.



Turunan gradien batch Berlatih dengan semua sampel pelatihan

Turunan gradien acak Berlatih dengan satu sampel pelatihan setiap kali

Turunan gradien batch Berlatih dengan jumlah sampel pelatihan tertentu

Gambar 2.27 Perbandingan efisiensi algoritma penurunan gradien

Setiap pembaruan SGD memilih sampel secara acak. Meskipun meningkatkan efisiensi komputasi, pembaruan ini juga menimbulkan ketidakstabilan, yang dapat menyebabkan fungsi kerugian menghasilkan turbulensi atau bahkan perpindahan terbalik selama proses penurunan ke titik terendah. MBGD adalah metode setelah SGD dan BGD diseimbangkan, dan juga merupakan algoritma penurunan gradien yang paling umum digunakan dalam pembelajaran mesin.

Set Validasi dan Pencarian Hiperparameter

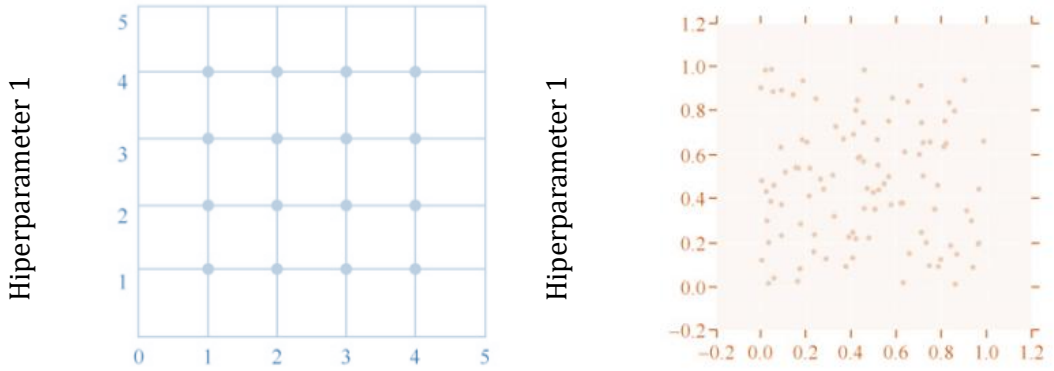
Set pelatihan adalah kumpulan sampel yang digunakan dalam pelatihan model. Selama proses pelatihan, algoritma penurunan gradien akan mencoba meningkatkan akurasi prediksi model untuk sampel-sampel dalam set pelatihan. Hal ini menyebabkan model berkinerja lebih baik pada set pelatihan dibandingkan pada set data yang tidak diketahui. Untuk mengukur kemampuan generalisasi model, orang sering kali secara acak memilih sebagian dari keseluruhan set data sebagai set uji sebelum pelatihan, seperti yang ditunjukkan pada Gambar 2.28. Sampel-sampel dalam set uji tidak terlibat dalam pelatihan, sehingga tidak diketahui oleh model. Dapat didekati bahwa kinerja model pada set uji adalah kinerja model pada sampel yang tidak diketahui.

Gambar 2.28 Set pelatihan, set validasi, dan set uji

Tujuan optimasi hiperparameter adalah untuk meningkatkan kemampuan generalisasi model. Ide yang paling intuitif adalah mencoba berbagai nilai hiperparameter, mengevaluasi kinerja model-model ini pada set uji, dan memilih model dengan kemampuan generalisasi terkuat. Masalahnya adalah set uji tidak dapat berpartisipasi dalam pelatihan model dalam bentuk apa pun, bahkan untuk pencarian hiperparameter. Oleh karena itu, beberapa sampel harus dipilih secara acak dari set pelatihan, dan set sampel ini disebut set validasi. Sampel dari set validasi juga tidak berpartisipasi dalam pelatihan, dan hanya digunakan untuk memverifikasi efek hiperparameter.

Secara umum, model perlu dioptimalkan berulang kali pada set pelatihan dan set validasi untuk akhirnya menentukan parameter dan hiperparameter serta dievaluasi pada set pengujian. Metode yang umum digunakan untuk mencari hiperparameter model meliputi pencarian grid, pencarian acak, pencarian cerdas heuristik, dan pencarian Bayesian. Pencarian grid mencoba mencari secara menyeluruh semua kemungkinan kombinasi hiperparameter untuk membentuk grid nilai hiperparameter, seperti yang ditunjukkan pada Gambar 2.29a. Dalam praktiknya, rentang dan panjang langkah grid seringkali perlu ditentukan secara

manual. Dalam kasus hiperparameter dengan jumlah yang relatif kecil, pencarian grid dapat diterapkan, sehingga pencarian grid layak dalam algoritma pembelajaran mesin umum.



(a) Pencarian grid (b) pencarian acak

Gambar 2.29 Pencarian grid dan pencarian acak

Namun, dalam kasus jaringan saraf, pencarian grid terlalu mahal dan memakan waktu, sehingga tidak diadopsi dalam banyak kasus. Dalam kasus ruang pencarian hiperparameter yang besar, hasil menggunakan pencarian acak akan lebih baik daripada pencarian grid, seperti yang ditunjukkan pada Gambar 2.29b. Pencarian acak mengimplementasikan pengambilan sampel parameter secara acak, di mana setiap pengaturan adalah untuk mengambil sampel dari distribusi nilai parameter yang mungkin, mencoba menemukan subset parameter terbaik. Untuk menggunakan pencarian acak, Anda perlu "penyesuaian kasar" terlebih dahulu dan kemudian "penyesuaian halus". Yaitu, mencari dalam rentang kasar terlebih dahulu, dan kemudian mempersempit rentang pencarian sesuai dengan posisi di mana hasil terbaik muncul. Perlu dicatat bahwa beberapa hiperparameter mungkin lebih penting daripada yang lain dalam operasi aktual. Dalam hal ini, hiperparameter yang paling penting akan secara langsung memengaruhi bias pencarian, sedangkan hiperparameter sekunder mungkin tidak dioptimalkan dengan baik.

Validasi Silang

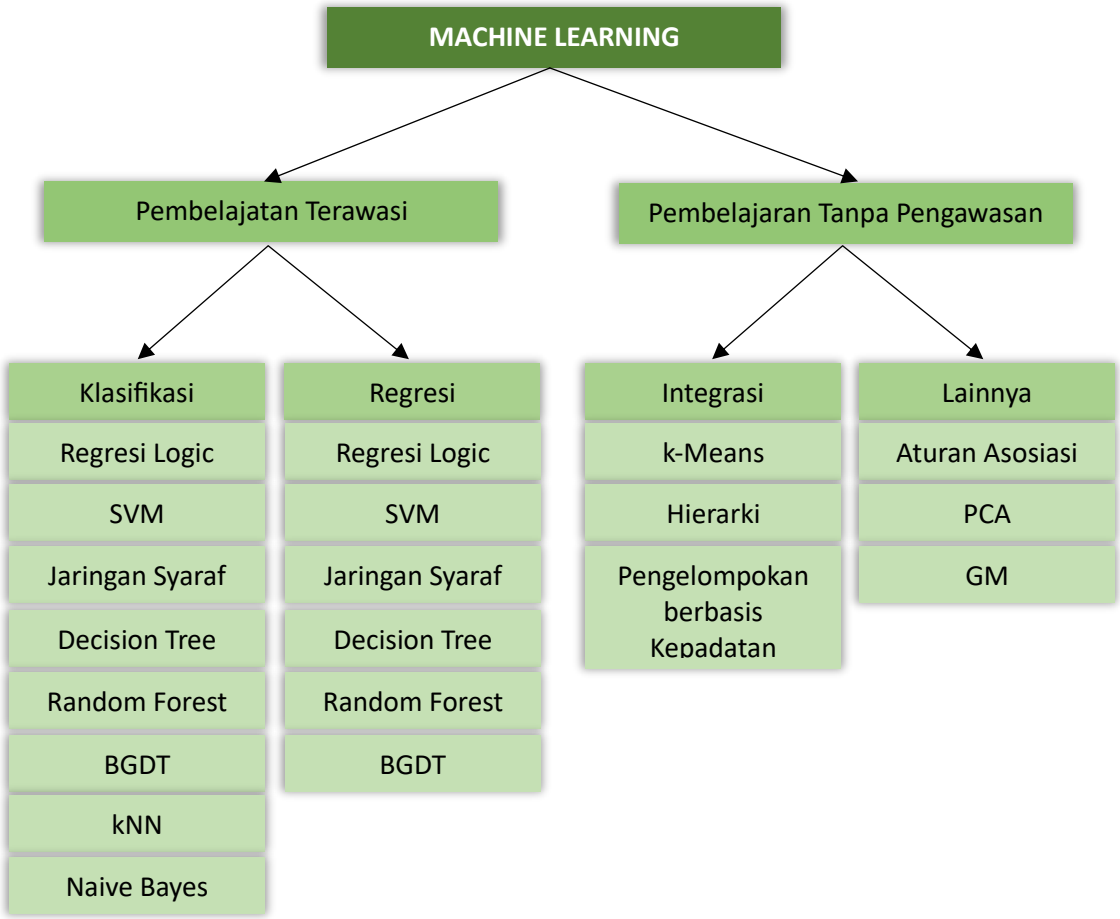
Metode pembagian set verifikasi yang disebutkan di atas memiliki dua masalah utama: peluang pembagian sampel yang besar, dan hasil verifikasi yang kurang meyakinkan; serta jumlah sampel yang dapat digunakan untuk pelatihan model semakin berkurang. Untuk mengatasi masalah ini, set pelatihan dapat dibagi menjadi k kelompok untuk validasi silang k-fold. Validasi silang k-fold akan melakukan k putaran pelatihan dan verifikasi, di mana satu set data digunakan sebagai set verifikasi secara bergantian, dan k set data sisanya digunakan sebagai set pelatihan. Ini akan menghasilkan k model dan akurasi klasifikasinya pada set validasi. Rata-rata dari k tingkat akurasi klasifikasi ini dapat digunakan sebagai indikator kinerja untuk kemampuan generalisasi model.

Validasi silang k-fold dapat menghindari kontingensi dalam proses pembagian set validasi, dan hasil validasi lebih meyakinkan. Namun, penggunaan validasi silang k-fold

membutuhkan pelatihan k model. Jika set data besar, pelatihan akan memakan waktu. Oleh karena itu, validasi silang k-fold umumnya berlaku untuk dataset yang lebih kecil. Nilai k dalam validasi silang k-fold juga merupakan hiperparameter, yang perlu ditentukan melalui eksperimen. Dalam kasus ekstrem, nilai k sama dengan jumlah sampel dalam set pelatihan. Pendekatan ini disebut validasi silang leave-one-out, karena satu sampel pelatihan dibiarkan sebagai set validasi selama setiap pelatihan. Hasil pelatihan dari validasi silang leave-one-out lebih baik, karena hampir semua sampel pelatihan terlibat dalam pelatihan. Namun, validasi silang leave-one-out membutuhkan waktu lebih lama, sehingga hanya cocok untuk dataset kecil.

2.5 ALGORITMA UMUM PEMBELAJARAN MESIN

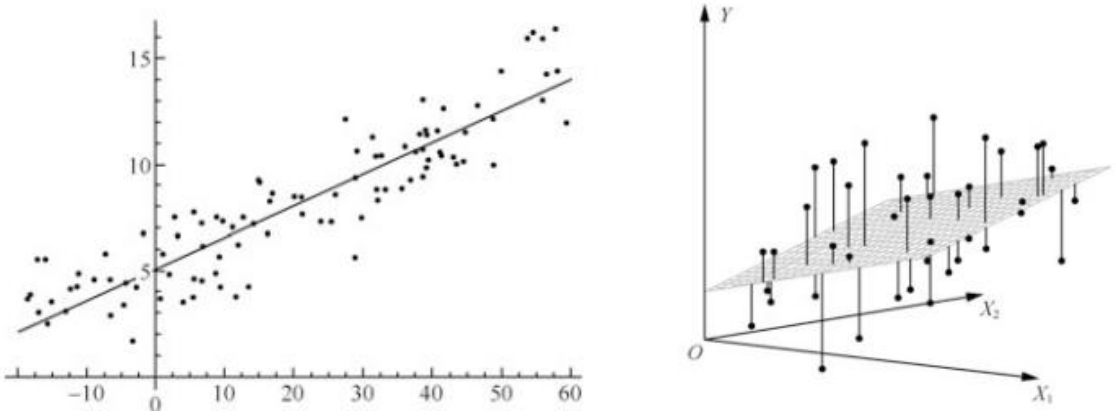
Seperti yang ditunjukkan pada Gambar 2.30, terdapat banyak algoritma umum untuk pembelajaran mesin, dan pengenalan mendetail tentang algoritma-algoritma ini mungkin membutuhkan waktu yang sama panjang seperti satu buku utuh. Oleh karena itu, bagian ini hanya memperkenalkan prinsip dan ide dasar dari algoritma-algoritma ini secara singkat. Pembaca yang tertarik dengan topik ini dapat merujuk ke buku-buku lain untuk pemahaman yang lebih mendalam.



Gambar 2.30 Algoritma umum pembelajaran mesin

Regresi Linier

Regresi linier adalah metode analisis statistik yang menggunakan analisis regresi dalam statistik matematika untuk menentukan hubungan kuantitatif antara dua variabel atau lebih. Metode ini termasuk dalam pembelajaran terbimbing. Seperti yang ditunjukkan pada Gambar 2.31, fungsi model regresi linier adalah sebuah hiperbidang:



(A) Regresi Linier Unary (B) Regresi Linier Berganda

Gambar 2.31 Regresi linier

$$h(x) = w^{\mathrm{T}}x + b$$

Di mana, w adalah parameter bobot, b adalah bias, dan x adalah sampel.
Hubungan antara nilai prediksi model dan nilai sebenarnya adalah sebagai berikut:

$$y = h(x) + \varepsilon$$

Di mana, y mewakili nilai sebenarnya dan mewakili kesalahan. Kesalahan dipengaruhi oleh banyak faktor. Menurut teorema limit pusat, kesalahan mengikuti distribusi normal.

$$\varepsilon \sim N(0, \sigma^2)$$

Di mana, distribusi probabilitas nilai sebenarnya dapat diperoleh.

$$y \sim N(h(x), \sigma^2)$$

Berdasarkan estimasi kemungkinan maksimum, tujuan optimasi model adalah

$$\arg\max_h \prod_{i=1}^{m} P(Y = y_i | X = x_i) = \arg\max_h \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(h(x_i) - y_i)^2}{2\sigma^2}\right)$$

Di mana, argmax mewakili titik maksimum, yaitu h yang memaksimalkan nilai fungsi objektif. Dalam fungsi objektif, $\left(\sqrt{2\pi\sigma}\right)^{-1}$adalah konstanta yang tidak ada hubungannya dengan h, dan perkalian atau pembagian fungsi objektif dengan konstanta tidak akan mengubah posisi titik maksimum, sehingga tujuan optimasi model dapat diubah menjadi karena fungsi logaritma bersifat monotonik, mengambil ln untuk fungsi objektif tidak akan mempengaruhi titik maksimum.

$$\arg\max_h \ln\left(\prod_{i=1}^{m}\exp\left(-\frac{(h(x_i)-y_i)^2}{2\sigma^2}\right)\right) = \arg\max_h \sum_{i=1}^{m}\frac{(h(x_i)-y_i)^2}{2\sigma^2}$$

Dengan mengambil negatif dari fungsi objektif, titik maksimum awal akan menjadi titik minimum. Pada saat yang sama, kita juga dapat mengalikan fungsi objektif dengan konstanta untuk mengubah tujuan optimasi model menjadi

$$\arg\max_h \frac{1}{2m}\sum_{i=1}^{m}(h(x_i)-y_i)^2$$

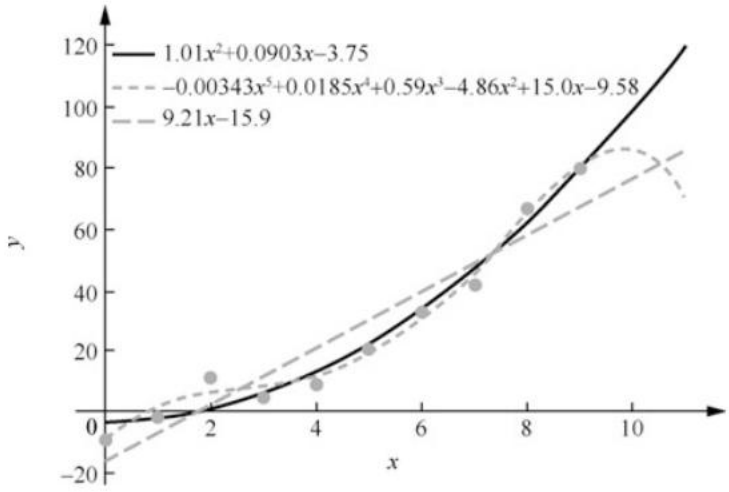
Jelasnya, fungsi kerugiannya adalah

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(h(x_i)-y_i)^2$$

Kami berharap nilai prediksi sedekat mungkin dengan nilai sebenarnya, yaitu untuk meminimalkan nilai kerugian. Metode penurunan gradien dapat digunakan untuk menemukan parameter bobot w ketika fungsi kerugian diminimalkan, dan kemudian konstruksi model selesai.

Regresi polinomial merupakan cabang dari regresi linier. Umumnya, kompleksitas dataset akan melebihi kemungkinan pencocokan dengan garis lurus, artinya, menggunakan model regresi linier asli jelas akan menghasilkan underfitting. Solusinya adalah dengan menggunakan regresi polinomial, seperti yang ditunjukkan pada Gambar 2.32, rumusnya adalah

$$h(x) = w_1x + w_2x^2 + \cdots + w_nx^n + b$$



Gambar 2.32 Perbandingan regresi linier dan regresi polinomial

Di mana, n merepresentasikan dimensi regresi polinomial. Dimensi regresi polinomial merupakan hiperparameter. Jika dipilih secara sembarangan, hal ini dapat menyebabkan overfitting. Menerapkan regularisasi membantu mengurangi overfitting. Metode regularisasi yang paling umum adalah menambahkan kerugian penjumlahan kuadrat di atas fungsi objektif.

$$h(x) = w_1 x + w_2 x^2 + \cdots + w_n x^n + b$$

Di mana $\|*\|_2$ mewakili suku reguler L2. Model regresi linier yang menggunakan fungsi kerugian ini juga disebut model regresi ridge. Demikian pula, model regresi linier dengan kerugian nilai absolut yang ditambahkan disebut model regresi Lasso, dan rumusnya adalah

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(h(x_i) - y_i)^2 + \lambda \sum \|\boldsymbol{w}\|_1$$

Di mana $\|*\|_1$ mewakili suku reguler L1.

Regresi Logistik

Model regresi logistik adalah model klasifikasi yang digunakan untuk menyelesaikan masalah klasifikasi. Definisi modelnya adalah sebagai berikut:

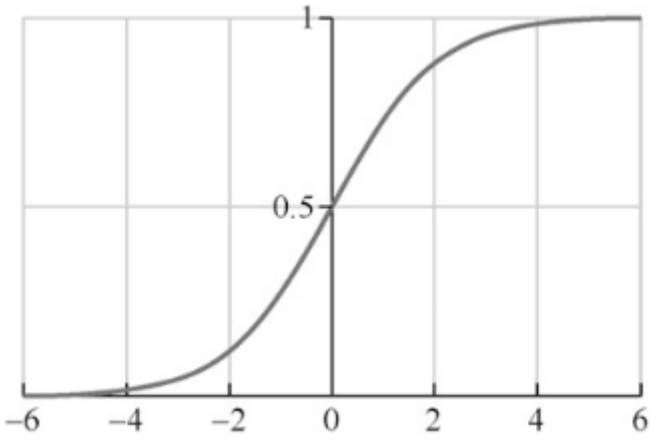
$$h(x) = P(Y = 1|X) = g(w^{\mathrm{T}}x + b)$$

Di mana g mewakili fungsi sigmoid, w mewakili bobot, dan b, bias. Dalam rumus tersebut, adalah fungsi linear dari x, sehingga regresi logistik, seperti regresi linear, termasuk dalam model linear umum.

Definisi fungsi sigmoid adalah sebagai berikut:

$$g(x) = \frac{1}{1 + \exp\{-x\}}$$

Citra fungsi sigmoid ditunjukkan pada Gambar 2.33. Dengan membandingkan hubungan magnitudo antara $\mathrm{P}(\mathrm{Y} = 1|\mathrm{X})$ dan ambang batas t, hasil klasifikasi yang sesuai dengan x dapat diperoleh. Ambang batas t di sini merupakan hiperparameter model, yang dapat dipilih secara acak. Dapat dilihat bahwa ketika ambang batas besar, model cenderung menilai sampel sebagai contoh negatif, sehingga tingkat presisi akan lebih tinggi; ketika ambang batas lebih kecil, model cenderung menilai sampel sebagai contoh positif, sehingga tingkat perolehan kembali akan lebih tinggi. Umumnya, 0,5 dapat digunakan sebagai ambang batas.



Gambar 2.33 Fungsi sigmoid

Menurut gagasan estimasi kemungkinan maksimum, ketika sampel merupakan contoh positif, kita mengharapkan $\mathrm{P}(\mathrm{Y} = 1|\mathrm{X})$ lebih besar; ketika sampel merupakan contoh negatif, kita mengharapkan $\mathrm{P}(\mathrm{Y} = 0|\mathrm{X})$ lebih besar. Dengan kata lain, kita mengharapkan persamaan berikut sebesar mungkin berapa pun ukuran sampelnya:

$$p = \mathrm{P}(\mathrm{Y} = 1|\mathrm{X})^{y}\mathrm{P}(\mathrm{Y} = 0|\mathrm{X})^{1-y}$$

Ganti $P(Y = 1|X)$ dan $P(Y = 0|X)$ dengan $h(x)$ untuk mendapatkan

$$P = h(x)^{y} \cdot \big(1 - h(x)\big)^{1-y}$$

Oleh karena itu, tujuan optimasi model adalah

$$\arg\max_{h} \prod_{i=1}^{m} P_i = \arg\max_{h} \prod_{i=1}^{m} h(x)^{y}\big(1 - h(x)\big)^{1-y}$$

Proses derivasi, mirip dengan regresi linier, dapat mengambil logaritma fungsi objektif tanpa mengubah posisi titik maksimum. Oleh karena itu, tujuan optimasi model ini setara dengan

$$\arg\max_h \sum_{i=1}^{m} \big(y \ln h(x) + (1-y) \ln\big(1-h(x)\big)\big)$$

Mengalikan fungsi objektif dengan konstanta $-1/m$ akan menyebabkan titik maksimum asli menjadi titik nilai minimum, yaitu

$$\arg\max_h -\frac{1}{m}\sum_{i=1}^{m} \big(y \ln h(x) + (1-y) \ln\big(1-h(x)\big)\big)$$

Oleh karena itu, fungsi kerugian dari regresi logistik adalah

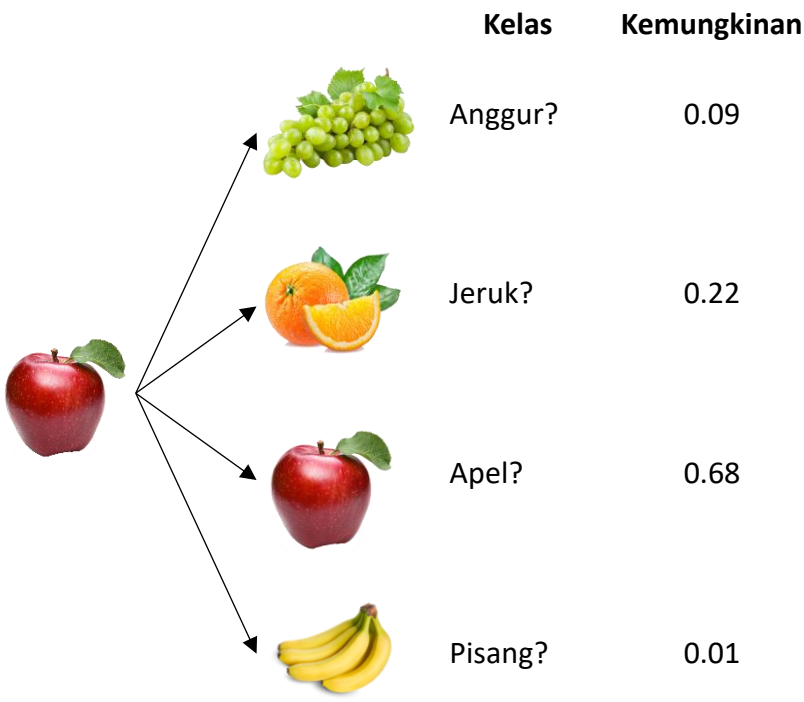
$$J(w) = -\frac{1}{m}\sum \big(y \ln h(x) + (1-y) \ln\big(1-h(x)\big)\big)$$

Dengan w mewakili parameter bobot, m adalah jumlah sampel, x adalah sampel, dan y adalah nilai sebenarnya. Nilai parameter bobot w juga dapat diperoleh melalui algoritma penurunan gradien.

Regresi Softmax merupakan generalisasi dari regresi logistik, yang dapat diterapkan untuk masalah klasifikasi k. Pada dasarnya, fungsi Softmax mengompresi (memetakan) vektor bilangan riil sembarang berdimensi k ke dalam vektor bilangan riil berdimensi k lainnya untuk merepresentasikan distribusi probabilitas kategori sampel. Fungsi kepadatan probabilitas regresi Softmax adalah sebagai berikut:

$$P(Y=c|x) = \frac{\exp\{w_c^T x + b\}}{\sum_{l=1}^{k} \exp\{w_c^T x + b\}}$$

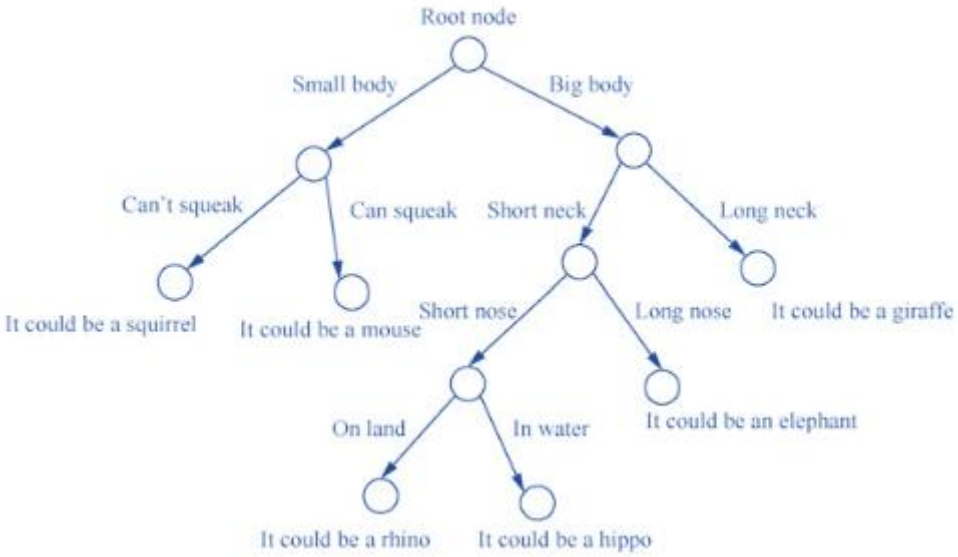
Seperti yang ditunjukkan pada Gambar 2.34, fungsi softmax menetapkan nilai probabilitas untuk setiap kategori dalam masalah multikelas, dan probabilitas ini berjumlah 1. Di antara kategori-kategori ini, nilai probabilitas kategori sampel yang berupa apel adalah yang terbesar, yaitu 0,68, sehingga nilai prediksi sampelnya seharusnya adalah apel.



Gambar 2.34 Contoh fungsi softmax

Pohon Keputusan

Pohon keputusan adalah pengklasifikasi berstruktur pohon (biner atau non-biner), seperti yang ditunjukkan pada Gambar 2.35. Setiap simpul non-daun merepresentasikan pengujian pada atribut fitur, setiap cabang merepresentasikan keluaran atribut fitur ini dalam rentang nilai tertentu, dan setiap simpul daun menyimpan sebuah kategori. Proses penggunaan pohon keputusan untuk membuat keputusan dimulai dari simpul akar, menguji atribut fitur yang sesuai pada item yang akan diklasifikasikan, dan memilih cabang keluaran sesuai nilainya hingga mencapai simpul daun, dan kategori yang tersimpan dalam simpul daun digunakan sebagai hasil keputusan.



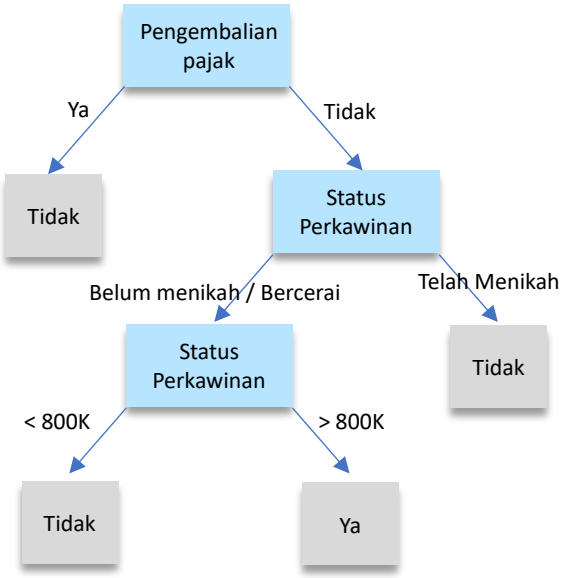
Gambar 2.35 Contoh Pohon Keputusan

Hal terpenting dalam model pohon keputusan adalah struktur pohonnya. Konstruksi dari apa yang disebut pohon keputusan adalah memilih atribut untuk menentukan struktur topologi antar setiap atribut fitur. Langkah kunci dalam membangun pohon keputusan adalah melakukan operasi pembagian berdasarkan semua atribut fitur, membandingkan kemurnian himpunan hasil dari semua operasi pembagian, dan memilih atribut dengan kemurnian tertinggi sebagai titik data dari himpunan data terpisah. Algoritma pembelajaran pohon keputusan adalah algoritma yang membangun pohon keputusan, dan algoritma yang umum digunakan meliputi ID3, C4.5, dan CART. Perbedaan antara algoritma-algoritma ini terutama terletak pada indikator kuantitatif kemurnian, seperti entropi informasi dan koefisien Gini:

$$H(X) = -\sum_{k=1}^{K} p_k \log_2 p_k$$

$$\text{Gini} = 1 - \sum_{k=1}^{K} {p^2}_k$$

Di mana, pk merepresentasikan probabilitas sampel termasuk dalam kelas k, dan K merepresentasikan jumlah total kategori. Semakin besar perbedaan kemurnian sebelum dan sesudah segmentasi, semakin kondusif penilaian fitur tertentu untuk peningkatan akurasi model, yang mengindikasikan bahwa fitur tersebut harus ditambahkan ke model pohon keputusan.



No	Pengembalian Pajak	Status Pernikahan	Pemasukan Pajak (Rp)	Penipuan Pajak
1	Ya	Lajang	1.250K	Tidak
2	Tidak	Menikah	1.000K	Tidak
3	Tidak	Lajang	700K	Tidak
4	Ya	Menikah	1.200K	Tidak
5	Tidak	Cerai	950K	Ya
6	Tidak	Menikah	600K	Tidak
7	Ya	Cerai	2.200K	Tidak
8	Tidak	Lajang	850K	Ya
9	Tidak	Menikah	750K	Tidak
10	Tidak	Lajang	900K	Ya

Gambar 2.36 Membangun pohon keputusan

Secara umum, proses membangun pohon keputusan terdiri dari tiga tahap berikut.

1. Pemilihan fitur: memilih fitur dari fitur-fitur data latih sebagai kriteria pemisahan untuk simpul saat ini (kriteria yang berbeda untuk pemilihan fitur menghasilkan algoritma pohon keputusan yang berbeda).
2. Pembangkitan pohon keputusan: Berdasarkan kriteria evaluasi fitur yang dipilih, simpul anak dibangkitkan secara rekursif dari atas ke bawah hingga dataset tidak dapat dipisahkan, kemudian pertumbuhan pohon keputusan dihentikan.
3. Pemangkasan: Dengan mengurangi ukuran pohon untuk menekan overfitting model, model dapat dibagi menjadi pra-pemangkasan dan pasca-pemangkasan.

Gambar 2.36 menunjukkan kasus klasifikasi menggunakan model pohon keputusan. Hasil klasifikasi dipengaruhi oleh tiga atribut: restitusi pajak, status perkawinan, dan penghasilan kena pajak. Dari contoh ini, kita dapat melihat bahwa model pohon keputusan tidak hanya dapat menangani kasus di mana atribut memiliki dua nilai, tetapi juga kasus di mana atribut memiliki beberapa nilai atau bahkan nilai kontinu. Selain itu, model pohon keputusan dapat diinterpretasikan, dan kita dapat secara intuitif menganalisis hubungan kepentingan antar atribut berdasarkan diagram struktur yang ditunjukkan pada Gambar 2.36b.

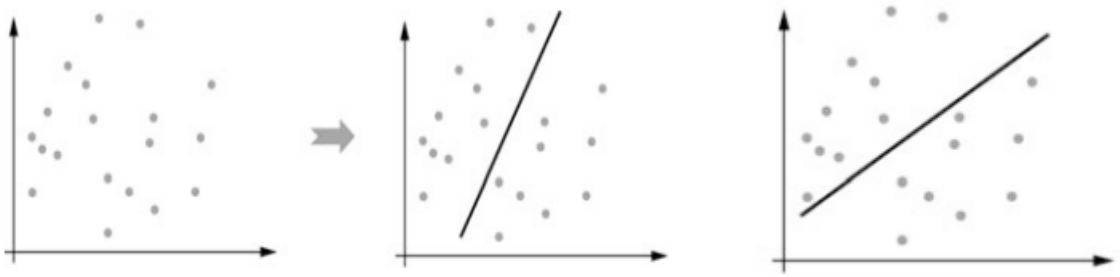
Mesin Vektor Pendukung

Mesin vektor pendukung (SVM) adalah pengklasifikasi linier dengan interval terbesar yang didefinisikan dalam ruang fitur. Algoritma pembelajaran SVM merupakan algoritma optimal untuk menyelesaikan pemrograman linier kuadratik konveks. Singkatnya, konsep inti SVM mencakup dua aspek berikut.

1. Mencari hiperbidang optimal dalam ruang fitur berdasarkan teori minimisasi risiko struktural, sehingga pembelajar memperoleh optimasi global, dan ekspektasi di seluruh ruang sampel memenuhi batas atas tertentu dengan probabilitas tertentu.
2. Untuk data yang tidak dapat dipisahkan secara linear, petakan sampel yang tidak dapat dipisahkan secara linear dari ruang input berdimensi rendah ke ruang fitur berdimensi

tinggi agar dapat dipisahkan secara linear berdasarkan algoritma pemetaan nonlinier, sehingga ruang fitur berdimensi tinggi mengadopsi algoritma linear untuk nonlinieritas sampel. Analisis linear fitur menjadi mungkin.

Garis lurus digunakan untuk membagi data ke dalam beberapa kategori, tetapi sebenarnya kita dapat menemukan beberapa garis lurus untuk memisahkan data, seperti yang ditunjukkan pada Gambar 2.37. Ide inti SVM adalah menemukan garis lurus yang memenuhi kondisi di atas, dan membuat titik-titik yang paling dekat dengan garis lurus tersebut sejauh mungkin dari garis lurus tersebut. Hal ini akan memberikan model kemampuan generalisasi yang kuat. Titik-titik yang paling dekat dengan garis lurus ini disebut vektor pendukung.

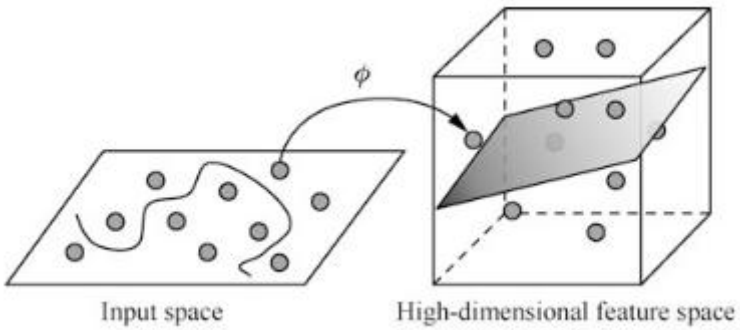


(a) dataset dua dimensi dengan dua klasifikasi

(b) Titik-titik di sisi kiri dan kanan dapat dipisahkan. Dataset mana yang benar ?

Gambar 2.37 Kinerja pengklasifikasi linier

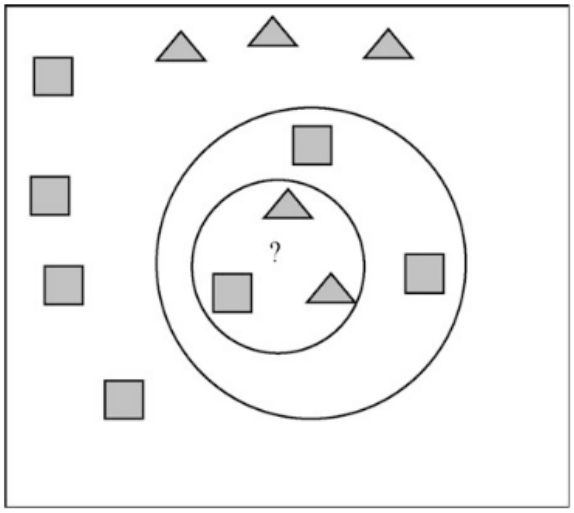
SVM linear dapat bekerja dengan baik pada dataset linear yang dapat dipisahkan, tetapi kita tidak dapat menggunakan garis lurus untuk membagi dataset non-linear. Pada saat ini, fungsi kernel diperlukan untuk membangun SVM non-linear. Fungsi kernel memungkinkan algoritma untuk menyesuaikan bidang-hiper dalam ruang fitur berdimensi tinggi yang telah ditransformasikan, seperti yang ditunjukkan pada Gambar 2.38. Fungsi kernel yang umum meliputi fungsi kernel linear, fungsi kernel polinomial, fungsi kernel sigmoid, dan fungsi kernel Gaussian. Fungsi kernel Gaussian dapat memetakan sampel ke ruang berdimensi tak hingga, sehingga efeknya juga lebih baik, dan merupakan salah satu fungsi kernel yang paling umum digunakan.



Gambar 2.38 Fungsi kernel

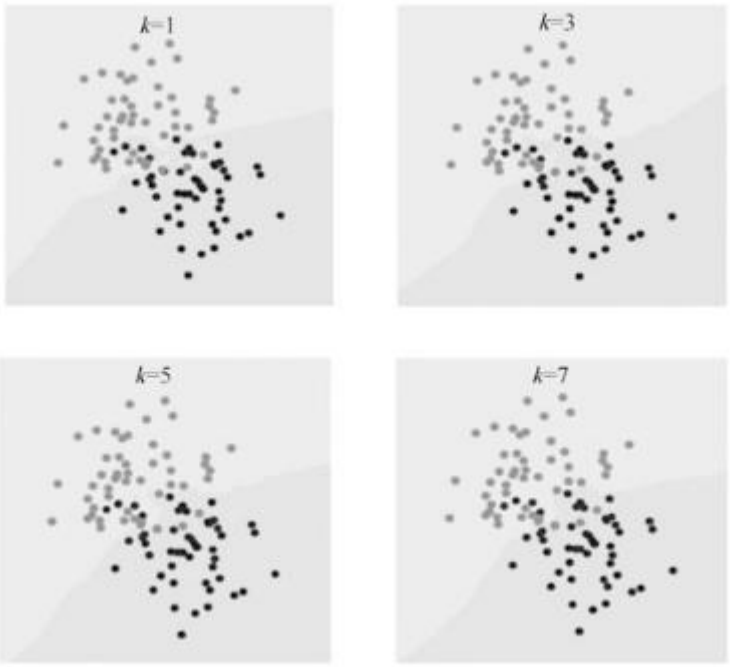
Algoritma K-Nearest Neighbor

Algoritma K-nearest neighbor (KNN) merupakan metode yang secara teoritis sudah matang dan merupakan salah satu algoritma pembelajaran mesin yang paling sederhana. Algoritma KNN merupakan metode non-parametrik, yang cenderung berkinerja lebih baik dalam kumpulan data dengan batas keputusan yang tidak teratur. Ide di balik metode ini adalah: jika sebagian besar K sampel terdekat (yaitu, tetangga terdekat dalam ruang fitur) dari suatu sampel dalam ruang fitur termasuk dalam kategori tertentu, maka sampel tersebut juga termasuk dalam kategori ini.



Gambar 2.39 Contoh algoritma KNN

Konsep inti dari algoritma KNN adalah "Apa yang ada di sekitar cinnabar menjadi merah, dan apa yang ada di sekitar tinta menjadi hitam", yang menampilkan logika ringkas. Namun, seperti validasi silang k-fold, K dalam algoritma KNN juga merupakan hiperparameter. Ini berarti sulit untuk memilih nilai K dengan tepat. Seperti yang ditunjukkan pada Gambar 2.39, ketika nilai K adalah 3, hasil prediksi pada tanda tanya akan berupa segitiga; dan ketika nilai K adalah 5, hasil prediksi pada tanda tanya akan menjadi persegi. Gambar 2.40 menunjukkan batas keputusan untuk berbagai nilai K. Dapat ditemukan bahwa ketika nilai K meningkat, batas keputusan akan menjadi lebih halus. Secara umum, nilai K yang lebih besar akan mengurangi dampak noise pada klasifikasi tetapi akan membuat batas kelas kurang jelas. Semakin besar nilai K, semakin besar kemungkinan menyebabkan under-fitting, karena batas keputusan terlalu kabur. Dengan demikian, semakin kecil nilai K, semakin mudah menyebabkan over-fitting, karena batas keputusan terlalu tajam.



Gambar 2.40 Pengaruh nilai K terhadap batas keputusan

Algoritma KNN tidak hanya dapat digunakan untuk masalah klasifikasi, tetapi juga untuk masalah regresi. Dalam masalah prediksi klasifikasi, biasanya mengadopsi metode voting mayoritas. Dalam masalah prediksi regresi, metode rata-rata banyak digunakan. Meskipun metode ini tampaknya hanya tentang sampel K dari tetangga terdekat, volume komputasi algoritma KNN sebenarnya sangat besar. Hal ini karena algoritma KNN perlu melintasi semua sampel untuk menentukan K sampel mana yang merupakan tetangga terdekat dengan sampel yang akan diuji.

Naive Bayes

Naive Bayes adalah algoritma multiklasifikasi sederhana berdasarkan teorema Bayes dan mengasumsikan bahwa fitur-fiturnya independen. Dengan fitur sampel X, probabilitas sampel tersebut termasuk dalam kelas c adalah

$$P(C = c|X) = \frac{P(X|C = c)P(C = c)}{P(X)}$$

Di mana, $P(C = c|X)$ disebut probabilitas posterior, $P(C = c)$ mewakili probabilitas sebelumnya dari target, dan $P(X)$ mewakili probabilitas sebelumnya dari fitur. Biasanya kita tidak mempertimbangkan $P(X)$, karena $P(X)$ dapat dilihat sebagai nilai tetap saat mengklasifikasikan, yaitu

$$P(C = c|X) \propto P(X|C = c)P(C = c)$$

$P(C = c)$ tidak ada hubungannya dengan X dan perlu ditentukan sebelum melatih model. Umumnya, proporsi sampel dengan kategori c dalam dataset dihitung sebagai $P(C = c)$.

Dapat dilihat bahwa inti dari klasifikasi adalah menemukan $P(X|\ C\ =\ c)$. Misalkan fitur X terdiri dari elemen-elemen berikut:

$$X = (X_1, X_2, \cdots, X_n)$$

Secara umum, dapat dengan mudah dihitung bahwa

$$\prod_{i=1}^{n} P(X_i|C = c)$$

Dengan menggabungkan asumsi independensi kondisional atribut, kita dapat membuktikan

$$P(X|C = c) = \prod_{i=1}^{n} P(X_i|C = c)$$

Asumsi independensi kondisional atribut menyatakan bahwa dengan klasifikasi sampel sebagai suatu kondisi, distribusi setiap nilai atribut bersifat independen terhadap distribusi nilai atribut lainnya. Alasan mengapa Naive Bayes bersifat naif justru karena asumsi independensi atribut yang digunakan dalam modelnya. Asumsi ini secara efektif menyederhanakan perhitungan dan memberikan pengklasifikasi Bayesian akurasi dan kecepatan pelatihan yang lebih tinggi pada basis data besar.

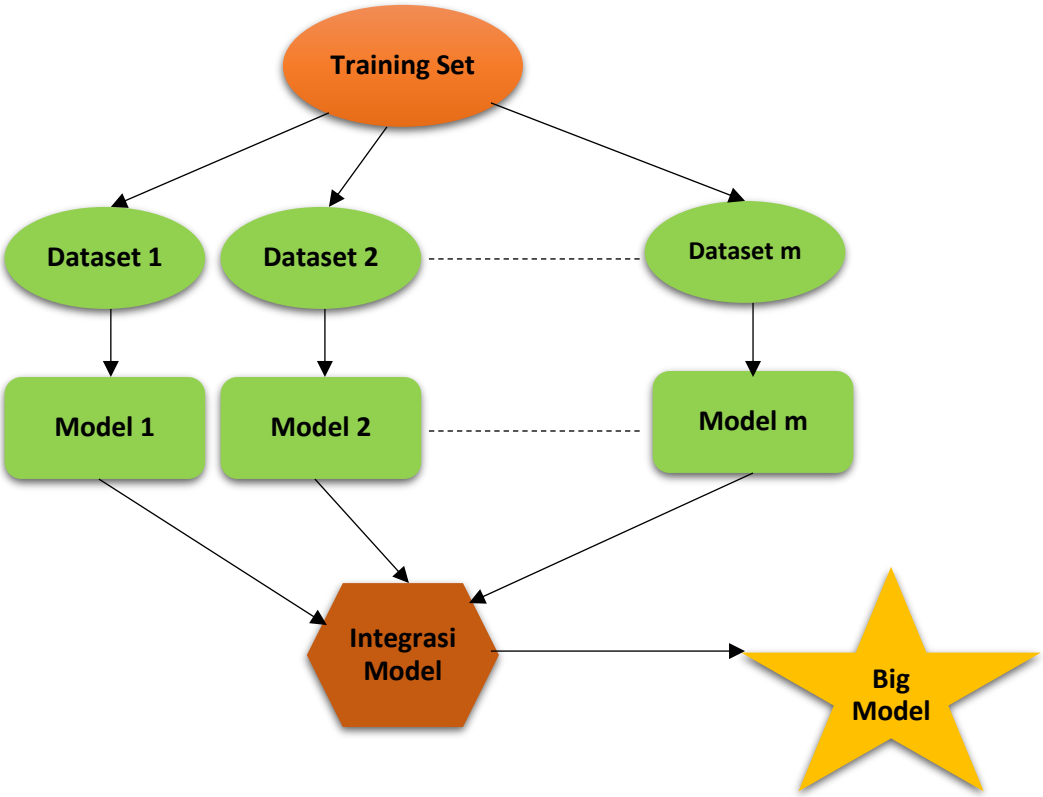
Berikut contohnya. Kita ingin menilai jenis kelamin seseorang C berdasarkan tinggi badan $X1$ dan berat badan $X2$. Misalkan probabilitas seseorang dengan tinggi badan 180 cm dan tinggi badan 150 cm adalah laki-laki masing-masing adalah 80% dan 20%, dan probabilitas seseorang dengan berat badan 80 kg dan 50 kg adalah laki-laki masing-masing adalah 70% dan 30%. Menurut model Naive Bayes, probabilitas seseorang dengan tinggi badan 180 cm dan berat badan 50 kg adalah laki-laki adalah $0{,}8 \cdot 0{,}3 \cdot 0{,}24$, sedangkan probabilitas seseorang dengan tinggi badan 150 cm dan berat badan 80 kg adalah laki-laki hanya $0{,}7 \cdot 0{,}2 \cdot 0{,}14$. Dapat diasumsikan bahwa kedua fitur, tinggi badan dan berat badan, secara independen berkontribusi terhadap probabilitas bahwa orang tersebut adalah laki-laki.

Kinerja model Naive Bayes biasanya bergantung pada tingkat kepuasan hipotesis independensi fitur. Sebagaimana disebutkan dalam contoh sebelumnya, kedua fitur, tinggi badan dan berat badan, tidak sepenuhnya independen. Korelasi ini pasti akan memengaruhi akurasi model, tetapi selama korelasinya tidak besar, kita dapat terus menggunakan model Naive Bayes. Dalam aplikasi praktis, fitur yang berbeda jarang sepenuhnya independen.

Pembelajaran Ensemble

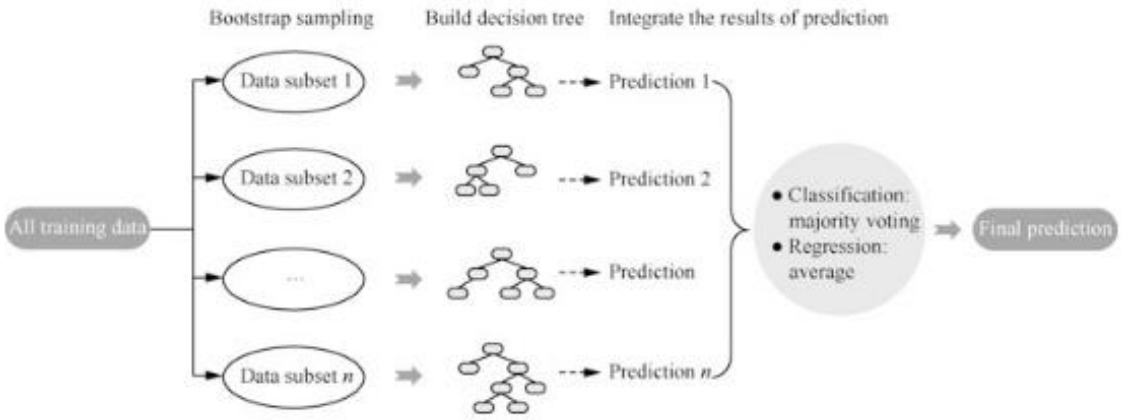
Pembelajaran terintegrasi adalah paradigma pembelajaran mesin. Dalam paradigma ini, beberapa pembelajar dilatih dan digabungkan untuk memecahkan masalah yang sama, seperti yang ditunjukkan pada Gambar 2.41. Berkat banyaknya pembelajar yang terlibat, kemampuan generalisasi pembelajaran ensemble bisa jauh lebih kuat daripada menggunakan

satu pembelajar. Bayangkan Anda secara acak mengajukan pertanyaan rumit kepada ribuan orang, lalu menggabungkan jawaban mereka. Dalam kebanyakan kasus, jawaban terintegrasi ini bahkan lebih baik daripada jawaban yang diberikan oleh seorang pakar. Inilah kecerdasan kolektif yang kita bicarakan.



Gambar 2.41 Pembelajaran ansambel

Metode implementasi pembelajaran ensemble dapat diklasifikasikan menjadi dua jenis bagging dan boosting. Bagging secara independen membangun beberapa pembelajar dasar, lalu merata-ratakan prediksi mereka. Model-model Bagging yang umum mencakup hutan acak dan sebagainya. Rata-rata, hasil prediksi pembelajar gabungan biasanya lebih baik daripada pembelajar dasar tunggal mana pun karena variansnya berkurang. Boosting membangun pembelajar dasar secara berurutan, dan secara bertahap mengurangi deviasi prediksi pembelajar komprehensif. Model-model Boosting yang umum meliputi Adaboost, GBDT, dan XGboost. Secara umum, Bagging dapat mengurangi varians, sehingga menekan over-fitting; sementara Boosting berfokus pada pengurangan deviasi, sehingga meningkatkan kapasitas model, tetapi dapat menyebabkan over-fitting.

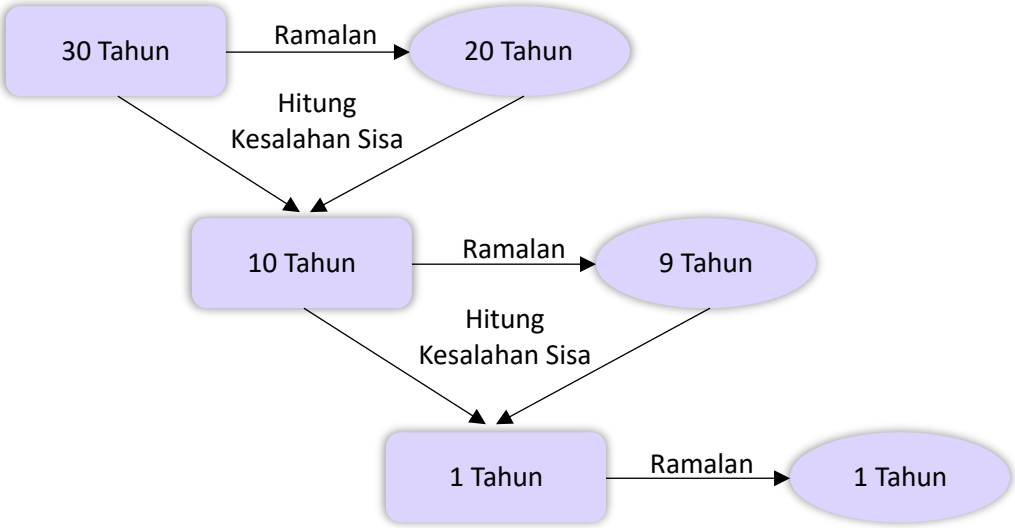


Gambar 2.42 Algoritma hutan acak

Algoritma hutan acak merupakan kombinasi dari metode bagging dan pohon keputusan CART. Proses keseluruhan algoritma ditunjukkan pada Gambar 2.42. Algoritma hutan acak dapat digunakan untuk masalah klasifikasi dan regresi. Prinsip dasarnya adalah membangun beberapa pohon keputusan dan menggabungkannya untuk menghasilkan prediksi yang lebih akurat dan stabil. Selama proses pelatihan pohon keputusan, pengambilan sampel dilakukan pada dua tingkat sampel dan fitur secara bersamaan. Pada tingkat sampel, pengambilan sampel bootstrap (pengambilan sampel dengan penggantian) digunakan untuk menentukan subset sampel yang digunakan untuk pelatihan pohon keputusan. Pada tingkat fitur, sebelum setiap simpul pohon keputusan dipecah, beberapa fitur dipilih secara acak untuk menghitung perolehan informasi.

Dengan mensintesis hasil prediksi beberapa pohon keputusan, model hutan acak dapat mengurangi varians dari model pohon keputusan tunggal, tetapi efek koreksi deviasinya tidak memuaskan. Oleh karena itu, model hutan acak mensyaratkan bahwa setiap pohon keputusan tidak boleh mengalami underfitting, meskipun persyaratan ini dapat menyebabkan beberapa pohon keputusan mengalami overfitting. Perlu dicatat juga bahwa setiap model pohon keputusan dalam hutan acak bersifat independen, sehingga proses pelatihan dan prediksi dapat dilakukan secara paralel.

Pohon keputusan penguat gradien (GBDT) adalah salah satu metode penguat. Nilai prediksi model adalah jumlah hasil dari semua pohon keputusan. Inti dari GBDT adalah terus menggunakan pohon keputusan baru untuk mempelajari residual dari semua pohon keputusan sebelumnya, yaitu, kesalahan antara nilai prediksi dan nilai sebenarnya. Seperti yang ditunjukkan pada Gambar 2.43, untuk sampel tertentu, hasil prediksi pohon keputusan pertama adalah 20 tahun, sedangkan usia sebenarnya dari sampel adalah 30. Perbedaan antara hasil prediksi dan nilai sebenarnya adalah 10. Jika kita dapat memprediksi perbedaan ini dengan pohon keputusan lain, kita dapat meningkatkan hasil prediksi 20 dan membuatnya lebih dekat ke 30. Berdasarkan ide ini, kami memperkenalkan pohon keputusan kedua untuk mempelajari kesalahan pohon keputusan pertama, dan seterusnya.

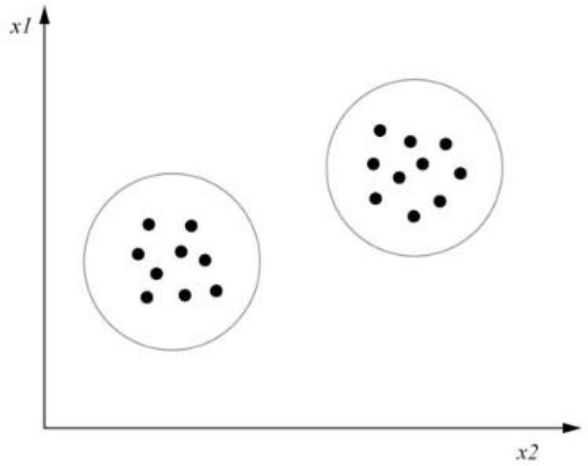


Gambar 2.43 Algoritma GBDT

Akhirnya, hasil prediksi dari tiga pembelajar ditambahkan bersama untuk mendapatkan nilai sebenarnya 30. GBDT meningkatkan akurasi dengan terus mengoreksi deviasi pohon keputusan, sehingga memungkinkan tingkat underfitting tertentu dari pohon keputusan. Namun, GBDT tidak dapat mengoreksi varians, sehingga umumnya tidak diperbolehkan untuk melakukan overfitting pada pohon keputusan. Hal ini juga merupakan salah satu perbedaan terbesar antara metode boosting dan bagging. Selain itu, data pelatihan setiap pohon keputusan dalam GBDT bergantung pada keluaran dari pohon keputusan sebelumnya, sehingga proses pelatihan tidak dapat diparalelkan.

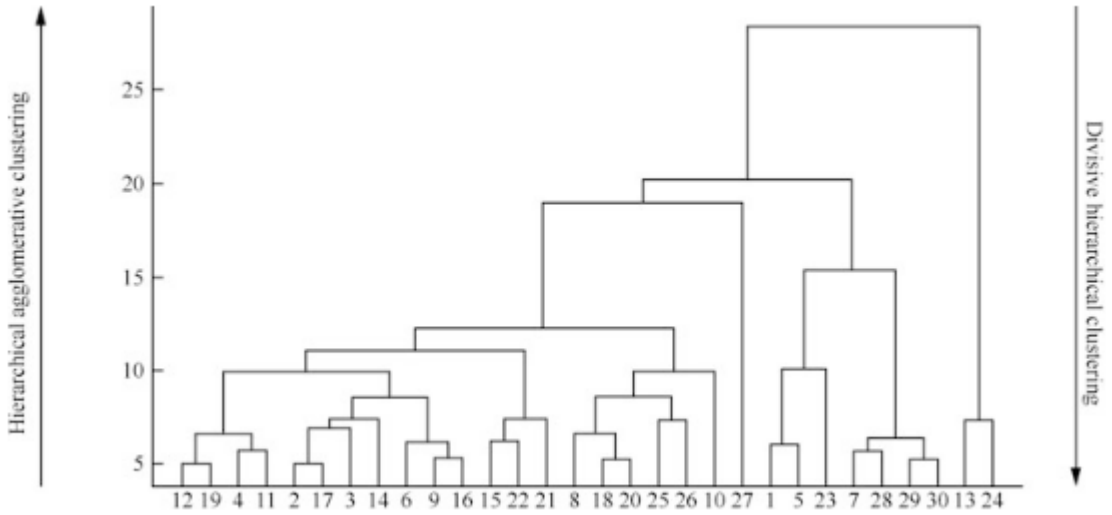
Algoritma Pengelompokan

Algoritma pengelompokan K-means (K-Means clustering) adalah algoritma yang memasukkan jumlah kluster K dan dataset yang berisi n objek data, dan menghasilkan K kluster yang memenuhi standar varians minimum, seperti yang ditunjukkan pada Gambar 2.44. Gambar ini menunjukkan bahwa kluster akhir yang diperoleh memenuhi: kesamaan objek dalam kluster yang sama lebih tinggi; dan kesamaan objek dalam kluster yang berbeda lebih rendah.



Gambar 2.44 Algoritma K-Means

Dibandingkan dengan algoritma K-Means, algoritma pengelompokan hierarkis juga menghasilkan hubungan seperti pohon antar sampel saat menghasilkan kluster. Seperti yang ditunjukkan pada Gambar 2.45, algoritma pengelompokan hierarkis mencoba membagi dataset pada tingkat yang berbeda sehingga membentuk struktur pengelompokan berbentuk pohon. Dataset dapat dibagi dengan strategi aglomerasi "bottom-up", atau strategi pembagian "top-down". Hirarki klaster direpresentasikan sebagai diagram pohon, di mana akar pohon merepresentasikan kelas leluhur semua sampel, dan daunnya merupakan klaster dengan hanya satu sampel.



Gambar 2.45 Algoritma Pengelompokan Hirarkis

2.6 STUDI KASUS

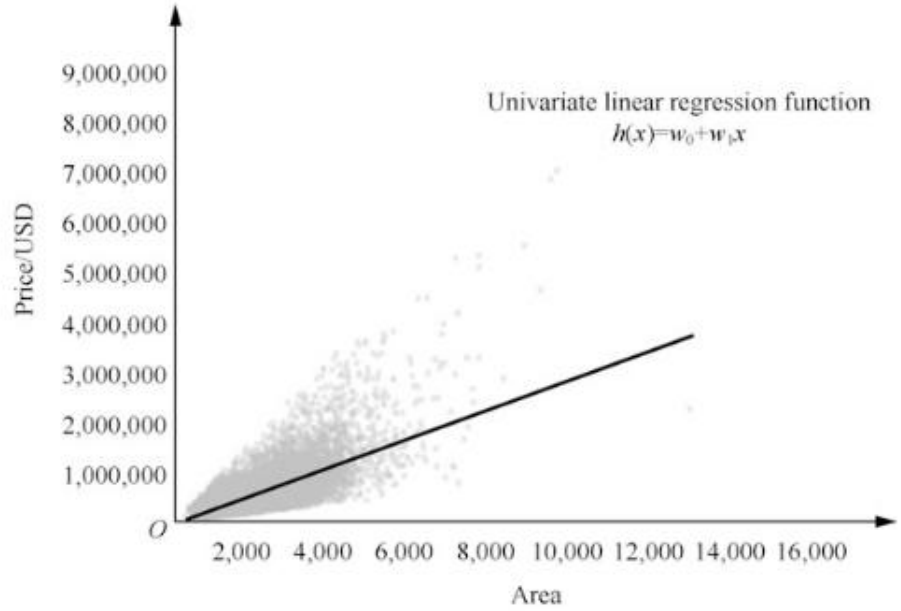
Di akhir bab ini, kita akan meninjau keseluruhan proses proyek pembelajaran mesin dengan satu kasus. Misalkan terdapat kumpulan data yang memberikan luas hunian (1 kaki persegi 0,09 meter persegi) dan harga 21.613 rumah yang terjual di suatu kota, seperti yang ditunjukkan pada Gambar 2.46. Berdasarkan data tersebut, kita berharap dapat melatih model untuk memprediksi harga rumah-rumah lain di kota tersebut.

x (Floor area/square foot)	y (Price/USD)
1180	221,900
2570	538,000
770	180,000
1960	604,000
1680	510,000
5420	1,225,000
1715	257,500

1060	291,850
1160	468,000
1430	310,000
1370	400,000
1810	530,000
…	…

Gambar 2.46 Dataset harga perumahan

Dari data dalam kumpulan data harga rumah, dapat disimpulkan bahwa input (luas rumah) dan output (harga) dalam data tersebut merupakan nilai kontinu, sehingga model regresi dalam pembelajaran terbimbing dapat digunakan. Tujuan proyek ini adalah membangun fungsi model h(x) agar model tersebut mendekati fungsi yang menyatakan distribusi sebenarnya dari kumpulan data tersebut secara tak terhingga. Gambar 2.47 menunjukkan diagram sebar data dan kemungkinan fungsi model.



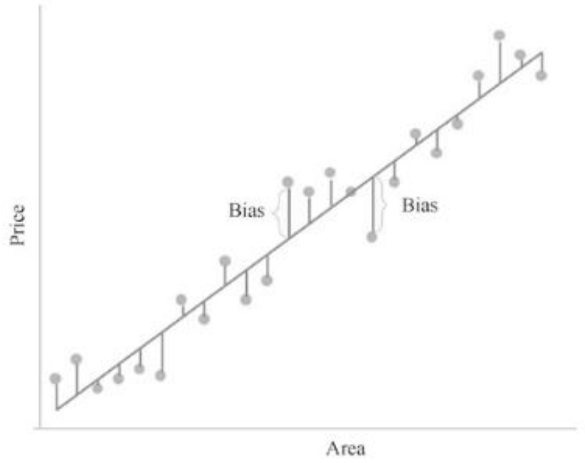
Gambar 2.47 Asumsi model/kaki persegi

Tujuan regresi linier adalah menemukan garis lurus yang paling sesuai dengan dataset, yaitu menentukan parameter dalam model. Untuk menemukan parameter terbaik, kita perlu membangun fungsi kerugian dan menemukan nilai parameter ketika fungsi kerugian mencapai nilai minimum. Persamaan fungsi kerugiannya adalah sebagai berikut:

$$J(w) = \frac{1}{2m}\sum(h(x) - y)^2$$

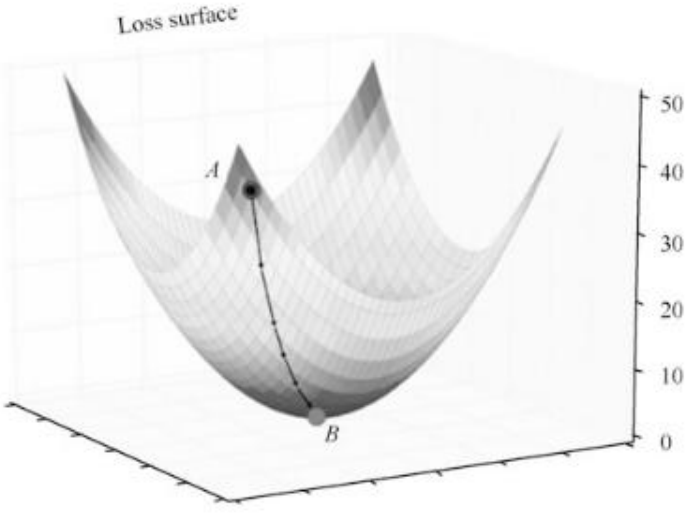
Dengan m menyatakan jumlah sampel, $h(x)$ adalah nilai prediksi, dan y adalah nilai sebenarnya. Secara intuitif, fungsi kerugian menyatakan jumlah galat kuadrat dari semua

sampel terhadap fungsi model, seperti yang ditunjukkan pada Gambar 2.48. Ketika fungsi kerugian ini dikurangi hingga minimum, semua sampel seharusnya terdistribusi secara merata di kedua sisi garis lurus yang telah disesuaikan. Pada saat ini, garis lurus yang telah disesuaikan adalah fungsi model yang kita butuhkan.



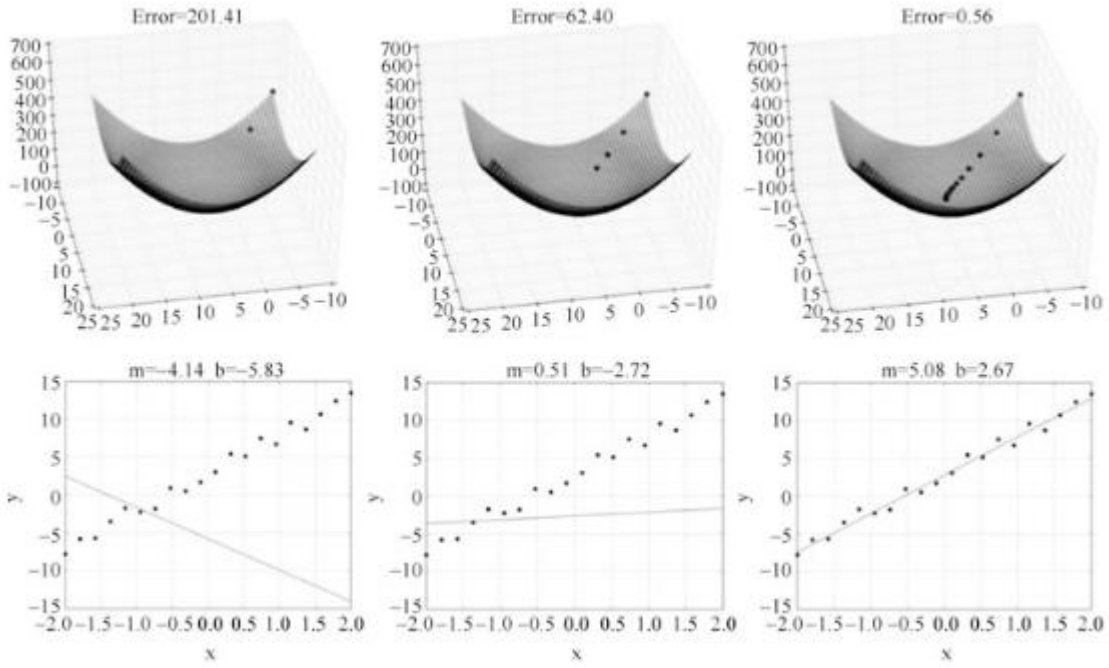
Gambar 2.48 Makna geometris kesalahan

Seperti yang telah disebutkan sebelumnya, algoritma penurunan gradien menggunakan metode iteratif untuk menemukan nilai minimum suatu fungsi. Algoritma penurunan gradien pertama-tama secara acak memilih titik awal pada fungsi kerugian, kemudian menemukan nilai minimum global dari fungsi kerugian berdasarkan arah gradien negatif. Nilai parameter pada saat ini adalah nilai parameter terbaik yang kita butuhkan, seperti yang ditunjukkan pada Gambar 2.49. Titik A menyatakan posisi di mana parameter w diinisialisasi secara acak; titik B menyatakan minimum global dari fungsi kerugian, yang merupakan nilai parameter akhir; garis terhubung antara A dan B menyatakan lintasan yang dibentuk oleh arah gradien negatif. Untuk setiap iterasi, nilai parameter w akan berubah, sehingga menghasilkan perubahan konstan pada garis regresi.



Gambar 2.49 Permukaan kerugian

Gambar 2.50 menunjukkan contoh proses iteratif menggunakan penurunan gradien. Dapat diamati bahwa ketika titik-titik pada permukaan kerugian secara bertahap mendekati titik terendah, garis kecocokan regresi linier semakin cocok dengan data. Akhirnya, kita bisa mendapatkan fungsi model terbaik $h(x) = 280{,}62x - 43{,}581$.



Gambar 2.50 Visualisasi proses penurunan gradien

Setelah pelatihan model selesai, kita perlu menggunakan set uji untuk pengujian guna memastikan bahwa model memiliki kemampuan generalisasi yang memadai. Jika terdapat overfitting dalam pengujian, kita dapat menambahkan suku reguler ke fungsi kerugian dan menyesuaikan hiperparameter. Jika underfitting, kita dapat menggunakan model regresi yang lebih kompleks, seperti GBDT. Setelah itu, model perlu dilatih ulang, dan set uji digunakan kembali untuk pengujian hingga kemampuan generalisasi model memenuhi harapan. Perlu dicatat bahwa karena data nyata digunakan dalam proyek ini, peran pembersihan data dan pemilihan fitur juga tidak dapat diabaikan.

2.7 KESIMPULAN

Bab ini terutama memperkenalkan definisi, klasifikasi, dan tantangan utama pembelajaran mesin. Selain itu, proses pembelajaran mesin secara keseluruhan (pengumpulan data, pembersihan data, ekstraksi dan pemilihan fitur, pelatihan model, evaluasi dan pengujian model, penerapan dan integrasi model, dll.), algoritma pembelajaran mesin umum (regresi linier, regresi logistik, pohon keputusan, mesin vektor pendukung, Naive Bayes, KNN, pembelajaran ensemble, K-Means, dll.), algoritma penurunan gradien, hiperparameter, dan pengetahuan penting lainnya tentang pembelajaran mesin akan dipilah

dan dijelaskan; akhirnya, melalui penggunaan regresi linier, kasus prediksi harga perumahan diselesaikan, yang menunjukkan keseluruhan proses pembelajaran mesin.

Latihan Soal

1. Pembelajaran mesin adalah teknologi inti dari kecerdasan buatan. Jelaskan definisi pembelajaran mesin.
2. Kesalahan generalisasi model dapat diklasifikasikan menjadi varians, bias, dan kesalahan tak terpecahkan. Apa perbedaan antara varians dan bias? Apa karakteristik varians dan bias dari model overfitting?
3. Sesuai dengan matriks konfusi yang ditunjukkan pada Gambar 2.25, carilah nilai F1.
4. Dalam pembelajaran mesin, seluruh dataset umumnya dibagi menjadi tiga bagian: set pelatihan, set validasi, dan set pengujian. Apa perbedaan antara set verifikasi dan set pengujian? Mengapa memperkenalkan set validasi?
5. Model regresi linier menggunakan fungsi linier untuk menyesuaikan data. Untuk data nonlinier, bagaimana cara menangani model regresi linier?
6. Banyak model klasifikasi hanya dapat menangani dua masalah klasifikasi. Dengan mengambil SVM sebagai contoh, cobalah mencari solusi untuk masalah multiklasifikasi.
7. Silakan merujuk pada informasi yang relevan dan jawab bagaimana fungsi kernel Gaussian dalam SVM memetakan fitur ke ruang berdimensi tak hingga?
8. Apakah penurunan gradien satu-satunya cara untuk melatih model? Apa saja keterbatasan metode ini?

BAB 3
TINJAUAN UMUM PEMBELAJARAN MENDALAM (DEEP LEARNING)

Sebagai model pembelajaran mesin berbasis jaringan saraf tiruan, pembelajaran mendalam sangat menguntungkan di bidang-bidang seperti visi komputer, pengenalan suara, dan pemrosesan bahasa alami. Bab ini terutama memperkenalkan pengetahuan dasar terkait pembelajaran mendalam, termasuk sejarah, komponen jaringan saraf tiruan, jenis-jenis jaringan saraf tiruan pembelajaran mendalam, dan masalah umum yang mungkin dihadapi peneliti dalam proyek pembelajaran mendalam.

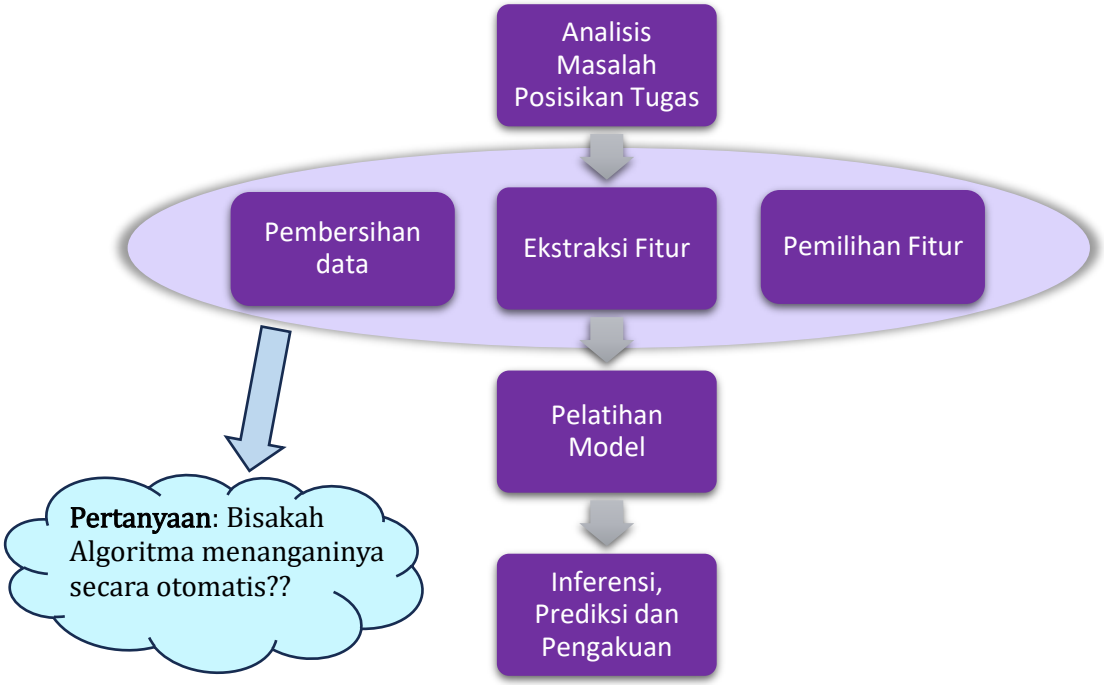
3.1 PENGANTAR PEMBELAJARAN MENDALAM

Dalam pembelajaran mesin tradisional, fitur-fitur dipilih secara manual. Semakin banyak fitur yang disertakan, semakin banyak informasi yang dapat ditransmisikan model ke dunia luar, sehingga model dianggap lebih ekspresif. Namun, seiring bertambahnya jumlah fitur, kompleksitas algoritma juga meningkat, dan ruang pencarian model juga akan meluas. Data pelatihan akan terlihat jarang di ruang fitur dan memengaruhi penilaian kesamaan. Inilah yang disebut ledakan dimensionalitas.

Yang lebih penting, jika fitur-fitur tersebut tidak kondusif untuk tugas tersebut, fitur-fitur tersebut dapat mengganggu efisiensi pembelajaran. Dibatasi oleh jumlah fitur, algoritma pembelajaran mesin tradisional lebih cocok untuk pelatihan data kecil. Ketika skala data terakumulasi hingga tingkat tertentu, akan sulit untuk meningkatkan kinerja hanya dengan menambahkan lebih banyak data. Oleh karena itu, pembelajaran mesin tradisional memiliki persyaratan perangkat keras komputer yang rendah dan kebutuhan kalkulasi yang terbatas, sehingga umumnya tidak memerlukan GPU dan kartu grafis untuk mendukung operasi paralel.

Gambar 3.1 menunjukkan proses umum pembelajaran mesin tradisional, di mana fitur-fiturnya sangat mudah diinterpretasikan karena dipilih secara manual. Namun, karena semakin banyak fitur belum tentu semakin baik, fitur berkualitas tinggi akan menentukan apakah model dapat melakukan pengenalan yang berhasil. Jumlah fitur yang dibutuhkan harus ditentukan oleh masalah yang ingin dipecahkan oleh model tersebut. Untuk menghindari bias internet terhadap pemilihan fitur oleh orang-orang, pembelajaran mendalam menggunakan algoritma yang dapat mengekstraksi fitur secara otomatis.

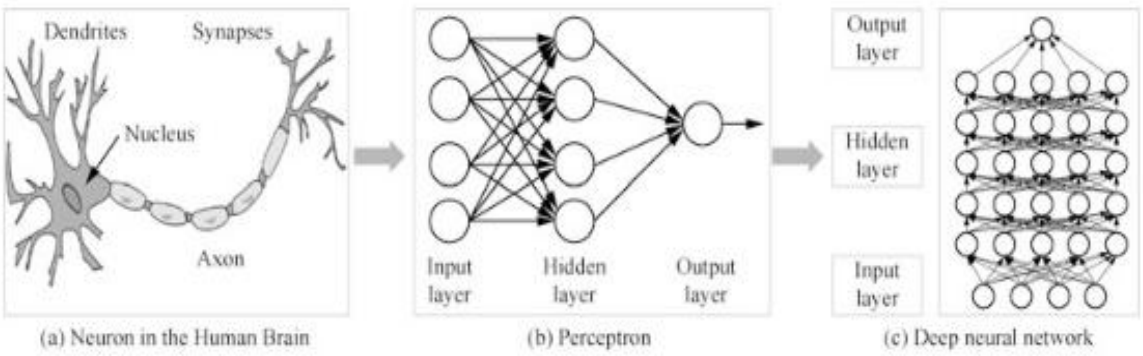
Meskipun hal ini mengurangi interpretabilitas fitur, hal ini meningkatkan kemampuan adaptasi model terhadap berbagai masalah. Selain itu, pembelajaran mendalam mengadopsi mode pembelajaran "end-to-end" dan dikombinasikan dengan parameter bobot berdimensi tinggi, yang memungkinkannya mencapai kinerja yang lebih tinggi dengan menangani data pelatihan berukuran besar dibandingkan cara tradisional. Data yang masif juga menuntut lebih banyak perangkat keras: sejumlah besar operasi matriks diproses secara lambat di CPU, sehingga GPU perlu dilibatkan untuk menyediakan akselerasi paralel.



Gambar 3.1 Proses umum pembelajaran mesin tradisional

Jaringan Saraf Tiruan Dalam

Secara umum, pembelajaran mendalam didasarkan pada jaringan saraf tiruan dalam, yaitu jaringan saraf tiruan berlapis. Ini adalah model yang dibangun untuk mensimulasikan jaringan saraf tiruan manusia. Seperti ditunjukkan pada Gambar 3.2, jaringan saraf tiruan dalam terdiri dari sekelompok perseptron yang ditumpuk, dan perseptron tersebut mensimulasikan neuron di otak manusia. Pada Gambar 3.2b, c, setiap lingkaran mewakili sebuah neuron. Dalam bagian selanjutnya, kita akan melihat kesamaan antara desain ini dan neuron-neuronnya. Penelitian desain dan aplikasi jaringan saraf tiruan biasanya perlu mempertimbangkan tiga hal, yaitu neuron sebagai fungsi, koneksi antar neuron, dan pembelajaran (pelatihan) jaringan.



Gambar 3.2 Neuron dalam otak manusia dan jaringan saraf tiruan

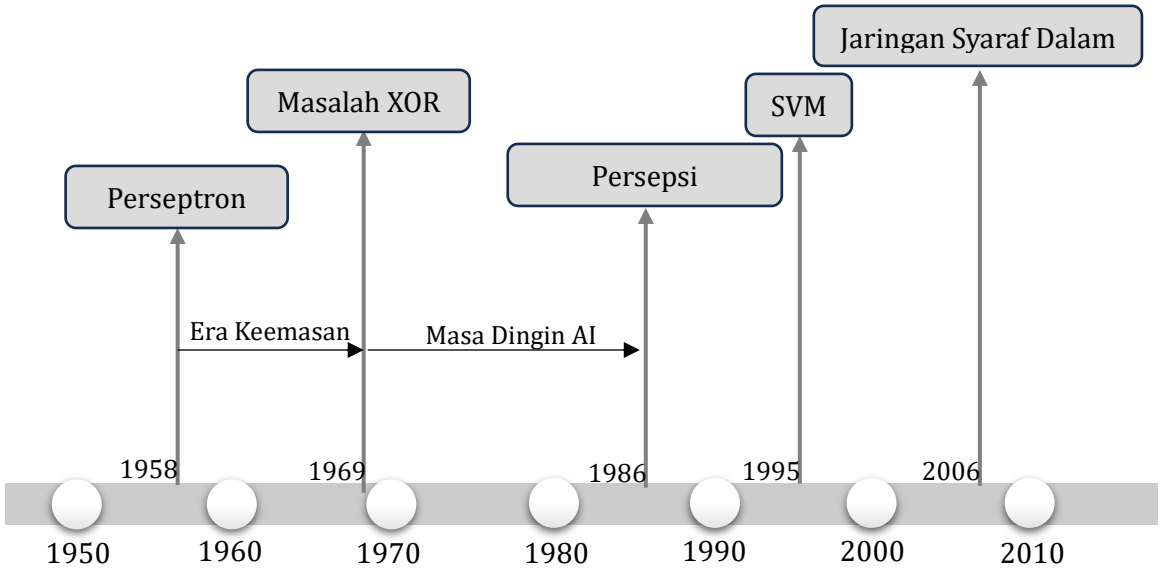
Apa sebenarnya jaringan saraf tiruan itu? Saat ini, belum ada konsensus tentang definisi jaringan saraf tiruan. Menurut ilmuwan jaringan saraf tiruan Amerika, Robert Hecht-Nielsen, jaringan saraf tiruan adalah sistem komputasi yang terdiri dari sejumlah elemen pemrosesan

yang sangat sederhana dan saling terhubung yang dibentuk dengan cara tertentu, yang memproses informasi melalui respons keadaan dinamisnya terhadap masukan eksternal.

Dengan menggabungkan asal-usul, karakteristik, dan berbagai interpretasi jaringan saraf tiruan, secara sederhana dapat dirumuskan sebagai berikut: jaringan saraf tiruan adalah sistem pemrosesan informasi yang dirancang untuk meniru struktur dan fungsi otak manusia. Jaringan saraf tiruan memiliki karakteristik dasar otak manusia, seperti pemrosesan informasi paralel, pembelajaran, asosiasi, klasifikasi, dan menghafal. Atau dapat dikatakan bahwa jaringan saraf tiruan adalah jaringan yang saling terhubung oleh sejumlah neuron buatan, versi abstrak dan sederhana dari otak manusia dalam hal struktur mikro dan fungsi, dan merupakan pendekatan penting untuk meniru kecerdasan manusia.

Perkembangan Pembelajaran Mendalam

Faktanya, sejarah pembelajaran mendalam adalah sejarah jaringan saraf tiruan. Sejak akhir 1950-an, seiring dengan perkembangan perangkat keras komputer yang berkelanjutan, jaringan saraf tiruan berkembang dari satu lapisan menjadi beberapa lapisan, dan akhirnya menjadi jaringan saraf tiruan dalam yang dikenal saat ini. Secara umum, perkembangan jaringan saraf tiruan dapat dibagi menjadi tiga tahap, seperti yang ditunjukkan pada Gambar 3.3.



Gambar 3.3 Sejarah jaringan saraf

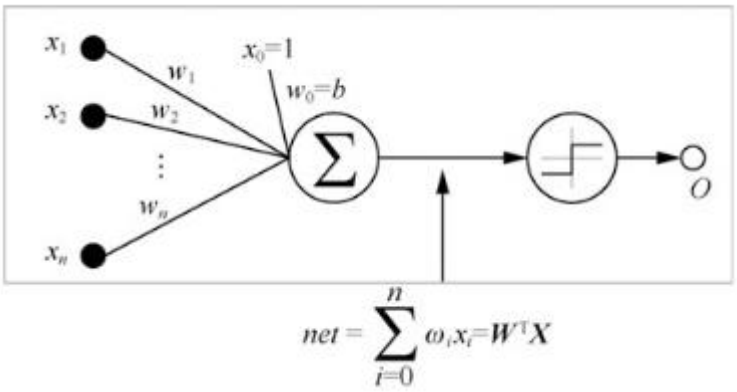
Pada tahun 1958, Frank Rosenblatt menemukan algoritma Perceptron, yang menandai dimulainya tahap awal pengembangan jaringan saraf tiruan. Namun, karena pembelajaran mesin belum menjadi subjek independen dalam penelitian kecerdasan buatan, pengembangan algoritma perceptron terhambat. Pada tahun 1969, Marvin Lee Minsky, seorang pelopor kecerdasan buatan dari AS, mempertanyakan bahwa perceptron hanya dapat menangani masalah klasifikasi linear, bahkan masalah XOR yang paling sederhana sekalipun. Pertanyaannya dan keraguan yang mengikutinya bagaikan hukuman mati bagi algoritma perceptron, dan juga menandai datangnya musim dingin pembelajaran mendalam yang berlangsung selama dua dekade.

Pada tahun 1986, Geoffrey Hinton menemukan Multi-Layer Perceptron (MLP) dan memperbaiki situasi tersebut. Hinton mengusulkan penggunaan fungsi sigmoid untuk melakukan pemetaan nonlinier pada keluaran perceptron, yang secara efektif memecahkan masalah klasifikasi dan pembelajaran nonlinier. Selain itu, Hinton juga menemukan algoritma Back Propagation (BP) untuk melatih MLP. Algoritma dan algoritma turunannya masih digunakan untuk pelatihan jaringan saraf tiruan dalam hingga saat ini. Pada tahun 1989, Robert Hecht-Nielsen membuktikan teorema aproksimasi universal (UAT).

Teorema tersebut menyatakan bahwa setiap fungsi kontinu f dalam rentang tertutup tertentu dapat didekati oleh jaringan BP dengan satu lapisan tersembunyi. Singkatnya, jaringan saraf tiruan dapat memuat fungsi kontinu apa pun. Pada tahun 1995, Vladimir Vapnik dan Corinna Cortes mengusulkan mesin vektor pendukung (SVM), salah satu terobosan paling menentukan dalam pembelajaran mesin. Algoritma ini tidak hanya kokoh dalam fondasi teoretis, tetapi juga memiliki kinerja yang luar biasa dalam eksperimen. Pada tahun 1998, sejumlah jaringan saraf tiruan diciptakan, termasuk jaringan saraf tiruan konvolusional (CNN) dan jaringan saraf tiruan rekuren (RNN) yang telah dikenal luas. Namun, karena masalah gradien menghilang dan ledakan gradien yang terkadang terjadi selama pelatihan jaringan saraf tiruan yang terlalu dalam, jaringan saraf tiruan tersebut perlahan-lahan dilupakan oleh masyarakat.

Tahun 2006 menandai dimulainya era pembelajaran mendalam. Pada tahun ini, Hinton mengusulkan kombinasi pra-pelatihan tanpa pengawasan dan fine-tuning terawasi sebagai solusi untuk masalah gradien menghilang pada pelatihan jaringan saraf tiruan dalam. Pada tahun 2012, AlexNet yang dirancang oleh tim Hinton memenangkan kompetisi pengenalan gambar terbaik dunia, ImageNet, dengan mengalahkan semua metode lainnya, yang memicu antusiasme pembelajaran mendalam. Pada tahun 2016, AlphaGo, program kecerdasan buatan pembelajaran mendalam dari Google, mengalahkan juara dunia Go dan pemain Go sembilan-dan, Lee Sedol. Kemenangan ini membawa perhatian dunia terhadap pembelajaran mendalam ke tingkat yang baru.

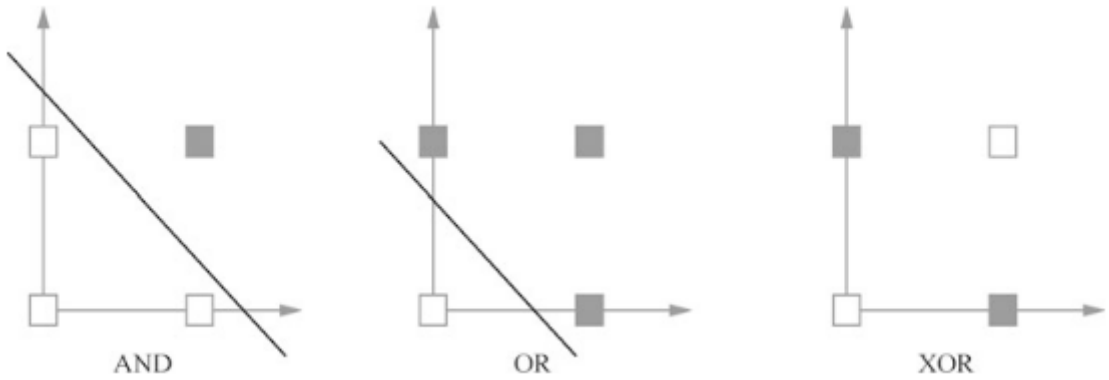
Perseptron lapis tunggal adalah jenis jaringan saraf tiruan yang paling sederhana. Seperti ditunjukkan pada Gambar 3.4, vektor masukan $X = [x_0, x_1, \cdots, x_n]^T$ pertama-tama menghitung produk titik dengan bobot $W = [w_0, w_1, \cdots, w_n]^T$, yang dilambangkan sebagai net. Di antara keduanya, x_0 sama dengan 1, dan w_0 disebut bias. Untuk regresi, net dapat langsung digunakan sebagai keluaran perseptron. Dan untuk masalah klasifikasi, net perlu diaktifkan oleh fungsi aktivasi yang disebut Sgn(net) agar dapat dijadikan keluaran. Fungsi Tanda sama dengan 1 dalam rentang $x > 0$, dan sama dengan -1 jika sebaliknya.



Gambar 3.4 Perseptron lapisan tunggal

Perseptron yang ditunjukkan pada Gambar 3.4 sebenarnya seperti pengklasifikasi yang mengadopsi vektor masukan berdimensi tinggi x untuk mengklasifikasikan sampel masukan dalam dikotomi dalam ruang berdimensi tinggi. Lebih spesifiknya, ketika Sgn(net) bernilai 1, berarti sampel diklasifikasikan sebagai positif, dan jika Sgn(net) bernilai -1, maka sampel diklasifikasikan sebagai negatif. Batas kedua situasi ini adalah $\boldsymbol{W^T X} = 0$, sebuah hiperbidang dalam ruang berdimensi tinggi.

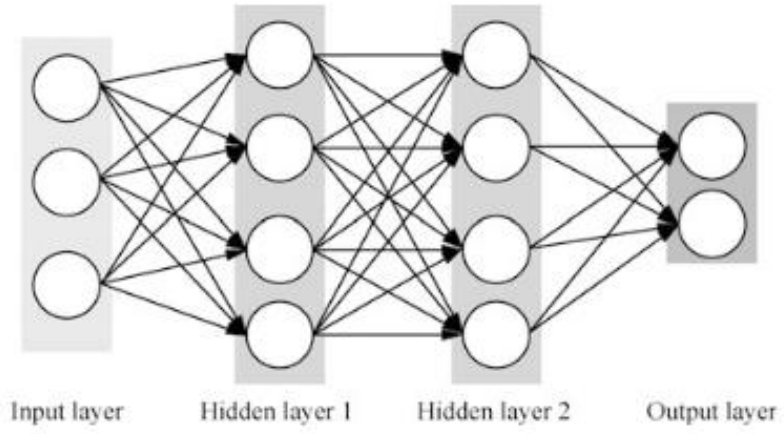
Di alam, perceptron hanyalah model linear, yang berarti ia hanya dapat mengimplementasikan klasifikasi linear tetapi tidak dapat menangani data nonlinier. Seperti ditunjukkan pada Gambar 3.5, untuk operator logika AND dan OR, perceptron dapat dengan mudah menemukan garis untuk mengklasifikasikannya dengan benar. Namun, ia tidak dapat melakukan apa pun terhadap operasi logika eksklusif OR (XOR), yang merupakan contoh yang dikutip Minsky untuk membuktikan keterbatasan perceptron secara langsung.



Gambar 3.5 Soal XOR

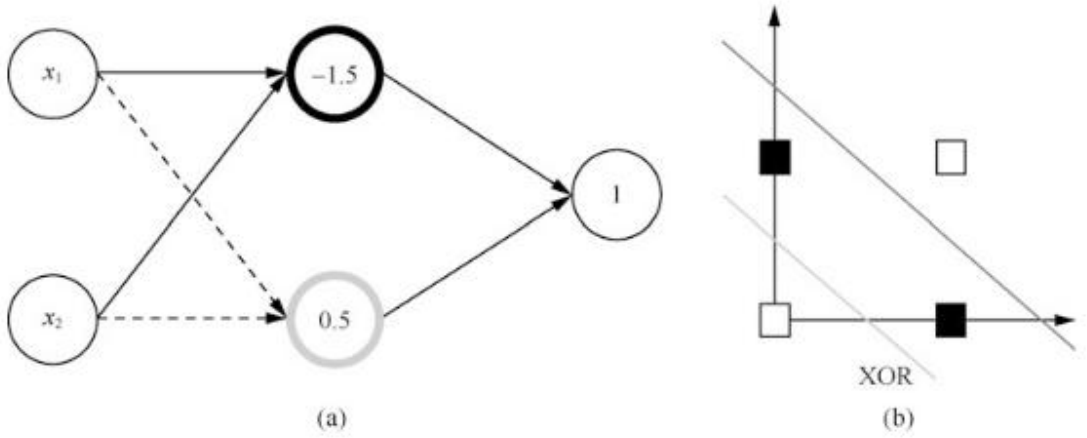
Agar perceptron dapat memproses data nonlinier, orang-orang menghasilkan perceptron multi-lapis, yang juga dikenal sebagai jaringan saraf tiruan umpan-maju, seperti yang ditunjukkan pada Gambar 3.6. Jaringan saraf tiruan umpan maju (feedforward neural network) adalah jaringan saraf tiruan versi dengan arsitektur paling sederhana, menampilkan neuron-neuron yang tersusun secara hierarkis (perceptron), dan saat ini merupakan salah satu jaringan saraf tiruan yang paling banyak diterapkan dan berkembang paling pesat. Pada Gambar 3.6,

tiga neuron paling kiri dalam perceptron multilapis membentuk lapisan masukan dari keseluruhan jaringan.



Gambar 3.6 Perseptron lapisan tunggal

Neuron-neuron pada lapisan masukan tidak menerapkan operasi apa pun. Mereka hanya merepresentasikan nilai setiap vektor masukan. Kecuali lapisan masukan, neuron pada setiap lapisan simpul yang memiliki fungsi komputasi disebut unit komputasi. Neuron-neuron pada setiap lapisan menerima nilai keluaran dari neuron-neuron pada lapisan sebelumnya sebagai masukan dan mengirimkannya ke lapisan berikutnya. Neuron-neuron pada lapisan yang sama tidak terhubung, dan informasi hanya dapat ditransmisikan satu arah antar lapisan yang berbeda.



Gambar 3.7 Soal XOR

Permasalahan XOR dapat diselesaikan hanya dengan sebuah perseptron multilapis yang sangat sederhana. Pada Gambar 3.7a, kita dapat menemukan struktur perseptron multilapis. Garis padat menunjukkan bobotnya adalah 1, dan garis putus-putus menunjukkan bobotnya adalah -1. Angka dalam lingkaran adalah offset. Misalnya, untuk titik (0,1)

$$x_1 = 0, x_2 = 1$$

Output dari neuron —1,5 adalah

$$\text{sgn}\,(x_1 + x_2 - 1.5) = \text{sgn}\,(-0.5) = -1$$

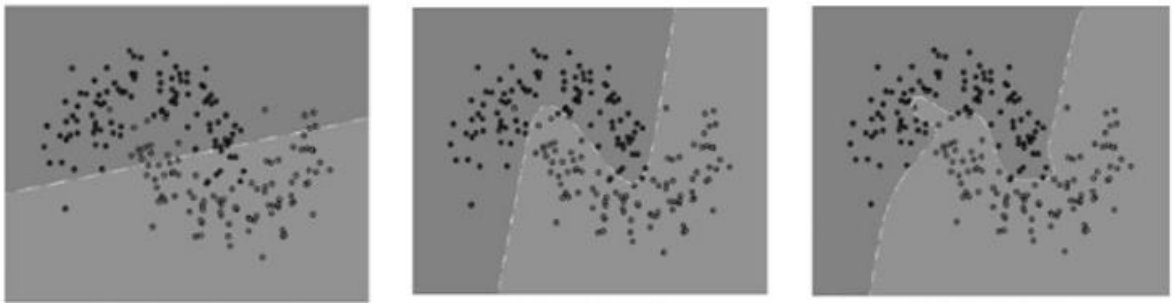
Karena kedua garis pada sisi kiri neuron $-1{,}5$ keduanya padat, kita dapat menyimpulkan koefisien x_1 dan x_2 keduanya 1. Jadi keluaran neuron $0{,}5$ ini adalah

$$\text{sgn}\,(-x_1 - x_2 + 1.5) = \text{sgn}\,(-0.5) = -1$$

Koefisien x_1 dan x_2 keduanya -1, karena dua garis di sisi kiri neuron $0{,}5$ keduanya merupakan garis putus-putus. Keluaran dari neuron paling kanan adalah

$$\text{sgn}\,(-1 - 1 + 1) = \text{sgn}\,(-1) = -1$$

Dua nilai —1 di sisi kiri persamaan merupakan keluaran dari neuron —1,5 dan neuron 0,5, dan +1 merupakan offset dari keluaran neuron tersebut. Pembaca kami dapat mencoba memverifikasi sendiri bahwa untuk titik (0, 0), (1, 0), dan (1, 1), keluaran dari multilayer perceptron masing-masing adalah 1, —1, dan 1, yang konsisten dengan hasil operasi XOR. Faktanya, kedua neuron —1,5 dan 0,5 masing-masing direpresentasikan oleh garis kanan atas dan kiri bawah yang ditunjukkan pada Gambar 3.7b, yang mengadopsi pengklasifikasi linier untuk mengimplementasikan klasifikasi sampel nonlinier. Dengan bertambahnya lapisan tersembunyi, klasifikasi nonlinier jaringan saraf tiruan secara bertahap ditingkatkan, seperti yang ditunjukkan pada Gambar 3.8.



Lapisan Tersembunyi = 0 Lapisan Tersembunyi = 3 Lapisan Tersembunyi = 20

Gambar 3.8 Jaringan saraf tiruan multi-lapisan tersembunyi

3.2 ATURAN PELATIHAN

Inti dari pelatihan model pembelajaran mesin adalah fungsi kerugian, dan untuk pembelajaran mendalam, hal yang sama juga berlaku. Bagian ini akan memperkenalkan aturan pelatihan model dalam pembelajaran mendalam dengan menguraikan fungsi kerugian, termasuk algoritma penurunan gradien dan propagasi balik.

Fungsi Kerugian

Saat melatih jaringan saraf tiruan dalam, pertama-tama perlu dibuat sebuah fungsi untuk mendeteksi kesalahan klasifikasi target. Fungsi ini disebut fungsi kerugian (atau fungsi kesalahan). Fungsi kerugian mencerminkan kesalahan antara nilai keluaran target dan nilai keluaran aktual dari perceptron. Fungsi kerugian yang paling umum diadopsi adalah kesalahan kuadrat rata-rata (MSE), dengan rumus sebagai berikut:

$$J(w) = \frac{1}{2n} \sum_{x \in X, d \in D} (t_d - O_d)^2$$

Dalam rumus ini, w = parameter model, X = himpunan contoh pelatihan, n = ukuran X, D = kumpulan neuron di lapisan keluaran, t = keluaran target, dan o = keluaran aktual. Meskipun parameter w tidak secara langsung dimasukkan di ruas kanan persamaan, keluaran aktual o perlu dihitung dalam model, oleh karena itu ditentukan oleh nilai parameter w. Setelah contoh pelatihan ditetapkan, t dan o keduanya merupakan konstanta.

Keluaran aktual dari fungsi kerugian berubah seiring dengan nilai w, sehingga w adalah variabel bebas dari fungsi galat. Fitur fungsi kerugian MSE adalah badan utamanya merupakan jumlah kuadrat galat, sedangkan galat adalah selisih antara keluaran target t dan keluaran aktual o. Dalam rumus ini, komponen lain yang lebih rumit adalah koefisien 1/2. Dalam bagian selanjutnya, kita akan mempelajari bahwa koefisien ini dapat menghasilkan bentuk yang lebih ringkas untuk derivasi fungsi kerugian, dengan meng-offset eksponen 2 menjadi satu. Kesalahan entropi silang adalah fungsi kerugian umum lainnya, dan rumusnya adalah sebagai berikut:

$$\boldsymbol{J(w)} = -\frac{1}{2} \sum_{x \in X, d \in D} (t_d \ln O_d + (1 - t_d) \ln(1 - O_d))$$

Simbol dalam rumus menunjukkan hal yang sama dengan simbol dalam rumus MSE. Galat entropi silang menggambarkan jarak antara dua distribusi probabilitas. Secara umum, fungsi MSE terutama berkaitan dengan regresi, sedangkan fungsi galat entropi silang lebih banyak digunakan untuk klasifikasi. Target pelatihan model adalah mencari vektor bobot yang meminimalkan fungsi kerugian. Namun, model jaringan saraf tiruan sangat rumit, dan tidak ada metode matematika yang efektif untuk menemukan solusi analitis. Oleh karena itu, kita perlu menggunakan penurunan gradien untuk menemukan solusi numerik guna meminimalkan fungsi kerugian.

Penurunan Gradien

Gradien fungsi multivariat $f(x_1, x_2, \ldots, x_n)$ pada x adalah

$$\nabla f(x_1, x_2, \ldots, x_n) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots \frac{\partial f}{\partial x_n}\right]^T |x$$

Vektor gradien menunjukkan arah peningkatan fungsi yang paling cepat. Vektor gradien negatif, dengan demikian, menunjukkan arah penurunan fungsi yang paling tajam. Algoritma penurunan gradien digunakan untuk membiarkan fungsi kerugian menelusuri arah gradien negatif, memperbarui parameter secara iteratif, dan meminimalkan fungsi kerugian. Setiap sampel dalam set contoh pelatihan X adalah <x, t>, dengan x adalah vektor masukan, t adalah keluaran target, o adalah keluaran aktual, dan η adalah laju pembelajaran. Gambar 3.9 menunjukkan pseudo-code untuk algoritma penurunan gradien batch (BGD).

Algoritma 1: Algoritma Batch Gradient Descent
Input: Kumpulan data pelatihan $D = \{< x, t >\}$, Laju pembelajaran η
Output: Nilai parameter optimal w
Langkah-langkah:
1. Inisialisasi $w \leftarrow$ vektor acak dengan nilai absolut relatif kecil;
2. Ulangi:
3. $\Delta w \leftarrow \vec{0}$
4. Untuk setiap x, t dalam D lakukan:
5. $\Delta w \leftarrow \Delta w + \frac{\partial J}{\partial w} |_{x,t}$
6. Selesai
7. $w \leftarrow w - \eta \Delta w$
8. Hingga model konvergen atau jumlah iterasi maksimum tercapai;
9. Kembalikan w

Gambar 3.9 Algoritma penurunan gradien batch

Sebagai turunan dari penerapan langsung penurunan gradien pada pembelajaran mendalam, algoritma BGD sebenarnya tidak terlalu sering digunakan. Kekurangan utama algoritma ini adalah setiap kali bobot diperbarui, algoritma ini perlu menghitung semua contoh pelatihan, sehingga memperlambat konvergensi. Untuk mengatasi masalah ini, varian umum dari penurunan gradien, yaitu algoritma penurunan gradien stokastik (SGD), diciptakan, yang juga dikenal sebagai algoritma penurunan gradien inkremental. Gambar 3.10 menunjukkan pseudo-code untuk algoritma SGD.

Algoritma 2: Algoritma *Stochastic Gradient Descent*
Input: Himpunan data latih $D = \{\langle x, t \rangle\}$, *Learning rate* η
Output: Nilai parameter optimal w
1. Inisialisasi www sebagai vektor acak dengan nilai absolut yang relatif kecil;
2. Ulangi:
3. Ambil sampel acak $(x, t) \in D$
4. Hitung perubahan bobot: $\Delta w \leftarrow \frac{\partial J}{\partial w} |_{x,t}$
5. Perbarui bobot: $w = w - \eta \Delta w$
6. Hingga model konvergen atau jumlah iterasi maksimum tercapai;
7. Kembalikan nilai w

Gambar 3.10 Algoritma penurunan gradien stokastik

Algoritma SGD memilih satu sampel pada satu waktu untuk memperbarui gradien. Salah satu keuntungan metode ini adalah dataset dapat diperluas selama pelatihan model. Dan mode pelatihan model selama pengumpulan data ini dikenal sebagai pembelajaran daring. Dibandingkan dengan algoritma penurunan gradien batch, algoritma penurunan gradien stokastik meningkatkan frekuensi pembaruan bobot tetapi berpindah dari satu ekstrem ke ekstrem lainnya. Biasanya, terdapat derau dalam sampel pelatihan. Penurunan gradien batch dapat mengurangi pengaruh derau dengan mengambil rata-rata gradien sampel. Namun, karena penurunan gradien stokastik menganalisis satu sampel pada satu waktu selama pembaruan bobot, ketika mencapai fase perkiraan nilai ekstrem yang akurat, gradien mungkin memantul di sekitarnya, dan tidak dapat konvergen ke nilai ekstrem.

Algoritma 3: Algoritma *Mini-Batch Gradient Descent*
Input: Himpunan data latih D={⟨x,t⟩}, *Learning rate* η
Output: Nilai parameter optimal w
1. Inisialisasi w sebagai vektor acak dengan nilai absolut yang relatif kecil;
2. Ulangi:
3. Untuk setiap *batch* dalam D / *batch* berarti sekumpulan sampel dalam D:
4. Inisialisasi perubahan bobot: $\Delta w \leftarrow 0$
5. Untuk setiap pasangan (x,t) dalam *batch* lakukan:
6. Hitung gradien dan akumulasikan: $\Delta w \leftarrow \Delta w + \frac{\partial J}{\partial w}|_{x,t}$
7. Selesai (end loop untuk sampel dalam batch)
8. Perbarui bobot: $w = w - \eta \Delta w$
9. Selesai (*end loop* untuk batch)
10. Ulangi hingga model konvergen atau jumlah iterasi maksimum tercapai;
11. Kembalikan nilai w

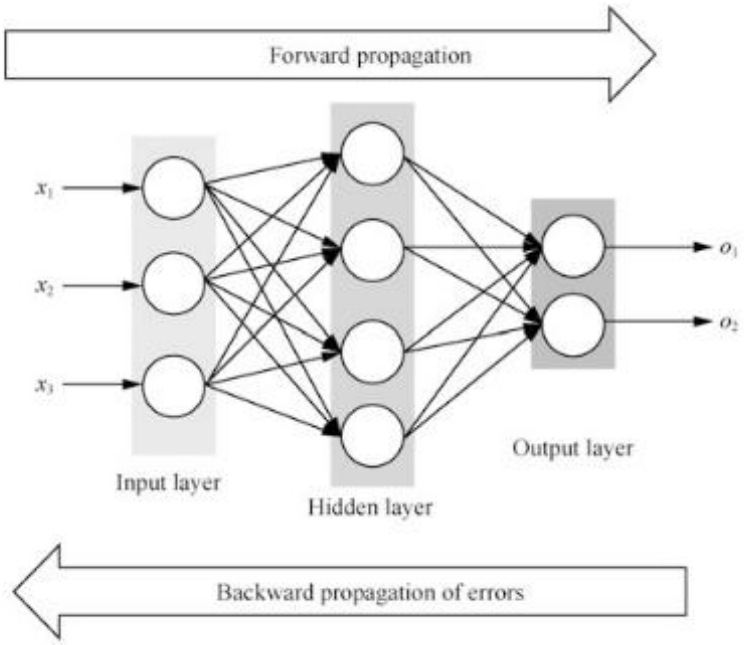
Gambar 3.11 Algoritma penurunan gradien mini-batch

Algoritma penurunan gradien yang paling umum digunakan dalam tugas-tugas praktis adalah algoritma penurunan gradien mini-batch (MBGD), seperti yang ditunjukkan pada Gambar 3.11. Untuk mengatasi kekurangan dari dua algoritma penurunan gradien di atas, algoritma penurunan gradien mini-batch menggunakan sejumlah kecil sampel pada satu waktu ketika memperbarui bobot, dengan mempertimbangkan efisiensi dan stabilitas gradien. Ukuran minibatch umumnya terdiri dari 128 elemen, tetapi juga bervariasi tergantung pada situasi yang berbeda.

Algoritma Backpropagation

Penerapan penurunan gradien perlu menghitung gradien fungsi kerugian. Untuk algoritma pembelajaran mesin tradisional, seperti regresi linier dan mesin vektor pendukung, menghitung gradien secara manual dapat dilakukan. Namun, fungsi model jaringan saraf

tiruan jauh lebih rumit, dan mustahil untuk menemukan gradien fungsi kerugian terhadap semua parameter hanya dengan satu rumus. Dengan latar belakang ini, Geoffrey Hinton menemukan algoritma backpropagation, yang dapat memperbarui bobot per lapisan secara terpisah melalui backpropagation, dan secara efektif mempercepat pelatihan jaringan saraf tiruan. Propagasi balik kesalahan bekerja dalam arah yang berlawanan dengan propagasi maju, seperti yang ditunjukkan pada Gambar 3.12.



Gambar 3.12 Propagasi kesalahan ke belakang

$$\boldsymbol{J(w) = \frac{1}{2n} \sum_{x \epsilon X, d \epsilon D} (t_d - o_d)^2}$$

Untuk setiap contoh pelatihan $< x, t >$ dalam himpunan contoh pelatihan X, keluaran model dilambangkan sebagai o. Dengan asumsi bahwa fungsi kerugian mengambil rata-rata kuadrat kesalahan:

$$\boldsymbol{J(w) = \frac{1}{2n} \sum_{x \epsilon X, d \epsilon D} (t_d - o_d)^2}$$

Misalkan ada L lapisan dalam model (tidak termasuk lapisan input), dan parameter lapisan pertama dicatat sebagai w_1. Kita dapat mengamati bahwa nilai $J(w)$ bukanlah minimum selama iterasi karena deviasi antara w dan nilai parameter optimal, dan pengamatan ini berlaku untuk setiap lapisan. Dengan kata lain, nilai fungsi kerugian disebabkan oleh kesalahan parameter.

Selama proses propagasi maju, setiap lapisan akan menyebabkan kesalahan tertentu. Karena kesalahan ini terakumulasi lapis demi lapis, kesalahan tersebut akhirnya terwujud dalam bentuk fungsi kerugian di lapisan keluaran. Ketika tidak ada fungsi model yang diberikan, kita tidak dapat yakin tentang hubungan antara fungsi kerugian dan parameter. Namun, hubungan antara fungsi kerugian dan keluaran model $\partial J/\partial o$ terlihat jelas. Ini adalah kunci untuk memahami algoritma backpropagation. Misalkan keluaran lapisan kedua terakhir adalah o', dan fungsi aktivasi lapisan keluaran adalah f, maka fungsi kerugian dapat diperluas menjadi:

$$J(w) = \frac{1}{2m} \sum_{x\epsilon X, d\epsilon D} (t_d - f(w_L o'_d))^2$$

Dalam rumus ini, o'_d hanya relevan dengan $w_1, w_2, \cdots, w_{L-1}$. Dapat diamati bahwa fungsi kerugian dapat dibagi menjadi dua bagian, bagian yang disebabkan oleh w_L dan bagian yang disebabkan oleh parameter lain. Yang terakhir bertindak pada fungsi kerugian sebagai keluaran di lapisan kedua terakhir melalui akumulasi kesalahan. Berdasarkan $\partial J/\partial o$ yang diperoleh di atas, kita dapat dengan mudah menghitung $\partial J/\partial o'$ dan $\partial J/\partial w_L$. Dengan cara ini, kita akan mendapatkan gradien fungsi kerugian terhadap parameter lapisan keluaran. Jelas, turunan dari fungsi aktivasi f ' wLo' terlibat dalam perhitungan $\partial J/\partial o'$ dan $\partial J/\partial w_L$ sebagai bobot. Ketika turunan dari fungsi aktivasi secara kekal kurang dari 1 (seperti cara kerja fungsi Sigmoid), nilai $\partial J/\partial o$ akan terus menurun selama backpropagation.

Fenomena ini dikenal sebagai masalah gradien menghilang, yang akan dijelaskan lebih spesifik pada bagian-bagian berikutnya. Untuk parameter di lapisan lain, kita dapat menyimpulkan hal yang sama dari hubungan antara ∂J/∂o' dan ∂J/∂o''. Singkatnya, backpropagation adalah proses mendistribusikan kesalahan lapis demi lapis dan pada dasarnya merupakan algoritma yang menerapkan aturan rantai untuk menghitung fungsi kerugian terhadap parameter setiap lapisan. Secara umum, algoritma backpropagation bekerja seperti yang ditunjukkan pada Gambar 3.13.

Algoritma 4: Algoritma Backpropagation
Input: Parameter dari setiap lapisan $w[1\ to\ L]$, Output dari setiap lapisan $o[0\ to\ L]$
Output: Parameter dari setiap lapisan setelah satu iterasi $w[1\ to\ L]$
1. $\delta L \leftarrow \frac{\partial J}{\partial o[L]} \odot f'(w[L]o[L-1])$
2. Untuk setiap lapisan dari $L-1\ ke\ 1$
3. $\delta_l \leftarrow w[l+1]^T \delta_{l+1} \odot f'(w[l]o[l-1])$;
4. dan
5. untuk $l \leftarrow 1\ untuk\ L\ lakukan$
6. $w[l] \leftarrow w[l] - \eta\delta_l o[l-1]^T$;
7. dan
8. kembali

Gambar 3.13 Algoritma propagasi mundur

Dalam kode di atas, π merepresentasikan perkalian dengan elemen, dan f adalah fungsi aktivasi. Perlu diperhatikan bahwa keluaran dari lapisan ke-i juga merupakan masukan dari lapisan ke$-i + 1$. Keluaran dari lapisan ke-0 dianggap sebagai masukan dari keseluruhan jaringan. Selain itu, ketika fungsi aktivasi adalah fungsi sigmoid, dapat dibuktikan bahwa:

$$f'(x) = f(x)\big(1 - f(x)\big)$$

Oleh karena itu, $f'(o[l - 1])$ dalam algoritma juga dapat ditulis sebagai $o[l](1 - o[l])$.

3.3 FUNGSI AKTIVASI

Fungsi aktivasi memainkan peran yang sangat penting dalam mempelajari model jaringan saraf tiruan dan menginterpretasikan fungsi nonlinier yang kompleks. Fungsi aktivasi menambahkan karakteristik nonlinier ke jaringan saraf tiruan. Tanpa fungsi aktivasi, jaringan saraf tiruan hanya dapat merepresentasikan satu fungsi linier, berapa pun lapisannya. Namun, kompleksitas fungsi linier terbatas, dan bahkan kurang mampu mempelajari pemetaan fungsi kompleks dari data. Bagian ini memperkenalkan fungsi aktivasi yang umum digunakan dalam pembelajaran mendalam dan menguraikan kelebihan dan kekurangannya. Pembaca dapat memilih dari daftar tersebut berdasarkan kebutuhan mereka saat mengerjakan proyek.

Seperti yang ditunjukkan pada Gambar 3.14a, fungsi sigmoid adalah fungsi aktivasi yang paling sering diadopsi pada tahap awal studi jaringan saraf tiruan umpan maju. Serupa dengan model regresi logistik, fungsi sigmoid dapat digunakan pada lapisan keluaran untuk klasifikasi biner. Secara umum, fungsi sigmoid bersifat monotonik dan kontinu, dengan turunan yang mudah dihitung dan keluarannya terbatas. Fitur-fitur ini memfasilitasi konvergensi jaringan.

Namun, kita dapat melihat turunan fungsi sigmoid mendekati 0 ketika jauh dari titik asal. Ketika jaringan sangat dalam, backpropagation akan mendorong semakin banyak neuron ke daerah saturasi, sehingga mengurangi cakupan gradien. Secara umum, jaringan sigmoid akan mengalami degenerasi menjadi 0 dalam 5 lapisan, yang menyebabkan jaringan saraf tidak dapat melakukan pelatihan lebih lanjut. Fenomena ini disebut masalah gradien menghilang. Kelemahan lain dari fungsi sigmoid adalah keluarannya tidak berpusat pada nol.

Seperti ditunjukkan pada Gambar 3.14b, fungsi tanh merupakan alternatif utama untuk Sigmoid. Fungsi aktivasi tanh memperbaiki kelemahan dari tidak berpusat pada nol pada keluaran dan gradiennya lebih mirip gradien alami dalam penurunan gradien, sehingga mengurangi jumlah iterasi yang diperlukan. Namun, fungsi tanh masih rentan terhadap saturasi seperti halnya fungsi sigmoid.

Seperti ditunjukkan pada Gambar 3.14c, fungsi Softsign mengurangi saturasi fungsi tanh dan fungsi sigmoid. Namun, baik Softsign, tanh, maupun Sigmoid, fungsi aktivasi lebih cenderung menyebabkan masalah gradien yang hilang. Pada posisi yang cukup jauh dari titik pusat (titik asal) fungsi, turunan fungsi aktivasi akan selalu mendekati 0, sehingga menghentikan pembaruan bobot.

Seperti ditunjukkan pada Gambar 3.14d, fungsi linear yang diperbaiki atau ReLU saat ini merupakan fungsi aktivasi yang paling banyak diterapkan. Berbeda dengan fungsi aktivasi seperti Sigmoid, ReLU tidak dibatasi oleh batas atas, sehingga neuron tidak akan pernah mencapai saturasi, yang secara efektif mengurangi gradien yang menghilang dan dapat konvergen jauh lebih cepat dalam penurunan gradien. Eksperimen menunjukkan bahwa jaringan saraf yang menggunakan fungsi aktivasi ReLU juga dapat menghasilkan kinerja yang baik bahkan tanpa pra-pelatihan tanpa pengawasan.

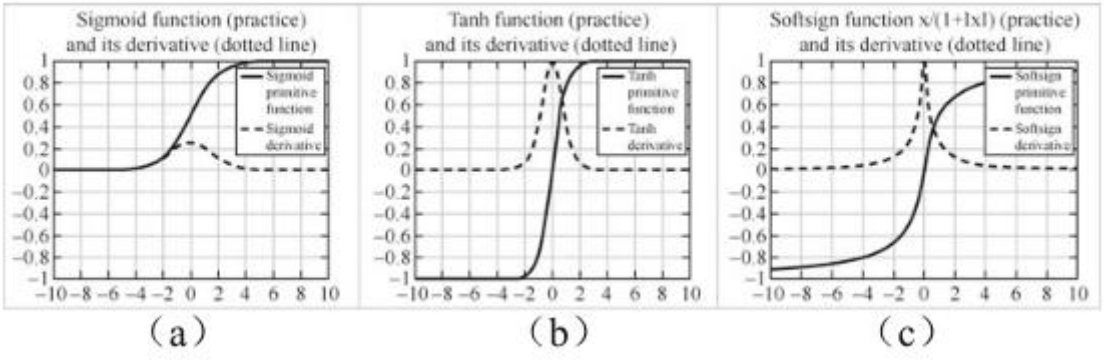
Selain itu, semua fungsi seperti Sigmoid memerlukan persamaan eksponensial, yang cukup intensif komputasi. Namun, fungsi aktivasi ReLU menghemat banyak upaya dalam kalkulasi dan komputasi. Meskipun ReLU memiliki banyak keuntungan, ia juga memiliki kekurangan. Karena fungsi ReLU tidak memiliki batas atas, ia mungkin menyimpang selama pelatihan. Kedua, fungsi ReLU tidak dapat diarahkan ke nol, sehingga tidak akan cukup mulus dalam beberapa masalah regresi. Yang lebih penting, nilai fungsi ReLU akan tetap nol secara konstan ketika input negatif, yang dapat menyebabkan "kematian" neuron.

Seperti yang ditunjukkan pada Gambar 3.14e, fungsi Softplus dimodifikasi berdasarkan ReLU. Meskipun fungsi softplus membutuhkan lebih banyak komputasi daripada ReLU, fungsi ini memiliki turunan kontinu, dan kurvanya relatif lebih halus. Fungsi softmax merupakan perluasan dari fungsi sigmoid dalam kasus multidimensi. Ekspresi fungsinya adalah sebagai berikut:

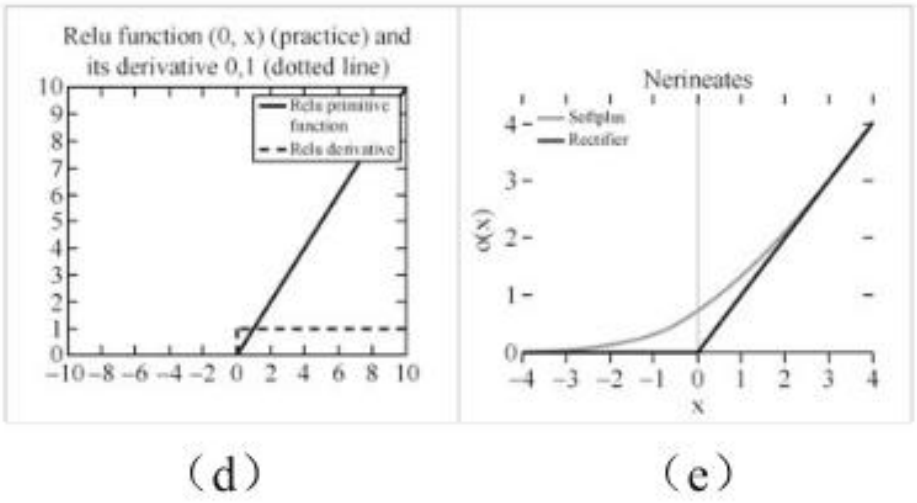
$$\sigma(z)_j = \frac{e_j^z}{\sum_k e_k^z}$$

Fungsi softmax dirancang untuk memetakan vektor sembarang bilangan riil berdimensi K ke distribusi probabilitas berdimensi K lainnya. Oleh karena itu, fungsi softmax sering digunakan sebagai lapisan keluaran untuk multiklasifikasi.

$$f(x) = \frac{1}{1+e^{-x}} \qquad \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad f(x) = \frac{x}{|x|+1}$$



$$y = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \quad f(x) = 1n(e^x + 1)$$



Gambar 3.14 Fungsi Aktivasi

3.4 REGULARISASI

Regularisasi merupakan langkah yang cukup penting dan sangat efektif untuk mengurangi kesalahan generalisasi dalam pembelajaran mesin. Dibandingkan dengan model pembelajaran mesin tradisional, kapasitas model pembelajaran mendalam lebih besar, sehingga lebih mungkin menyebabkan overfitting. Untuk tujuan ini, para peneliti telah mengembangkan beberapa teknik yang berguna untuk mencegah overfitting, termasuk:

1. Penalti norma parameter, yang menambahkan batasan norma seperti L1 dan L2.
2. Ekspansi himpunan data, seperti menambahkan derau dan mengubah data.
3. Dropout.
4. Penghentian dini.

Bagian ini akan memperkenalkan teknik-teknik ini secara berurutan.

Penalti Norma Parameter

Untuk banyak metode regularisasi, mereka membatasi kemampuan pembelajaran model dengan menambahkan penalti norma parameter $Z(w)$ ke fungsi tujuan J:

$$\tilde{J} = J + aZ(w)$$

Di mana a adalah koefisien suku penalti non-negatif yang nilainya memberi bobot pada kontribusi relatif suku penalti norma Z dan fungsi objektif standar J terhadap fungsi objektif umum. Menetapkan a ke nol menghasilkan tidak adanya regularisasi; dan semakin besar nilai a, semakin besar pula regularisasinya. Jadi, a adalah hiperparameter. Perlu diperhatikan bahwa dalam pembelajaran mendalam, kita biasanya hanya memberikan penalti pada parameter model w, alih-alih bias.

Hal ini karena, secara umum, bias membutuhkan lebih sedikit data untuk pencocokan yang akurat, sementara menambahkan penalti pada parameter bias sering kali menyebabkan underfitting. Z yang berbeda akan menghasilkan metode regularisasi yang berbeda. Bagian ini terutama akan memperkenalkan dua di antaranya: regularisasi L1 dan regularisasi L2. Di antara

model regresi linier, regularisasi L1 dapat mengarah pada regresi Lasso, dan regularisasi L2 dapat mengarah pada regresi ridge. Faktanya, yang disebut L1 dan L2 menunjukkan norma. Norma L1 dari suatu vektor didefinisikan sebagai

$$||\boldsymbol{w}||_1 = \sum_i |w_i|$$

Artinya, jumlah nilai absolut semua komponen vektor. Dapat dibuktikan bahwa gradien norma L1 adalah Sgn(w). Akibatnya, algoritma penurunan gradien dapat digunakan untuk menghitung model regularisasi L1.

Norma L2 adalah jarak Euclidean yang umum dikenal:

$$||\boldsymbol{w}||_2 = \sqrt{\sum_i w_i^2}$$

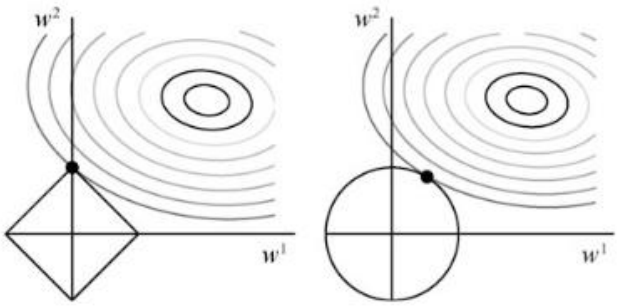
Karena norma L2 memiliki penerapan yang cukup luas, maka sering disingkat menjadi ||w|| dengan subskrip dihilangkan. Namun, karena gradien norma L2 relatif rumit, regularisasi L2 umumnya ditulis sebagai:

$$Z(w) = \frac{1}{2}||w||^2$$

Kita dapat melihat bahwa suku penalti regularisasi L2 terbukti menjadi w setelah turunan. Oleh karena itu, ketika melakukan penurunan gradien pada model regularisasi L2, persamaan untuk memperbarui bobot harus diperbaiki sebagai berikut:

$$\boldsymbol{w} = (1 - \eta a)\boldsymbol{w} - \eta \nabla J$$

Dibandingkan dengan persamaan pembaruan gradien umum, persamaan ini setara dengan perkalian parameter dengan faktor reduksi, sehingga dapat menahan pertumbuhan parameter. Gambar 3.15 menampilkan perbedaan antara regularisasi L1 dan regularisasi L2.



Gambar 3.15 Signifikansi geometrik penalti parameter

Garis kontur merepresentasikan fungsi objektif standar J, dan berlian atau lingkaran yang berpusat di titik asal merepresentasikan regularizer. Signifikansi geometris dari penalti norma parameter adalah bahwa untuk setiap titik dalam ruang fitur, yang harus dipertimbangkan bukan hanya nilai fungsi objektif standar yang bersesuaian dengan titik tersebut, tetapi juga ukuran bentuk geometris yang bersesuaian dengan regularizer titik tersebut. Tidak sulit untuk menyimpulkan bahwa semakin besar koefisien penalti a, semakin besar kemungkinan berlian atau lingkaran akan mengecil, dan semakin dekat parameter akan miring ke titik asal.

Seperti yang dapat dilihat dari Gambar 3.15, parameter yang dapat menstabilkan model regularisasi L1 sangat mungkin muncul di sudut-sudut bentuk berlian, yang menunjukkan bahwa parameter model regularisasi L1 kemungkinan besar berupa matriks renggang. Contoh pada gambar menunjukkan bahwa parameter optimal yang berkorespondensi dengan w1 dilambangkan sebagai nol. Oleh karena itu, regularisasi L1 dapat berfungsi sebagai metode seleksi fitur. Untuk menganalisisnya dari perspektif distribusi probabilitas, untuk banyak kendala norma, yang dilakukan hanyalah menambahkan distribusi prior ke parameter. Norma L2 setara dengan parameter yang mematuhi distribusi prior Gaussian, dan norma L1 adalah parameter yang mematuhi distribusi prior Laplace.

Ekspansi Dataset

Metode paling efektif untuk mencegah overfitting adalah dengan meningkatkan ukuran set pelatihan, karena probabilitas overfitting akan berkurang seiring bertambahnya set pelatihan. Namun, pengumpulan data (terutama data pelabelan) merupakan tugas yang memakan waktu dan biaya. Untuk tujuan ini, ekspansi dataset merupakan alternatif yang lebih hemat waktu namun tetap efisien, meskipun hampir tidak ada metode ekspansi universal untuk berbagai dataset dalam tugas yang berbeda.

Dalam pengenalan target, metode ekspansi dataset yang populer meliputi rotasi gambar, penskalaan gambar, dan sebagainya. Premis transformasi bentuk gambar adalah mengubah kategori gambar. Misalnya, saat mengenali angka tulisan tangan, 6 dan 9 adalah dua kategori yang mudah tertukar setelah rotasi sehingga memerlukan perhatian khusus. Dalam pengenalan ucapan, seringkali menambahkan derau acak ke dalam data masukan. Dan dalam pengenalan bahasa alami, metode yang umum adalah mengganti sinonim.

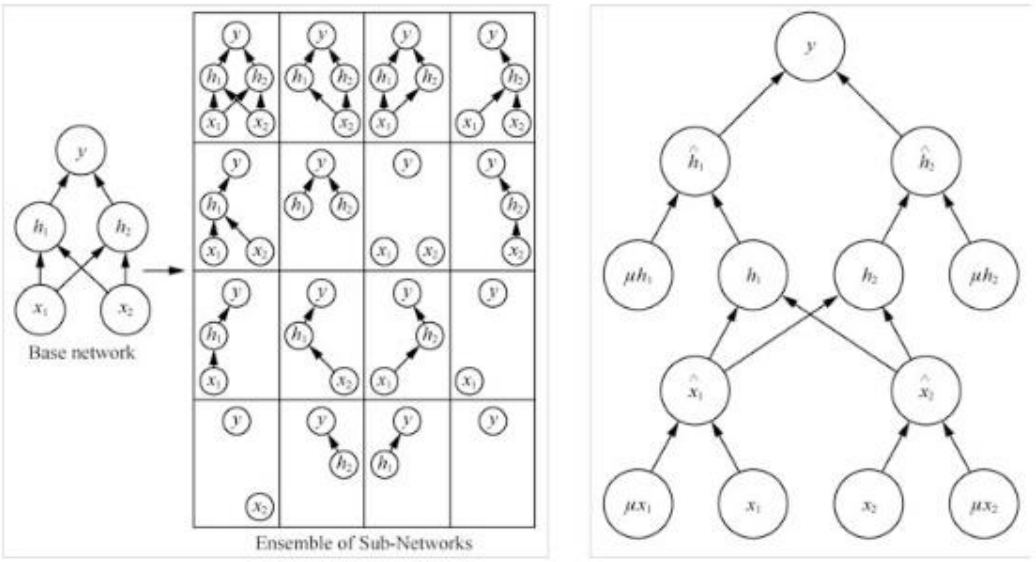
Injeksi derau adalah pendekatan ekspansi dataset yang populer. Derau dapat disuntikkan ke lapisan masukan, lapisan tersembunyi, atau lapisan keluaran. Untuk klasifikasi Softmax, derau saya tambahkan ke lapisan keluaran melalui Perataan Label. Misalkan terdapat K kelas kandidat untuk tugas klasifikasi, keluaran standar yang disediakan oleh dataset umumnya direpresentasikan sebagai vektor K-dimensi yang dikodekan satu-panas. Elemen yang mewakili kelas yang benar adalah "1", dan "0" untuk sisanya.

Dengan menyuntikkan derau, elemen yang sesuai dengan kategori yang benar akan berubah menjadi $1 - (k - 1)e/k$, dan sisanya akan menjadi e/k, di mana e mewakili konstanta yang cukup kecil. Secara intuitif, perataan label mengurangi perbedaan antara sampel yang benar dan sampel yang salah dalam hal nilai label, yang meningkatkan kesulitan

pelatihan model. Untuk model overfitting, hal ini secara efektif akan mengurangi masalah overfitting, sehingga memfasilitasi kinerja model.

Dropout

Dropout adalah metode regularisasi yang sangat populer dan sederhana dalam komputasi, telah diterapkan secara luas sejak diusulkan pada tahun 2014. Sederhananya, Dropout adalah membuang sebagian keluaran neuron secara acak selama fase pelatihan. Parameter neuron yang dibuang ini tidak akan diperbarui. Dengan membuang keluaran secara acak, Dropout membangun serangkaian subnet dengan struktur yang berbeda, seperti yang ditunjukkan pada Gambar 3.16.

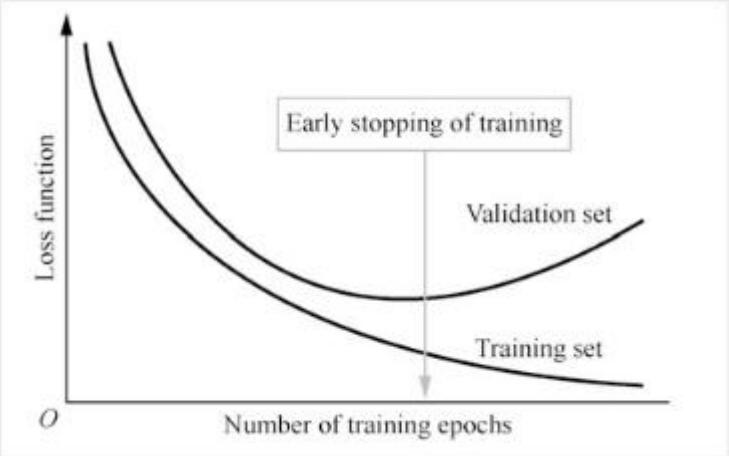


Gambar 3.16 Dropout

Subnet-subnet ini akan konvergen dengan cara tertentu dalam jaringan saraf dalam yang sama, yang setara dengan menggunakan metode pembelajaran terintegrasi. Saat menjalankan model, kami akan memanfaatkan kecerdasan kolektif dari semua subnet yang telah dilatih, sehingga keluaran neuron tidak akan lagi dibuang. Dropout lebih sederhana dalam komputasi dan lebih mudah diimplementasikan dibandingkan dengan penalti norma parameter. Proses acak Dropout selama pelatihan bukanlah syarat cukup maupun syarat perlu, sehingga dapat menghasilkan parameter shield yang konstan dan menghasilkan model yang kompetitif.

Penghentian Awal

Penghentian awal pelatihan sebaiknya diperbolehkan. Seperti ditunjukkan pada Gambar 3.17, dengan memeriksa dataset validasi secara berkala, dan ketika fungsi kerugian dataset validasi mulai meningkat, pelatihan dapat dihentikan lebih awal untuk menghindari overfitting. Namun, hal ini juga dapat menimbulkan risiko underfitting. Hal ini disebabkan oleh kurangnya sampel dalam dataset validasi, sehingga sulit untuk menghentikan pelatihan tepat pada saat kesalahan generalisasi model mencapai minimum. Dalam beberapa kasus ekstrem, kesalahan generalisasi model pada dataset validasi dapat turun dengan cepat segera setelah sedikit peningkatan, sementara penghentian awal pelatihan akan menyebabkan underfitting pada model.



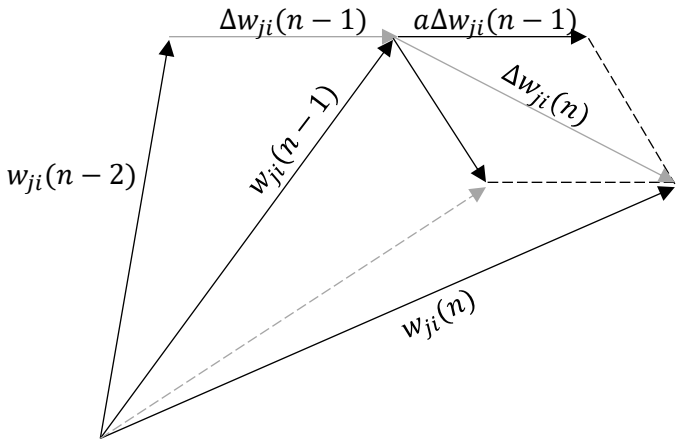
Gambar 3.17 Penghentian Awal Pelatihan

3.5 PENGOPTIMAL

Terdapat berbagai versi optimal dari algoritma penurunan gradien. Dalam implementasi bahasa berorientasi objek, algoritma ini seringkali merangkum berbagai algoritma penurunan gradien ke dalam satu objek, yang disebut pengoptimal. Pengoptimal yang umum kami gunakan meliputi SGD, Momentum, Nesterov, Adagrad, Adadelta, RMSprop, Adam, Adamax, dan Nadam, dll. Pengoptimal ini terutama meningkatkan algoritma dalam hal kecepatan konvergensi, stabilitas setelah konvergensi ke ekstrem lokal, dan efisiensi penyesuaian hiperparameter. Bagian ini akan memperkenalkan pengoptimal yang paling umum digunakan dan desainnya.

Momentum

Pengoptimal Momentum merupakan penyempurnaan mendasar dari algoritma penurunan gradien, yang melengkapi suku momentum pada persamaan pembaruan bobot, seperti yang ditunjukkan pada Gambar 3.18.



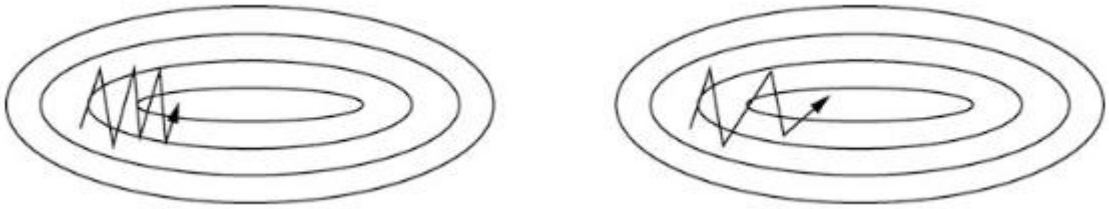
Gambar 3.18 Peran suku momentum

Misalkan nilai perubahan bobot pada iterasi ke-n adalah d(n), maka persamaan pembaruan bobot akan berubah menjadi:

$$d(n) = -\eta \nabla_w J + ad(n-1)$$

Di mana a adalah angka konstan antara 0 dan 1, yang dikenal sebagai momentum, dan $ad(n-1)$ disebut suku momentum. Bayangkan sebuah bola kecil menggelinding ke bawah sepanjang permukaan galat dari suatu titik acak. Algoritma penurunan gradien biasa akan menggerakkan bola tersebut bergerak sepanjang kurva gaya, tetapi sebenarnya hal ini bertentangan dengan hukum fisika. Situasi sebenarnya adalah momentum bola kecil tersebut akan terakumulasi saat menggelinding ke bawah, dan kecepatannya akan meningkat lebih cepat di sepanjang arah menurun.

Di tempat-tempat dengan gradien yang relatif stabil, bola kecil tersebut akan menggelinding jauh lebih cepat sehingga dapat dengan cepat melewati daerah datar. Akibatnya, konvergensi model akan semakin cepat. Di sisi lain, seperti yang ditunjukkan pada Gambar 3.19, suku momentum memperbaiki arah gradien dan mengurangi terjadinya perubahan mendadak. Selain itu, bola kecil yang dibawa dengan inersia lebih mungkin untuk mengatasi titik ekstrem lokal yang sempit, sehingga model akan lebih kecil kemungkinannya untuk terjebak di titik-titik ekstrem lokal. Kerugian dari pengoptimal momentum adalah momentum dapat membawa bola kecil begitu jauh sehingga mungkin melewatkan solusi optimal, sehingga diperlukan iterasi tambahan untuk konvergensi. Kedua, laju pembelajaran dan momentum a dari pengoptimal momentum diatur secara manual. Artinya, banyak eksperimen harus dilakukan untuk menemukan nilai yang tepat untuk digunakan.



Gambar 3.19 Suku momentum mempercepat konvergensi model

Pengoptimal Adagrad

Karakteristik umum dari penurunan gradien stokastik, penurunan gradien mini-batch, dan pengoptimal momentum adalah bahwa setiap parameter diperbarui dengan satu laju pembelajaran yang sama. Namun, untuk pengoptimal Adagrad, parameter yang berbeda harus mengadopsi laju pembelajaran yang berbeda. Persamaan pembaruan gradien dari pengoptimal Adagrad umumnya ditulis sebagai:

$$\Delta w = -\frac{\eta}{e + \sqrt{\sum_{i=1}^{n} \mathrm{g}^2(i)}} \mathrm{g}(n)$$

Dalam persamaan tersebut, $g(n)$ merepresentasikan gradien dJ/dw dari fungsi biaya pada iterasi $ke - n$, dan e adalah konstanta kecil. Dengan peningkatan n, penyebut persamaan akan meningkat, dan derajat pembaruan bobot akan menurun secara bertahap, yang setara dengan laju pembelajaran yang telah dikurangi secara dinamis. Pada tahap awal pelatihan model, nilai awal bahkan tidak mendekati solusi optimal fungsi kerugian, sehingga memerlukan laju pembelajaran yang jauh lebih tinggi.

Namun, seiring meningkatnya frekuensi pembaruan, parameter bobot akan mendekati solusi optimal, dan laju pembelajaran akan terus menurun. Keuntungan pengoptimal Adagrad adalah dapat memperbarui laju pembelajaran secara otomatis, tetapi fitur ini juga memiliki beberapa kelemahan. Karena pembaruan laju pembelajaran ditentukan oleh gradien iterasi sebelumnya, kemungkinan besar laju pembelajaran telah dikurangi menjadi nol ketika parameter bobot masih jauh dari solusi optimal. Jika demikian, proses optimasi akan menjadi tidak berarti.

Pengoptimal RMSprop

Pengoptimal Root Mean Square Propagation (RMSprop) merupakan versi perbaikan dari pengoptimal Adagrad dengan menambahkan koefisien atenuasi ke dalam algoritmanya, yang mengatenuasi gradien catatan historis sebesar persentase tertentu untuk setiap iterasi. Persamaan pembaruan gradien dari pengoptimal RMSprop adalah sebagai berikut:

$$r(n) = br(n-1) + (1-b)g^2(n)$$
$$\Delta w = -\frac{\eta}{e + \sqrt{r(n)}} g^2(n)$$

Dalam persamaan, b mengacu pada faktor atenuasi dan e adalah konstanta kecil. Karena pengaruh faktor atenuasi, r tidak harus meningkat secara monoton seiring dengan peningkatan n. Hal ini secara efektif dapat mencegah pengoptimal Adagrad menghentikan pembelajaran terlalu dini, dan sangat cocok untuk menangani target non-stasioner, terutama yang kondusif untuk jaringan saraf rekuren.

Pengoptimal Adam

Pengoptimal Estimasi Momen Adaptif (Adam) saat ini merupakan pengoptimal yang paling banyak digunakan, yang merupakan evolusi dari Adagrad dan Adadelta. Adam bertujuan untuk mengidentifikasi laju pembelajaran adaptif untuk setiap parameter, yang berguna dalam struktur jaringan yang kompleks, karena sensitivitas setiap bagian jaringan terhadap penyesuaian bobot berbeda, dan laju pembelajaran untuk bagian sensitif umumnya harus lebih rendah. Sulit dan rumit bagi orang untuk mengidentifikasi bagian sensitif dan menghitung laju pembelajaran spesifik secara manual. Persamaan pembaruan gradien pengoptimal Adam serupa dengan pengoptimal RMSprop, seperti yang ditunjukkan di bawah ini:

$$\Delta w = -\frac{\eta}{e + \sqrt{v(n)}} m(n)$$

Di mana m dan v masing-masing mewakili estimasi momen pertama (rata-rata) dan momen kedua (varians tak terpusat) dari gradien masa lalu. Mirip dengan rumus berbasis atenuasi yang diusulkan oleh pengoptimal RMSprop, m dan v dapat didefinisikan sebagai:

$$m\ (n) = am¼\ (n-1) + (1-a)\ g\ (n)$$
$$v(n) = bv(n-1) + (1-b)\ g^2(n)$$

Secara bentuk, m dan v masing-masing adalah rata-rata bergerak gradien dan kuadrat gradien. Namun, mendefinisikannya demikian akan membuat algoritma menjadi tidak stabil dalam beberapa iterasi pertama. Mari kita asumsikan bahwa m(0) dan v(0) bernilai nol, sehingga ketika a dan b mendekati satu, m dan v akan sangat mendekati nol pada iterasi awal. Untuk mengatasi masalah ini, persamaan akhir yang digunakan adalah sebagai berikut:

$$\tilde{m}(n) = \frac{m(n)}{1-a^n}$$
$$\tilde{v}(n) = \frac{v(n)}{1-b^n}$$

Meskipun laju pembelajaran, a, dan b dalam pengoptimal Adam perlu diatur secara manual, kesulitan implementasinya telah berkurang secara signifikan. Eksperimen telah membuktikan bahwa nilai default a dan b masing-masing adalah 0,9 dan 0,999, dan laju pembelajaran adalah 0,0001. Dalam praktiknya, pengoptimal Adam akan memfasilitasi konvergensi yang cepat. Ketika algoritma mencapai titik saturasi, laju pembelajaran dapat diturunkan, dan parameter lainnya dapat tetap tidak disesuaikan. Dengan pengurangan laju pembelajaran beberapa kali, model akan konvergen ke nilai ekstrem yang tepat dan memuaskan.

3.6 JENIS JARINGAN SARAF TIRUAN

Sejak jaringan saraf tiruan BP (BP) pertama, manusia telah merancang berbagai macam jaringan saraf tiruan untuk menangani berbagai masalah. Dalam bidang visi komputer, jaringan saraf tiruan konvolusional merupakan model mendalam yang paling banyak digunakan saat ini. Dalam bidang pemrosesan bahasa alami (NLP), jaringan saraf tiruan rekuren pernah mengungguli model-model lainnya. Bagian ini juga akan memperkenalkan model generatif berdasarkan teori permainan jaringan adversarial generatif.

Jaringan Saraf Tiruan Konvolusional

Jaringan saraf tiruan konvolusional (CNN) merupakan jenis jaringan saraf tiruan umpan maju. Tidak seperti jaringan saraf tiruan terhubung penuh (FCNN), neuron buatan pada jaringan saraf tiruan konvolusional dapat merespons sebagian unit dalam rentang cakupan, dan memiliki kinerja yang baik dalam pemrosesan gambar. Secara umum, jaringan saraf tiruan konvolusional terdiri dari lapisan konvolusional, lapisan penyatuan, dan lapisan terhubung penuh.

Pada tahun 1960-an, dalam studi mereka tentang neuron korteks kucing yang sensitif terhadap lokalitas dan digunakan untuk memilih arah, David Hubel dan Torsten Wiesel menemukan bahwa struktur jaringan unik neuron kucing dapat secara efektif mengurangi kompleksitas jaringan saraf umpan balik. Terinspirasi oleh penemuan mereka, jaringan saraf konvolusional diusulkan. Saat ini, jaringan saraf konvolusional telah menjadi salah satu topik yang paling hangat dibahas di berbagai bidang ilmiah, khususnya dalam pengenalan pola. Karena struktur CNN dapat menghindari prapemrosesan gambar yang rumit dan dapat langsung memasukkan gambar asli, CNN telah diterapkan secara luas di berbagai bidang.

Nama "jaringan saraf konvolusional" berasal dari operasi matematika yang disebut konvolusi. Konvolusi adalah operasi melakukan produk dalam terhadap gambar (atau peta fitur) dan matriks filter (juga disebut kernel filter dan konvolusi). Gambar adalah input dari jaringan saraf, dan peta fitur adalah output dari lapisan konvolusional atau lapisan penyatuan jaringan saraf. Perbedaannya terletak pada nilai dalam peta fitur yang merupakan keluaran neuron, sehingga secara teoritis tidak memiliki batasan.

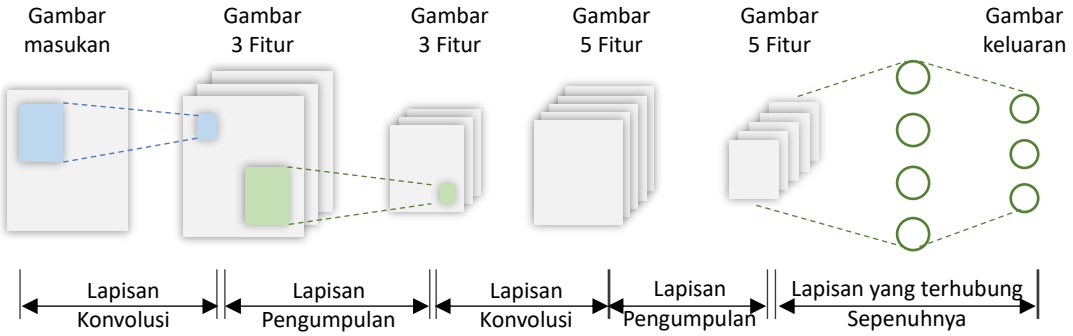
Namun, nilai gambar sejalan dengan kecerahan tiga kanal RGB, yang bernilai dari 0 hingga 255. Setiap lapisan konvolusional dalam jaringan saraf tiruan berkorespondensi dengan satu atau lebih matriks filter, dan cara kerjanya berbeda dari jaringan saraf tiruan yang terhubung penuh. Setiap neuron dalam lapisan konvolusional hanya dapat menerima keluaran neuron di jendela lokal tertentu dari lapisan sebelumnya sebagai masukan, alih-alih dapat menerima keluaran semua neuron di lapisan sebelumnya secara bersamaan. Fitur konvolusional ini dikenal sebagai persepsi lokal.

Secara umum, persepsi orang terhadap dunia luar berevolusi dari fitur lokal ke fitur global. Hubungan spasial suatu gambar juga sama, yaitu piksel lokal lebih erat hubungannya, sedangkan piksel jauh kurang terkait. Oleh karena itu, sebenarnya, setiap neuron tidak perlu mempersepsi gambar global. Neuron hanya perlu melakukan persepsi lokal, dan mensintesis data yang dipersepsikan secara lokal pada tingkat yang lebih tinggi untuk mendapatkan informasi global. Gagasan jaringan yang terhubung sebagian terinspirasi oleh struktur sistem visual biologis di mana neuron di korteks visual juga hanya merespons stimulus di wilayah tertentu dan menerima informasi secara lokal.

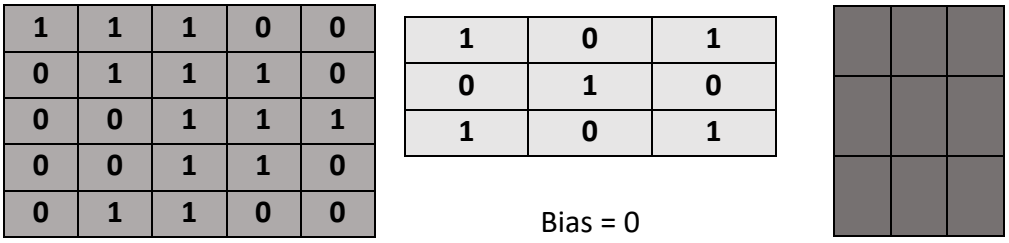
Fitur lain dari konvolusi adalah pembagian parameter. Untuk citra masukan, citra tersebut dapat dipindai oleh satu atau lebih kernel konvolusi. Parameter dalam kernel konvolusi sama dengan bobot model. Dalam lapisan konvolusional, semua neuron berbagi kernel konvolusi yang sama dan oleh karena itu juga berbagi bobot yang sama. Pembagian bobot berarti bahwa setiap kali kernel konvolusi melintasi citra, parameternya tidak akan berubah. Misalnya, lapisan konvolusional memiliki tiga kernel konvolusi fitur, dan setiap kernel konvolusi akan memindai seluruh citra. Selama pemindaian, nilai parameter kernel konvolusi tetap, dan piksel dari seluruh citra berbagi bobot yang sama. Ini berarti bahwa fitur yang dipelajari di bagian tertentu dari citra juga dapat diterapkan ke bagian lain atau citra lain. Fitur ini dikenal sebagai invariansi posisi.

1. Lapisan Konvolusional

Pada gambar 3.20 menampilkan arsitektur umum jaringan saraf tiruan konvolusional. Citra paling kiri adalah masukan. Citra masukan pertama-tama melewati lapisan konvolusional yang terdiri dari tiga kernel konvolusional untuk memperoleh tiga peta fitur. Parameter ketiga kernel konvolusional ini independen satu sama lain, dan dapat dihitung serta dioptimalkan dengan algoritma propagasi balik. Saat melakukan operasi konvolusi, sebuah jendela citra masukan diproyeksikan ke neuron pada peta fitur. Operasi konvolusional dilakukan untuk mendeteksi berbagai fitur masukan. Lapisan konvolusional pertama mungkin hanya menampilkan beberapa fitur tingkat rendah, seperti tepi, garis, dan sudut citra. Lapisan-lapisan berikutnya dapat mempelajari fitur yang lebih kompleks dari fitur tingkat rendah melalui iterasi.



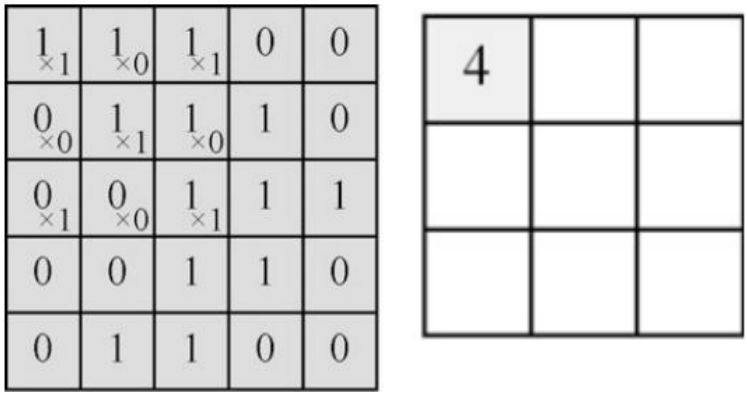
Gambar 3.20 Arsitektur jaringan saraf tiruan konvolusional



Gambar 3.21 Contoh operasi konvolusional

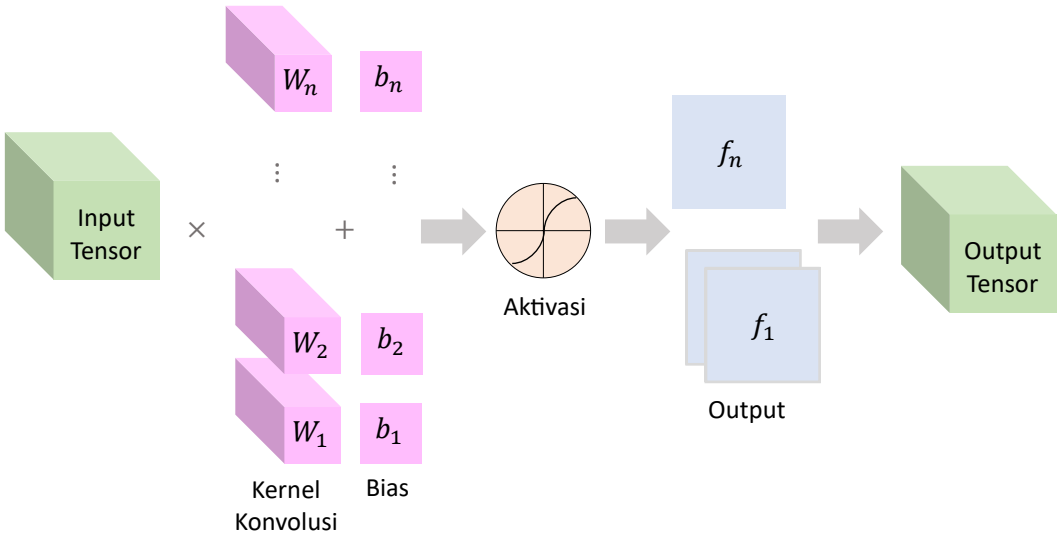
Dengan melihat operasi konvolusional yang ditunjukkan pada Gambar 3.21, Anda dapat membagi hingga 3 × 3 wilayah berbeda dalam matriks persegi 5 × 5, dan bentuk wilayah-wilayah ini identik dengan kernel konvolusional. Oleh karena itu, dimensi peta fitur adalah 3 × 3. Seperti yang ditunjukkan pada Gambar 3.22, setiap elemen dalam peta fitur dihitung dengan mengalikan satu bagian dari citra asli dan kernel konvolusi. Dalam matriks di sebelah kiri Gambar 3.22, wilayah 3 × 3 di sudut kiri atas adalah bagian yang relevan dengan elemen kiri atas dalam peta fitur. Dengan mengalikan setiap elemen dari bagian ini dan elemen kernel

konvolusi yang bersesuaian, dan menjumlahkan hasilnya, kita akan mendapatkan elemen pertama, 4, dari peta fitur. Contoh yang ditunjukkan di sini tidak menyertakan suku bias, yaitu, bias sama dengan nol. Dalam operasi konvolusi yang lebih umum, seringkali diperlukan untuk menjumlahkan hasil dan suku bias setelah perkalian (perkalian titik), sehingga menghasilkan peta fitur. Peran suku bias di sini serupa dengan yang ada dalam regresi linier.



Gambar 3.22 Contoh operasi konvolusi

Lapisan konvolusional sebagian besar terstruktur oleh konvolusi multikanal. Seperti ditunjukkan pada Gambar 3.23, lapisan konvolusional dapat menampung beberapa kernel konvolusi dan bias. Kombinasi kernel konvolusi dan bias mampu memproyeksikan tensor masukan ke peta fitur. Peran konvolusi multikanal adalah untuk menggabungkan semua peta fitur yang dihasilkan oleh kernel konvolusi dan suku bias untuk membentuk matriks tiga dimensi sebagai keluaran. Secara umum, tensor masukan dan keluaran serta kernel konvolusi adalah matriks tiga dimensi, yang masing-masing memiliki tiga dimensi yang mengacu pada lebar, tinggi, dan kedalaman.



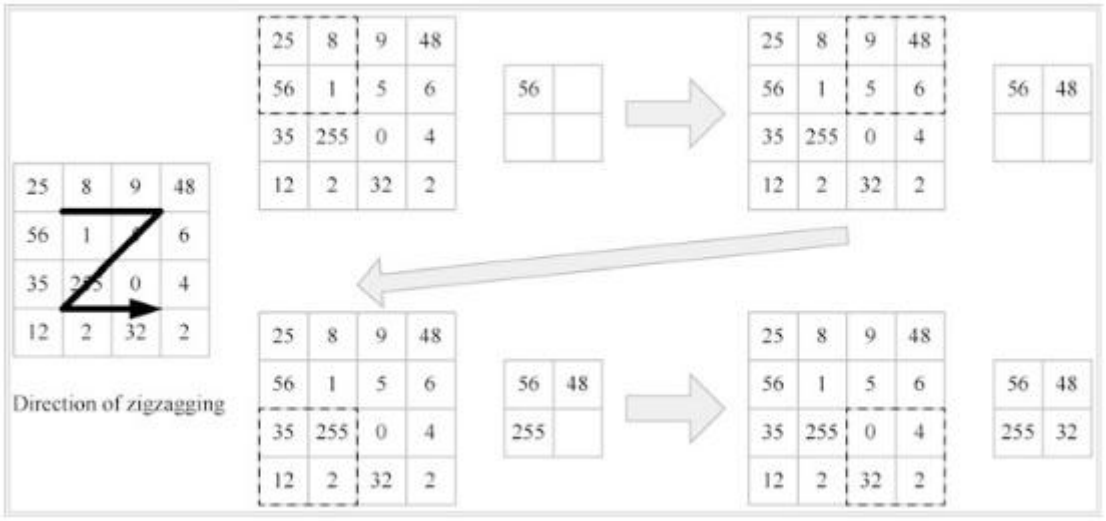
Gambar 3.23 Struktur lapisan konvolusional

Untuk memperluas konvolusi yang disebutkan di atas ke bidang tiga dimensi, perlu dibuat aturan bahwa kedalaman setiap kernel konvolusi harus sama dengan tensor masukan. Hal ini memastikan bahwa kedalaman peta fitur yang sesuai dengan setiap kernel konvolusi tunggal adalah satu. Operasi konvolusi tidak memiliki persyaratan khusus terkait lebar dan tinggi kernel konvolusi, tetapi keduanya umumnya dinilai sama dalam operasi demi kenyamanan. Selain itu, agar peta fitur yang dihasilkan oleh kernel konvolusi yang berbeda dapat ditumpuk bersama-sama, peta fitur tersebut harus memiliki lebar dan tinggi yang sama. Dengan kata lain, semua kernel konvolusi dalam lapisan konvolusi yang sama harus berukuran sama.

Sebagai keluaran dari lapisan konvolusional, peta fitur perlu diaktifkan. Terkadang, fungsi aktivasi dapat dianggap sebagai bagian yang membentuk lapisan konvolusional. Namun, karena fungsi aktivasi dan operasi konvolusi tidak begitu relevan, kita juga dapat menganggap fungsi aktivasi sebagai lapisan independen. Lapisan aktivasi yang paling umum digunakan adalah lapisan linier terkoreksi, yang menggunakan fungsi aktivasi ReLU.

2. Lapisan Penggabungan

Lapisan penggabungan berfungsi sebagai alat reduksi dimensionalitas dengan mengelompokkan unit-unit di sekitarnya dan mengurangi ukuran peta fitur. Lapisan penggabungan yang paling umum digunakan meliputi lapisan penggabungan maksimum dan lapisan penggabungan rata-rata. Seperti ditunjukkan pada Gambar 3.24, lapisan penggabungan maksimum membagi peta fitur menjadi beberapa patch persegi panjang dan mengambil nilai maksimum setiap patch sebagai nilai karakteristik keseluruhan wilayah. Sementara itu, penggabungan rata-rata serupa dengan penggabungan maksimum, hanya saja lapisan ini mengganti nilai karakteristik maksimum dengan nilai rata-rata untuk merepresentasikan wilayah tersebut.



Gambar 3.24 Contoh operasi pooling

Dalam peta fitur, ukuran setiap patch disebut ukuran jendela penggabungan. Dalam jaringan saraf tiruan konvolusional yang sebenarnya, lapisan konvolusional dan lapisan penggabungan pada dasarnya saling terhubung. Seperti halnya konvolusi, penggabungan juga dapat

meningkatkan skala fitur, sehingga mengekstraksi fitur dari lapisan sebelumnya. Namun, tidak seperti operasi konvolusi, lapisan penggabungan tidak memiliki parameter apa pun dan tidak ada hubungannya dengan susunan elemen dalam patch-patch kecil. Lapisan ini hanya relevan dengan karakteristik statistik elemen-elemen tersebut.

Lapisan pooling terutama bertujuan untuk mengurangi ukuran data masukan dari lapisan berikutnya agar dapat menyederhanakan parameter secara efektif dan menurunkan intensitas komputasi, yang pada saat yang bersamaan mencegah terjadinya over-fitting. Fungsi lain dari lapisan pooling adalah dapat memetakan masukan dengan ukuran acak apa pun ke keluaran dengan panjang tetap tertentu dengan memberikan ukuran dan panjang langkah yang wajar pada jendela pooling. Misalkan ukuran masukan adalah $a \times a$, ukuran jendela pooling adalah $a/4$, dan panjang langkah adalah $a/4$.

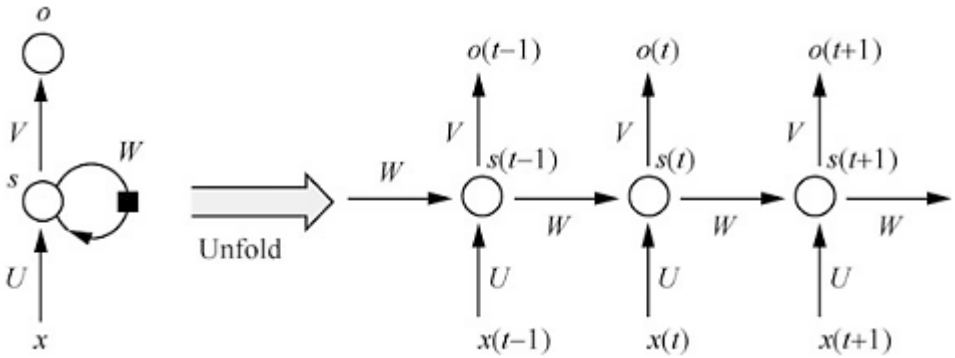
Jika a adalah kelipatan 4, maka ukuran jendela pengumpulan sama dengan panjang langkah, dan mudah untuk menemukan bahwa ukuran keluaran lapisan pengumpulan adalah 4×4. Ketika a adalah bilangan bulat yang cukup besar tetapi tidak habis dibagi 4, maka ukuran jendela pengumpulan selalu lebih besar dari panjang langkah sebanyak satu satuan, dan dapat dibuktikan bahwa ukuran keluaran lapisan pengumpulan tetap 4×4. Fitur pembeda dari lapisan pengumpulan memungkinkan jaringan saraf konvolusional untuk menangani citra masukan dengan ukuran berapa pun.

3. Lapisan Terhubung Penuh

Lapisan terhubung penuh biasanya digunakan untuk keluaran jaringan saraf konvolusional. Dalam pengenalan pola, klasifikasi atau regresi adalah dua tugas yang umum dilakukan. Lebih spesifiknya, klasifikasi berarti menentukan kategori objek, atau memberi peringkat objek dalam citra. Untuk mengatasi masalah ini, menggunakan peta fitur sebagai keluaran tampaknya kurang tepat. Peta fitur harus diproyeksikan ke vektor yang memenuhi persyaratan tertentu, dan ini akan memerlukan penyematan peta fitur, yang berarti menyusun neuron dalam peta fitur ke dalam vektor dalam urutan tetap.

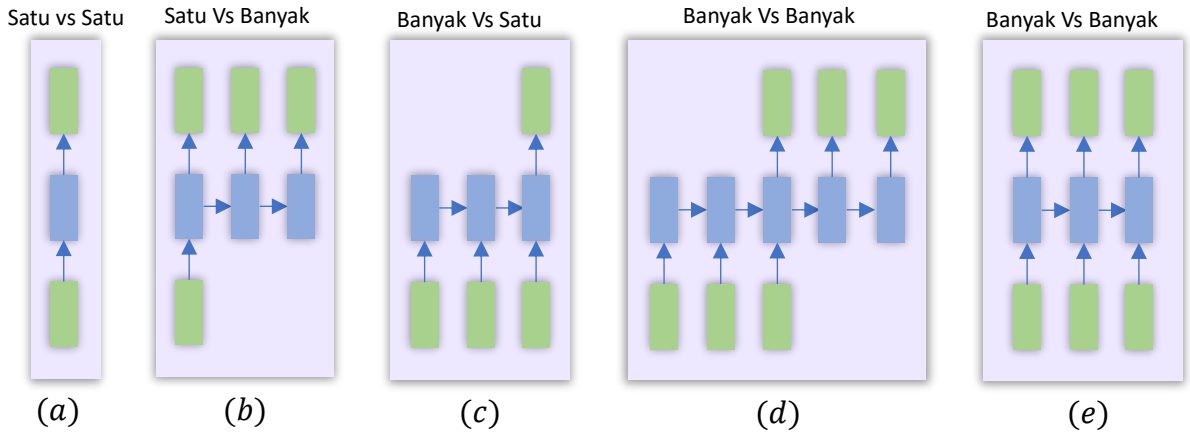
Jaringan Saraf Tiruan Berulang

Jaringan Saraf Tiruan Berulang (RNN) adalah jenis jaringan saraf tiruan yang menangkap informasi dinamis dalam data serial dengan menghubungkan simpul-simpul pada lapisan tersembunyi secara berkala, untuk mengklasifikasikan data serial. Tidak seperti jaringan saraf tiruan umpan maju lainnya, jaringan saraf tiruan berulang dapat mempertahankan status konteks data serial. Jaringan saraf tiruan berulang tidak lagi dibatasi oleh batas spasial seperti jaringan saraf tiruan tradisional, tetapi dapat diperluas dalam dimensi waktu. Sederhananya, simpul-simpul sel memori pada saat ini dan selanjutnya dapat dihubungkan.



Gambar 3.25 Struktur jaringan saraf tiruan rekuren

Jaringan saraf tiruan berulang banyak digunakan dalam skenario yang menampilkan data sekuensial, seperti video, audio, dan kalimat. Apa yang ditunjukkan pada Gambar 3.25 adalah struktur khas dari jaringan saraf berulang, di mana $x(t)$ mewakili nilai urutan input pada langkah waktu t, dan $s-(t)$ mewakili keadaan sel memori pada langkah waktu t, $o(t)$ mewakili output dari lapisan tersembunyi pada langkah waktu t, dan U, V dan W mewakili bobot model. Dapat disimpulkan bahwa pembaruan lapisan tersembunyi tidak hanya bergantung pada input saat ini $x(t)$, tetapi juga bergantung pada keadaan sel memori pada langkah waktu sebelumnya $s(t-1)$. Persamaan harus $s(t) = f(Ux(t) + Ws(t-1))$, di mana f mewakili fungsi aktivasi. Lapisan output dari jaringan saraf berulang sama dengan multilayer perceptron, dan bab ini tidak akan membahas detailnya.



Gambar 3.26 Berbagai struktur jaringan saraf tiruan rekuren

Seperti yang ditunjukkan pada Gambar 3.26, terdapat beragam struktur untuk berbagai jaringan saraf tiruan rekuren. Gambar 3.26a menggambarkan jaringan saraf tiruan BP umum yang tidak terkait dengan deret waktu. Gambar 3.26b adalah model generatif yang dapat menghasilkan sekuens sesuai dengan persyaratan spesifik berdasarkan setiap masukan. Gambar 3.26c menunjukkan model yang dapat digunakan untuk klasifikasi keseluruhan atau regresi sekuens, dan merupakan struktur jaringan saraf tiruan rekuren yang paling klasik.

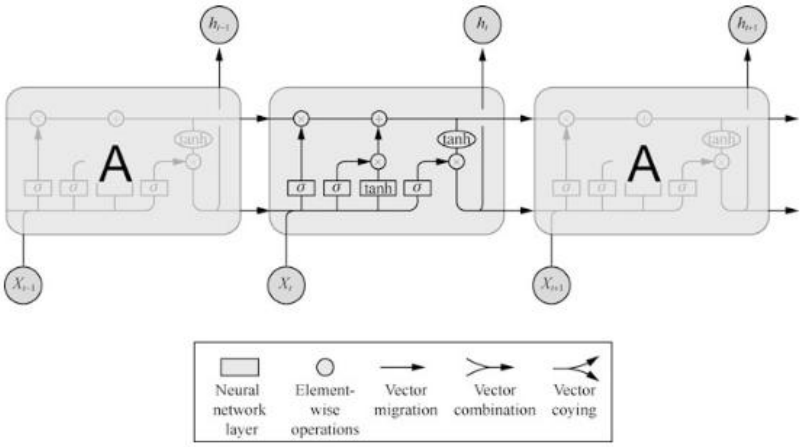
Gambar 3.26d, e menunjukkan model yang dapat digunakan untuk translasi sekuens. Struktur pada Gambar 3.26d juga dikenal sebagai arsitektur encoder-decoder.

Perhitungan jaringan saraf tiruan rekuren dioperasikan melalui algoritma Backpropagation Trough Time (BPTT), yang merupakan perluasan dari algoritma backpropagation tradisional dalam domain waktu. Algoritma BP tradisional hanya mempertimbangkan propagasi ketidakpastian (atau propagasi kesalahan) antara lapisan tersembunyi yang berbeda, sedangkan algoritma BPTT perlu mempertimbangkan propagasi ketidakpastian dalam lapisan tersembunyi yang sama pada langkah waktu yang berbeda. Secara khusus, kesalahan sel memori pada langkah waktu t terdiri dari dua bagian: komponen yang disebarkan oleh lapisan tersembunyi pada langkah waktu t, dan komponen yang disebarkan oleh unit memori pada langkah waktu $t + 1$.

Ketika kedua komponen ini disebarkan secara terpisah, metode komputasinya sama dengan algoritma BP tradisional. Ketika mereka disebarkan ke sel memori, jumlah dari dua komponen akan diambil sebagai kesalahan sel memori pada langkah waktu t. Menurut kesalahan lapisan tersembunyi dan sel memori pada langkah waktu t, mudah untuk menyimpulkan gradien parameter U, V, dan W pada langkah waktu t. Setelah melintasi semua langkah waktu dalam backpropagation, setiap parameter termasuk U, V, dan W akan mendapatkan gradien T, dengan T adalah total panjang waktu. Gradien T yang dijumlahkan adalah gradien dari parameter U, V, dan W. Ketika kita mendapatkan gradien setiap parameter, persamaan algoritma penurunan gradien dapat dengan mudah dipecahkan.

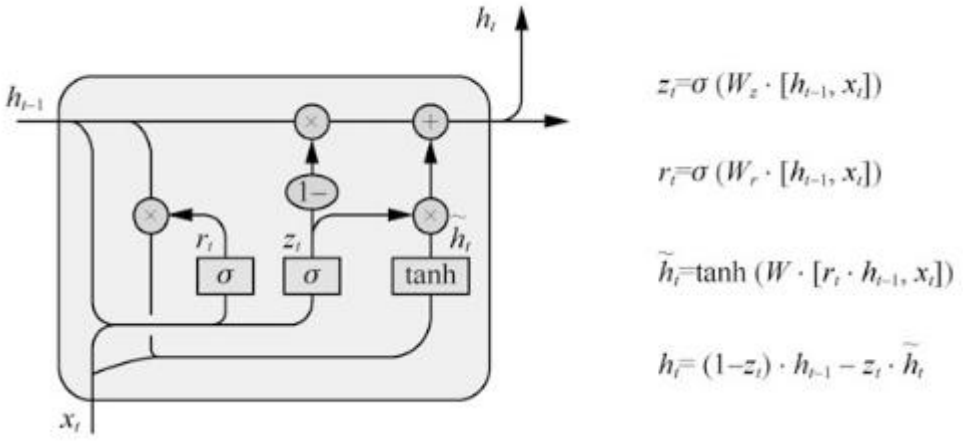
Masih banyak masalah yang harus diatasi dengan jaringan saraf rekuren. Karena sel memori akan menerima keluaran dari waktu sebelumnya setiap saat, masalah hilangnya gradien dan ledakan gradien yang menjadi ciri jaringan saraf terhubung penuh dalam juga mengganggu jaringan saraf rekuren. Di sisi lain, keadaan sel memori pada langkah waktu tertentu t tidak dapat dipertahankan untuk waktu yang lama, dan keadaan sel memori perlu dipetakan oleh fungsi aktivasi untuk setiap langkah waktu yang dilaluinya. Untuk urutan yang relatif panjang, ketika berulang hingga akhir, masukan di awal urutan mungkin telah lenyap bersamaan dengan pemetaan fungsi aktivasi. Dengan kata lain, memori jangka panjang jaringan saraf rekuren pada akhirnya akan meluruh.

Namun untuk banyak tugas, kami berharap model tersebut dapat menyimpan informasi lebih lama, seperti halnya bayangan dalam fiksi detektif yang hanya dapat terungkap hingga akhir cerita. Namun, jika sel memori memiliki kapasitas terbatas, jaringan saraf rekuren pasti tidak akan mampu mengingat semua informasi dalam keseluruhan rangkaian. Oleh karena itu, kami ingin sel memori mengingat informasi kunci secara selektif, yang dapat dicapai melalui Memori Jangka Panjang dan Jangka Pendek (LSTM).



Gambar 3.27 Jaringan memori jangka panjang dan jangka pendek

Seperti ditunjukkan pada Gambar 3.27 (Memahami Jaringan LSTM), inti dari jaringan memori jangka panjang dan jangka pendek adalah blok LSTM, yang merupakan alternatif dari lapisan tersembunyi jaringan saraf rekuren. Blok LSTM umum berisi tiga unit komputasi: gerbang masukan, gerbang lupa, dan gerbang keluaran, yang memungkinkan jaringan LSTM untuk mengingat, melupakan, dan mengeluarkan secara selektif, sehingga mewujudkan memori selektif. Kita dapat melihat bahwa blok LSTM yang berdekatan dihubungkan oleh dua garis, yang mewakili status sel dan status tersembunyi jaringan LSTM. Seperti ditunjukkan pada Gambar 3.28, Gated Recurrent Unit (GRU) merupakan varian dari jaringan LSTM, yang menggabungkan forget gate dan input gate menjadi update gate, serta menggabungkan cell state dan hidden state menjadi satu hidden state. Sebagai model yang sangat populer, struktur model GRU lebih ringkas dibandingkan model LSTM standar.



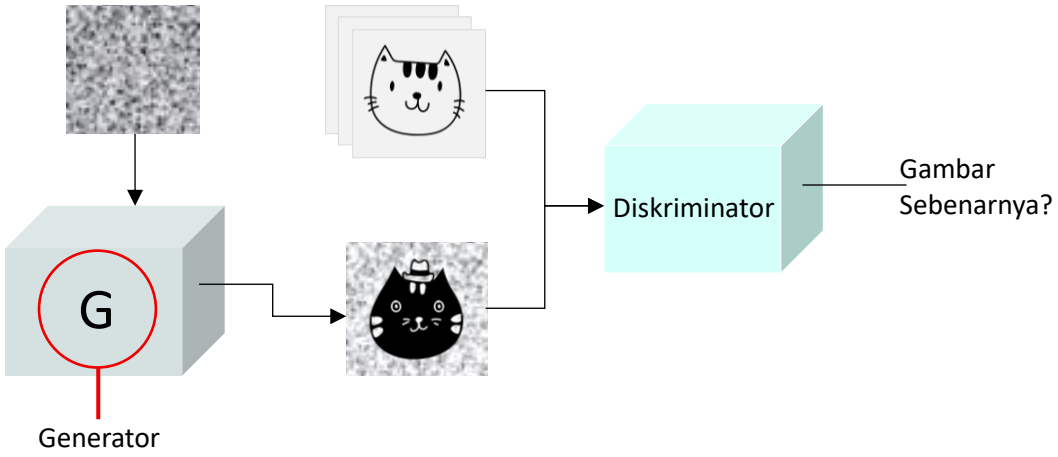
Gambar 3.28 Unit rekursif berpagar (GRU)

Jaringan Adversarial Generatif

Jaringan Adversarial Generatif (GAN) adalah jenis kerangka kerja yang digunakan dalam skenario seperti pembangkitan gambar, segmentasi semantik, pembangkitan teks, augmentasi data, bot obrolan, pengambilan informasi, dan pemeringkatan. Sebelum diperkenalkannya

jaringan adversarial generatif, model generatif mendalam selalu memerlukan rantai Markov atau estimasi kemungkinan maksimum perkiraan, yang dapat menyebabkan banyak masalah ketidakpastian yang tak terduga. Melalui proses adversarial, GAN melatih generator G dan diskriminator D secara bersamaan dan membiarkan keduanya memainkan permainan zero-sum. Yaitu, diskriminator D dirancang untuk menentukan apakah suatu sampel nyata, atau dihasilkan oleh generator G. Sementara generator G bertujuan untuk menghasilkan sampel yang dapat mengelabui diskriminator D. Metode yang diadopsi untuk melatih GAN adalah algoritma BP yang matang.

Seperti yang ditunjukkan pada Gambar 3.29, masukan dari generator dianggap sebagai derau z. Derau z mengikuti distribusi probabilitas prior yang dipilih secara artifisial, seperti distribusi seragam dan distribusi Gaussian. Ruang masukan dapat dipetakan ke ruang sampel berdasarkan arsitektur jaringan tertentu. Masukan diskriminator adalah sampel asli x atau sampel palsu G(z), dan keluarannya adalah keaslian sampel. Diskriminator dapat dirancang berdasarkan model klasifikasi apa pun, seperti jaringan saraf tiruan konvolusional dan jaringan saraf tiruan terhubung penuh yang umum digunakan sebagai diskriminator. Misalnya, kita mungkin ingin menghasilkan gambar bertema kucing, dan gambar tersebut harus senyata mungkin. Target diskriminator adalah untuk membedakan apakah gambar tersebut asli atau tidak.



Gambar 3.29 Struktur jaringan adversarial generatif

Fungsi objektif jaringan adversarial generatif adalah:

$$G = min_G max_D E_{X \sim P_{datat}}[logD(x)] + E_{z \sim P_z}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

Fungsi objektif terdiri dari dua bagian. Bagian pertama hanya terkait dengan diskriminator D. Ketika sampel nyata adalah input, semakin dekat output diskriminator D ke 1, semakin besar nilai bagian pertama. Bagian kedua terkait dengan generator G dan diskriminator D. Jika gangguan acak adalah input, generator G dapat menghasilkan sampel. Diskriminator D menerima sampel ini sebagai input, dan semakin dekat output ke 0, semakin besar nilai bagian

kedua. Karena diskriminator D bertujuan untuk memaksimalkan fungsi objektif, ia perlu mengeluarkan nilai 1 di bagian pertama dan nilai 0 di bagian kedua, sehingga dapat mengklasifikasikan sampel dengan benar.

Meskipun generator bertujuan untuk meminimalkan fungsi objektif, bagian pertama dari fungsi objektif tidak terkait dengan generator, sehingga generator hanya dapat meminimalkan bagian kedua sebanyak mungkin. Untuk mencapai hal tersebut, generator perlu mengeluarkan sampel yang memastikan nilai keluaran diskriminator adalah 1, yang berarti membuat diskriminator tidak dapat membedakan antara benar dan salah sebanyak mungkin.

Sejak diperkenalkannya GAN pada tahun 2014, GAN telah menghasilkan lebih dari 200 varian, dan telah diterapkan secara luas untuk mengatasi berbagai masalah pembangkitan. Namun, GAN asli juga memiliki banyak masalah, seperti ketidakstabilan prosedur pelatihan. Jaringan saraf tiruan terhubung penuh, jaringan saraf tiruan konvolusional, dan jaringan saraf tiruan berulang yang telah dijelaskan sebelumnya, semuanya dirancang untuk meminimalkan fungsi biaya dengan mengoptimalkan parameter melalui prosedur pelatihan. Pelatihan GAN sedikit berbeda, terutama karena hubungan adversarial antara generator G dan diskriminator D yang menyulitkan pencapaian keseimbangan.

Prosedur pelatihan umum adalah melakukan pelatihan alternatif D dan G, hingga D(G(z)) stabil pada nilai sekitar 0,5. Pada titik ini, D dan G mencapai status Ekuilibrium Nash, yang menandakan bahwa pelatihan telah selesai. Namun, dalam beberapa kasus, sulit bagi model untuk mencapai kesetimbangan Nash, dan hal ini dapat menyebabkan masalah seperti mode collapsing. Oleh karena itu, cara meningkatkan stabilitas model GAN selalu menjadi topik utama dalam bidang studi ini. Secara umum, GAN memang memiliki beberapa masalah, tetapi masalah tersebut tidak memengaruhi peran penting GAN di antara model-model generatif.

3.7 MASALAH UMUM

Model pembelajaran mendalam sangat rumit, sehingga mungkin terdapat berbagai masalah selama prosedur pelatihan. Bagian ini merangkum masalah-masalah yang paling umum terlihat selama pelatihan, sehingga pembaca kami dapat menghabiskan lebih sedikit waktu untuk menemukan dan mengatasinya ketika mereka menghadapi situasi tersebut.

Data Tidak Seimbang

Data tidak seimbang mengacu pada masalah dalam kumpulan data tugas klasifikasi di mana jumlah sampel dalam setiap kategori tidak selalu sama. Masalah ketidakseimbangan data tidak seimbang terutama terjadi ketika sampel dari satu atau lebih kategori untuk prediksi sangat jarang. Misalnya, jika terdapat 4251 citra latih untuk klasifikasi, mungkin terdapat lebih dari 2000 kelas yang memiliki satu citra, dan beberapa kelas berisi dua hingga lima citra. Dalam kondisi ini, model tidak dapat memeriksa setiap kelas secara merata, yang dapat memengaruhi kinerja model. Metode umum untuk memperbaiki data yang tidak seimbang meliputi under-sampling acak, over-sampling acak, dan sampling sintetis.

Under-sampling acak mengacu pada penghapusan sampel secara acak dari kelas-kelas dengan jumlah observasi yang memadai. Metode ini dapat menghemat waktu operasi dan memenuhi persyaratan penyimpanan ketika set pelatihan terlalu besar. Namun, saat menghapus sampel, beberapa informasi penting yang terkandung di dalamnya juga dapat dibuang secara acak. Dan sampel yang tersisa mungkin terlalu bias untuk mewakili mayoritas kelas secara akurat. Oleh karena itu, penggunaan under-sampling acak mungkin tidak memberikan hasil yang akurat pada set pengujian yang sebenarnya seperti yang diharapkan.

Over-sampling acak mengacu pada duplikasi sampel yang ada dalam kelas yang tidak seimbang untuk meningkatkan jumlah observasi. Tidak seperti under-sampling acak, metode ini tidak akan menyebabkan hilangnya data, sehingga seringkali mengungguli under-sampling acak pada set uji yang sebenarnya. Namun, karena sampel yang baru diduplikasi sebenarnya sama dengan sampel asli, hal ini lebih mungkin mengakibatkan overfitting.

Synthetic Minority Over-sampling Technique (SMOTE) adalah penggunaan metode sintetis untuk mewujudkan observasi kelas yang tidak seimbang, sangat mirip dengan metode klasifikasi tetangga terdekat. SMOTE pertama-tama memilih subset data dari kelas minoritas, kemudian menghasilkan sampel baru berdasarkan subset yang dipilih, dan contoh-contoh yang disintesis ini akan ditambahkan ke set data asli. Keuntungan metode ini adalah tidak akan merusak data yang berharga dan secara efektif dapat mempermudah overfitting dengan menghasilkan sampel sintetis melalui pengambilan sampel acak. Namun secara umum, kinerjanya dalam menangani data berdimensi tinggi tidak begitu menjanjikan. Selain itu, saat menghasilkan sampel sintetis, SMOTE tidak akan mempertimbangkan contoh dari kelas lain, yang dapat memperparah tumpang tindih kelas dan menimbulkan gangguan tambahan.

Gradien Menghilang dan Gradien Meledak

Ketika terdapat lapisan jaringan yang cukup, gradien parameter model selama proses backpropagation dapat menjadi sangat kecil atau sangat besar, dan kedua situasi ini masing-masing disebut gradien menghilang dan gradien meledak. Pada dasarnya, kedua masalah ini berasal dari persamaan backpropagation. Mari kita asumsikan bahwa model memiliki tiga lapisan, dan setiap lapisan hanya berisi satu neuron, maka persamaan backpropagation dapat ditulis sebagai:

$$\delta_1 = \delta_3 f'_2(o_1) w_3 f'_1(o_0) w_2$$

Di mana f adalah fungsi aktivasi. Dalam contoh ini, kami menggunakan fungsi Sigmoid. Seiring bertambahnya jumlah lapisan jaringan, frekuensi kemunculan $f(o)w$ dalam persamaan juga akan meningkat. Berdasarkan pertidaksamaan rata-rata aritmatika dan geometri, kita dapat memperoleh nilai maksimum $f'(x) = f(x)(1 - f(x))$ adalah 1/4. Oleh karena itu, dapat disimpulkan bahwa ketika w lebih kecil dari 4, $f(o)w$ pasti lebih kecil dari 1. Dengan mengalikan beberapa suku bernilai kurang dari 1, nilai δ_1 pasti akan mendekati 0, yang menyebabkan gradien menghilang. Gradien meledak disebabkan oleh alasan serupa, terutama terjadi ketika nilai w sangat besar. Artinya, kita mengalikan beberapa suku bernilai lebih besar dari 1, yang menghasilkan nilai δ_1 yang sangat besar.

Masalah gradien menghilang dan gradien meledak muncul terutama karena jaringan terlalu dalam, dan pembaruan bobot jaringan tidak stabil, yang pada dasarnya disebabkan oleh efek perkalian selama backpropagation gradien. Pendekatan populer untuk mengatasi gradien menghilang dan meledak termasuk pra-pelatihan, mengadopsi fungsi aktivasi ReLU, jaringan saraf LSTM dan modul residual dan sebagainya. (ResNet, pemenang ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015, meningkatkan kedalaman model hingga 152 lapisan dengan memperkenalkan modul residual ke dalam model. Sebagai perbandingan, kedalaman model dari kedalaman pemenang ILSVRC 2014 GoogLeNet hanya 27 lapisan.) Solusi utama untuk menangani gradien meledak adalah pemotongan gradien, yang dirancang untuk menetapkan ambang batas gradien, dan menurunkan gradien yang melebihi ambang batas ke batas tertentu untuk mencegah gradien meledak.

Overfitting

Overfitting mengacu pada masalah di mana model berkinerja baik pada set pelatihan tetapi berkinerja buruk pada set pengujian. Ada banyak alasan yang menyebabkan overfitting, seperti dimensi fitur yang tinggi, asumsi model yang kompleks, parameter yang berlebihan, data pelatihan yang terbatas, dan noise yang berlebihan. Namun pada dasarnya, masalah overfitting terutama disebabkan oleh model yang melakukan overfitting pada set pelatihan tanpa mempertimbangkan kemampuan generalisasi jaringan. Akibatnya, model dapat memprediksi set pelatihan dengan baik tetapi selalu gagal dalam prediksi data baru. Untuk mengatasi overfitting yang disebabkan oleh data pelatihan yang tidak mencukupi, kita dapat mencoba mendapatkan lebih banyak data. Salah satu pilihannya adalah mendapatkan lebih banyak data dari sumber data, tetapi akan membutuhkan banyak waktu dan tenaga. Pilihan umum lainnya adalah augmentasi data.

Jika model terlalu kompleks untuk menyebabkan overfitting, ada banyak cara untuk mengatasinya. Metode paling sederhana adalah menyesuaikan hiperparameter model, mengurangi jumlah lapisan dan jumlah neuron dalam jaringan, dll., sehingga membatasi kemampuan jaringan untuk melakukan fitting. Anda juga dapat mempertimbangkan untuk menambahkan teknologi regularisasi ke dalam model. Konten terkait telah diperkenalkan sebelumnya, jadi saya tidak akan mengulanginya di sini.

3.8 KESIMPULAN

Bab ini terutama menguraikan definisi dan perkembangan jaringan saraf tiruan, aturan pelatihan perseptron, dan pengetahuan terkait jaringan saraf tiruan populer seperti CNN, RNN, dan GAN. Selain itu, bab ini juga memperkenalkan permasalahan jaringan saraf tiruan yang umum ditemui dalam rekayasa kecerdasan buatan dan solusinya.

Latihan Soal

1. Pembelajaran mendalam merupakan cabang baru yang berasal dari pembelajaran mesin. Menurut Anda, apa perbedaan pembelajaran mendalam dengan pembelajaran mesin tradisional?

2. Pada tahun 1986, pengenalan perseptron multilapis mengakhiri musim dingin pertama AI dalam sejarah pembelajaran mesin. Mengapa menurut Anda perseptron multilapis dapat menyelesaikan masalah XOR? Apa peran fungsi aktivasi dalam prosedur ini?
3. Fungsi sigmoid merupakan jenis fungsi aktivasi yang banyak diadopsi pada tahap awal penelitian jaringan saraf tiruan. Apa saja kekurangannya? Bagaimana fungsi aktivasi tanh memecahkan masalah-masalah ini?
4. Teknik regularisasi banyak diterapkan dalam model pembelajaran mendalam. Apa tujuannya? Bagaimana Dropout menghasilkan regularisasi?
5. Dikatakan bahwa pengoptimal merupakan enkapsulasi dari algoritma pelatihan model. Pengoptimal yang populer antara lain SGD, Adam, dan lain-lain. Cobalah bandingkan perbedaan kinerjanya.
6. Silakan lihat contoh dan selesaikan operasi konvolusi pada Gambar 3.22.
7. Jaringan saraf tiruan rekuren dapat mempertahankan status konteks dalam data serial. Bagaimana fungsi memori ini direalisasikan? Masalah apa yang mungkin dihadapi ketika menangani sekuens yang panjang?
8. Jaringan adversarial generatif adalah jenis arsitektur jaringan generatif dalam. Jelaskan secara singkat prinsip kerjanya.
9. Gradien menghilang dan gradien meledak adalah dua masalah umum dalam pembelajaran mendalam. Apa yang menyebabkan gradien menghilang dan meledak? Bagaimana cara menghindarinya?

BAB 4
KERANGKA KERJA PEMBELAJARAN MENDALAM (DEEP LEARNING)

Bab ini pertama-tama memperkenalkan kerangka kerja pengembangan yang banyak digunakan dalam pembelajaran mendalam beserta karakteristiknya, dan mengilustrasikan salah satu kerangka kerja representatif, TensorFlow, secara detail. Bab ini bertujuan untuk membantu pembaca memperdalam pemahaman mereka tentang praktik-praktik setelah mempelajari konsep tersebut pada tingkat teoretis, dan untuk mengatasi permasalahan praktis. Di bagian akhir bab ini, kerangka kerja MindSpore yang dikembangkan oleh Huawei diperkenalkan. Kerangka kerja ini memiliki beberapa keunggulan yang tidak dapat dilampaui oleh banyak kerangka kerja saat ini. Setelah membaca bab ini, pembaca dapat memutuskan untuk membaca bagian ini berdasarkan kebutuhan mereka sendiri.

4.1 PENGANTAR KERANGKA KERJA PEMBELAJARAN MENDALAM

Pengantar PyTorch

PyTorch adalah kerangka kerja pengembangan pembelajaran mendalam yang diluncurkan oleh Facebook. Ini adalah paket komputasi ilmiah pembelajaran mesin yang dikembangkan dari Torch. Torch adalah kerangka kerja komputasi ilmiah yang didukung oleh sejumlah besar algoritma pembelajaran mesin dan pustaka operasi tensor yang mirip dengan NumPy. Meskipun Torch dicirikan oleh fleksibilitasnya yang luar biasa, ia mengadopsi Lua, bahasa pemrograman yang kurang populer, sehingga tidak dapat digunakan secara luas. Oleh karena itu, PyTorch berbasis Python diperkenalkan.

PyTorch memiliki karakteristik berikut.

1. Dahulukan Python

PyTorch tidak hanya membangun pengikatan Python ke seluruh kerangka kerja C++. PyTorch mendukung akses Python secara mendetail. Anda dapat menggunakan PyTorch secepat menggunakan NumPy atau SciPy, yang tidak hanya membantu pengguna memahami Python dengan lebih mudah, tetapi juga menjamin bahwa kode tersebut pada dasarnya konsisten dengan kode Python asli.

2. Jaringan Saraf Dinamis

Banyak kerangka kerja arus utama saat ini tidak mendukung jaringan saraf dinamis, seperti TensorFlow 1.x. Menjalankan TensorFlow 1.x memerlukan pembuatan grafik komputasi statis terlebih dahulu, lalu menjalankan grafik tersebut berulang kali melalui metode feed dan run(). Namun, menjalankan PyTorch jauh lebih sederhana. Program PyTorch dapat secara dinamis membangun dan menyesuaikan grafik komputasi saat runtime.

3. Mudah Di-debug

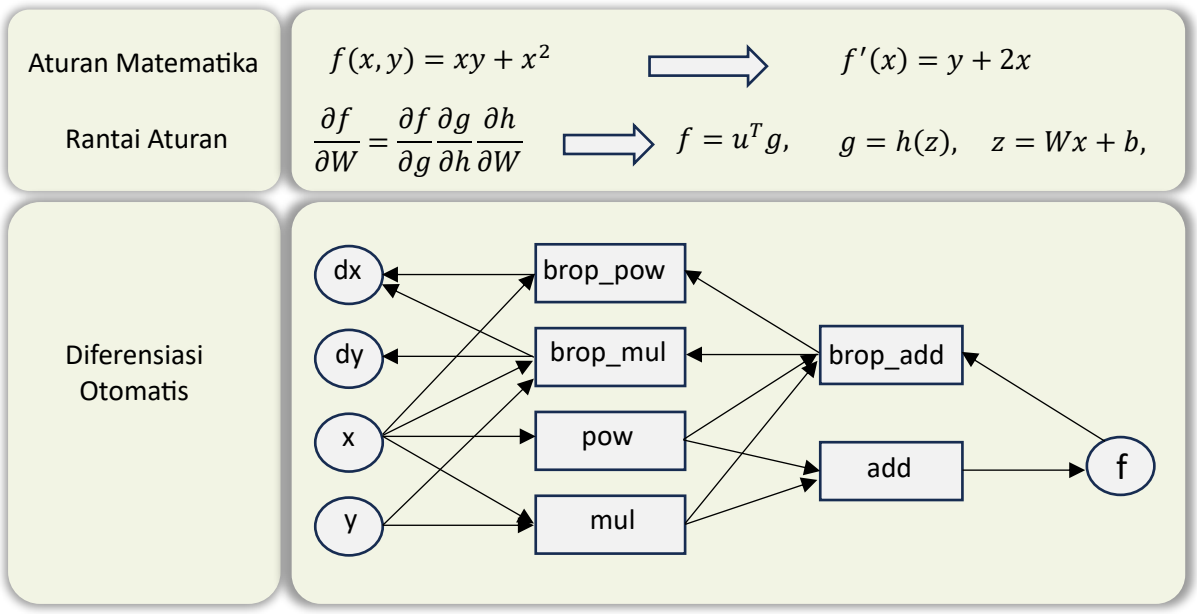
PyTorch dapat menghasilkan grafik dinamis saat sedang berjalan. Oleh karena itu, pengembang dapat menghentikan interpreter dalam sesi debugger dan memeriksa keluaran pada simpul tertentu.

Sementara itu, tensor yang disediakan PyTorch untuk mendukung CPU dan GPU juga dapat mempercepat komputasi secara substansial.

Pengantar MindSpore

Huawei mengembangkan arsitektur esensial kerangka kerja MindSpore berdasarkan kemudahan penggunaan, efisiensi, dan fleksibilitas, yang terdiri dari empat lapisan: lapisan teratas adalah ekspresi grafik komputasi asli MindSpore; lapisan kedua adalah lapisan eksekusi Pipeline paralel, yang terutama dirancang untuk mengoptimalkan komputasi citra kedalaman dan fusi operator; lapisan ketiga mencakup arsitektur distribusi kolaboratif sesuai permintaan, pustaka komunikasi, dan kerangka kerja dasar penjadwalan dan penerapan tugas terdistribusi; dan lapisan terbawah adalah lapisan efisiensi eksekusi. Arsitektur inti MindSpore memungkinkan diferensiasi otomatis, paralelisasi otomatis, dan penyetelan otomatis, menawarkan dukungan yang solid untuk API semua skenario yang sejalan dengan tujuan desain Huawei yaitu kemudahan pengembangan, pengoperasian yang efisien, dan penerapan yang fleksibel.

Inti dan peran penting dalam paradigma pemrograman kerangka kerja AI adalah teknologi diferensiasi otomatis dari kerangka kerja tersebut. Model pembelajaran mendalam dilatih melalui komputasi maju dan mundur. Mari kita lihat ekspresi matematika yang ditunjukkan pada Gambar 4.1. Proses komputasi maju persamaan ini ditunjukkan oleh panah gelap pada Gambar 4.1 yang mencari keluaran f melalui komputasi maju, dan mendapatkan nilai diferensial x dan y melalui komputasi mundur berdasarkan aturan rantai. Ketika para insinyur algoritma merancang model, mereka hanya mencakup komputasi maju, dan komputasi mundur diimplementasikan oleh teknologi diferensiasi otomatis yang dimiliki kerangka kerja tersebut.



Gambar 4.1 Ekspresi matematika

Terlebih lagi, dengan perluasan model NPL, overhead memori untuk melatih model besar seperti Bert (340M) dan GPT-2 (1542M) telah melampaui kapasitas kartu tunggal, sehingga model tersebut perlu didistribusikan ke beberapa GPU untuk dieksekusi. Saat ini, metode yang umum diadopsi adalah paralelisme model yang direkayasa secara manual, yang memerlukan perancangan partisi model dan memperhatikan topologi klaster. Mengembangkan model semacam itu sangat sulit, belum lagi memastikan kinerja dan penyetelan yang tinggi.

MindSpore mengadopsi partisi grafik otomatis, yang mempartisi grafik berdasarkan dimensi input/output operator, dan menggabungkan paralelisme data dengan paralelisme model. Sistem ini menggunakan penjadwalan berbasis topologi klaster dan meminimalkan overhead komunikasi melalui kesadaran akan topologi klaster dan penjadwalan eksekusi subgraf secara otomatis. Sistem ini dapat mempertahankan logika kode mandiri dan menerapkan paralelisme model, sehingga meningkatkan efisiensi pengembangan paralelisme model hingga sepuluh kali lipat dibandingkan dengan paralelisme yang dirancang secara manual.

Saat ini, eksekusi model dalam konteks chip daya komputasi super menghadapi tantangan besar, seperti masalah dinding memori, overhead interaksi yang meningkat, dan pasokan data yang bermasalah. Karena sebagian operasi dieksekusi di host dan sebagian lagi dieksekusi di perangkat terminal, overhead interaksi host-perangkat bahkan telah melampaui overhead eksekusi, sehingga mengurangi okupansi akselerator. MindSpore menawarkan optimasi grafik kedalaman berorientasi chip yang menampilkan sinkronisasi yang mengurangi waktu tunggu dan memaksimalkan paralelisme "data-komputasi-komunikasi", yang membuat data dan grafik komputasi tersinkronisasi ke prosesor Ascend AI.

MindSpore juga menggunakan metode eksekusi di perangkat untuk desentralisasi. Dengan menerapkan optimasi partisi grafik adaptif mandiri berbasis data gradien, MindSpore mewujudkan desentralisasi AllReduce yang independen, mensintesis kecepatan agregasi gradien, dan menyediakan pipeline komputasi dan komunikasi yang memadai. MindSpore juga mengadopsi arsitektur terdistribusi kolaboratif sesuai permintaan yang mensinergikan edge perangkat dan cloud. Representasi perantara (IR) dari model kolektif menjamin pengalaman penerapan yang konsisten, dan memblokir perbedaan skenario melalui teknologi optimasi grafik yang mengolaborasikan perangkat lunak dan perangkat keras. Strategi meta-pembelajaran federal kolaboratif perangkat-cloud mendobrak batasan antara perangkat dan cloud, serta mewujudkan pembaruan real-time dari model kolaboratif multi-perangkat.

Pengantar TensorFlow

TensorFlow adalah kerangka kerja pembelajaran mendalam yang dikembangkan oleh Google. Ini adalah generasi kedua dari pustaka perangkat lunak sumber terbuka yang dirancang untuk komputasi digital oleh Google. Kerangka kerja ini dapat mendukung berbagai algoritma dan platform pembelajaran mendalam dengan stabilitas sistem yang relatif tinggi. TensorFlow memiliki karakteristik berikut.

1. Mendukung Berbagai Platform

Semua platform lingkungan Python dapat mendukung TensorFlow. Namun, TensorFlow harus mengakses GPU yang didukung melalui perangkat lunak lain seperti NVIDIA CUDA Toolkit dan cuDNN.

2. Mendukung GPU

TensorFlow mendukung GPU NVIDIA tertentu yang kompatibel dengan versi terkait dari CUDA Toolkit yang memenuhi kriteria kinerja tertentu.

3. Mendukung Komputasi Terdistribusi

TensorFlow mendukung komputasi terdistribusi. Hal ini memungkinkan sebagian grafik untuk dihitung pada proses yang berbeda, yang mungkin berada di server yang sama sekali berbeda.

4. Mendukung Berbagai Bahasa

Bahasa pemrograman utama TensorFlow adalah Python. Pengembang juga dapat menggunakan C++, Java, dan Go, tetapi bahasa-bahasa ini tidak menjanjikan stabilitas, begitu pula banyak binding pihak ketiga untuk C#, Haskell, Julia, Rust, Ruby, Scala, R, dan bahkan PHP. Google baru-baru ini merilis pustaka TensorFlow-Lite yang dioptimalkan untuk perangkat seluler agar dapat menjalankan aplikasi TensorFlow di sistem Android.

5. Fleksibel dan Dapat Diperluas

Salah satu keuntungan utama menggunakan TensorFlow adalah desainnya yang modular, dapat diperluas, dan fleksibel. Pengembang dapat dengan mudah memindahkan model di berbagai prosesor CPU, GPU, atau TPU dengan melakukan beberapa modifikasi kode. Pengembang Python dapat menggunakan API tingkat rendah (atau API inti) TensorFlow untuk mengembangkan model mereka sendiri, dan menggunakan API tingkat tinggi untuk model bawaan. TensorFlow memiliki banyak pustaka bawaan dan pustaka terdistribusi, dan dapat melapisi kerangka kerja pembelajaran mendalam tingkat tinggi seperti Keras untuk berfungsi sebagai API tingkat tinggi.

6. Performa Komputasi yang Kuat

Meskipun TensorFlow berkinerja terbaik pada Tensor Processing Unit (TPU) Google, TensorFlow juga berhasil mencapai performa yang lebih tinggi pada platform lain, tidak hanya server dan sistem desktop, tetapi juga sistem tertanam dan perangkat seluler. Penerapan terdistribusi TensorFlow memungkinkannya berjalan di berbagai sistem komputer. Model pelatihan dapat dihasilkan secara real-time pada sistem sekecil ponsel pintar atau sebesar kluster komputer. Lingkungan Windows dibangun dengan mode GPU tunggal, dan sebagian besar kerangka kerja pembelajaran mendalam mengandalkan cuDNN. Jadi, selama tidak ada perbedaan yang signifikan dalam daya komputasi perangkat keras atau alokasi memori, kecepatan pelatihan kerangka kerja ini tidak akan terlalu berbeda satu sama lain. Namun, untuk pembelajaran mendalam skala besar, jumlah data yang sangat besar akan menyulitkan satu mesin untuk menyelesaikan pelatihan tepat waktu. Namun, TensorFlow mendukung pelatihan terdistribusi.

TensorFlow diyakini sebagai salah satu pustaka yang paling ramah pengguna untuk pembelajaran mendalam. Dengan bantuan TensorFlow, pengembangan pembelajaran

mendalam akan menjadi tugas yang jauh lebih mudah. Fitur sumber terbukanya memungkinkan siapa pun untuk memelihara dan memperbarui TensorFlow guna meningkatkan efisiensinya. Keras, yang menerima Bintang ketiga terbanyak (yaitu, ditandai) di GitHub, dienkapsulasi sebagai API tingkat tinggi untuk TensorFlow 2.0. Berkat Keras, TensorFlow 2.0 menjadi lebih fleksibel dan lebih mudah di-debug.

Di TensorFlow 1.0, setelah tensor dibuat, Anda tidak dapat langsung kembali ke hasilnya, tetapi untuk membuat sesi yang menyertakan konsep grafik, Anda perlu menjalankan session.run untuk operasi. Gaya ini lebih mirip dengan bahasa pemrograman perangkat keras VHDL. Dibandingkan dengan framework yang lebih sederhana seperti PyTorch, langkah-langkah yang tidak perlu di atas yang diwajibkan oleh TensorFlow 1.0 tidak ada gunanya selain menciptakan lebih banyak rintangan bagi developer dalam penggunaannya. TensorFlow 1.0 sering dikritik karena pengalaman debugging-nya yang rumit, API yang membingungkan, dan tidak mudah untuk memulai. Dan TensorFlow 1.0 masih sulit digunakan bahkan oleh developer yang berpengalaman, sehingga banyak developer beralih ke PyTorch.

4.2 DASAR-DASAR TENSORFLOW 2.0

Pengantar TensorFlow 2.0

Fungsi inti TensorFlow 2.0 adalah eager execution, sebuah mekanisme grafik dinamis yang memungkinkan pengguna menulis dan men-debug model layaknya pemrograman normal, sehingga memudahkan pembelajaran dan penerapan TensorFlow. TensorFlow 2.0 mendukung lebih banyak platform dan bahasa pemrograman, serta meningkatkan kompatibilitas di seluruh komponen dengan menstandardisasi format pertukaran dan menyelaraskan API. TensorFlow versi 2.0 membersihkan API yang usang dan mengurangi API yang terduplikasi agar tidak membingungkan pengguna. TensorFlow 2.0 menyediakan modul yang kompatibel dengan TensorFlow 1.x, dan modul tf.contrib tidak akan digunakan lagi, modul yang dipertahankan akan dipindahkan ke tempat lain, dan sisanya akan dihapus.

Pengantar Tensor

Struktur data paling mendasar dalam TensorFlow adalah tensor, yang merangkum semua data. Menurut definisinya, tensor adalah larik multidimensi. Di antaranya, tensor peringkat 0 adalah skalar, tensor peringkat 1 adalah vektor, dan tensor peringkat 2 adalah matriks. Dalam TensorFlow, tensor dapat dibagi menjadi konstanta dan variabel.

Eksekusi Eager TensorFlow 2.0

TensorFlow 1.0 mengadopsi mekanisme grafik statis yang memisahkan definisi komputasi dari eksekusinya melalui grafik (juga dikenal sebagai grafik komputasional). Ini adalah semacam model pemrograman deklaratif. Dengan mekanisme grafik statis, Anda perlu terlebih dahulu membuat grafik, lalu menjalankan sesi, dan memasukkan data untuk mendapatkan hasil eksekusi. Mekanisme grafik statis memiliki banyak keunggulan dalam pelatihan terdistribusi, optimasi kinerja, dan penerapan, tetapi tidak mudah digunakan dalam debugging, seperti halnya memanggil dari program bahasa C yang dikompilasi di mana kita tidak dapat melakukan debugging internal. Oleh karena itu, eksekusi eager berbasis grafik

dinamis (AutoGraph) diperkenalkan. Ease execution adalah lingkungan pemrograman imperatif yang konsisten dengan Python asli. Hasil eksekusi akan segera dikembalikan setelah operasi dilakukan.

TensorFlow 2.0 AutoGraph

Di TensorFlow 2.0, eager execution diaktifkan secara default. Bagi pengguna, eager execution mudah dan fleksibel, menawarkan operasi yang lebih mudah dan cepat. Namun, hal ini dapat dicapai dengan mengorbankan kinerja dan penerapan. Untuk mendapatkan kinerja terbaik dan memastikan bahwa model dapat diterapkan di mana saja, kita dapat menerapkan dekorator @tf.function untuk membangun grafik dalam program, yang meningkatkan efisiensi kode Python. Fitur tf.function yang sangat menarik adalah AutoGraph, yang dapat mengonversi fungsi operasi TensorFlow menjadi grafik, sehingga mengeksekusi fungsi tersebut dalam mode Grafik. Dengan cara ini, fungsi tersebut dienkapsulasi menjadi grafik operasi TensorFlow.

4.3 PENGANTAR MODUL TENSORFLOW 2.0

Pengantar Modul Umum

Fungsi-fungsi dalam tf.modules TensorFlow 2.0 dirancang untuk menangani operasi-operasi umum. Misalnya, sebagian besar operasi dalam tf.abs (menghitung nilai absolut), tf.add (penjumlahan elemen per elemen), tf.concat (penggabungan tensor), dll. dapat dilakukan oleh NumPy.tf.modules juga mencakup:

1. tf.errors: jenis pengecualian untuk kesalahan TensorFlow.
2. tf.data: melakukan operasi pada set data. Misalnya, gunakan alur input yang dibuat oleh tf.data untuk membaca data pelatihan. Modul ini juga mendukung input data yang mudah dari memori (seperti NumPy).
3. tf.gfile: melakukan operasi pada berkas. Fungsi-fungsi dalam modul ini dapat melakukan operasi I/O berkas, serta menyalin dan mengganti nama berkas.
4. tf.image: melakukan operasi pada gambar. Fungsi-fungsi dalam modul ini dapat memproses gambar seperti OpenCV, dengan serangkaian fungsi seperti kecerahan gambar, saturasi, inversi, pemotongan, pengubahan ukuran, konversi format gambar (dari format RGB ke format HSV, YUV, YIQ, Gray), rotasi, dan deteksi tepi Sobel. Modul ini setara dengan perangkat pemrosesan gambar OpenCV skala kecil.
5. tf.keras: memanggil API Python dari perangkat Keras. Modul ini relatif besar, yang berisi semua jenis operasi jaringan.
6. tf.nn: modul pendukung fungsional jaringan saraf tiruan. Modul ini paling umum digunakan untuk membangun jaringan konvolusional klasik. Modul ini juga berisi sub-modul rnn_cell, yang diterapkan dalam pembangunan jaringan saraf tiruan rekuren. Fungsi yang sering digunakan dalam modul ini meliputi: pengumpulan rata-rata avg_pool(), normalisasi batch batch_normalization(), penambahan bias bias_add(), konvolusi dua dimensi conv2d(), sel jaringan neural dropout acak dropout(), lapisan aktivasi ReLu relu(), entropi silang sigmoid setelah aktivasi sigmoid_cross_entropy_with_logits(), lapisan aktivasi softmax softmax().

Antarmuka Keras

Keras adalah program yang direkomendasikan dalam TensorFlow 2.0 untuk membangun jaringan. Modul keras.layer telah mencakup semua jaringan neural populer. Keras adalah API tingkat tinggi yang dirancang untuk membangun dan melatih model pembelajaran mendalam. API ini dapat digunakan untuk pembuatan prototipe cepat, riset tingkat lanjut, dan produksi. Keras memiliki tiga keunggulan utama berikut.

1. Ramah Pengguna

Keras memiliki antarmuka yang sederhana dan konsisten yang dioptimalkan untuk kasus penggunaan umum yang memberikan umpan balik yang jelas dan dapat ditindaklanjuti untuk kesalahan pengguna.

2. Modular dan Komposabel

Anda dapat membangun model Keras dengan menghubungkan blok penyusun yang dapat dikonfigurasi, hampir tanpa batasan.

3. Mudah Diperluas

Anda dapat menulis blok penyusun khusus untuk mengekspresikan ide penelitian baru, membuat lapisan baru, fungsi kerugian, dan mengembangkan model lanjutan.

Berikut adalah modul yang umum digunakan di Keras.

1. **tf.keras.layers**

   Namespace tf.keras.layers menyediakan antarmuka yang luas untuk lapisan jaringan umum, seperti lapisan terhubung penuh, lapisan fungsi aktivasi, lapisan penggabungan, lapisan konvolusional, lapisan jaringan saraf tiruan berulang, dll. Untuk kelas-kelas lapisan jaringan ini, komputasi maju dapat dilakukan dengan menentukan parameter yang relevan dari lapisan jaringan saat membuatnya, dan memanggil metode _call_. Keras akan secara otomatis memanggil logika propagasi maju setiap lapisan sambil memanggil metode _call_, yang diimplementasikan oleh fungsi panggilan kelas.

2. **Kontainer Jaringan**

   Dalam jaringan yang sering digunakan, pengguna perlu memanggil instans kelas setiap lapisan secara manual untuk melakukan propagasi maju. Ketika jumlah lapisan jaringan bertambah, kode bagian ini akan menjadi sangat redundan. Kontainer jaringan Sequential yang disediakan oleh Keras dapat merangkum beberapa lapisan jaringan menjadi model jaringan yang besar, di mana pengguna hanya perlu memanggil instance model jaringan sekali untuk menyelesaikan operasi sekuensial data dari lapisan pertama hingga terakhir.

4.4 MEMULAI DENGAN TENSORFLOW 2.0

Pengaturan Lingkungan

Untuk menyiapkan lingkungan pengembangan TensorFlow 2.0, Anda perlu melakukan hal berikut.

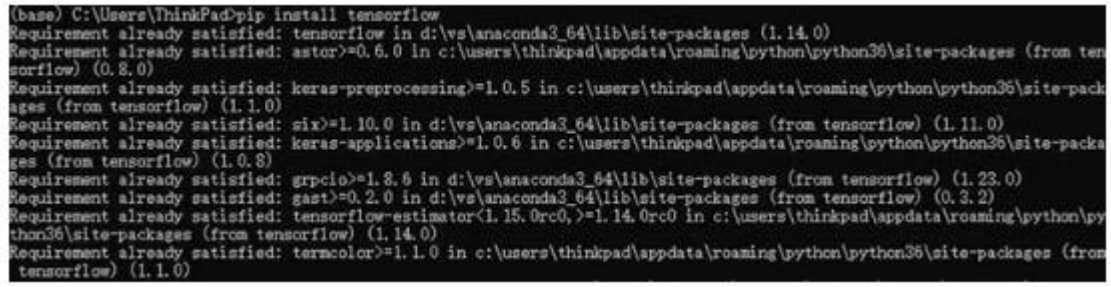
1. Pengaturan di Lingkungan Windows

(a) Sistem operasi: Windows 10.

(b) Lingkungan pengembangan Python: Anaconda3 (versi yang diadaptasi untuk Python 3) yang dilengkapi dengan perangkat lunak pip.

Instal TensorFlow: Buka Anaconda Prompt, dan instal TensorFlow dengan langsung menjalankan perintah pip. Seperti yang ditunjukkan pada Gambar 4.2, ketik perintah berikut di Anaconda Prompt.

```
pip install tensorflow
```



Gambar 4.2 Perintah Instalasi

2. Pengaturan Lingkungan Linux

Cara termudah untuk menginstal TensorFlow di lingkungan Linux adalah menggunakan pip. Jika instalasi berjalan lambat, Anda dapat menggunakan Tsinghua Open Source Mirror untuk menjalankan perintah berikut di terminal.

```
 pip install pip-U
 pip config set global.index-url
 https://pypi.tuna.tsinghua.edu.cn/
simple
```

Terakhir, jalankan perintah instalasi pip.

```
pip install tensorflow==2.0.0
```

Proses Pengembangan

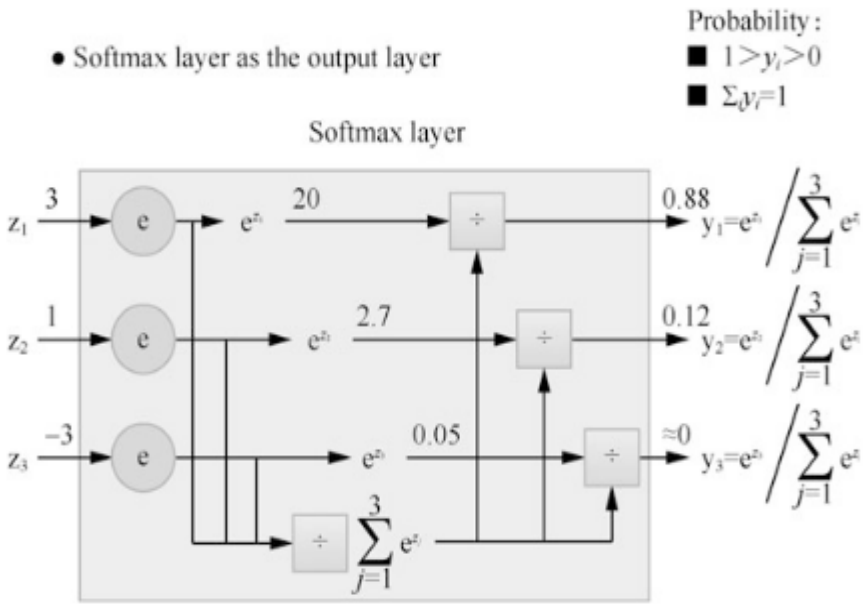
Proses pengembangan TensorFlow 2.0 mencakup lima langkah.

1. Persiapan data, termasuk eksplorasi dan pemrosesan data.
2. Membangun jaringan, termasuk mendefinisikan struktur jaringan, mendefinisikan fungsi kerugian, memilih pengoptimal, dan mendefinisikan standar evaluasi model.
3. Melatih dan memvalidasi model.
4. Menyimpan model.
5. Memulihkan dan memanggil model.

Proses yang disebutkan di atas akan diuraikan lebih lanjut dalam bagian berikut oleh sebuah proyek nyata pengenalan digit tulisan tangan MNIST.

Pengenalan digit tulisan tangan merupakan tugas umum pengenalan gambar di mana komputer diharuskan mengenali digit dari gambar digit tulisan tangan. Tidak seperti cetakan, tulisan tangan setiap orang memiliki gaya yang berbeda, dan ukuran digit tulisan tangan bervariasi dari satu orang ke orang lain, yang membuat pengenalan digit tulisan tangan oleh komputer jauh lebih sulit. Dalam proyek ini, pembelajaran mendalam dan kerangka kerja TensorFlow digunakan untuk melakukan pengenalan digit tulisan tangan pada dataset MNIST.

Gambar 4.3 Contoh dari dataset



Gambar 4.4 Perhitungan fungsi Softmax

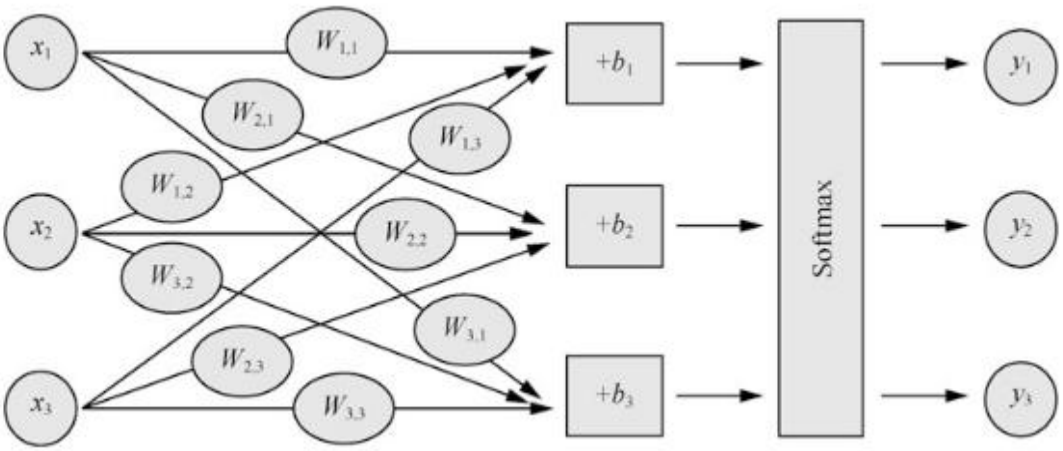
1. Persiapan Data

- ✓ Unduh dataset MNIST.
- ✓ Dataset MNIST terdiri dari set pelatihan dan set pengujian.
- ✓ Set pelatihan berisi 60.000 gambar digit tulisan tangan dan label yang sesuai.
- ✓ Set pengujian berisi 10.000 gambar digit tulisan tangan dan label yang sesuai.

Gambar 4.3 menunjukkan contoh dari dataset.

2. Membangun Jaringan

Fungsi aktivasi yang digunakan dalam proyek ini adalah model regresi Softmax. Fungsi Softmax juga dikenal sebagai fungsi eksponensial ternormalisasi, yang merupakan turunan dari fungsi biner Sigmoid dalam klasifikasi multikelas. Gambar 4.4 menunjukkan bagaimana fungsi Softmax dihitung.



Gambar 4.5 Proses perhitungan model

Proses pembangunan model merupakan kunci dari konstruksi jaringan. Gambar 4.5 menunjukkan proses perhitungan model, yang mendefinisikan bagaimana model dibangun, dan bagaimana keluaran dihasilkan berdasarkan masukan tersebut.

Kode inti TensorFlow untuk mengimplementasikan model regresi Softmax disajikan pada Gambar 4.6. Membuat model terutama perlu menentukan dua hal berikut terlebih dahulu.

- Fungsi kerugian: Dalam pembelajaran mesin maupun pembelajaran mendalam, kita sering kali perlu mendefinisikan fungsi kerugian sebagai indikator untuk menyatakan kesesuaian suatu model, dan kemudian meminimalkan fungsi kerugian tersebut. Indikator ini disebut biaya atau kerugian. Fungsi kerugian yang digunakan dalam proyek ini adalah fungsi kerugian entropi silang.
- Pengoptimal: Setelah fungsi kerugian didefinisikan, kita perlu mengoptimalkan fungsi kerugian dengan pengoptimal, untuk menemukan parameter optimal dan meminimalkan nilai fungsi kerugian. Pengoptimal yang paling sering digunakan untuk menemukan parameter optimal pembelajaran mesin adalah pengoptimal berbasis penurunan gradien.

```
## Import TensorFlow
import tensorflow as tf
## Define the input variables with operator symbol variables
'''
Where x is not a particular value, but a placeholder, which will be entered when running TensorFlow. For the
computation, each input image will be flattened into a vector of 784 dimensions, featuring a shape of [None,
784]. None refers to the first dimension of the vector. It can be of any length.
'''
x = tf.placeholder(tf.float32, [None, 784])
'''
Weight (w) and bias (b) are expressed by the Variable that is changeable. The initial values of the two are both zero.
'''
w = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
'''
Use tf.matmul (x,w) to express x multiplying w. The equation of Softmax regression is y=softmax (wx+b)
'''
y = tf.nn.softmax(tf.matmul(x,w) + b)
```

Gambar 4.6 Kode implementasi Softmax

3. Pelatihan dan Validasi Model

Melatih semua data secara batch atau iterasi massal. Dalam proyek ini, kami melatih data secara langsung dengan model.fit, dan melatih data dalam iterasi massal sebanyak lima kali, seperti yang ditunjukkan pada Gambar 4.7. Epoch mewakili jumlah iterasi pelatihan. Seperti yang ditunjukkan pada Gambar 4.8, kami menguji dan memvalidasi model dengan set uji dan membandingkan hasil prediksi dengan hasil aktual, untuk menemukan label yang tepat, dan memperkirakan akurasi model pada set uji.

```
# Fitting the model on training data through model.fit
model.fit(mnist.train.images, mnist.train.labels, epochs=5)

Epoch 1/5
55000/55000 [==============================] - 4s 74us/sample - loss: 0.3043 - categorical_accuracy: 0.9110
Epoch 2/5
55000/55000 [==============================] - 4s 73us/sample - loss: 0.1460 - categorical_accuracy: 0.9569
Epoch 3/5
55000/55000 [==============================] - 4s 79us/sample - loss: 0.1104 - categorical_accuracy: 0.9669
Epoch 4/5
55000/55000 [==============================] - 4s 74us/sample - loss: 0.0881 - categorical_accuracy: 0.9722
Epoch 5/5
55000/55000 [==============================] - 4s 73us/sample - loss: 0.0767 - categorical_accuracy: 0.9760
```

Gambar 4.7 Proses pelatihan

Gambar 4.8 Validasi pengujian

4.5 KESIMPULAN

Bab ini memperkenalkan kerangka kerja pengembangan yang umum digunakan dalam industri AI beserta karakteristiknya, dengan penekanan khusus pada komposisi modul dan prosedur dasar pengembangan kerangka kerja TensorFlow. Lebih lanjut, bab ini menawarkan proyek untuk memperkenalkan penerapan fungsi dan modul TensorFlow dalam kasus praktis. Pembaca dapat menjadikan bab ini sebagai panduan saat menyiapkan lingkungan kerangka kerja dan mengoperasikan proyek contoh. Dengan langkah-langkah ini, kami berharap pembaca dapat memiliki pemahaman yang lebih mendalam tentang AI.

Latihan Soal

1. Seiring dengan semakin meluasnya implementasi AI, apa saja framework pengembangan AI yang paling populer saat ini? Apa saja karakteristiknya?
2. TensorFlow adalah framework pengembangan representatif untuk AI yang telah menarik banyak pengguna. Perubahan terpenting selama pemeliharaannya adalah transformasi dari TensorFlow 1.0 ke TensorFlow 2.0. Jelaskan perbedaan antara kedua versi tersebut.
3. TensorFlow memiliki beragam modul yang dirancang sesuai kebutuhan pengguna. Jelaskan tiga modul umum Tensor Flow.
4. Dibandingkan dengan framework lain, Keras cukup istimewa sebagai framework front-end. Jelaskan secara singkat fitur-fitur antarmuka Keras.
5. Cobalah mengonfigurasi framework pengembangan AI sesuai dengan panduan yang diberikan dalam bab ini.

BAB 5
KERANGKA KERJA PENGEMBANGAN AI HUAWEI MINDSPORE

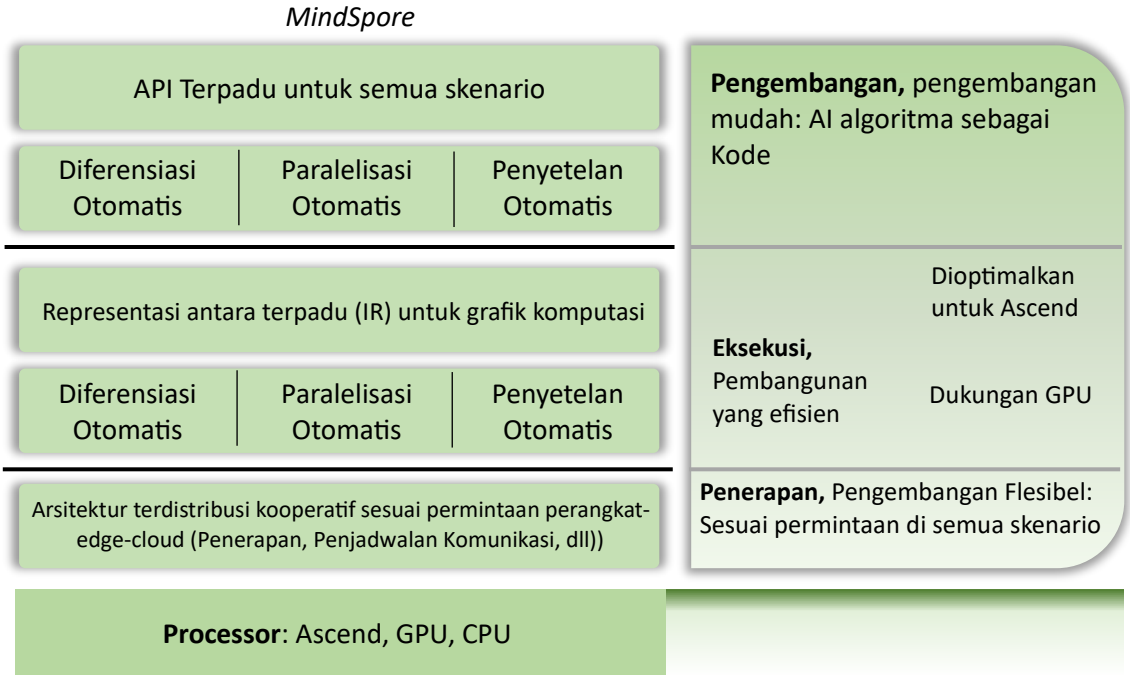
Bab ini berfokus pada kerangka kerja pengembangan AI yang dikembangkan oleh Huawei MindSpore. Pertama, bab ini memperkenalkan arsitektur MindSpore dan bagaimana ia dirancang. Selanjutnya, bab ini menganalisis permasalahan dan kesulitan kerangka kerja pengembangan AI. Terakhir, bab ini menyajikan lebih lanjut kerangka kerja berdasarkan pengembangan dan penerapan MindSpore.

5.1 PENDAHULUAN KERANGKA KERJA PENGEMBANGAN MINDSPORE

MindSpore adalah kerangka kerja pengembangan AI yang dikembangkan sendiri oleh Huawei yang dapat mencapai kolaborasi sesuai permintaan semua skenario perangkat-edge-cloud. MindSpore menyediakan API semua skenario terpadu yang memungkinkan pengembangan, eksekusi, dan penerapan model secara menyeluruh. MindSpore mengadopsi arsitektur terdistribusi kolaboratif sesuai permintaan perangkat-edge-cloud, paradigma baru pemrograman diferensial asli, dan mode eksekusi AI Native baru untuk mencapai efisiensi sumber daya, keamanan, dan keandalan yang lebih tinggi, sekaligus menurunkan standar masuk untuk pengembangan AI dan memberikan daya komputasi penuh pada prosesor AI Ascend untuk mewujudkan aplikasi AI yang inklusif.

Arsitektur MindSpore

Arsitektur MindSpore terdiri dari tiga bagian: pengembangan, eksekusi, dan penerapan. Seperti yang ditunjukkan pada Gambar 5.1, prosesor yang dapat diterapkan meliputi CPU, GPU, dan prosesor Ascend AI (Ascend310, Ascend910).



Gambar 5.1 Arsitektur MindSpore

Pengembangan ini merupakan API terpadu untuk semua skenario (API Python) yang menyediakan antarmuka pelatihan, inferensi, dan ekspor model terpadu bagi pengguna, serta antarmuka pemrosesan, peningkatan, dan konversi format data terpadu. Pengembangan ini mencakup fungsi-fungsi seperti optimasi tingkat tinggi grafik (GHLO), optimasi independen perangkat keras (misalnya, penghapusan kode mati, dll.), paralelisme otomatis, dan diferensiasi otomatis. Fungsi-fungsi ini juga mendukung konsep desain API terpadu untuk semua skenario.

Integrasi (IR) MindSpore dalam eksekusi memiliki representasi grafik komputasional asli, dan representasi antara (IR) terpadu, yang menjadi dasar MindSpore melakukan optimasi pada proses kompiler. Eksekusi ini mencakup optimasi terkait perangkat keras, lapisan eksekusi Pipeline paralel, dan optimasi mendalam terkait kombinasi perangkat lunak dan perangkat keras, seperti fusi operator dan fusi buffer. Fitur-fitur ini memungkinkan diferensiasi otomatis, paralel otomatis, dan penyetelan otomatis.

Penerapan ini mengadopsi arsitektur terdistribusi dari kolaborasi perangkat-tepi-cloud sesuai permintaan, dengan penerapan, penjadwalan, dan komunikasi yang bekerja pada lapisan yang sama, berkontribusi pada kolaborasi sesuai permintaan untuk semua skenario. Singkatnya, MindSpore mengintegrasikan kemudahan pengembangan (algoritma AI sebagai kode), pengoperasian yang efisien (mendukung optimalisasi prosesor dan GPU Ascend AI), dan penerapan yang fleksibel (kolaborasi sesuai permintaan untuk semua skenario) dalam satu kerangka kerja.

Bagaimana MindSpore Dirancang

Menanggapi tantangan yang dihadapi oleh pengembang AI di industri ini, seperti standar awal pengembangan yang tinggi, biaya operasional yang tinggi, dan kesulitan penerapan, MindSpore mengusulkan tiga inovasi teknologi: paradigma pemrograman baru, mode eksekusi baru, dan mode kolaborasi baru untuk membantu para pengembang mewujudkan pengembangan dan penerapan aplikasi AI dengan cara yang lebih mudah dan efisien.

1. Paradigma Pemrograman Baru

Konsep desain paradigma pemrograman baru diusulkan untuk mengatasi tantangan pengembangan AI.

Tantangan pengembangan ini terutama meliputi:

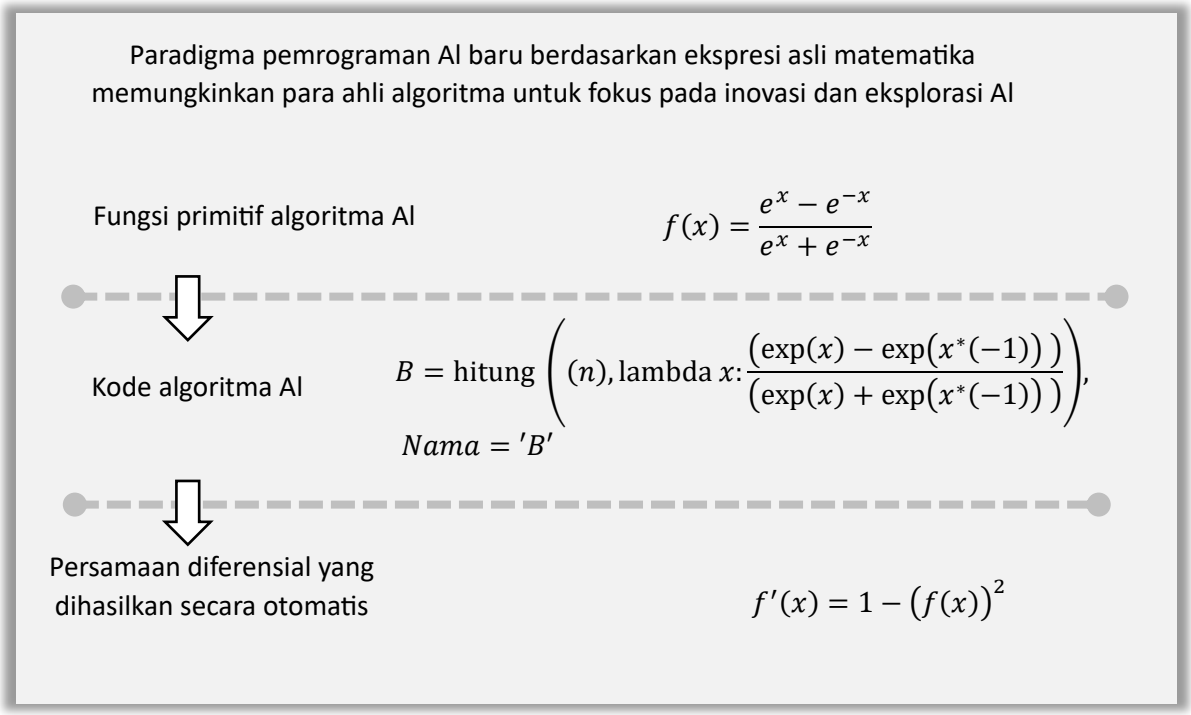
(a) Persyaratan keterampilan yang tinggi. Pengembang diharuskan memiliki pengetahuan teoretis dan keterampilan matematika yang relevan tentang AI, sistem komputer, dan perangkat lunak untuk terlibat dalam pengembangan AI, sehingga meningkatkan standar masuk.

(b) Optimasi kotak hitam yang sulit. Ketidakmampuan interpretasi dan fitur "kotak hitam" dari algoritma AI mempersulit penyetelan parameter.

(c) Perencanaan paralel yang sulit. Dipengaruhi oleh kemajuan teknologi, volume data semakin besar, dan model semakin besar, sehingga komputasi paralel menjadi tak terelakkan. Perencanaan paralel sangat bergantung pada pengalaman pribadi teknisi,

yang mengharuskan teknisi tidak hanya menjadi profesional data dan model tetapi juga memahami arsitektur sistem terdistribusi.

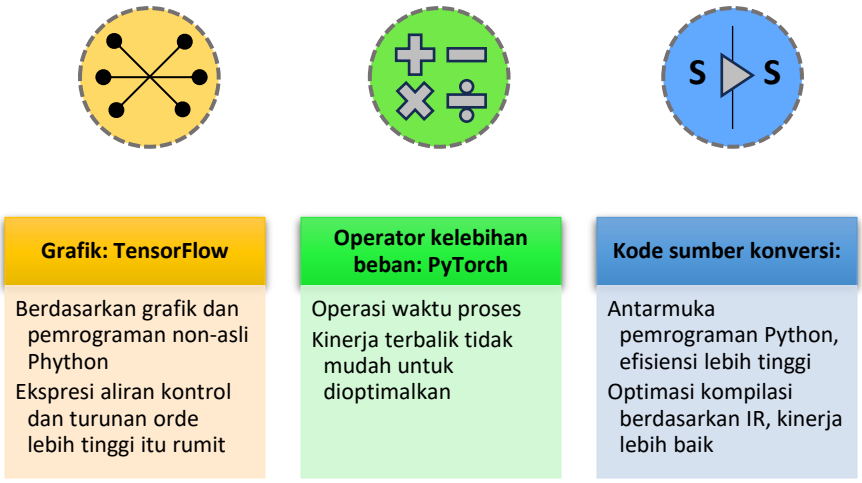
Algoritma AI dalam paradigma pemrograman baru adalah kode itu sendiri, yang menurunkan standar pengembangan AI. Paradigma pemrograman AI baru berdasarkan ekspresi asli matematika memungkinkan para ahli algoritma untuk berfokus pada inovasi dan eksplorasi AI, seperti yang ditunjukkan pada Gambar 5.2.



Gambar 5.2 Paradigma pemrograman baru MindSpore

(a) Diferensiasi Otomatis

Diferensiasi otomatis (AD) adalah inti dari kerangka kerja pembelajaran mendalam. Secara umum, AD mengacu pada metode komputasi turunan suatu fungsi secara otomatis. Dalam pembelajaran mesin, turunan ini dapat memperbarui bobot. Dalam ilmu pengetahuan alam dengan konotasi yang lebih luas, turunan ini juga dapat digunakan dalam komputasi selanjutnya. Perkembangan diferensiasi otomatis ditunjukkan pada Gambar 5.3.



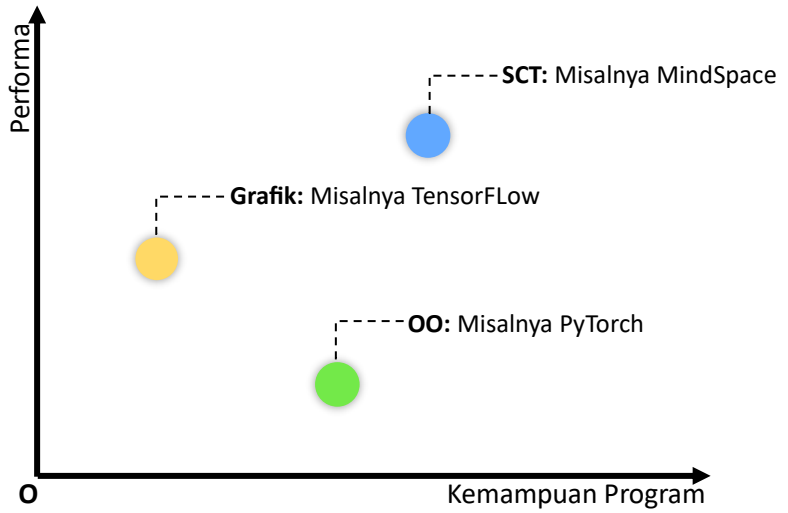
Gambar 5.3 Pengembangan diferensiasi otomatis

Dalam pengembangan diferensiasi otomatis, terdapat tiga jenis teknik diferensiasi otomatis sebagai berikut.

- Konversi berdasarkan grafik kalkulasi statis: Jaringan dikonversi menjadi diagram kalkulasi statis pada waktu kompilasi, kemudian aturan rantai diterapkan pada diagram kalkulasi untuk mewujudkan diferensiasi otomatis. Misalnya, Tensorflow dapat mengoptimalkan kinerja jaringan dengan teknologi kompilasi statis. Namun, membangun atau men-debug jaringan sangatlah rumit.
- Konversi berdasarkan grafik kalkulasi dinamis: Jalur operasi jaringan selama propagasi maju direkam oleh operator yang kelebihan beban, kemudian aturan rantai diterapkan pada grafik kalkulasi yang dihasilkan secara dinamis untuk mewujudkan diferensiasi otomatis. Misalnya, PyTorch sangat mudah digunakan, tetapi sulit untuk mengoptimalkan kinerjanya hingga tingkat tertinggi.
- Konversi berdasarkan kode sumber: Berdasarkan kerangka kerja pemrograman fungsional, PyTorch mengadopsi kompilasi *just-in-time* (JIT) untuk melakukan transformasi diferensiasi otomatis pada representasi antara (representasi program selama kompilasi). PyTorch juga mendukung diferensiasi otomatis dari alur kontrol otomatis. Jadi, sangat mudah untuk membangun model, sama seperti PyTorch. Di saat yang sama, MindSpore dapat melakukan kompilasi statis dan optimasi jaringan neural, sehingga kinerjanya juga sangat baik. Perbandingan teknologi diferensiasi otomatis ditunjukkan pada Tabel 5.1, dan perbandingan kinerja dan pemrograman ditunjukkan pada Gambar 5.4.

Tabel 5.1 Perbandingan teknologi diferensiasi otomatis

Sekolah AD	Umum	Cepat	Portabel	Dapat Didiferensiasi	Framework Tipikal
Graph	Tidak	√	√	Sebagian	TensorFlow
OO	√	Sebagian	Sebagian	√	PyTorch
SCT	√	√	√	√	MindSpore



Gambar 5.4 Perbandingan kinerja diferensiasi otomatis dan kemampuan pemrograman

- Pemrograman: Diferensiabilitas dapat dicapai dengan bahasa universal Python berbasis primitif IR (setiap operasi primitif dalam MindSpore IR sesuai dengan fungsi dasar dalam aljabar dasar).
- Performa: Optimalisasi kompiler, penyetelan otomatis balik operator.
- Debugging: Antarmuka visual yang beragam dan mendukung eksekusi dinamis.

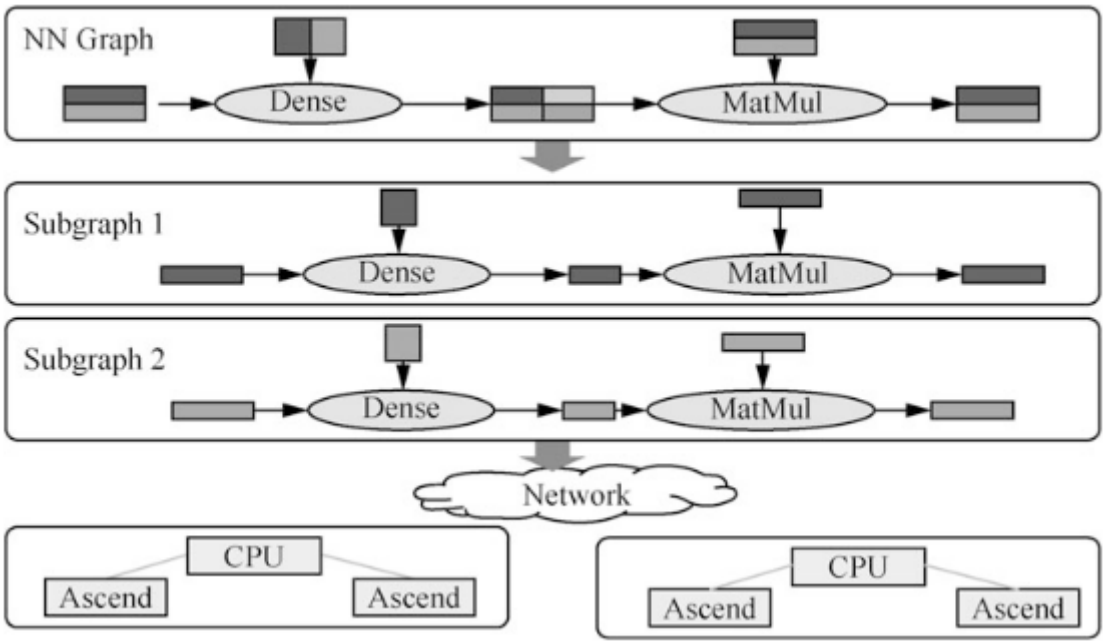
Teknologi diferensiasi otomatis MindSpore memiliki keunggulan sebagai berikut.

(b) Paralel Otomatis

Model pembelajaran mendalam saat ini seringkali perlu diparalelkan karena ukurannya yang besar. Saat ini, paralelisme model manual diadopsi, yang membutuhkan segmentasi model desain dan pemahaman topologi klaster, yang sulit diwujudkan. Selain itu, memastikan kinerja tinggi dengan paralelisme manual juga sulit, apalagi dengan penyetelannya.

MindSpore dapat secara otomatis memparalelkan kode yang ditulis secara serial, mewujudkan pelatihan paralel terdistribusi secara otomatis, dan mempertahankan kinerja tinggi. Secara umum, pelatihan paralel dapat dibagi menjadi paralel model dan paralel data. Paralel data lebih mudah dipahami, yang berarti setiap sampel dapat menyelesaikan propagasi maju secara independen, dan akhirnya merangkum hasil propagasi. Sebaliknya, paralel model lebih rumit, dan kita perlu menulis semua bagian yang perlu diparalelkan secara manual dengan logika "berpikir paralel".

MindSpore memperkenalkan teknologi inovasi utama segmentasi grafik penuh otomatis. Seperti ditunjukkan pada Gambar 5.5, graf lengkap disegmentasi berdasarkan data masukan dan keluaran operator. Artinya, setiap operator dalam graf dibagi menjadi beberapa klaster dan kemudian menyelesaikan operasi paralel. Teknologi ini menggabungkan paralel data dan paralel model. Topologi klaster diadopsi untuk menjadwalkan eksekusi subgraf secara otomatis guna meminimalkan overhead komunikasi.



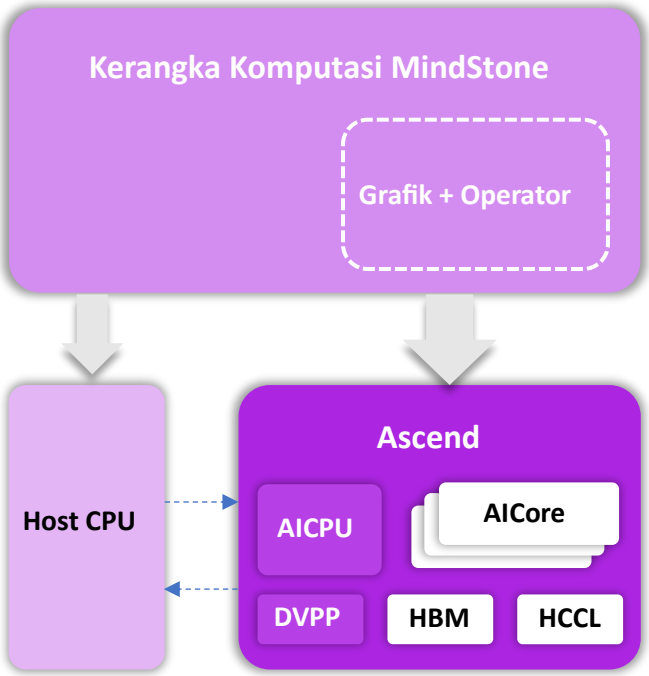
Gambar 5.5 Segmentasi Grafik Penuh Otomatis

Tujuan paralel otomatis MindSpore adalah membangun metode pelatihan yang menggabungkan paralelisme data, paralelisme model, dan paralelisme hibrida. Metode ini akan secara otomatis memilih metode segmentasi model yang memiliki biaya terendah untuk mengimplementasikan pelatihan paralel terdistribusi otomatis. Metode granularitas halus operator dari segmentasi MindSpore sangat rumit. Namun, pengembang tidak perlu memperhitungkan implementasi yang mendasarinya, selama mereka melengkapi API tingkat atas dengan efisiensi perhitungan yang tinggi. Secara umum, paradigma pemrograman baru ini tidak hanya mewujudkan "algoritma AI sebagai kode", yang menurunkan ambang batas pengembangan AI, tetapi juga memungkinkan pengembangan dan penelusuran kesalahan yang efisien. Misalnya, metode ini dapat menyelesaikan diferensiasi otomatis secara efisien, mewujudkan paralelisme otomatis satu baris kode, dan menyelesaikan penelusuran kesalahan serta menjalankan satu baris kode.

Transformer, algoritma klasik yang digunakan oleh pengembang untuk mengimplementasikan pemrosesan bahasa alami, diwujudkan oleh kerangka kerja MindSpore. Selama proses pengembangan dan penelusuran kesalahan, implementasi dinamis dan statis digabungkan, dan penelusuran kesalahan menjadi transparan dan sederhana. Berdasarkan struktur akhir, jumlah kode pada kerangka kerja MindSpore adalah 2000 baris, yang jika dibandingkan dengan 2500 baris pada Tensorflow, sekitar 20% lebih sedikit, tetapi dengan peningkatan efisiensi lebih dari 50%.

2. Mode Eksekusi Baru

Konsep desain mode eksekusi baru diusulkan sebagai respons terhadap tantangan eksekusi. Tantangan yang dihadapi dalam eksekusi adalah sebagai berikut:



Gambar 5.6 Eksekusi pada perangkat

(a) Kompleksitas komputasi AI dan keragaman daya komputasi: berbagai jenis daya komputasi, seperti inti CPU, unit kubus, unit vektor; komputasi besaran skalar, besaran vektor, dan besaran tensor; komputasi presisi campuran; komputasi matriks padat dan matriks jarang.

(b) Dalam kasus operasi multi-GPU, seiring bertambahnya jumlah node, kinerja sulit ditingkatkan secara linear, sehingga menghasilkan overhead kontrol paralel yang tinggi.

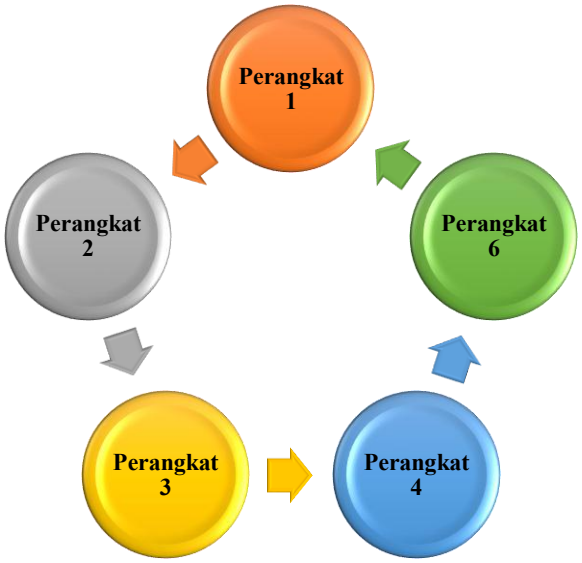
Mode eksekusi baru mengadopsi mesin eksekusi Ascend Native dan mengusulkan eksekusi pada perangkat, seperti yang ditunjukkan pada Gambar 5.6. Eksekusi pemindahan grafik penuh dan optimasi peta kedalaman dimanfaatkan, memberikan daya komputasi yang kuat dari prosesor Ascend AI secara maksimal.

Ada dua inti teknis implementasi pada perangkat, sebagai berikut.

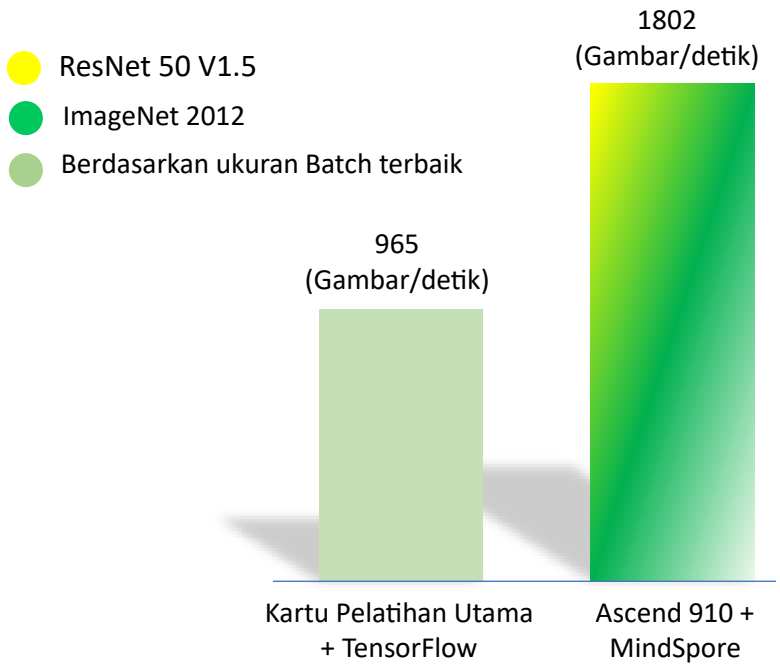
(a) Eksekusi pemindahan grafik penuh, memberikan daya komputasi yang kuat dari prosesor Ascend AI secara maksimal. Teknik ini mengatasi tantangan yang dihadapi oleh eksekusi model di bawah daya komputasi super chip, seperti dinding memori, overhead interaksi yang tinggi, kesulitan dalam penyediaan data, dll. Eksekusi sebagian dilakukan pada host dan sebagian lagi pada perangkat terminal, yang menyebabkan overhead interaksi yang jauh lebih tinggi daripada overhead eksekusi, sehingga menghasilkan tingkat hunian akselerator yang rendah. MindSpore, yang mengadopsi teknologi optimasi peta kedalaman berorientasi chip, meminimalkan waktu tunggu sinkronisasi dan memaksimalkan derajat paralel sinergi data-kalkulasi-komunikasi, dengan menenggelamkan seluruh data + kalkulasi grafik penuh ke prosesor Ascend AI, sehingga menawarkan hasil terbaik dan optimal. Hasil akhirnya adalah peningkatan 10

kali lipat dalam kinerja pelatihan dibandingkan dengan metode penjadwalan grafik di sisi host.

(b) Agregasi gradien terdistribusi skala besar berbasis data. Teknologi ini mengatasi tantangan agregasi gradien terdistribusi di bawah daya komputasi super chip, yaitu overhead sinkronisasi kontrol pusat dan overhead komunikasi dari sinkronisasi yang sering dalam satu iterasi ResNet-50 dalam 20 ms. Metode tradisional memerlukan tiga langkah sinkronisasi untuk menyelesaikan AllReduce dan AllReduce otonom berbasis data tanpa overhead kontrol.



Gambar 5.7 Algoritma AllReduce otonom terdesentralisasi



Gambar 5.8 Perbandingan antara MindSpore dan TensorFlow

MindSpore mencapai algoritma AllReduce otonom terdesentralisasi melalui optimasi segmentasi grafik adaptif yang digerakkan oleh data gradien, sehingga menjaga agregasi gradien pada langkah yang sama dan paralel antara komputasi dan komunikasi, seperti yang ditunjukkan pada Gambar 5.7. Gambar 5.8 menunjukkan contoh dalam visi komputer. Jaringan saraf tiruan ResNet-50 V1.5 diadopsi untuk melakukan pelatihan pada dataset ImageNet 2012 berdasarkan ukuran batch terbaiknya. Terlihat bahwa kecepatan kerangka kerja MindSpore pada Ascend 910 jauh lebih tinggi dibandingkan dengan kerangka kerja lain dan kartu pelatihan arus utama lainnya.

3. Model Kolaboratif Baru

Konsep desain model kolaborasi baru diusulkan sebagai respons terhadap tantangan penerapan. Tantangan penerapan adalah sebagai berikut.

(a) Skenario aplikasi perangkat, edge, dan cloud memiliki persyaratan, tujuan, dan kendala yang berbeda. Misalnya, perangkat ponsel mungkin lebih menyukai model yang lebih ringan, sementara cloud mungkin memerlukan presisi yang lebih tinggi.

(b) Perangkat keras yang berbeda memiliki akurasi dan kecepatan yang berbeda, seperti yang ditunjukkan pada Gambar 5.9.

(c) Keragaman arsitektur perangkat keras menyebabkan perbedaan penerapan dan ketidakpastian kinerja di semua skenario, dan pemisahan pelatihan dan inferensi menyebabkan isolasi model.

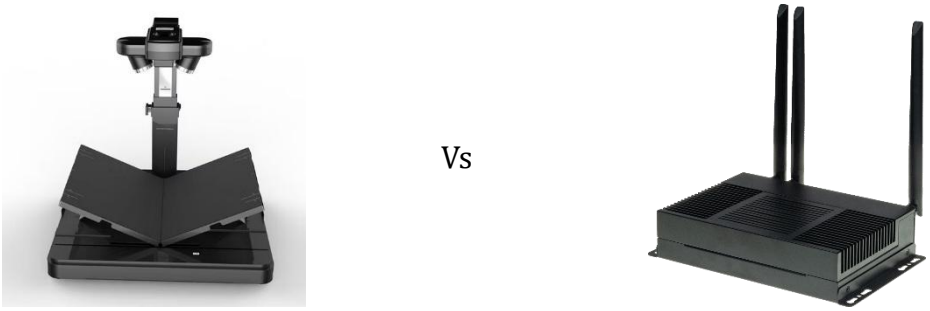
Dalam mode kolaborasi baru, semua skenario dapat dikoordinasikan sesuai permintaan, menghasilkan efisiensi sumber daya dan perlindungan privasi, keamanan, dan kredibilitas yang lebih baik, serta mendukung pengembangan satu kali dan beberapa penerapan. Model dapat berukuran besar atau kecil dan dapat diterapkan secara fleksibel, menawarkan pengalaman pengembangan yang konsisten.

MindSpore memiliki tiga teknologi kunci berikut terkait model kolaboratif baru.

(a) IR model terpadu dapat mengatasi perbedaan dalam berbagai skenario bahasa, kompatibel dengan struktur data yang disesuaikan, sehingga menghadirkan pengalaman penerapan yang konsisten.

Vs

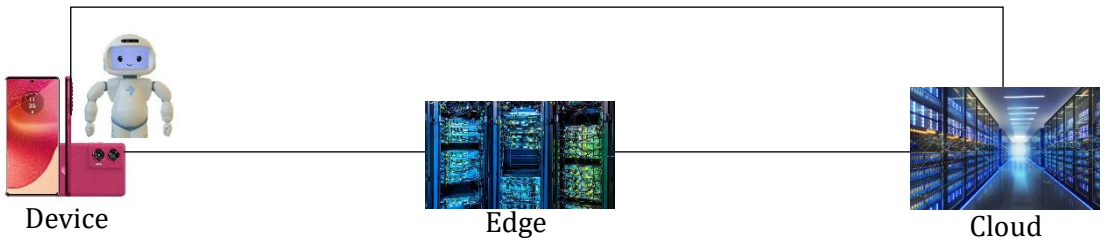
Skenario Aplikasi Di Perangkat, Edge, Dan Cloud Memiliki Tuntutan, Target, Dan Batasan Yang Berbeda



Perangkat Keras Yang Berbeda Memiliki Presisi Dan Kecepatan Yang Berbeda

Gambar 5.9 Tantangan dalam penerapan

Kolaborasi sesuai permintaan semua skenario dan praktik pengembangan yang

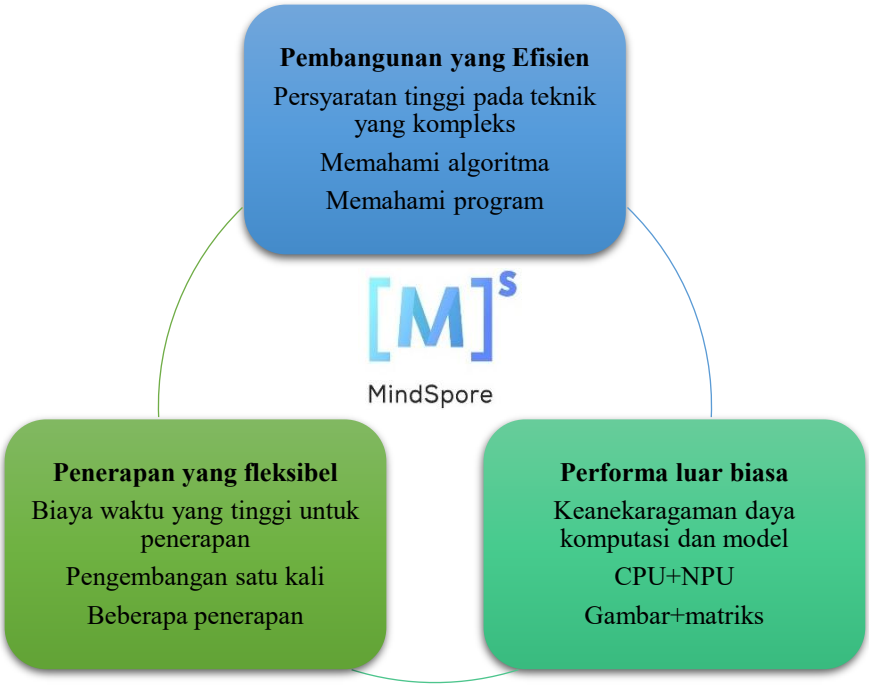


Gambar 5.10 Kolaborasi sesuai permintaan dan pengembangan yang konsisten

(b) Perangkat keras kerangka kerja ini juga dikembangkan oleh Huawei, dan teknologi optimasi grafik kolaboratif perangkat lunak-perangkat keras dapat menutupi perbedaan skenario.

(c) Strategi meta-pembelajaran terfederasi kolaboratif perangkat-awan mendobrak batasan antara perangkat dan awan, mewujudkan pembaruan waktu nyata dari model kolaboratif multi-perangkat.

Efek akhir dari ketiga teknologi kunci ini berada di bawah arsitektur terpadu, kinerja penerapan model semua skenario konsisten, dan akurasi model yang dipersonalisasi meningkat secara signifikan, seperti yang ditunjukkan pada Gambar 5.10. Visi dan nilai MindSpore adalah menyediakan platform komputasi AI dengan pengembangan yang efisien, kinerja yang sangat baik, dan penerapan yang fleksibel, untuk menurunkan standar masuk untuk pengembangan AI, melepaskan daya komputasi prosesor AI Ascend, dan mewujudkan AI yang inklusif, seperti yang ditunjukkan pada Gambar 5.11.



Gambar 5.11 Visi dan Nilai MindSpore

Keunggulan MindSpore

MindSpore memiliki keunggulan sebagai berikut.

1. Kemudahan Penggunaan dalam Pengembangan

(a) Diferensiasi otomatis, pemrograman jaringan dan operator terpadu, ekspresi asli rumus dan algoritma fungsional, serta pembuatan operator jaringan terbalik secara otomatis.

(b) Paralelisme otomatis, paralelisme model dengan efisiensi optimal yang dicapai melalui segmentasi model otomatis.

(c) Penyetelan otomatis, rangkaian kode yang sama untuk grafik dinamis dan grafik statis.

2. Efisiensi Tinggi dalam Eksekusi

(a) Eksekusi pada perangkat, memaksimalkan daya komputasi prosesor AI Ascend yang tangguh.

(b) Optimasi pipeline, memaksimalkan kinerja paralel.

(c) Optimasi peta kedalaman, daya komputasi, dan presisi Inti AI adaptif (arsitektur Da Vinci, lihat Bab 6 untuk detailnya).

3. Fleksibilitas dalam Penerapan

(a) Komputasi kolaboratif sesuai permintaan untuk perangkat, edge, dan cloud guna melindungi privasi dengan lebih baik.

(b) Arsitektur terpadu perangkat, edge, dan cloud memungkinkan pengembangan dan penerapan satu kali sesuai permintaan.

4. Setara dengan Kerangka Kerja Sumber Terbuka Industri

MindSpore sejajar dengan kerangka kerja sumber terbuka industri, mendukung CPU, GPU, dan perangkat keras lainnya, dengan chip dan layanan cloud yang dikembangkan sendiri diprioritaskan dalam layanan.

5. Ke Atas

MindSpore dapat terhubung dengan kerangka kerja pihak ketiga. Ia dapat terhubung dengan kerangka kerja pihak ketiga melalui Graph IR (docking front-end pelatihan, docking model inferensi), dan pengembang dapat memperluasnya.

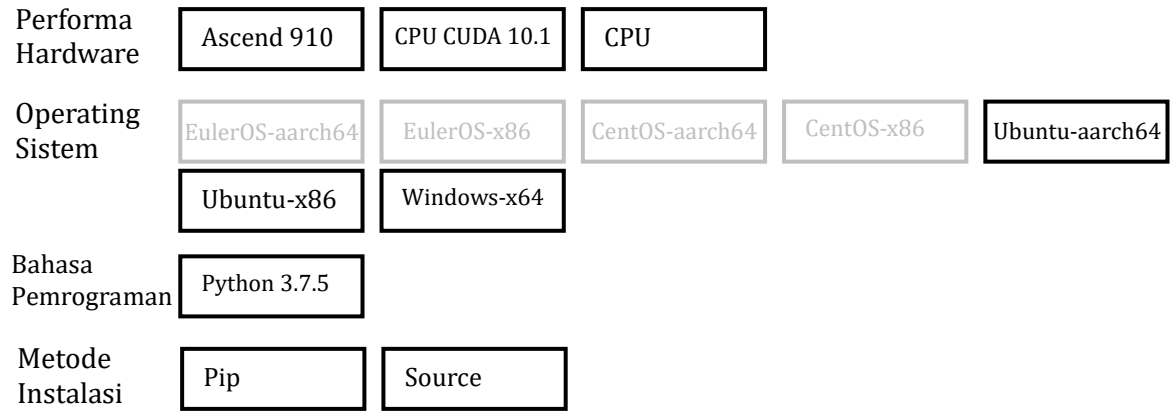
6. Ke Bawah

MindSpore dapat berinteraksi dengan chip pihak ketiga untuk membantu pengembang memperluas skenario aplikasi MindSpore dan memakmurkan ekosistem AI.

5.2 PENGEMBANGAN DAN APLIKASI MINDSPORE

Pengaturan Lingkungan

Pengaturan lingkungan pengembangan MindSpore memerlukan instalasi Python versi 3.7.5 atau yang lebih baru. MindSpore mendukung platform perangkat keras seperti CPU, GPU, dan Ascend910, serta sistem operasi seperti Ubuntu. MindSpore dapat diinstal langsung dengan paket instalasi atau kode sumber yang dikompilasi, seperti yang ditunjukkan pada Gambar 5.12. Sekarang, mari kita ambil contoh instalasi pada sistem Ubuntu 18.04 dengan lingkungan CPU untuk memperkenalkan langkah-langkah instalasi. Persyaratan versi CPU MindSpore pada sistem dan perangkat lunak ditunjukkan pada Tabel 5.2.



Gambar 5.12 Instalasi MindSpore

Tabel 5.2 Persyaratan sistem dan perangkat lunak Mindspore

Item	Persyaratan
Nomor Versi	MindSpore master
Sistem Operasi	• Ubuntu 18.04 × 86_64 • Ubuntu 18.04 aarch64
Ketergantungan Instalasi Berkas Eksekusi	• Python 3.7.5 • Untuk ketergantungan lain, periksa requirements.txt (lihat Gambar 5.13)

Ketergantungan Kompilasi dan Instalasi Kode Sumber	**Ketergantungan Kompilasi**: • Python 3.7.5 • wheel 0.32.0 atau lebih tinggi • GCC 7.3.0 • CMake 3.14.1 atau lebih tinggi • patch 2.5 atau lebih tinggi **Ketergantungan Instalasi**: • Sama dengan ketergantungan instalasi berkas eksekusi

GCC 7.3.0 dapat diinstal langsung melalui perintah apt.

Jika Python sudah terinstal, pastikan untuk menambahkan Python ke variabel sistem. Anda juga dapat menggunakan perintah "Python-Version" untuk memeriksa apakah versi Python memenuhi persyaratan.

1. Instalasi dengan Pip

Gunakan perintah pip berikut untuk menginstal MindSpore.

```
pip install -y MindSpore-cpu
```

Harap tambahkan pip ke variabel sistem untuk memastikan bahwa toolkit terkait Python dapat diinstal langsung melalui pip. Jika pip belum terinstal di sistem saat ini, Anda dapat mengunduhnya dari situs web resmi dan menginstalnya. Isi requirements.txt ditunjukkan pada Gambar 5.13. Ketika internet terhubung, item Reliance akan diunduh secara otomatis selama instalasi paket whl. Dalam kasus lain, Anda perlu menginstal item Reliance secara manual.

```
numpy >= 1.17.0, <= 1.17.5
protobuf >= 3.8.0
asttokens >= 1.1.13
pillow >= 6.2.0
scipy >= 1.3.3
easydict >= 1.9
sympy >= 1.4
cffi >= 1.13.2
wheel >= 0.32.0
decorator >= 4.4.0
setuptools >= 40.8.0
matplotlib >= 3.1.3        # for ut test
opencv-python >= 4.1.2.30  # for ut test
sklearn >= 0.0             # for st test
pandas >= 1.0.2            # for ut test
bs4
astunparse
packaging >= 20.0
pycocotools >= 2.0.0       # for st test
tables >= 3.6.1
```

Gambar 5.13 Isi requirements.txt

2. Instalasi melalui Kompilasi Kode Sumber

Prosedur untuk menginstal MindSpore melalui kode sumber adalah sebagai berikut.

(a) Kode sumber untuk mengunduh kode sumber dari repositori kode adalah sebagai berikut:

(b) Jalankan perintah berikut di direktori root kode sumber untuk mengkompilasi MindSpore:

- Sebelum menjalankan perintah di atas, pastikan jalur tempat berkas eksekusi cmake dan path store telah ditambahkan ke variabel lingkungan PATH.
- Pastikan alat git telah terinstal. Kemudian, perintah "git clone" akan dijalankan di build.sh untuk mendapatkan kode dari repositori kode pihak ketiga.
- Jika kinerja kompiler kuat, Anda dapat menambahkan -j{Jumlah utas} ke dalam skrip untuk menambah jumlah utas. Misalnya, bash build.sh -e cpu -j12.

(c) Jalankan perintah berikut untuk menginstal MindSpore.

```
  chmod +x build/package/MindSpore-{version}-cp37-cp37m-linux_
{arch}.whl
  pip install build/package/MindSpore-{version}-cp37-cp37m
linux_{arch}.whl
```

(d) Jalankan perintah berikut. Jika outputnya adalah "Tidak ada modul bernama 'mindspore'", artinya instalasi tidak berhasil.

```
python -c 'import MindSpore'
```

Komponen dan Konsep Terkait MindSpore

1. Komponen

Beberapa komponen dan deskripsi yang umum digunakan oleh MindSpore ditunjukkan pada Tabel 5.3.

Tabel 5.3 Komponen dan deskripsi MindSpore

Komponen	Deskripsi
Tensor	Penyimpanan data
model_zoo	Definisi model jaringan umum
communication	Modul pemuatan data, termasuk definisi *dataloader*, *dataset* serta beberapa fungsi pemrosesan data gambar, teks, dan lainnya
dataset	Modul pemrosesan *dataset*, seperti pembacaan data dan prapemrosesan data
common	Definisi tensor, parameter, *dtype*, dan inisialisasi
context	Definisi kelas Context, mengatur parameter operasi model, seperti beralih antara mode grafik dan mode pynative
akg	Diferensiasi otomatis dan pustaka operator kustom

nn	Definisi unit jaringan saraf MindSpore, fungsi kerugian (*loss function*), dan pengoptimal (*optimizer*)
ops	Definisi dasar operator dan pendaftaran operator balik (*reverse operator*)
train	Modul yang terkait dengan pelatihan model dan fungsi ringkasan
utils	Utilitas, terutama untuk verifikasi parameter (digunakan secara internal oleh kerangka kerja)

Dalam MindSpore, struktur data paling dasar juga merupakan tensor. Terdapat beberapa operasi tensor yang umum, yaitu:

```
asnumpy()
size()
dim()
dtype()
set_dtype()
tensor_add(other: Tensor)

tensor_mul(ohter: Tensor)
shape()
__str__# convert into string
```

Operasi-operasi tensor ini pada dasarnya dapat dipahami dengan membaca namanya, misalnya asnumpy() berarti konversi ke array NumPy, tensor_add() berarti penjumlahan tensor.

2. Konsep Pemrograman: Operasi

Ada beberapa operasi umum di MindSpore sebagai berikut.

(a) Array: Operator yang berhubungan dengan array sebagai berikut:

```
 -ExpandDims - Squeeze
-Concat   - OnesLike
-Select   - StridedSlice
-ScatterNd ...
```

(b) Matematika: operator terkait perhitungan matematika sebagai berikut:

```
-AddN        - Cos
-Sub         - Sin
-Mul         - LogicalAnd
-MatMul      - LogicalNot
-RealDiv     - Less
-ReduceMean  - Greater
...
```

(c) Nn: Operator jaringan sebagai berikut:

```
-Conv2D       - MaxPool
-Flatten      - AvgPool
-Softmax      - TopK
-ReLU         - SoftmaxCrossEntropy
-Sigmoid      - SmoothL1Loss
-Pooling      - SGD
-BatchNorm    - SigmoidCrossEntropy
...
```

(d) Kontrol: kendalikan operator sebagai berikut:

```
ControlDepend
```

3. Konsep Pemrograman: Sel

(a) Sel mendefinisikan modul-modul dasar untuk melakukan perhitungan. Objek-objek sel dapat dieksekusi secara langsung.

- init, menginisialisasi parameter (parameter), sub-modul (sel), operator (primitif), dan komponen lainnya, serta melakukan verifikasi inisialisasi.
- Construct, yang mendefinisikan proses eksekusi. Dalam mode grafik, fungsi ini akan dikompilasi menjadi grafik untuk dieksekusi. Terdapat beberapa batasan tata bahasa.
- Bprop (opsional), kebalikan dari modul kustom. Ketika fungsi ini tidak didefinisikan, diferensiasi otomatis akan digunakan untuk menghitung kebalikan dari bagian konstruk.

(b) Sel-sel yang telah ditentukan sebelumnya di MindSpore terutama mencakup: fungsi kerugian yang umum digunakan (seperti SoftmaxCrossEntropyWithLogits, MSELoss), pengoptimal yang umum digunakan (seperti Momentum, SGD, Adam), dan fungsi pengemasan jaringan yang umum digunakan (seperti TrainOneStepCell, WithGradCell).

4. Konsep Pemrograman: MindSporeIR

(a) MindSporeIR adalah IR fungsional berbasis graf yang ringkas, efisien, dan fleksibel yang dapat mengekspresikan semantik fungsional seperti variabel bebas, fungsi orde tinggi, dan rekursi.

(b) Setiap graf merepresentasikan definisi fungsi, dan graf tersebut terdiri dari ParameterNode, ValueNode, dan ComplexNode (CNode).

Realisasi Pengenalan Digit Tulis Tangan dengan MindSpore

1. Ikhtisar

Tutorial ini menggunakan contoh klasifikasi gambar sederhana untuk mendemonstrasikan fungsi dasar MindSpore. Untuk pengguna umum, dibutuhkan waktu 20–30 menit untuk menyelesaikan seluruh contoh praktik. Ini adalah proses aplikasi yang

sederhana dan mendasar, dan aplikasi lanjutan dan kompleks lainnya dapat diperluas berdasarkan proses dasar ini.

Contoh ini akan mengimplementasikan klasifikasi gambar sederhana, dan keseluruhan prosesnya adalah sebagai berikut:

(a) Memproses dataset yang dibutuhkan. Dataset MNIST digunakan dalam contoh ini.
(b) Mendefinisikan jaringan. Jaringan LeNet digunakan dalam contoh ini.
(c) Tentukan fungsi kerugian dan pengoptimalnya.
(d) Muat dataset dan lakukan pelatihan. Setelah pelatihan selesai, periksa hasilnya dan simpan berkas model.
(e) Muat model yang telah disimpan untuk inferensi.
(f) Validasi model, muat dataset uji dan model yang telah dilatih, dan validasi akurasi hasilnya.

2. Persiapan

Sebelum latihan, periksa apakah MindSpore telah terpasang dengan benar. Jika belum, instal MindSpore di komputer. Selain itu, Anda harus memiliki pengetahuan matematika dasar seperti dasar-dasar pemrograman Python, probabilitas, dan matriks.
Selanjutnya, mari kita mulai perjalanan MindSpore.

(a) Unduh Dataset MNIST.

Dataset MNIST yang digunakan dalam contoh ini terdiri dari 10 kelas gambar skala abu-abu berukuran 28 × 28 piksel. Dataset ini memiliki 60.000 contoh dalam satu set pelatihan, dan 10.000 contoh dalam satu set pengujian. Halaman unduh MNIST menyediakan empat tautan unduh berkas set data. Dua tautan pertama diperlukan untuk pelatihan data, dan dua tautan terakhir diperlukan untuk pengujian data.

Unduh berkas-berkas tersebut, dekompresi, dan simpan di direktori ruang kerja: `./MNIST_Data/train dan ./MNIST_Data/test.`

Struktur direktorinya adalah sebagai berikut:

```
└─MNIST_Data
  ├ ─test
  │ t10k-images.idx3-ubyte
  │ t10k-labels.idx1-ubyte
  └─train
    train-images.idx3-ubyte
    train-labels.idx1-ubyte
```

Untuk memudahkan pengoperasian, kami menambahkan fungsi pengunduhan dataset secara otomatis pada contoh kode skrip.

(b) Impor Pustaka dan Modul Python Sebelum penggunaan, impor pustaka Python.

Sekarang pustaka telah digunakan. Untuk pemahaman yang lebih baik, pustaka lain yang diperlukan akan diperkenalkan saat kami menggunakannya dalam praktik.

import os

(c) Konfigurasi Informasi yang Berjalan

Sebelum mengompilasi kode secara resmi, kita perlu mengetahui informasi dasar tentang perangkat keras dan backend yang diperlukan untuk MindSpore saat run-time.

Kita dapat menggunakan context.set_context untuk mengonfigurasi informasi yang diperlukan saat run-time, seperti mode berjalan, informasi backend, dan informasi perangkat keras. Impor modul konteks dan konfigurasikan informasi yang diperlukan. Kode-kode tersebut dicontohkan sebagai berikut.

```
      import argparse
      from MindSpore import context

if __name__ == "__main__":
  parser = argparse.ArgumentParser(description='MindSpore LeNet
Example')
  parser.add_argument('--device_target', type=str,
default="Ascend",     choices=['Ascend',     'GPU',     'CPU'],
help='device
where the code will be implemented (default: Ascend)')
   args = parser.parse_args()
 context.set_context(mode=context.GRAPH_MODE,device_target=arg
 s.
device_target, enable_ mem_reuse=False)
...
```

Kami menggunakan mode grafik saat menerapkan contoh yang disebutkan di atas. Anda dapat mengonfigurasi informasi perangkat keras sesuai dengan persyaratan situs. Misalnya, jika kode berjalan pada prosesor Ascend AI, maka device_target memilih Ascend. Aturan ini juga berlaku untuk kode yang berjalan pada CPU dan GPU. Untuk detail tentang parameter, lihat antarmuka `context.set_context()`.

3. Memproses Data

Kumpulan data penting untuk pelatihan. Kumpulan data yang baik dapat secara efektif meningkatkan akurasi dan efisiensi pelatihan. Secara umum, sebelum memuat kumpulan data, Anda perlu melakukan beberapa operasi pada kumpulan data tersebut. Tentukan fungsi create_dataset() untuk membuat kumpulan data. Dalam fungsi ini, kita perlu mendefinisikan augmentasi data dan proses yang akan dilakukan.

(a) Tentukan kumpulan data.
(b) Tentukan parameter yang diperlukan untuk augmentasi dan pemrosesan data.
(c) Hasilkan augmentasi data yang sesuai sesuai dengan parameter.
(d) Gunakan fungsi pemetaan map() untuk menerapkan proses data ke kumpulan data.

(e) Proses kumpulan data yang dihasilkan.

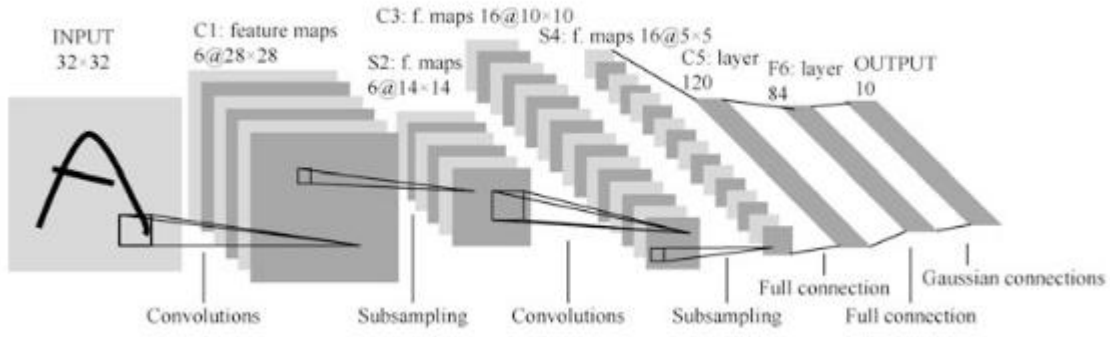
Contoh kode untuk memproses dataset adalah sebagai berikut:

```
    import MindSpore.dataset as ds
    import MindSpore.dataset.transforms.c_transforms as C
    import MindSpore.dataset.transforms.vision.c_transforms as
    CV
    from MindSpore.dataset.transforms.vision import Inter
    from MindSpore.common import dtype as mstype

def create_dataset(data_path, batch_size=32, repeat_size=1,
    num_parallel_workers=1):
  """ create dataset for train or test
  Args:

    data_path: Data path
    batch_size: The number of data records in each group
    repeat_size: The number of replicated data records
    num_parallel_workers: The number of parallel workers
"""
# define dataset
mnist_ds = ds.MnistDataset(data_path)

    # define operation parameters
    resize_height, resize_width = 32, 32
    rescale = 1.0 / 255.0
    shift = 0.0
    rescale_nml = 1 / 0.3081
    shift_nml = -1 * 0.1307 / 0.3081

    # define map operations
    resize_op = CV.Resize((resize_height, resize_width),
interpolation=Inter.LINEAR) # resize images to (32, 32)
    rescale_nml_op = CV.Rescale(rescale_nml, shift_nml) #
    normalize
images
    rescale_op = CV.Rescale(rescale, shift) # rescale images
    hwc2chw_op = CV.HWC2CHW() # change shape from (height,
    width,
channel) to (channel, height, width) to fit network.
    type_cast_op = C.TypeCast(mstype.int32) # change data type
    of label
to int32 to fit network
```

```
    # apply map operations on images
    mnist_ds = mnist_ds.map(input_columns="label",
operations=type_cast_op, num_parallel_
workers=num_parallel_workers)
    mnist_ds = mnist_ds.map(input_columns="image",
operations=resize_op,                          num_parallel_
workers=num_parallel_workers)
    mnist_ds = mnist_ds.map(input_columns="image",
operations=rescale_op,                         num_parallel_
workers=num_parallel_workers)
    mnist_ds = mnist_ds.map(input_columns="image",
operations=rescale_nml_op, num_parallel_
workers=num_parallel_workers)
    mnist_ds = mnist_ds.map(input_columns="image",
operations=hwc2chw_op, num_parallel_
workers=num_parallel_workers)

    # apply DatasetOps
    buffer_size = 10000
    mnist_ds = mnist_ds.shuffle(buffer_size=buffer_size) #
    10000 as in
LeNet train script
    mnist_ds = mnist_ds.batch(batch_size,
    drop_remainder=True)
    mnist_ds = mnist_ds.repeat(repeat_size)
    return mnist_ds
```

Catatan untuk kode-kode tersebut adalah sebagai berikut.

- Ukuran batch: jumlah data dalam setiap grup. Saat ini, setiap grup berisi 32 rekaman data.



Gambar 5.14 Arsitektur LeNet-5

- Repeat_size: jumlah replikasi dataset.

Pertama, implementasikan operasi shuffle dan batch, lalu lakukan operasi repeat untuk memastikan data tidak berulang selama satu epoch.
MindSpore mendukung beberapa operasi pemrosesan dan augmentasi data, yang biasanya digunakan secara kombinasi.

4. Mendefinisikan Jaringan

Di sini kita memilih jaringan LeNet yang relatif sederhana. Selain lapisan input, jaringan LeNet memiliki tujuh lapisan, termasuk dua lapisan konvolusional, dua lapisan down-sampling (lapisan pooling), dan tiga lapisan terhubung penuh. Setiap lapisan berisi jumlah parameter pelatihan yang berbeda, seperti yang ditunjukkan pada Gambar 5.14:

Kita perlu menginisialisasi lapisan terhubung penuh dan lapisan konvolusional. MindSpore mendukung beberapa metode inisialisasi parameter, seperti TruncatedNormal, Normal, dan Uniform. Untuk detailnya, lihat ilustrasi di modul Mindspore.common.initializer dari API MindSpore. Kasus ini menggunakan metode inisialisasi parameter TruncatedNormal. Berikut adalah contoh kode yang dicontohkan untuk menginisialisasi lapisan terhubung penuh dan lapisan konvolusional:

```
     import MindSpore.nn as nn
     from MindSpore.common.initializer import TruncatedNormal

def weight_variable():
  """
  weight initial
  """
  return TruncatedNormal(0.02)

  def conv(in_channels, out_channels, kernel_size, stride=1,
padding=0):
  """
  conv layer weight initial
  """
  weight = weight_variable()
  return nn.Conv2d(in_channels,out_channels, _size=kernel_size,

stride=stride,     padding=     padding     weight_init=weight,
has_bias=False,
pad_mode="valid")
 def fc_with_initialize(input_channels, out_channels):
   """
   fc layer weight initial
   """
   weight = weight_variable()
   bias = weight_variable()
   return nn.Dense(input_channels, out_channels, weight, bias)
```

Untuk menggunakan MindSpore guna mendefinisikan jaringan saraf tiruan, diperlukan pewarisan MindSpore.nn.Cell. Cell adalah kelas dasar dari semua jaringan saraf tiruan (seperti Conv2d). Setiap lapisan jaringan saraf tiruan perlu didefinisikan terlebih dahulu oleh metode init(), lalu didefinisikan dengan metode construct() untuk menyelesaikan konstruksi jaringan saraf tiruan selanjutnya. Berdasarkan struktur jaringan LeNet, definisikan lapisan-lapisan jaringan sebagai berikut:

5. Mendefinisikan Fungsi Kerugian dan Pengoptimal

(a) Konsep fungsi kerugian dan pengoptimal.

Fungsi kerugian: Juga dikenal sebagai fungsi objektif, fungsi kerugian digunakan untuk mengukur perbedaan antara nilai prediksi dan nilai aktual. Pembelajaran mendalam mengurangi nilai fungsi kerugian melalui iterasi. Mendefinisikan fungsi kerugian yang baik dapat meningkatkan kinerja model secara efektif.

- ✓ Pengoptimal: Digunakan untuk meminimalkan fungsi kerugian agar model dapat dioptimalkan selama pelatihan.

Setelah fungsi kerugian didefinisikan, gradien terkait bobot dari fungsi kerugian juga dapat diperoleh. Gradien digunakan untuk menunjukkan arah optimasi bobot bagi pengoptimal dan meningkatkan kinerja model.

(b) Mendefinisikan Fungsi Kerugian

Fungsi kerugian yang didukung oleh MindSpore meliputi SoftmaxCrossEntropyWithLogits, L1Loss, dan MSELoss. Fungsi kerugian SoftmaxCrossEntropyWithLogits digunakan dalam contoh kode berikut:

```
class LeNet5(nn.Cell):
  """
  Lenet network structure
  """
  #define the operator required
  def __init__(self):
    super(LeNet5, self).__init__()
    self.batch_size = 32
    self.conv1 = conv(1, 6, 5)
    self.conv2 = conv(6, 16, 5)
    self.fc1 = fc_with_initialize(16 * 5 * 5, 120)
    self.fc2 = fc_with_initialize(120, 84)
    self.fc3 = fc_with_initialize(84, 10)
    self.relu = nn.ReLU()
    self.max_pool2d = nn.MaxPool2d(kernel_size=2, stride=2)
    self.flatten = nn.Flatten()
#use the preceding operators to construct networks
def construct(self, x):
  x = self.conv1(x)
```

```
  x = self.relu(x)
  x = self.max_pool2d(x)
  x = self.conv2(x)
  x = self.relu(x)
  x = self.max_pool2d(x)
  x = self.flatten(x)
  x = self.fc1(x)
  x = self.relu(x)
  x = self.fc2(x)
  x = self.relu(x)
  x = self.fc3(x)
  return x
from MindSpore.nn.loss import SoftmaxCrossEntropyWithLogits
```

Panggil fungsi kerugian yang telah ditentukan di fungsi utama. Contoh kodenya adalah sebagai berikut.

```
if __name__ == "__main__":
   ...
   #define the loss function
   net_loss = SoftmaxCrossEntropyWithLogits(is_grad=False,
sparse=True, reduction='mean')
   ...
```

(c) Mendefinisikan Pengoptimal

Pengoptimal yang didukung oleh MindSpore meliputi Adam, AdamWeightDecay, dan Momentum. Contoh kode berikut didasarkan pada pengoptimal Momentum, yang cukup populer:

```
if __name__ == "__main__":
   ...
   #learning rate setting
   lr = 0.01
   momentum = 0.9
   #create the network
   network = LeNet5()
   #define the optimizer
   net_opt = nn.Momentum(network.trainable_params(), lr,
momentum)
  ...
```

6. Melatih Jaringan

(a) Menyimpan Model yang Dikonfigurasi

MindSpore menyediakan mekanisme panggilan balik untuk menjalankan logika yang disesuaikan selama pelatihan. Di sini, kami mencantumkan ModelCheckpoint dan LossMonitor yang disediakan oleh kerangka kerja sebagai contoh.

```
        from MindSpore.train.callback import ModelCheckpoint,
      CheckpointConfig

if __name__ == "__main__":
   ...
   # set parameters of check point
   config_ck = CheckpointConfig(save_checkpoint_steps=1875,
keep_checkpoint_max=10)
    # apply parameters of check point
    ckpoint_cb = ModelCheckpoint(prefix="checkpoint_lenet",
config=config_ck)
   ...
```

ModelCheckpoint dapat menyimpan model dan parameter jaringan untuk fine-tuning selanjutnya. LossMonitor dapat memantau perubahan nilai kerugian selama pelatihan.

(b) Konfigurasi Pelatihan Jaringan

Melalui API model.train() yang disediakan oleh MindSpore, kita dapat melatih jaringan dengan mudah. Di sini, kita menetapkan epoch-size ke 1 untuk melatih dataset selama 1 iterasi.

```
     from MindSpore.nn.metrics import Accuracy
     from MindSpore.train.callback import LossMonitor
     from MindSpore.train import Model
     ...
     def train_net(args, model, epoch_size, mnist_path,
   repeat_size, ckpoint_cb):
      """define the training method"""
      print("============== Starting Training ==============")
      #load training dataset
      ds_train    =    create_dataset(os.path.join(mnist_path,
      "train"),
   32, repeat_size)
       model.train(epoch_size, ds_train, callbacks=[ckpoint_cb,
   LossMonitor()], dataset_ sink_mode=False)
    ...
if __name__ == "__main__":
   ...
   epoch_size = 1
   mnist_path = "./MNIST_Data"
```

```
   repeat_size = epoch_size
   model=Model(network, net_loss, net_opt, metrics={"Accuracy":
Accuracy()})
train_net(args, model, epoch_size, mnist_path, repeat_size,
ckpoint_cb)
...
```

Di sini, dalam metode train_net(), kita memuat dataset pelatihan yang telah kita unduh sebelumnya, dan mnist_path adalah jalur dataset MNIST.

7. Jalankan dan Lihat Hasilnya

Jalankan kode skrip menggunakan perintah berikut:

```
python lenet.py --device_target=CPU
```

Penjelasannya adalah sebagai berikut. lenet.py: berkas skrip yang Anda tulis.

- ✓ --device_target CPU: Tentukan platform perangkat keras. Parameternya adalah 'CPU', 'GPU', atau 'Ascend'.

Nilai kerugian dicetak selama pelatihan, seperti yang ditunjukkan pada gambar berikut. Meskipun nilai kerugian dapat berfluktuasi, nilainya akan menurun secara bertahap. Nilai kerugian untuk setiap proses mungkin berbeda karena sifat acaknya. Berikut adalah contoh keluaran nilai kerugian selama pelatihan.

```
epoch: 1 step: 262, loss is 1.9212162
epoch: 1 step: 263, loss is 1.8498616
epoch: 1 step: 264, loss is 1.7990671
epoch: 1 step: 265, loss is 1.9492403
epoch: 1 step: 266, loss is 2.0305142
epoch: 1 step: 267, loss is 2.0657792
epoch: 1 step: 268, loss is 1.9582214
epoch: 1 step: 269, loss is 0.9459006
epoch: 1 step: 270, loss is 0.8167224
epoch: 1 step: 271, loss is 0.7432692
...
```

Berikut ini adalah contoh file model yang disimpan setelah pelatihan, yaitu menyimpan file parameter model:

```
checkpoint_lenet-1_1875.ckpt
```

Penjelasannya adalah sebagai berikut.

- ✓ checkpoint_lenet-1_1875.ckpt: Merujuk pada berkas parameter model yang tersimpan. checkpoint_network berarti nama jaringan, nomor epoch berarti nomor seri epoch, dan .ckpt berarti nomor seri langkah.

8. Validasi Model

Setelah mendapatkan berkas model, kami memverifikasi kemampuan generalisasi model. Gunakan antarmuka model.eval() untuk memuat dataset pengujian. Inferensi dilakukan dengan menggunakan model yang tersimpan setelah pelatihan.

Contoh kodenya adalah sebagai berikut:

```
   from MindSpore.train.serialization import load_checkpoint,
load_param_into_net
...
def test_net(args,network,model,mnist_path):
"""define the evaluation method"""

   print("============== Starting Testing ==============")
   #load the saved model for evaluation
   param_dict = load_checkpoint("checkpoint_lenet-1_1875.ckpt")
   #load parameter to the network
   load_param_into_net(network, param_dict)
   #load testing dataset
   ds_eval = create_dataset(os.path.join(mnist_path, "test"))
   acc = model.eval(ds_eval, dataset_sink_mode=False)
   print("============ Accuracy:{}=========".format(acc))

if __name__ == "__main__":
  ...
  test_net(args, network, model, mnist_path)
```

Penjelasannya sebagai berikut.

load_checkpoint(): Antarmuka ini digunakan untuk memuat berkas parameter model CheckPoint dan mengembalikan kamus parameter.
checkpoint_lenet-1_1875.ckpt: nama berkas model CheckPoint yang disimpan. load_param_into_net(): Antarmuka ini digunakan untuk memuat parameter ke jaringan.

Jalankan kode skrip dengan perintah berikut:

```
python lenet.py --device_target=CPU
```

Contoh hasil menjalankan kode contoh model verifikasi disajikan sebagai berikut:

```
============== Starting Testing ==============
========== Accuracy:{'Accuracy':0.9742588141025641} ==========
```

Di sini kita dapat menemukan bahwa akurasi model ditampilkan dalam konten keluaran. Dalam contoh ini, akurasinya mencapai 97,4%, yang menunjukkan bahwa kualitas model memuaskan.

5.3 KESIMPULAN

Bab ini terutama memperkenalkan kerangka kerja pembelajaran mendalam MindSpore yang dikembangkan secara independen oleh Huawei. Pertama, bab ini menyajikan tiga inovasi teknologi MindSpore dalam desainnya, yaitu mode eksekusi baru, mode kolaborasi baru, dan keunggulan seperti pengembangan yang ramah pengguna, eksekusi yang efisien, dan penerapan yang fleksibel. Terakhir, bab ini menjelaskan secara singkat tentang pengembangan dan penerapan MindSpore, dan menggunakan contoh klasifikasi gambar untuk mengilustrasikan prosedur pengembangan MindSpore.

Karena MindSpore masih dalam tahap iterasi versi yang cepat, silakan merujuk ke materi resmi terbaru untuk mempelajari operasi spesifiknya.

Latihan Soal

1. MindSpore adalah kerangka kerja komputasi AI kolaboratif sesuai permintaan untuk semua skenario perangkat-edge-cloud yang dikembangkan oleh Huawei. Kerangka kerja ini menyediakan API terpadu untuk semua skenario, dan menyediakan kapabilitas antar-perangkat untuk pengembangan model, operasi model, dan penerapan model AI untuk semua skenario. Apa saja fitur utama arsitektur MindSpore?
2. Untuk mengatasi tantangan yang dihadapi oleh pengembang AI, termasuk standar masuk yang tinggi, biaya operasional yang tinggi, dan penerapan yang sulit, MindSpore mengusulkan tiga inovasi teknologi untuk mencapai pengembangan dan penerapan aplikasi AI yang lebih mudah dan efisien. Apa saja inovasi tersebut?
3. Tantangan mode eksekusi dengan daya komputasi super chip: masalah dinding memori, overhead interaksi yang tinggi, pasokan data yang sulit, dan tingkat hunian akselerator yang rendah karena overhead interaksi bahkan melampaui overhead eksekusi karena sebagian operasi dieksekusi di host dan sebagian di perangkat. Apa solusi yang ditawarkan MindSpore?
4. Gunakan MindSpore untuk melakukan pengenalan digit tulisan tangan MNIST.

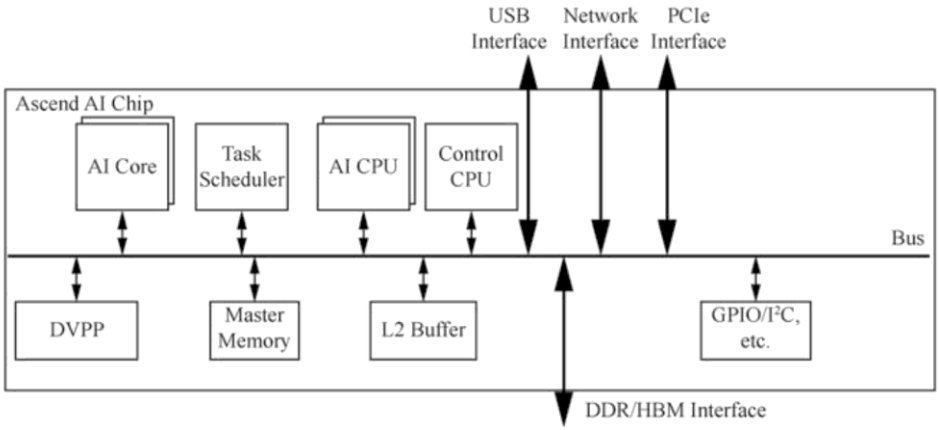
BAB 6
SOLUSI KOMPUTASI AI HUAWEI ATLAS

Bab ini terutama memperkenalkan prosesor AI Huawei Ascend dan solusi komputasi AI Huawei Atlas, dengan fokus pada arsitektur perangkat keras dan perangkat lunak prosesor AI Ascend serta solusi AI full-stack dan semua skenario Huawei.

6.1 ARSITEKTUR PERANGKAT KERAS PROSESOR AI ASCEND

Arsitektur Logika Perangkat Keras Prosesor AI Ascend

Arsitektur logika perangkat keras prosesor AI Ascend terutama terdiri dari empat modul, yaitu CPU Kontrol, mesin komputasi AI (termasuk AI Core dan AI CPU), cache atau Buffer sistem-pada-chip multi-level, Digital Vision Pre-Processing (DVPP), dll., seperti yang ditunjukkan pada Gambar 6.1. Berikut ini akan berfokus pada Inti AI dari mesin komputasi AI, yaitu Arsitektur Da Vinci.



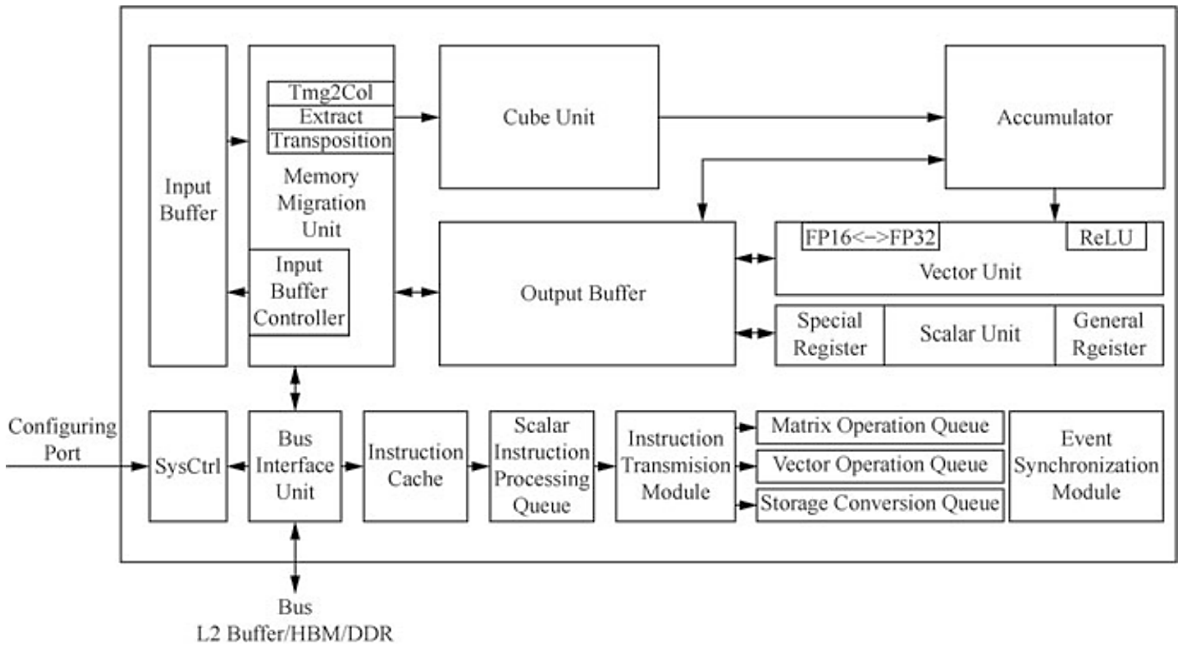
Gambar 6.1 Arsitektur logika perangkat keras prosesor AI Ascend

Arsitektur Da Vinci

Arsitektur Da Vinci, yang merupakan mesin komputasi AI sekaligus inti dari prosesor AI Ascend, dikembangkan secara khusus untuk meningkatkan daya komputasi AI. Arsitektur Da Vinci terutama terdiri dari tiga bagian: unit komputasi, sistem memori, dan unit kontrol.

a. Unit komputasi mencakup tiga sumber daya komputasi dasar: Unit Kubus, Unit Vektor, dan Unit Skalar.
b. Sistem memori mencakup unit memori on-chip AI Core dan jalur data terkait.
c. Unit kontrol menyediakan kendali perintah untuk seluruh proses kalkulasi, yang setara dengan kantor pusat inti AI dan bertanggung jawab atas pengoperasian seluruh Inti AI.

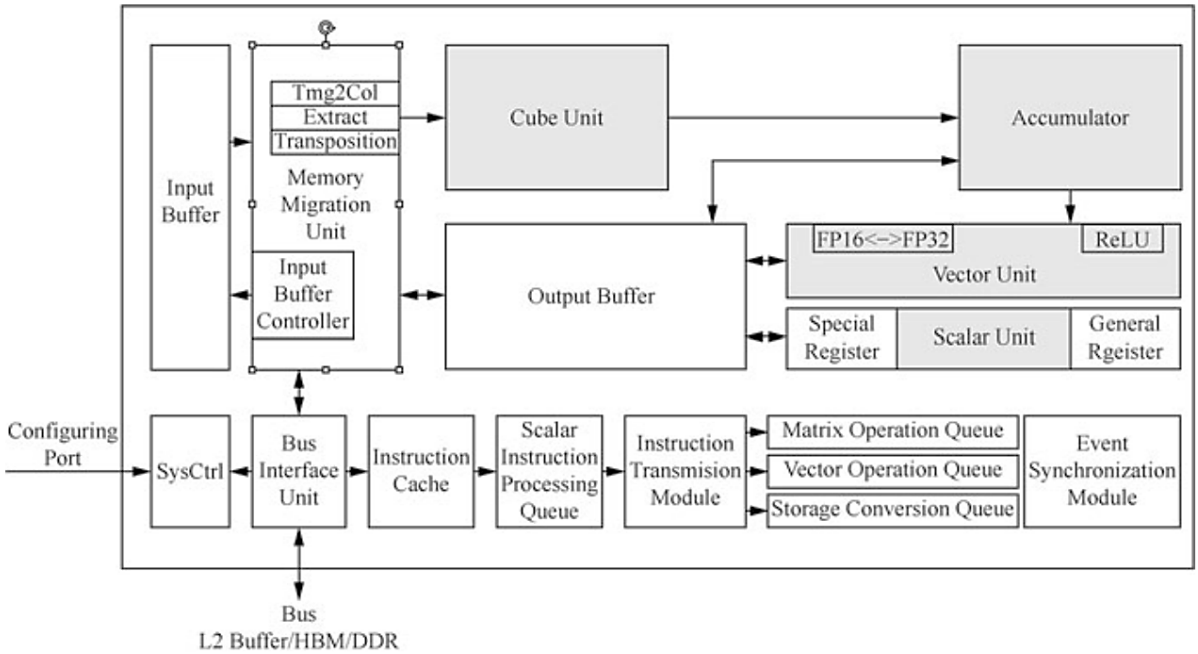
Arsitektur Da Vinci ditunjukkan pada Gambar 6.2.



Gambar 6.2 Arsitektur Da Vinci

1. Unit Komputasi

Terdapat tiga unit komputasi dasar dalam Arsitektur Da Vinci: Unit Kubus, Unit Vektor, dan Unit Skalar, yang masing-masing sesuai dengan tiga mode komputasi umum, yaitu kubus, vektor, dan skalar, seperti yang ditunjukkan pada Gambar 6.3.

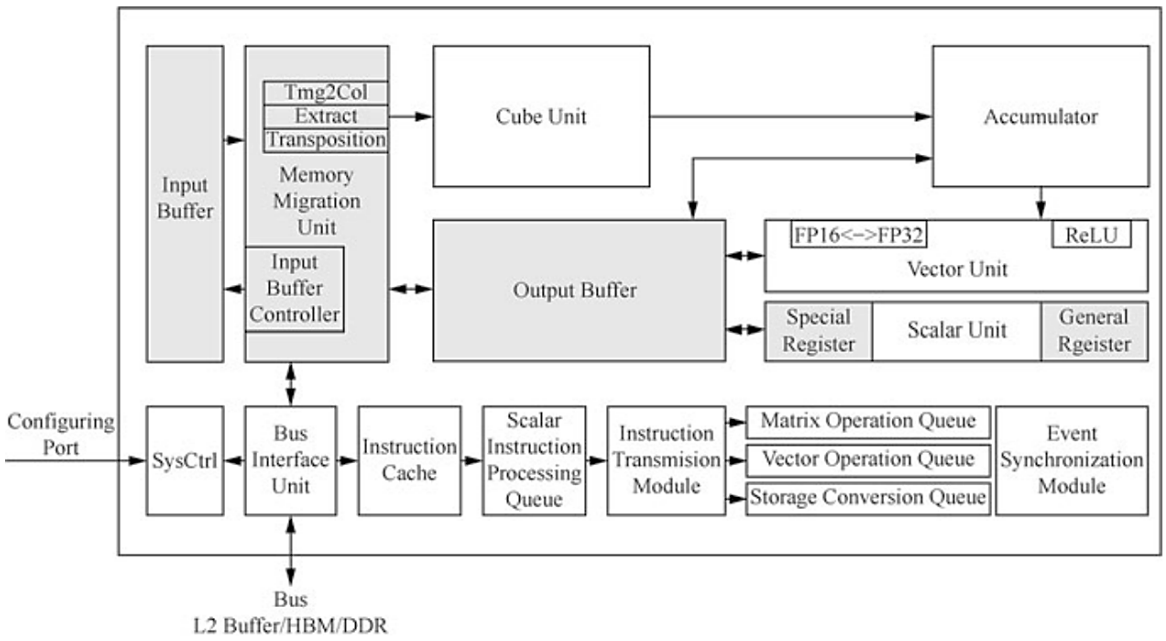


Gambar 6.3 Arsitektur Da Vinci—unit komputasi

- Unit Kubus: Fungsi utama Unit Kubus dan akumulator adalah untuk menyelesaikan operasi korelasi matriks. Satu ketukan menyelesaikan perkalian matriks 16 × 16 dan 16 × 16 (4096) dari satu FP16; Jika data masukan bertipe Int 8, maka satu ketukan akan menyelesaikan perkalian matriks 16 × 32 dan 32 × 16 (8192).
- Unit Vektor: Unit ini mengimplementasikan kalkulasi antara vektor dan skalar, serta vektor ganda. Fungsinya mencakup berbagai tipe kalkulasi dasar dan berbagai tipe kalkulasi khusus, seperti FP16, FP32, Int32, Int8, dan tipe data lainnya.
- Unit Skalar: Unit ini setara dengan CPU mikro, yang mengendalikan seluruh operasi AI Core, menyelesaikan kontrol siklus dan penilaian cabang dari keseluruhan program, menyediakan alamat data dan kalkulasi parameter terkait untuk matriks dan vektor, serta operasi aritmatika dasar.

2. Sistem Memori

Unit Memori dan jalur data yang sesuai membentuk Sistem Memori Arsitektur Da Vinci, seperti yang ditunjukkan pada Gambar 6.4.



Gambar 6.4 Arsitektur Da Vinci—sistem memori

(a) Unit Memori terdiri dari unit kontrol memori, buffer, dan register.

- Unit Kontrol Penyimpanan: Akses langsung ke cache tingkat rendah selain AI Core dapat dicapai melalui antarmuka bus. Unit ini juga dapat mengakses memori secara langsung melalui Double Data Rate Synchronous Dynamic Random Access Memory (SDRAM, disingkat DDR) atau High Bandwidth Memory (HBM). Unit migrasi memori, yang bertanggung jawab atas manajemen baca-tulis data internal AI Core antar buffer yang berbeda, juga berperan sebagai pengontrol transmisi jalur data internal AI Core, yang menyelesaikan serangkaian operasi konversi format, seperti pengisian, Img2Col, transposisi, ekstraksi, dll.

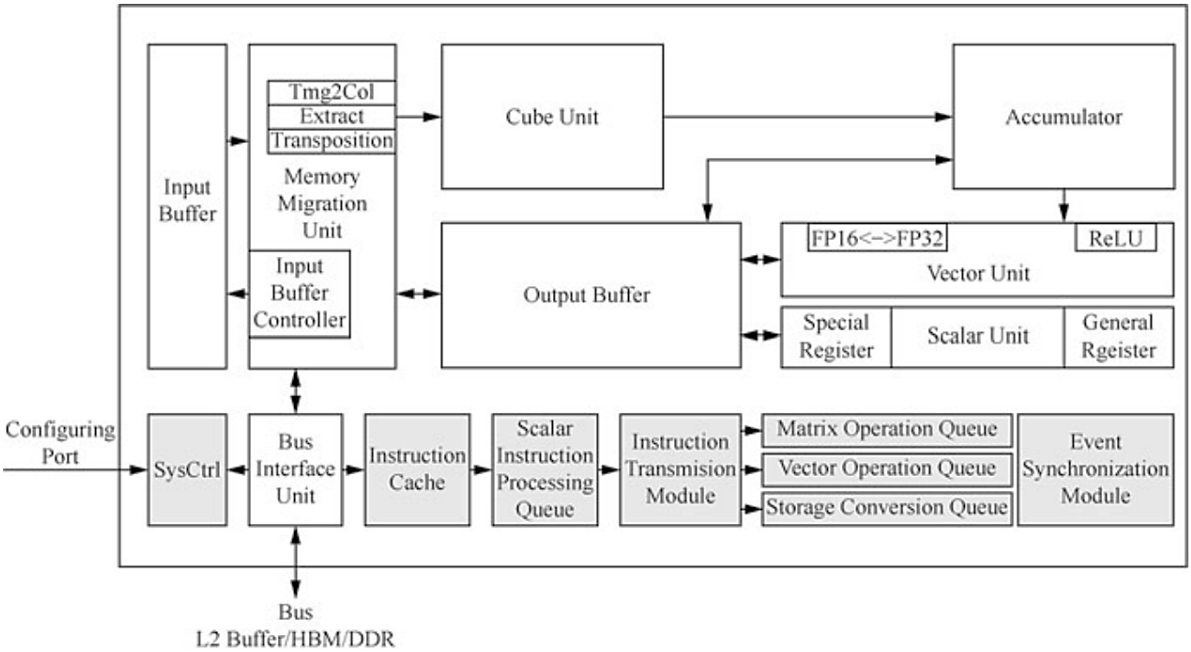
- Buffer Input: Digunakan untuk menyimpan sementara data yang sering digunakan, sehingga menghindari pembacaan eksternal ke AI Core melalui antarmuka bus setiap saat. Frekuensi akses data pada bus berkurang serta risiko kongesti data pada bus, sehingga meminimalkan disipasi daya dan meningkatkan kinerja.
- Buffer Keluaran: Digunakan untuk menyimpan hasil antara setiap lapisan kalkulasi dalam jaringan saraf tiruan, agar data dapat diperoleh dengan mudah saat memasuki lapisan berikutnya. Di sisi lain, bandwidth pembacaan data melalui bus rendah dan penundaannya besar, sehingga efisiensi kalkulasi dapat ditingkatkan secara signifikan melalui buffer keluaran.
- Register: Semua jenis sumber daya register di AI Core terutama digunakan oleh unit skalar.

(b) Jalur data mengacu pada jalur aliran data di AI Core ketika AI Core menyelesaikan tugas komputasi.

Jalur data Arsitektur Da Vinci dicirikan oleh banyak masukan dan satu keluaran, yang terutama disebabkan oleh banyaknya masukan dalam proses komputasi jaringan saraf tiruan. Masukan paralel dapat digunakan untuk meningkatkan efisiensi aliran data masuk. Sebaliknya, hanya matriks karakteristik keluaran yang dihasilkan setelah pemrosesan beberapa masukan, dan tipe datanya relatif tunggal. Oleh karena itu, jalur data satu keluaran dapat menghemat sumber daya perangkat keras chip.

3. Unit Kontrol

Unit kontrol terdiri dari modul kontrol sistem, cache perintah, antrean pemrosesan perintah skalar, modul transmisi perintah, antrean eksekusi matriks, dan modul sinkronisasi peristiwa, seperti yang ditunjukkan pada Gambar 6.5.



Gambar 6.5 Arsitektur Da Vinci—unit kontrol

(a) Modul Kontrol Sistem: Modul ini mengontrol proses eksekusi blok tugas (granularitas tugas komputasi minimum dalam AI Core). Setelah blok tugas dieksekusi, modul kontrol sistem akan menghentikan pemrosesan dan mendeklarasikan statusnya. Jika terjadi kesalahan dalam proses eksekusi, status kesalahan akan dilaporkan ke Penjadwal Tugas.

(b) Cache Perintah: Dalam proses eksekusi perintah, perintah-perintah selanjutnya dapat diambil terlebih dahulu, dan beberapa perintah dapat dibaca ke dalam cache sekaligus untuk meningkatkan efisiensi eksekusi perintah.

(c) Antrean Pemrosesan Perintah Skalar: Setelah perintah didekodekan, perintah tersebut akan diimpor ke antrean skalar untuk melakukan dekode alamat dan kontrol komputasi. Perintah-perintah tersebut meliputi perintah komputasi matriks, perintah komputasi vektor, dan perintah konversi memori.

(d) Modul Transmisi Perintah: Modul ini membaca alamat perintah dan parameter dekode yang dikonfigurasi dalam antrean perintah skalar, lalu mengirimkannya ke antrean eksekusi perintah yang sesuai berdasarkan jenis perintahnya. Perintah skalar tersebut berada dalam antrean pemrosesan perintah skalar untuk eksekusi selanjutnya.

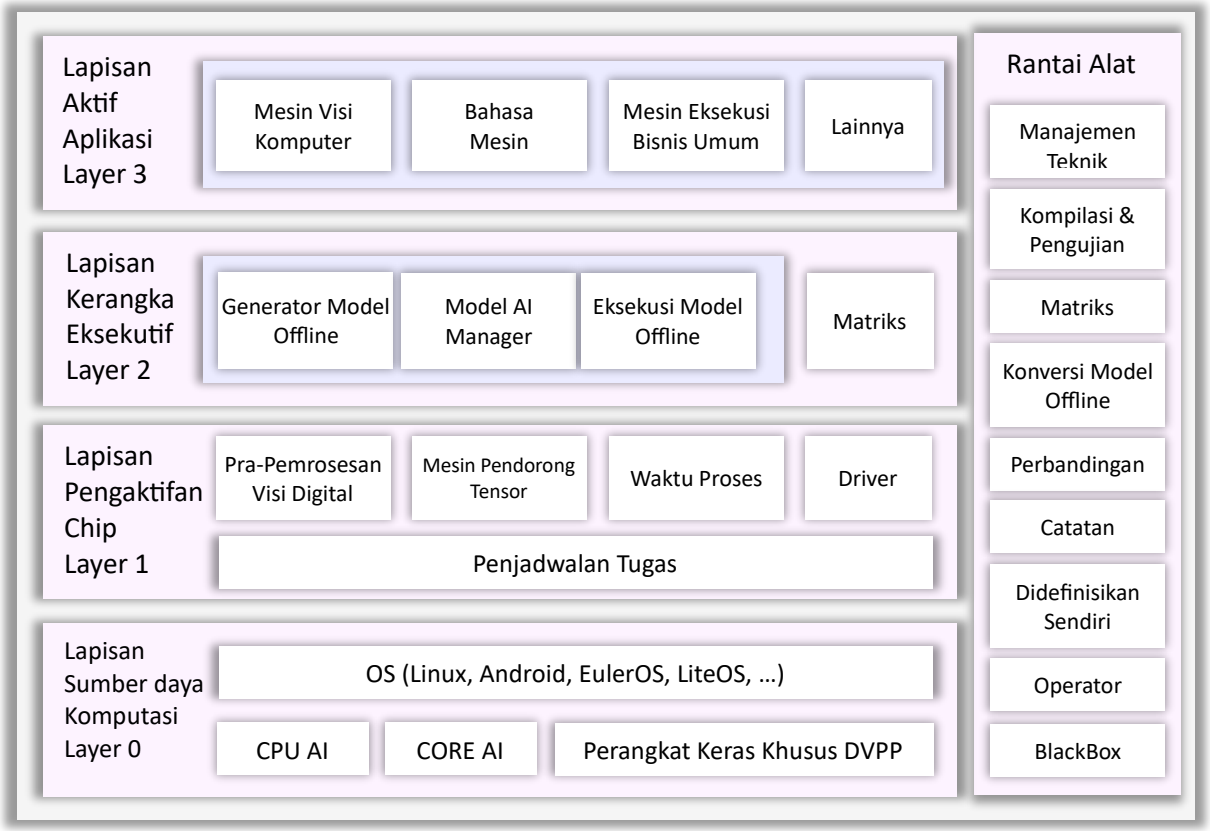
(e) Antrean Eksekusi Perintah: Modul ini terdiri dari antrean matriks, antrean vektor, dan antrean konversi memori. Perintah yang berbeda memasuki antrean yang berbeda, dan perintah dalam antrean dieksekusi sesuai urutan entri.

(f) Modul Sinkronisasi Peristiwa: Modul ini mengontrol status eksekusi setiap jalur perintah setiap saat, menganalisis dependensi dari berbagai jalur perintah, untuk menyesuaikan ketergantungan data dan sinkronisasi antar jalur perintah.

6.2 ARSITEKTUR PERANGKAT LUNAK PROSESOR AI ASCEND

Arsitektur Logika Perangkat Lunak Prosesor AI Ascend

Tumpukan perangkat lunak prosesor AI Ascend terutama dibagi menjadi empat tingkat dan satu rantai alat bantu. Keempat lapisan tersebut adalah lapisan pengaktifan aplikasi L3, lapisan kerangka kerja eksekusi L2, lapisan pengaktifan chip L1, dan lapisan sumber daya komputasi L0. Rantai alat ini terutama menyediakan kapabilitas bantu seperti manajemen rekayasa, kompilasi dan debugging, matriks, log, dan pembuatan profil. Komponen utama tumpukan perangkat lunak saling bergantung dalam fungsi, aliran data pembawa, aliran kalkulasi, dan aliran kontrol, seperti yang ditunjukkan pada Gambar 6.6.



Gambar 6.6 Arsitektur logika perangkat lunak prosesor AI Ascend

1. Lapisan Pengaktifan Aplikasi L3

Lapisan pengaktifan aplikasi L3 adalah enkapsulasi tingkat aplikasi, yang menyediakan berbagai algoritma pemrosesan untuk aplikasi tertentu, selain menyediakan mesin komputasi dan pemrosesan untuk berbagai bidang. Lapisan ini juga dapat secara langsung menggunakan kapabilitas penjadwalan kerangka kerja yang disediakan oleh lapisan kerangka kerja eksekusi L2, dan menghasilkan jaringan saraf tiruan yang sesuai melalui kerangka kerja umum untuk mewujudkan fungsi mesin tertentu. Lapisan pengaktifan aplikasi L3 mencakup mesin visi komputer, mesin bahasa, mesin eksekusi bisnis umum, dll.

(a) Mesin Visi Komputer: Mesin ini menyediakan enkapsulasi beberapa algoritma pemrosesan video atau gambar, yang dikhususkan untuk menangani algoritma dan aplikasi di bidang visi komputer.

(b) Mesin Bahasa: Menyediakan enkapsulasi algoritma pemrosesan dasar untuk suara, teks, dan data lainnya, yang memungkinkan pemrosesan bahasa dilakukan sesuai dengan skenario aplikasi tertentu.

(c) Mesin Eksekusi Bisnis Umum: Menyediakan kemampuan penalaran jaringan saraf tiruan umum.

2. Lapisan Kerangka Kerja Eksekutif L2

Lapisan kerangka kerja eksekutif L2 merupakan enkapsulasi kemampuan pemanggilan kerangka kerja dan kemampuan pembuatan model luring. Setelah lapisan pengaktifan aplikasi L3 mengembangkan algoritma aplikasi dan mengenkapsulasinya ke dalam sebuah mesin,

pemanggilan (seperti Caffe atau Tensorflow) yang sesuai untuk kerangka kerja pembelajaran mendalam akan dilakukan sesuai dengan karakteristik algoritma yang relevan, kemudian jaringan saraf tiruan dengan fungsi yang sesuai diperoleh, dan model luring (OM) dihasilkan oleh Pengelola Kerangka Kerja.

Lapisan kerangka kerja eksekutif L2 berisi pengelola kerangka kerja dan koreografer proses.

(a) Pengelola Kerangka Kerja berisi Generator Model Luring (OMG), Pelaksana Model Luring (OME), dan Antarmuka Penalaran Model Luring, yang mendukung pembuatan, pemuatan, pembongkaran, dan penalaran model. Kerangka kerja daring umumnya menggunakan kerangka kerja sumber terbuka pembelajaran mendalam arus utama (seperti Caffe, Tensorflow, dll.) untuk mempercepat komputasi pada prosesor Ascend AI melalui transformasi dan pemuatan model luring.

Meskipun kerangka kerja luring mengacu pada prosesor Ascend AI, lapisan kerangka kerja eksekutif L2 menyediakan kemampuan pembangkitan luring dan eksekutif jaringan saraf tiruan, yang dapat dipisahkan dari kerangka kerja sumber terbuka pembelajaran mendalam (seperti Caffe, TensorFlow, dll.) untuk memungkinkan model luring memiliki kemampuan yang sama (terutama kemampuan penalaran).

- Generator Model Luring bertanggung jawab untuk mengubah model yang dilatih oleh Caffe atau TensorFlow menjadi model luring yang didukung oleh prosesor Ascend AI.
- Eksekutor Model Luring bertanggung jawab untuk memuat dan membongkar model luring, mengonversi berkas model yang berhasil dimuat menjadi urutan instruksi yang dapat dieksekusi pada prosesor Ascend AI, dan menyelesaikan kompilasi program sebelum dieksekusi.

(b) Matriks: Menyediakan platform pengembangan bagi pengembang untuk komputasi pembelajaran mendalam, termasuk sumber daya komputasi, kerangka kerja operasional, dan perangkat pendukung terkait, yang memungkinkan pengembang untuk menulis aplikasi AI yang berjalan pada perangkat keras tertentu dengan mudah dan efisien, yang bertanggung jawab atas pembuatan, pemuatan, dan penjadwalan model. Setelah lapisan kerangka kerja eksekutif L2 mengubah model asli jaringan saraf tiruan menjadi model luring yang dapat berjalan pada prosesor AI Ascend, Pelaksana Model Luring mengirimkan model luring tersebut ke lapisan pengaktifan chip L1 untuk alokasi tugas.

3. Lapisan Pengaktif Chip L1

Lapisan pengaktif chip L1 merupakan jembatan antara model offline dan prosesor Ascend AI. Setelah menerima model offline yang dihasilkan oleh lapisan kerangka kerja eksekusi L2, lapisan pengaktif chip L1 akan menyediakan fungsi akselerasi untuk kalkulasi model offline melalui Pustaka Akselerasi berdasarkan berbagai tugas komputasi. Lapisan pengaktif chip L1 merupakan lapisan yang paling dekat dengan sumber daya komputasi yang mendasarinya, bertanggung jawab untuk mengirimkan tugas tingkat operator ke perangkat

keras. Lapisan ini terutama terdiri dari Digital Vision Pre-Processing (DVPP), Tensor Boost Engine (TBE), Runtime, Driver, dan Task Scheduler.

Pada lapisan pengaktif chip L1, TBE chip bertindak sebagai inti, yang mendukung akselerasi model online dan offline, termasuk pustaka akselerasi operator standar dan kapabilitas operator khusus. Mesin akselerasi tensor mencakup pustaka akselerasi operator standar, dan operator ini memiliki kinerja yang baik setelah optimasi. Operator berinteraksi dengan Runtime pada lapisan atas pustaka akselerator operator dalam proses eksekutif. Pada saat yang sama, Runtime berkomunikasi dengan lapisan kerangka kerja eksekutif L2, menyediakan antarmuka pustaka akselerasi operator standar untuk melakukan pemanggilan, sehingga model jaringan tertentu dapat menemukan operator yang dioptimalkan, dapat dieksekusi, dan dipercepat untuk implementasi fungsi yang optimal.

Jika tidak ada operator yang dibutuhkan oleh lapisan kerangka kerja eksekutif L2 dalam pustaka akselerasi operator standar lapisan pengaktifan chip L1, operator kustom baru dapat ditulis oleh TBE untuk memenuhi kebutuhan lapisan kerangka kerja eksekutif L2. Oleh karena itu, TBE menyediakan operator yang berfungsi penuh untuk lapisan kerangka kerja eksekutif L2 dengan menawarkan pustaka operator standar dan kemampuan untuk mengkustomisasi operator. Di bawah TBE terdapat Penjadwal Tugas. Setelah fungsi kernel komputasi tertentu dihasilkan berdasarkan operator yang sesuai, Penjadwal Tugas memproses dan mendistribusikan fungsi kernel komputasi yang sesuai ke CPU AI atau Inti AI sesuai dengan jenis tugas tertentu, mengaktifkan perangkat keras melalui driver. Penjadwal Tugas sendiri berjalan pada inti CPU khusus. Modul Pra-Pemrosesan Visi Digital (DVPP) adalah kapsul multifungsi untuk bidang gambar dan video. Dalam kasus pra-pemrosesan gambar atau video umum, modul menyediakan berbagai kemampuan pra-pemrosesan data kepada lapisan atas dengan menggunakan perangkat keras khusus di lapisan bawah.

4. Lapisan Sumber Daya Komputasi L0

Lapisan sumber daya komputasi L0 adalah fondasi daya komputasi perangkat keras untuk prosesor Ascend AI, yang menyediakan sumber daya komputasi dan menjalankan tugas komputasi tertentu. Setelah lapisan pengaktifan chip L1 menyelesaikan distribusi tugas yang berkaitan dengan operator, eksekusi tugas komputasi tertentu dimulai oleh lapisan sumber daya komputasi L0. Lapisan sumber daya komputasi L0 terdiri dari sistem operasi, CPU AI, AI Core, dan modul perangkat keras khusus DVPP.

AI Core adalah inti komputasi prosesor Ascend AI, yang utamanya menyelesaikan kalkulasi korelasi matriks jaringan saraf tiruan, sementara CPU AI melakukan kalkulasi umum operator kontrol, skalar, dan vektor. Jika data masukan perlu dipra-pemrosesan, modul perangkat keras khusus DVPP akan diaktifkan untuk melakukan pra-pemrosesan data gambar dan video, menyediakan format data kepada AI Core untuk memenuhi kebutuhan komputasi dalam skenario tertentu.

AI Core terutama bertanggung jawab atas tugas komputasi besar, sementara AI CPU terutama bertanggung jawab atas komputasi kompleks dan kontrol eksekutif, sementara perangkat keras DVPP bertanggung jawab atas pra-pemrosesan data. Peran sistem operasi adalah untuk memastikan ketiganya saling mendukung dan membentuk sistem perangkat

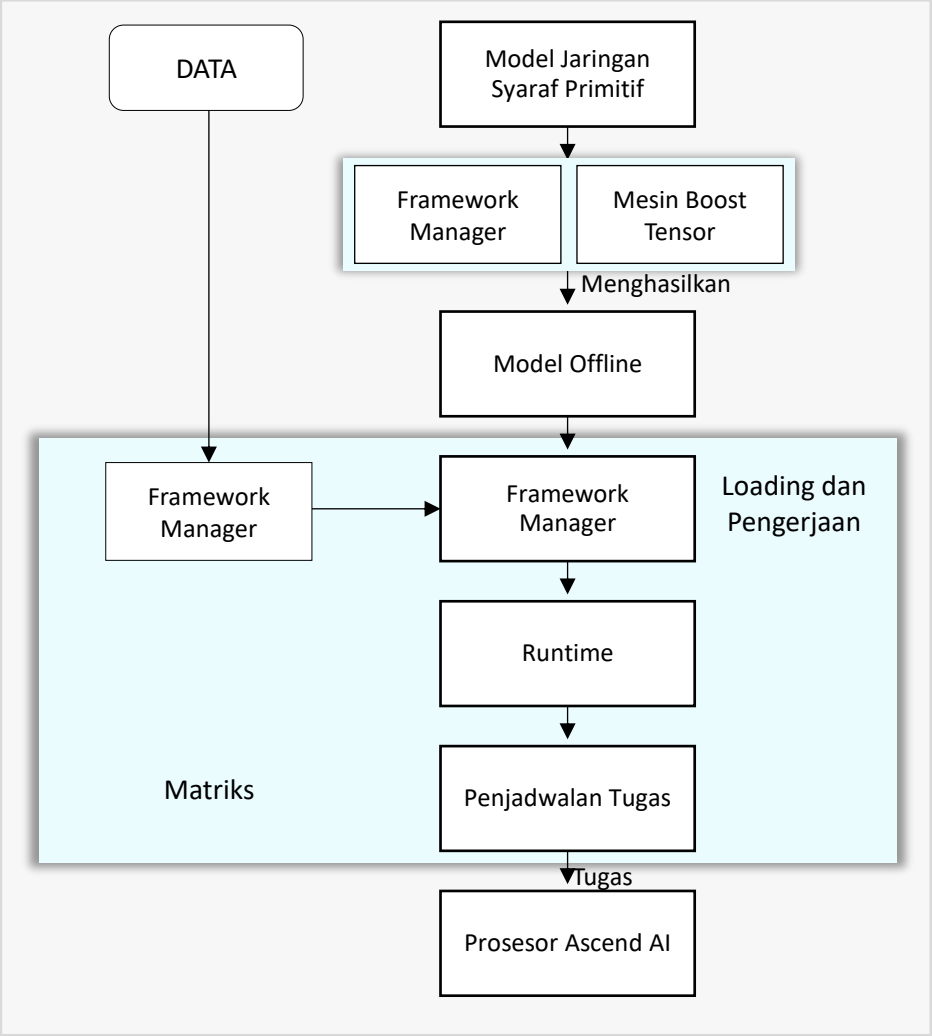
keras yang sempurna, yang memberikan jaminan eksekutif untuk komputasi jaringan saraf dalam prosesor Ascend AI.

5. Rantai Alat

Rantai Alat, yang dirancang untuk kenyamanan programmer, merupakan seperangkat platform alat yang mendukung prosesor Ascend AI. Platform ini menyediakan dukungan untuk pengembangan, debugging, transplantasi jaringan, optimasi, dan analisis operator khusus. Selain itu, serangkaian layanan pemrograman desktop disediakan dalam antarmuka pemrograman berorientasi programmer, yang menurunkan hambatan masuk untuk pengembangan aplikasi terkait jaringan saraf dalam.

Rantai alat terdiri dari manajemen rekayasa, kompilasi dan pengujian, matriks, konversi model offline, perbandingan, log, pembuatan profil, operator khusus, dll. Oleh karena itu, rantai alat menyediakan layanan multi-level dan multifungsi yang nyaman untuk pengembangan dan implementasi aplikasi pada platform ini.

Alur Perangkat Lunak Jaringan Saraf Ascend AI Processor



Gambar 6.7 Alur perangkat lunak jaringan saraf tiruan prosesor Ascend AI

Alur perangkat lunak jaringan saraf Ascend AI Processor merupakan jembatan antara kerangka kerja pembelajaran mendalam dan prosesor Ascend AI, yang menyediakan jalan pintas bagi jaringan saraf untuk bertransformasi dari model asli ke representasi grafik komputasi menengah, dan kemudian ke model luring independen. Alur perangkat lunak jaringan saraf Ascend AI Processor terutama digunakan untuk menghasilkan, memuat, dan mengeksekusi model luring aplikasi jaringan saraf. Alur perangkat lunak jaringan saraf Ascend AI Processor mengumpulkan modul-modul fungsional seperti Matriks, modul DVPP, Tensor Boost Engine, Framework, Runtime, dan Task Scheduler, sehingga membentuk klaster fungsional yang lengkap.

Alur perangkat lunak jaringan saraf Ascend AI Processor ditunjukkan pada Gambar 6.7.

1. Matriks: Bertanggung jawab atas pendaratan dan implementasi jaringan saraf pada prosesor Ascend AI, mengoordinasikan seluruh proses efektif jaringan saraf, serta mengatur proses pemuatan dan eksekusi model luring.
2. Modul DVPP: Melakukan pemrosesan dan modifikasi data sebelum input untuk memenuhi kebutuhan format komputasi.
3. Tensor Boost Engine: Sebagai gudang operator jaringan neural, modul ini menyediakan aliran operator komputasi canggih yang stabil untuk model jaringan neural.
4. Kerangka Kerja: Kerangka kerja ini membangun model jaringan saraf tiruan asli ke dalam bentuk yang didukung oleh prosesor Ascend AI, dan model tersebut terintegrasi dengan prosesor Ascend AI untuk memandu jaringan saraf tiruan agar berjalan dan memaksimalkan kinerjanya.
5. Waktu Proses: Menyediakan berbagai saluran manajemen sumber daya untuk distribusi dan alokasi tugas jaringan saraf tiruan.
6. Penjadwal Tugas: Sebagai penggerak tugas untuk eksekusi perangkat keras, kerangka kerja ini menyediakan tugas target spesifik untuk prosesor Ascend AI; Waktu Proses dan Penjadwal Tugas berinteraksi bersama untuk membentuk sistem aliran tugas jaringan saraf tiruan ke sumber daya perangkat keras, pemantauan waktu nyata, dan pendistribusian berbagai jenis tugas eksekutif secara efektif.

Seluruh perangkat lunak jaringan saraf tiruan menyediakan proses eksekutif yang berfungsi penuh untuk prosesor Ascend AI yang menggabungkan perangkat keras dan perangkat lunak, membantu pengembangan aplikasi AI terkait. Modul fungsional yang terkait dengan jaringan saraf tiruan akan diperkenalkan secara terpisah sebagai berikut.

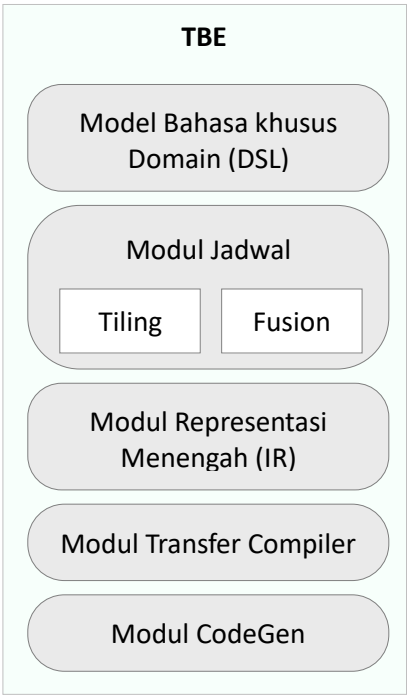
Pengantar Modul Fungsional Alur Perangkat Lunak Ascend AI Processor

1. Tensor Boost Engine

Dalam konstruksi jaringan neural, operator membentuk struktur jaringan dengan fungsi aplikasi yang berbeda. Sebagai gudang operator, Tensor Boost Engine (TBE) menyediakan kemampuan pengembangan operator untuk jaringan neural berbasis prosesor Ascend AI. Operator yang ditulis dalam bahasa TBE digunakan untuk membangun berbagai model jaringan neural. Pada saat yang sama, TBE juga menyediakan kemampuan bagi operator untuk melakukan wrap calling. TBE berisi pustaka operator standar TBE yang

dioptimalkan untuk jaringan neural, yang dapat langsung dimanfaatkan oleh pengembang untuk mencapai komputasi jaringan neural berkinerja tinggi. Selain itu, TBE juga menyediakan kemampuan fusi operator TBE, yang membuka jalur unik untuk optimasi jaringan neural.

TBE menyediakan kemampuan untuk mengembangkan operator kustom berdasarkan Tensor Virtual Machine (TVM). Pengguna dapat menyelesaikan pengembangan operator jaringan neural yang sesuai melalui bahasa TBE dan antarmuka pengembangan pemrograman operator kustom. TBE mencakup Modul Bahasa Domain Spesifik (DSL), Modul Jadwal, Modul Representasi Menengah (IR), Modul Transfer Kompiler, dan Modul CodeGen. Struktur TBE ditunjukkan pada Gambar 6.8.



Gambar 6.8 Struktur TBE

Pengembangan operator TBE dibagi menjadi penulisan logika komputasi dan pengembangan penjadwalan. Modul DSL menyediakan antarmuka pemrograman logika komputasi operator, yang secara langsung didasarkan pada proses perhitungan dan penjadwalan operator berdasarkan bahasa domain spesifik. Proses perhitungan operator menjelaskan metode dan langkah perhitungan operator, sedangkan proses penjadwalan menjelaskan rencana segmentasi data dan aliran data. Setiap perhitungan operator diproses berdasarkan bentuk data tetap, yang memerlukan segmentasi bentuk data terlebih dahulu.

Operator yang dieksekusi pada unit komputasi yang berbeda dalam prosesor Ascend AI, seperti Unit Matriks, Unit Vektor, dan operator yang dieksekusi pada CPU AI, memiliki persyaratan yang berbeda untuk bentuk data input. Setelah proses implementasi dasar operator didefinisikan, Sub Modul Tiling dari Modul Penjadwalan perlu dijalankan, melakukan segmentasi data dalam operator sesuai deskripsi penjadwalan, dan menentukan proses penanganan data untuk memastikan eksekusi optimal pada perangkat keras. Selain

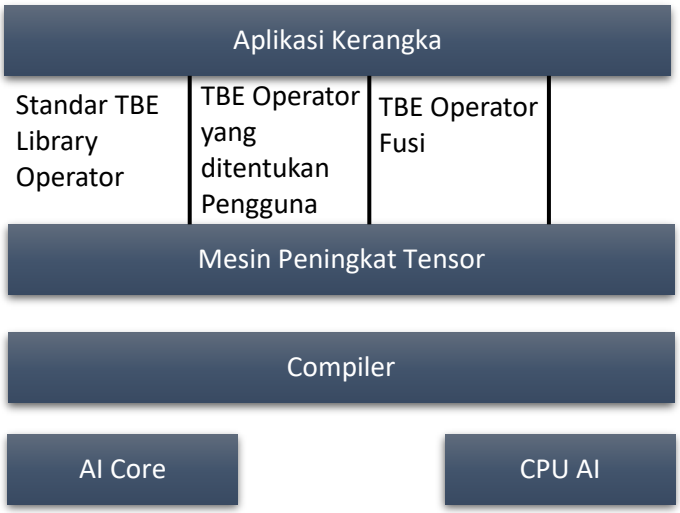
segmentasi bentuk data, kemampuan fusi dan optimasi operator TBE juga disediakan oleh Sub Modul Fusion dari Modul Penjadwalan.

Setelah operator ditulis, representasi antara perlu dihasilkan untuk optimasi lebih lanjut, dan Modul IR menghasilkan representasi antara melalui format IR yang mirip dengan TVM. Setelah representasi antara dihasilkan, modul perlu dikompilasi dan dioptimalkan untuk berbagai skenario aplikasi, dengan berbagai metode optimasi seperti Double Buffer, Sinkronisasi Pipeline, Manajemen Alokasi Memori, Pemetaan Perintah, Unit Kubus Adaptor Tiling, dll.

Setelah operator diproses oleh Modul Transfer Kompiler, berkas sementara berisi kode mirip C dihasilkan oleh Modul CodeGen. Berkas kode sementara tersebut dapat ditransfer ke berkas implementasi operator oleh kompiler, yang dapat langsung dimuat dan dieksekusi melalui Offline Model Executor. Singkatnya, operator yang ditentukan pengguna secara lengkap melengkapi seluruh proses pengembangan melalui sub-modul TBE. Setelah prototipe operator dibentuk oleh logika kalkulasi operator dan deskripsi penjadwalan yang disediakan oleh modul bahasa khusus domain, modul penjadwalan melakukan segmentasi data dan fusi operator, yang kemudian memasuki modul representasi antara untuk menghasilkan representasi antara dari operator tersebut.

Modul transfer kompiler menggunakan representasi antara untuk mengoptimalkan alokasi memori. Terakhir, modul pembangkit kode menghasilkan kode mirip C untuk dikompilasi langsung oleh kompiler. Dalam proses pendefinisian operator, TBE tidak hanya menyelesaikan kompilasi operator, tetapi juga melakukan optimasi terkait, yang meningkatkan kinerja operator.

Tiga skenario aplikasi TBE ditunjukkan pada Gambar 6.9.



Gambar 6.9 Tiga Skenario Aplikasi TBE

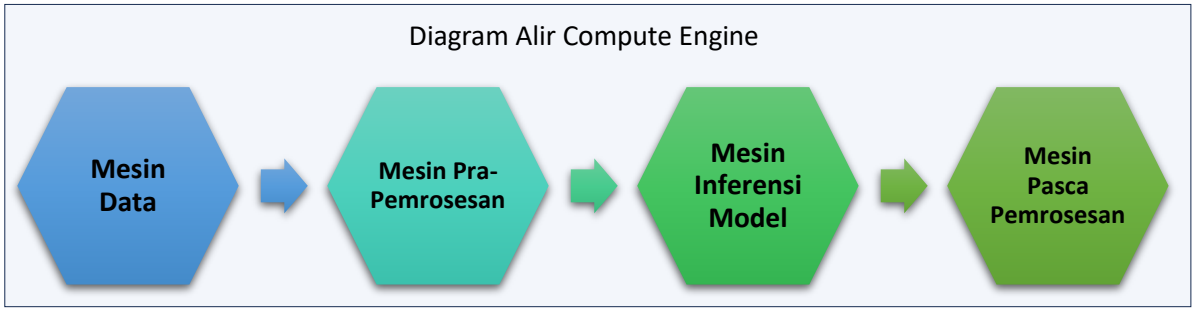
(a) Umumnya, model jaringan saraf tiruan yang diimplementasikan oleh operator standar dalam kerangka kerja pembelajaran mendalam telah dilatih oleh GPU atau jenis

prosesor jaringan saraf tiruan lainnya. Jika model jaringan saraf tiruan terus berjalan pada prosesor Ascend AI, kinerjanya diharapkan dapat dimaksimalkan tanpa mengubah kode aslinya. Oleh karena itu, TBE menyediakan satu set lengkap pustaka penguat operator TBE. Fungsi operator dalam pustaka tersebut menjaga korespondensi satu-ke-satu dengan operator standar umum dalam jaringan saraf tiruan, dan tumpukan perangkat lunak menyediakan antarmuka pemrograman untuk memanggil operator, yang meningkatkan berbagai kerangka kerja atau aplikasi dalam pembelajaran mendalam tingkat atas, tanpa perlu mengembangkan kode adaptasi yang mendasari prosesor Ascend AI.

(b) Jika operator baru muncul dalam konstruksi model jaringan saraf tiruan, pustaka operator standar yang disediakan oleh TBE tidak akan memenuhi persyaratan pengembangan. Saat ini, perlu untuk mengembangkan operator khusus melalui bahasa TBE, yang mirip dengan CUDA C++ pada GPU. Operator yang lebih serbaguna dapat direalisasikan dan berbagai model jaringan dapat diprogram secara fleksibel. Operator yang telah selesai akan ditransfer ke kompiler untuk dikompilasi, dan eksekusi akhir memanfaatkan kemampuan akselerasi chip pada AI Core atau AI CPU.

(c) Dalam skenario yang sesuai, kemampuan fusi operator yang disediakan oleh TBE dapat meningkatkan kinerja operator, sehingga operator jaringan saraf tiruan dapat melakukan fusi cache multi-level berdasarkan buffer dengan level yang berbeda. Prosesor AI Ascend dapat meningkatkan tingkat pemanfaatan sumber daya secara signifikan saat melakukan operator fusi.

Singkatnya, berkat kemampuan pengembangan operator, pemanggilan operator standar, dan optimasi fusi operator yang disediakan oleh TBE, prosesor AI Ascend dapat memenuhi kebutuhan diversifikasi fungsional dalam aplikasi jaringan saraf tiruan yang sebenarnya. Selain itu, cara konstruksi jaringan akan lebih fleksibel dan kemampuan optimasi fusi akan menghasilkan kinerja yang lebih baik.



Gambar 6.10 Diagram alir mesin komputasi dalam aplikasi jaringan saraf tiruan dalam

2. Matriks

(a) Pengantar Singkat Fungsi Matriks.

Prosesor AI Ascend membagi tingkat eksekusi jaringan, dengan mempertimbangkan eksekusi fungsi-fungsi spesifik sebagai unit eksekusi dasar Mesin Komputasi. Setiap mesin komputasi melengkapi fungsi operasi dasar dari pengaturan data dalam proses, seperti

klasifikasi gambar dan sebagainya. Mesin Komputasi dikustomisasi oleh pengembang untuk melengkapi fungsi-fungsi spesifik yang dibutuhkan.

Melalui pemanggilan terpadu Matriks, seluruh aplikasi jaringan saraf tiruan dalam umumnya mencakup empat mesin: Mesin Data, Mesin Pra-pemrosesan, Mesin Inferensi Model, dan Mesin Pasca-pemrosesan, seperti yang ditunjukkan pada Gambar 6.10.

- Mesin Data terutama menyiapkan set data yang dibutuhkan oleh jaringan saraf tiruan (seperti set data MNIST) dan memproses data terkait (seperti pemfilteran gambar, dll.) sebagai sumber data untuk mesin komputasi selanjutnya.
- Umumnya, data media masukan perlu melalui pra-pemrosesan format untuk memenuhi persyaratan komputasi prosesor Ascend AI. Mesin Pra-pemrosesan terutama digunakan untuk melakukan pra-pemrosesan data media, menyelesaikan pengodean dan dekodean gambar dan video, konversi format, dan operasi lainnya, dan setiap modul fungsi pra-pemrosesan visi digital perlu dipanggil melalui Matrix secara seragam.
- Mesin Inferensi Model digunakan dalam inferensi aliran data jaringan saraf tiruan. Mesin ini terutama menggunakan model yang dimuat dan aliran data masukan untuk menyelesaikan algoritma penerusan jaringan saraf tiruan.
- Setelah Mesin Penalaran Model mengeluarkan hasilnya, Mesin Pasca-pemrosesan melakukan pemrosesan selanjutnya pada data keluaran, seperti pembingkaian dan pelabelan pengenalan gambar.

Gambar 6.10 menunjukkan diagram alir umum Compute Engine. Setiap simpul pemrosesan data spesifik dalam diagram alir mesin komputasi merupakan mesin komputasi. Ketika data mengalir melalui setiap mesin sesuai dengan jalur yang telah diatur, pemrosesan dan perhitungan terkait masing-masing dilakukan, dan hasil yang dibutuhkan akhirnya dikeluarkan. Keluaran akhir dari keseluruhan diagram alir merupakan hasil yang sesuai dari keluaran kalkulasi jaringan saraf tiruan. Koneksi antara dua simpul mesin komputasi yang berdekatan dibuat melalui berkas konfigurasi dalam diagram alir mesin komputasi, dan aliran data aktual antar simpul akan mengalir sesuai dengan mode koneksi simpul dari model jaringan tertentu. Setelah konfigurasi atribut simpul selesai, seluruh proses mesin komputasi dimulai dengan menuangkan data ke simpul awal diagram alir mesin komputasi.

Matrix berjalan di atas lapisan pengaktifan chip L1 dan di bawah lapisan pengaktifan aplikasi L3, menyediakan antarmuka perantara standar terpadu untuk berbagai sistem operasi (Linux, Android, dll.), yang bertanggung jawab atas pembentukan, penghapusan, dan daur ulang keseluruhan diagram alir mesin komputasi. Dalam proses pembentukan diagram alir mesin komputasi, Matrix menyelesaikan pembentukan diagram alir mesin komputasi sesuai dengan berkas konfigurasi mesin komputasi. Sebelum eksekusi, Matrix menyediakan data masukan. Jika data masukan seperti video, gambar, dan format lain tidak memenuhi persyaratan pemrosesan, antarmuka pemrograman yang sesuai dapat digunakan untuk memanggil modul pra-pemrosesan visi digital untuk pra-pemrosesan data.

Jika data memenuhi persyaratan pemrosesan, Offline Model Executor akan langsung dipanggil melalui antarmuka untuk perhitungan penalaran. Dalam proses eksekusi, Matrix

memiliki fungsi penjadwalan multi-simpul dan manajemen multi-proses, yang bertanggung jawab atas pengoperasian proses komputasi di sisi perangkat, menjaga proses komputasi, dan mengumpulkan informasi eksekusi yang relevan. Setelah model dieksekusi, Matrix akan menyediakan aplikasi di host dengan fungsi untuk memperoleh hasil keluaran.

(b) Skenario Aplikasi Matrix.

Karena prosesor Ascend AI ditargetkan untuk kebutuhan bisnis yang berbeda, platform perangkat keras yang berbeda dapat dibangun dengan spesifisitas yang berbeda pula. Tergantung pada kolaborasi perangkat keras dan sisi host tertentu, aplikasi Matrix digunakan secara berbeda dalam skenario umum seperti Accelerator dan Atlas 200 DK.

- Skenario Aplikasi Bentuk Akselerator

  Akselerator PCIe berbasis prosesor Ascend AI terutama berorientasi pada skenario pusat data dan server edge, seperti yang ditunjukkan pada Gambar 6.11. Akselerator PCIe mendukung berbagai akurasi data, yang telah meningkatkan kinerjanya dibandingkan dengan akselerator serupa lainnya, memberikan daya komputasi yang lebih besar untuk komputasi jaringan neural. Dalam skenario akselerator, sebuah host diperlukan untuk terhubung dengan akselerator. Host dapat mendukung berbagai server kartu plug-in PCIe dan komputer pribadi, melakukan pemrosesan yang sesuai dengan memanfaatkan daya komputasi jaringan neural akselerator.

Gambar 6.11 Akselerator PCIe

Fungsi Matrix dalam skenario akselerator diwujudkan oleh tiga sub-proses: Matrix Agent, Matrix Daemon, dan Matrix Service. Matrix Agent biasanya berjalan di host. Matrix Agent dapat mengontrol dan mengelola Data Engine dan Post-processing Engine, lengkap dengan interaksi data dengan aplikasi host, mengontrol aplikasi, dan berkomunikasi dengan proses pemrosesan di sisi perangkat.

Matrix Daemon berjalan di sisi perangkat. Matrix Daemon dapat menyelesaikan pembentukan proses pada perangkat sesuai dengan berkas konfigurasi. Matrix Daemon bertanggung jawab untuk memulai dan mengelola proses koreografi proses pada perangkat, serta menghentikan proses kalkulasi dan daur ulang sumber daya setelah kalkulasi selesai.

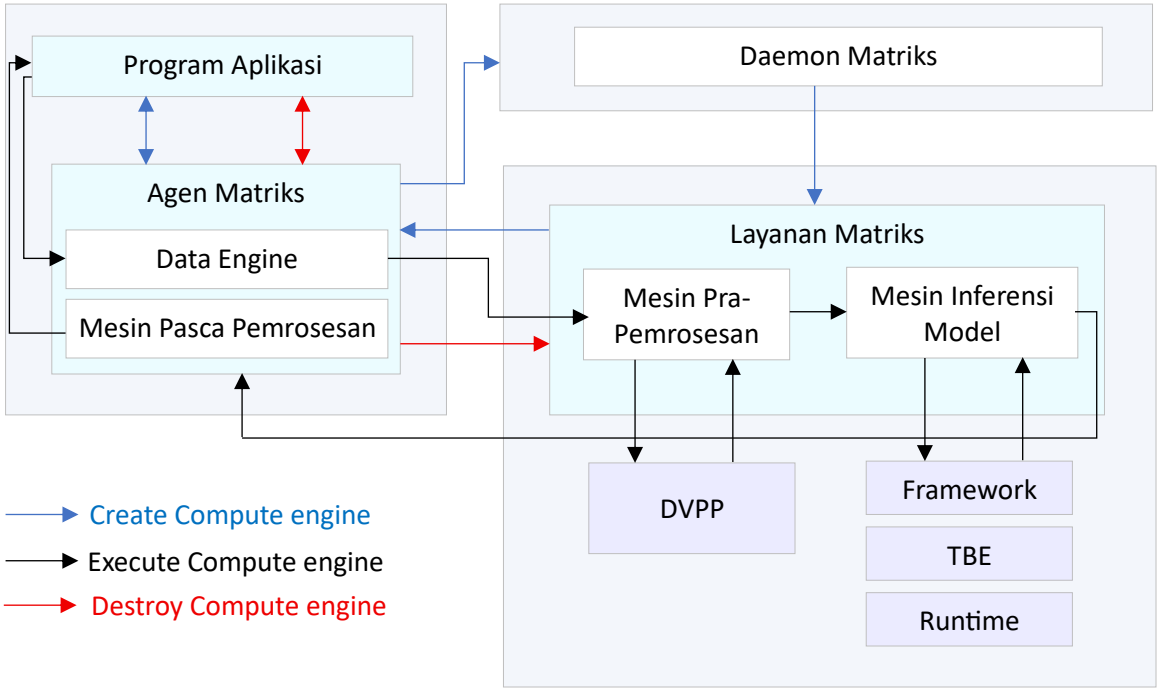
Matrix Service berjalan di sisi perangkat. Matrix Service mengendalikan Pre-processing Engine dan Model Reasoning Engine di sisi perangkat. Matrix Service dapat mengendalikan Pre-processing Engine untuk memanggil antarmuka pemrograman Modul Pra-Pemrosesan Digital Vision guna mewujudkan fungsi pra-pemrosesan data video dan gambar. Ia juga dapat memanggil antarmuka pemrograman manajer model di Offline Model Executor untuk memuat dan menalar model offline. Model offline jaringan saraf tiruan dialar melalui Matrix, dan proses komputasinya ditunjukkan pada Gambar 6.12.

Model offline jaringan saraf tiruan dapat dibagi menjadi tiga langkah berikut.

- ✓ *Langkah 1*: Diagram alir untuk membuat mesin komputasi. Melalui Matrix, proses eksekusi jaringan saraf tiruan diatur menggunakan berbagai mesin komputasi.
- ✓ *Langkah 2*: Diagram alir untuk menjalankan mesin komputasi. Berdasarkan diagram alir mesin komputasi yang telah ditentukan, fungsi jaringan saraf tiruan dihitung dan diimplementasikan.

  Setelah model offline dimuat, aplikasi akan memberi tahu Agen Matrix di sisi host untuk memasukkan data aplikasi. Program aplikasi akan langsung mengirimkan data ke mesin data untuk pemrosesan yang sesuai. Jika data media yang masuk tidak memenuhi persyaratan komputasi prosesor Ascend AI, mesin pra-pemrosesan akan segera memulai dan memanggil antarmuka modul DVPP untuk melakukan pra-pemrosesan data media, seperti pengodean, dekode, penskalaan, dll.

  Setelah pra-pemrosesan, data dikembalikan ke mesin pra-pemrosesan, yang kemudian data tersebut ditransmisikan ke mesin inferensi model. Pada saat yang sama, mesin inferensi model memanggil antarmuka pemrosesan dari manajer model untuk menyelesaikan kalkulasi inferensial dengan menggabungkan data dengan model luring yang telah dimuat. Setelah hasil keluaran diperoleh, mesin inferensi model memanggil antarmuka pengiriman data dari unit koreografi proses untuk mengembalikan hasil inferensial ke mesin pasca-pemrosesan, yang menyelesaikan operasi pasca-pemrosesan data, dan akhirnya mengembalikan data pasca-pemrosesan ke program aplikasi melalui unit koreografer proses. Dengan demikian, diagram alir mesin kalkulasi eksekusi selesai.



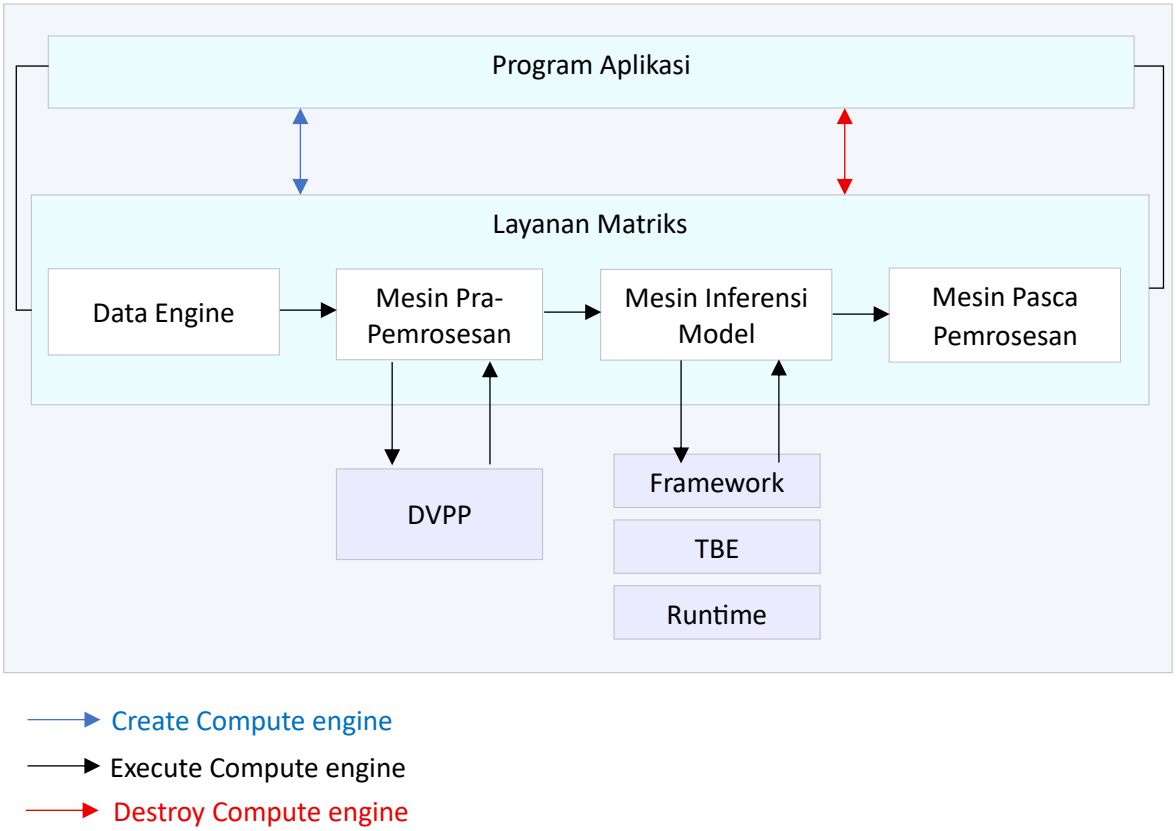
Gambar 6.12 Proses komputasi model offline yang dialiri alasan matriks

- ✓ *Langkah 3*: Diagram alir untuk menghancurkan mesin komputasi. Setelah semua kalkulasi selesai, sumber daya sistem yang digunakan oleh mesin komputasi dilepaskan. Setelah semua data mesin diproses dan dikembalikan, program aplikasi memberi tahu Matrix Agent untuk melepaskan sumber daya perangkat keras komputasi mesin data dan mesin pasca-pemrosesan, sementara Matrix Agent memberi tahu Matrix Service untuk melepaskan sumber daya mesin pra-pemrosesan dan mesin inferensi model. Setelah semua sumber daya dilepaskan, diagram alir mesin komputasi dihancurkan, dan kemudian Matrix Agent memberi tahu aplikasi untuk mengimplementasikan jaringan saraf tiruan berikutnya.

➲ **Skenario Aplikasi Atlas 200 DK**

Skenario Aplikasi Atlas 200 DK mengacu pada skenario Atlas 200 Developer Kit (Atlas 200 DK) yang berbasis prosesor Ascend AI, seperti yang ditunjukkan pada Gambar 6.13.

Gambar 6.13 Kit pengembang Atlas 200 (Atlas 200 DK)



Gambar 6.14 Arsitektur logis Kit Pengembang Atlas 200

Atlas 200 DK membuka fungsi inti prosesor Ascend AI melalui antarmuka periferal pada papan pengembang, yang memfasilitasi kontrol eksternal langsung dan pengembangan chip, memungkinkan kemampuan pemrosesan jaringan saraf prosesor Ascend AI untuk dijalankan dengan mudah dan intuitif. Oleh karena itu, Atlas 200 DK, yang berbasis prosesor Ascend AI, dapat digunakan di berbagai bidang kecerdasan buatan, yang akan menjadi tulang punggung perangkat keras terminal seluler di masa mendatang. Untuk Skenario Aplikasi Atlas 200 DK, fungsi kontrol host juga terdapat pada papan pengembang, dan arsitektur logisnya ditunjukkan pada Gambar 6.14. Sebagai antarmuka fungsi prosesor Ascend AI, Matrix dapat menyelesaikan interaksi data antara diagram alir mesin komputasi dan program aplikasi. Berdasarkan berkas konfigurasi, Matrix menetapkan diagram alir mesin komputasi, yang bertanggung jawab atas penjadwalan, pengendalian, dan pengelolaan proses.

Setelah perhitungan, Matrix juga menghapus diagram alir mesin komputasi dan mendaur ulang sumber daya setelah perhitungan. Dalam proses pra-pemrosesan, Matrix memanggil antarmuka mesin pra-pemrosesan untuk menjalankan fungsi pra-pemrosesan media. Dalam proses inferensi, Matrix juga dapat memanggil antarmuka pemrograman manajer model untuk menjalankan pemuatan dan inferensi model offline. Dalam Skenario Aplikasi Atlas 200 DK, Matrix mengoordinasikan implementasi seluruh diagram alir mesin komputasi tanpa berinteraksi dengan perangkat lain.

3. Penjadwal Tugas

Bersama dengan Runtime, Penjadwal Tugas (TS) membentuk sistem penghubung antara perangkat keras dan perangkat lunak. Selama eksekusi, Penjadwal Tugas mengendalikan tugas-tugas perangkat keras, menyediakan tugas-tugas target spesifik untuk prosesor Ascend AI, menyelesaikan proses penjadwalan tugas dengan Runtime, dan mengirimkan data keluaran kembali ke Runtime yang bertindak sebagai saluran untuk pengiriman tugas dan pengembalian data.

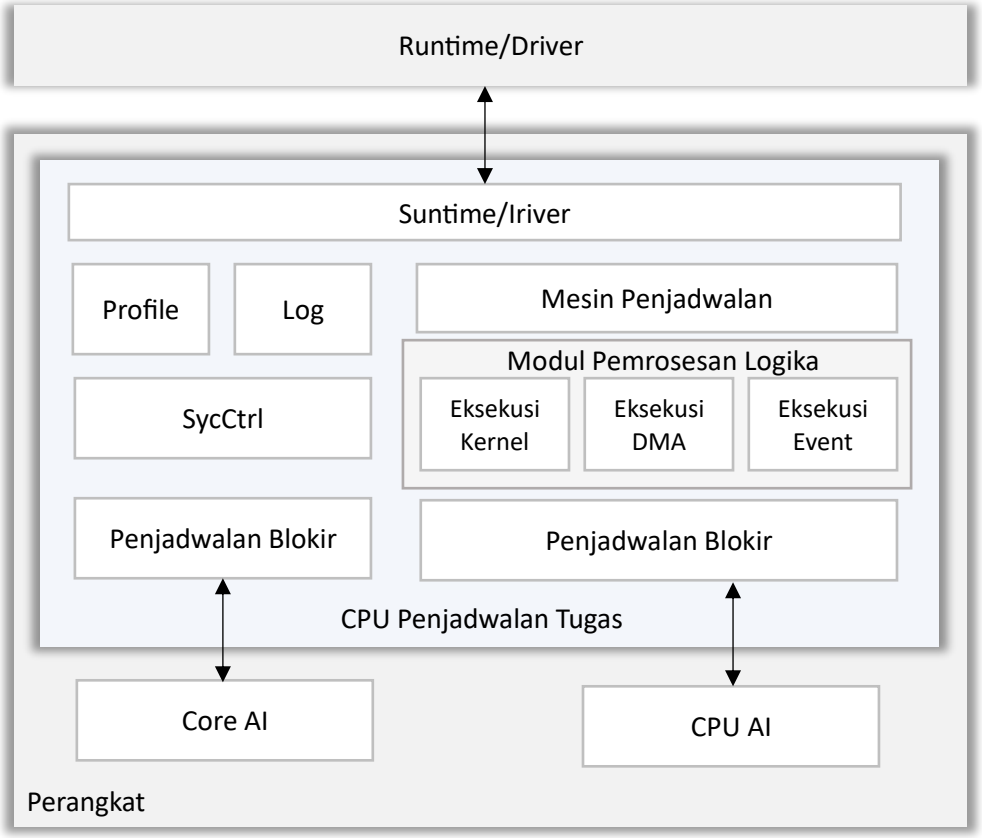
(a) Pengantar Fungsi Penjadwal Tugas

Penjadwal Tugas berjalan pada CPU penjadwal tugas di sisi perangkat dan bertanggung jawab untuk selanjutnya mengirimkan tugas-tugas spesifik yang didistribusikan oleh Runtime ke CPU AI. Penjadwal Tugas juga dapat menetapkan tugas ke AI Core untuk dieksekusi melalui Block Scheduler (BS) perangkat keras, dan mengembalikan hasil eksekusi tugas ke Runtime setelah eksekusi. Secara umum, tugas-tugas utama Penjadwal Tugas meliputi tugas AI Core, tugas AI CPU, Replikasi Memori, Perekaman Peristiwa, Penantian Peristiwa, Pemeliharaan, dan Pembuatan Profil.

Replikasi Memori terutama dilakukan secara asinkron. Perekaman Peristiwa terutama merekam informasi kejadian peristiwa. Jika ada tugas yang menunggu peristiwa tersebut, tugas-tugas ini dapat melepaskan penantian tersebut dan melanjutkan eksekusi setelah perekaman peristiwa selesai, sehingga menghilangkan pemblokiran alur eksekusi yang disebabkan oleh perekaman peristiwa. Event Waiting berarti jika event waiting telah terjadi, tugas waiting akan langsung diselesaikan; jika event waiting belum terjadi, tugas waiting akan ditambahkan ke daftar tunggu, dan pemrosesan semua tugas selanjutnya dalam alur eksekusi tempat event waiting berada akan dihentikan. Ketika event waiting terjadi, pemrosesan event waiting akan dieksekusi.

Setelah task selesai, Maintenance akan melakukan pemeliharaan yang sesuai berdasarkan berbagai parameter task dan memulihkan sumber daya komputasi. Dalam proses eksekusi, dimungkinkan juga untuk merekam dan menganalisis kinerja kalkulasi, yang memerlukan Profiling untuk mengontrol awal dan jeda operasi profiling. Kerangka kerja fungsional Task Scheduler ditunjukkan pada Gambar 6.15. Task Scheduler biasanya terletak di sisi perangkat, dengan fungsinya diselesaikan oleh task scheduling CPU. Task scheduling CPU terdiri dari Interface, Engine, Logic Processing, AI CPU Scheduler, Block Scheduler, SysCtrl, Profile, dan Log. CPU penjadwalan tugas mewujudkan komunikasi dan interaksi antara Runtime dan Driver melalui antarmuka penjadwalan. Tugas ditransmisikan ke mesin penjadwalan tugas melalui hasilnya.

Sebagai badan utama penjadwalan tugas, mesin penjadwalan tugas bertanggung jawab atas proses pengorganisasian tugas, ketergantungan tugas, dan kontrol penjadwalan tugas, serta mengelola proses eksekusi seluruh CPU penjadwalan tugas. Berdasarkan jenis tugas spesifiknya, mesin penjadwalan tugas membagi tugas menjadi tiga jenis: kalkulasi, penyimpanan, dan kontrol, yang didistribusikan ke berbagai modul pemrosesan logika penjadwalan untuk memulai manajemen dan penjadwalan tugas fungsi kernel tertentu, tugas memori, dan dependensi peristiwa antar alur eksekusi.



Gambar 6.15 Kerangka kerja fungsional Penjadwal Tugas

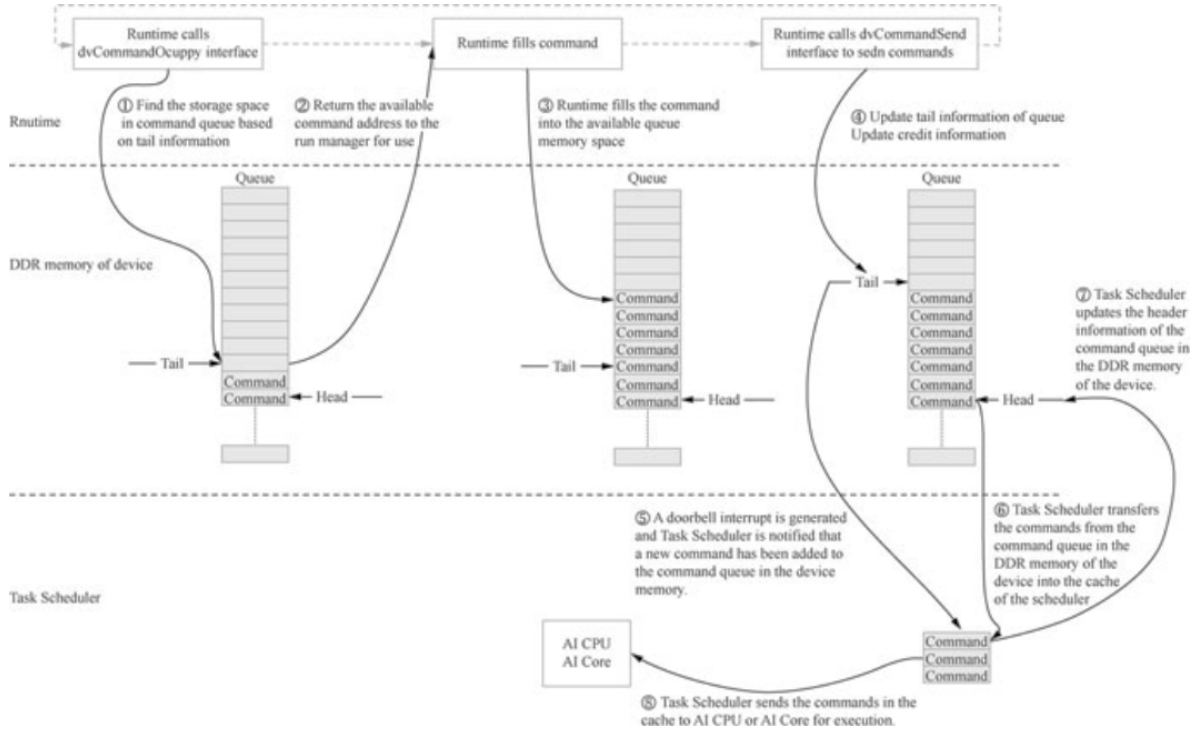
Modul pemrosesan logika dibagi menjadi Kernel Execute, DMA Execute, dan Event Execute. Kernel Execute menjalankan pemrosesan penjadwalan tugas komputasi dan mewujudkan logika penjadwalan tugas pada AI CPU dan AI Core, serta menjadwalkan fungsi kernel tertentu. DMA Execute mengimplementasikan logika penjadwalan tugas penyimpanan dan mengirimkan tugas seperti replikasi memori dan tugas lainnya. Event Execute bertanggung jawab atas logika penjadwalan tugas kontrol sinkron dan pemrosesan logis dependensi peristiwa antaralur eksekusi. Setelah menyelesaikan pemrosesan logika penjadwalan berbagai jenis tugas, proses tersebut langsung diserahkan ke unit kontrol terkait untuk dieksekusi oleh perangkat keras. Untuk eksekusi tugas AI CPU, penjadwal AI CPU dalam CPU penjadwalan tugas melakukan manajemen status dan penjadwalan tugas AI CPU melalui perangkat lunak. Untuk eksekusi tugas AI Core, CPU penjadwalan tugas mendistribusikan tugas yang telah diproses ke AI Core melalui perangkat keras penjadwal blok terpisah, dan kalkulasi spesifik dilakukan oleh AI Core. Hasil perhitungan juga dikembalikan ke CPU penjadwal tugas oleh penjadwal blok.

Dalam proses penjadwalan tugas oleh CPU, SysCtrl mengonfigurasi sistem dan menginisialisasi fungsi chip. Pada saat yang sama, Profil dan Log memantau seluruh proses eksekusi dan mencatat parameter eksekusi utama serta detail eksekusi spesifik. Di akhir seluruh proses eksekusi atau ketika kesalahan dilaporkan, profil kinerja spesifik atau lokasi

kesalahan dapat dilakukan, yang menyediakan dasar untuk evaluasi selanjutnya atas akurasi dan efisiensi implementasi.

(b) Proses Penjadwalan Penjadwal Tugas.

Dalam proses eksekusi model luring jaringan neural, Penjadwal Tugas menerima tugas-tugas spesifik dari Pelaksana Model Luring. Terdapat ketergantungan di antara tugas-tugas ini, yang perlu dihilangkan terlebih dahulu, diikuti oleh penjadwalan tugas dan langkah-langkah lainnya, dan akhirnya didistribusikan ke AI Core atau AI CPU sesuai dengan jenis tugas spesifik untuk menyelesaikan perhitungan atau eksekusi perangkat keras tertentu. Dalam proses penjadwalan tugas, sebuah tugas terdiri dari beberapa perintah (CMD). Penjadwal Tugas dan Runtime berinteraksi satu sama lain untuk menyelesaikan penjadwalan yang tertib untuk seluruh perintah tugas. Runtime berjalan pada CPU host, karena antrean perintah terletak di memori perangkat, dan Penjadwal Tugas mengeluarkan perintah-perintah spesifik. Alur detail proses penjadwalan penjadwal tugas ditunjukkan pada Gambar 6.16.



Gambar 6.16 Kolaborasi Runtime dan Penjadwal Tugas

Pertama, Runtime memanggil antarmuka dvCommandOcuppy driver untuk memasuki antrean perintah, menanyakan ruang memori yang tersedia dalam antrean perintah berdasarkan informasi ekor perintah, dan mengembalikan alamat ruang memori yang tersedia ke Runtime. Setelah menerima alamat tersebut, Runtime mengisi perintah yang telah disiapkan ke dalam ruang memori antrean perintah, dan memanggil antarmuka dvCommandSend driver untuk memperbarui informasi Ekor dan informasi Kredit antrean perintah saat ini. Setelah antrean menerima perintah baru, interupsi bel pintu akan dihasilkan dan Penjadwal Tugas akan diberi tahu bahwa perintah baru telah ditambahkan ke antrean perintah di memori perangkat.

Setelah Penjadwal Tugas menerima pemberitahuan, ia akan memasuki memori perangkat, mentransfer perintah ke dalam cache penjadwal, dan memperbarui informasi header antrean perintah di memori DDR perangkat. Terakhir, Penjadwal Tugas mengirimkan perintah dalam cache ke AI CPU atau AI Core untuk dieksekusi. Mirip dengan arsitektur tumpukan perangkat lunak di sebagian besar akselerator, Runtime, Driver, dan Penjadwal Tugas dalam prosesor Ascend AI bekerja sama erat untuk menyelesaikan tugas secara tertib dan mendistribusikan tugas ke sumber daya perangkat keras yang sesuai untuk dieksekusi. Proses penjadwalan ini menyediakan proses komputasi jaringan saraf tiruan dalam dengan pengiriman tugas yang padat dan tertib, yang memastikan kontinuitas dan efisiensi eksekusi tugas.

4. Runtime

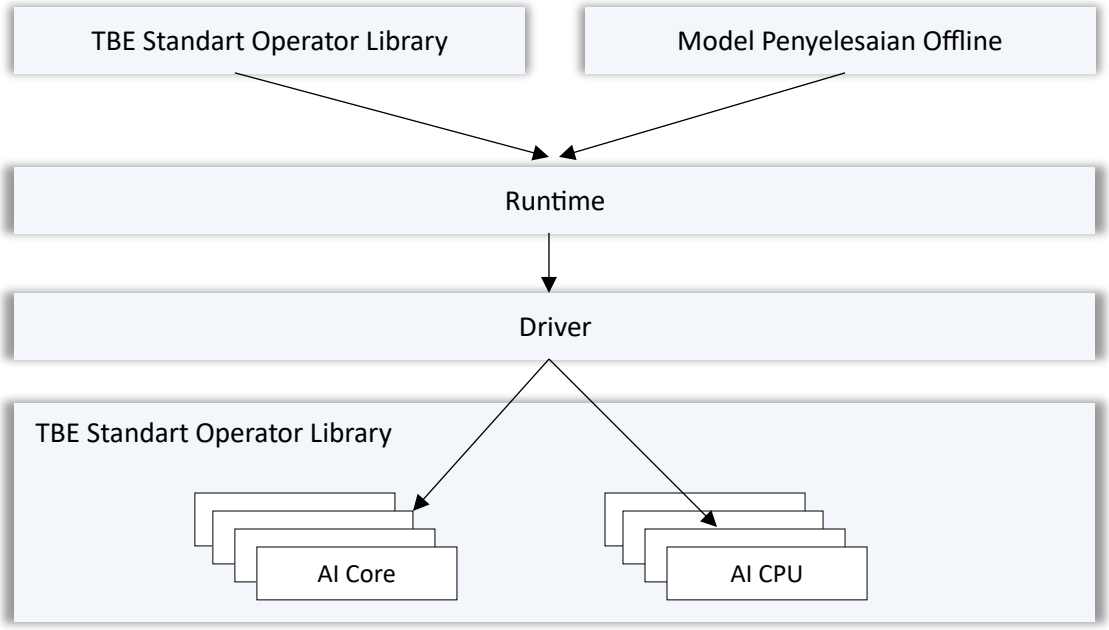
Konteks Runtime dalam tumpukan perangkat lunak ditunjukkan pada Gambar 6.17. Lapisan atas Runtime adalah pustaka operator standar TBE dan Pelaksana Model Offline. Pustaka operator standar TBE menyediakan operator jaringan saraf tiruan untuk prosesor Ascend AI, dan Pelaksana Model Offline secara khusus digunakan untuk memuat dan mengeksekusi model offline. Lapisan bawah Runtime adalah driver, yang berinteraksi dengan prosesor Ascend AI. Runtime menyediakan berbagai antarmuka panggilan, seperti antarmuka memori, antarmuka perangkat, antarmuka alur eksekusi, antarmuka peristiwa, dan antarmuka kontrol eksekusi. Berbagai antarmuka dikendalikan oleh Runtime Engine untuk menyelesaikan berbagai fungsi, seperti yang ditunjukkan pada Gambar 6.18.

Antarmuka memori menyediakan aplikasi, pelepasan, dan replikasi Memori Bandwidth Tinggi (HBM) atau Memori Laju Data Ganda (DDR) pada perangkat, termasuk replikasi data perangkat-ke-host, dari host-ke-perangkat, dan dari perangkat-ke-perangkat. Replikasi memori ini dapat dibagi menjadi mode sinkron dan asinkron. Replikasi sinkron berarti replikasi memori selesai sebelum operasi berikutnya dilakukan, sementara replikasi asinkron berarti operasi lain dapat dilakukan secara bersamaan.

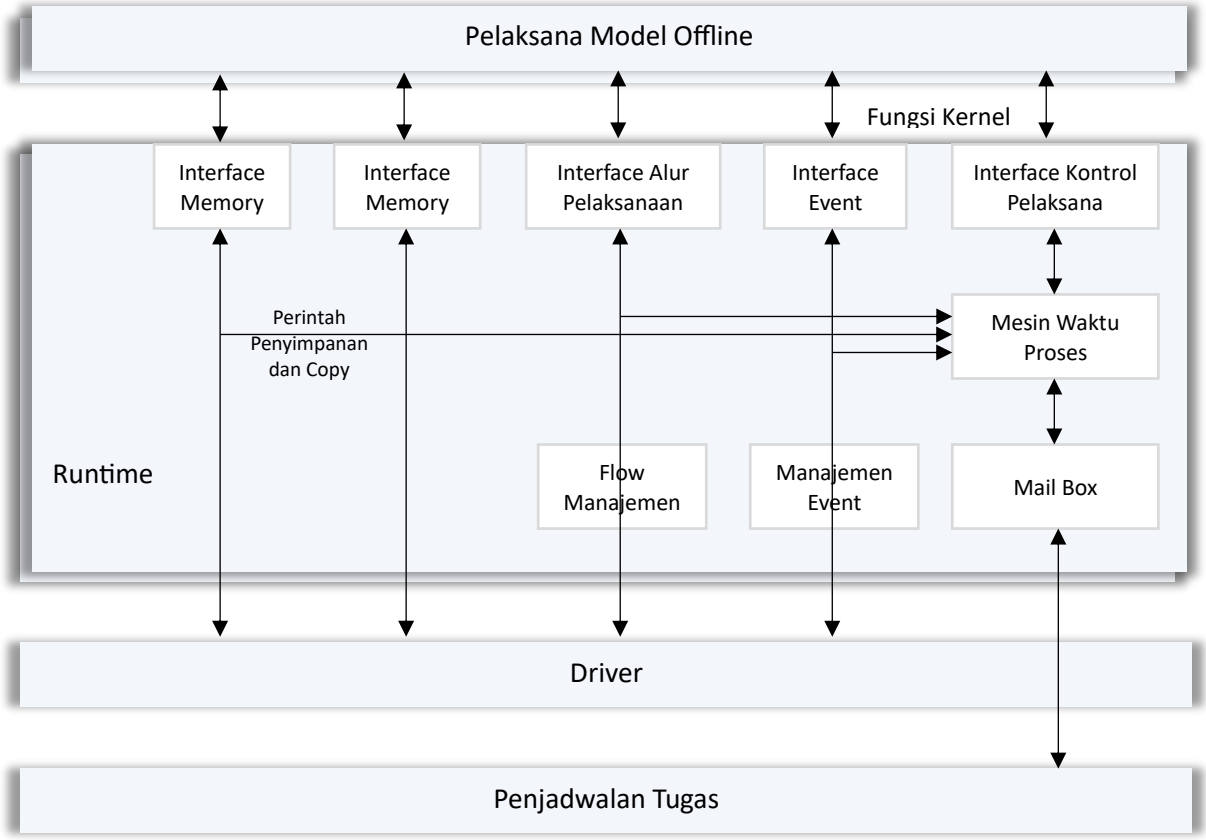
Antarmuka perangkat menyediakan kueri jumlah dan properti perangkat yang mendasarinya, serta operasi pemilihan dan pengaturan ulang. Setelah model offline memanggil antarmuka perangkat dan memilih perangkat fitur, semua tugas dalam model akan dieksekusi pada perangkat yang dipilih. Jika tugas perlu dikirim ke perangkat lain selama eksekusi, antarmuka perangkat perlu dipanggil kembali untuk pemilihan perangkat.

Antarmuka alur eksekusi menyediakan pembuatan, pelepasan, definisi prioritas, pengaturan fungsi panggilan balik, definisi dependensi peristiwa, dan sinkronisasi alur eksekusi. Fungsi-fungsi ini terkait dengan eksekusi tugas dalam alur eksekusi, dan tugas-tugas dalam satu alur eksekusi harus dieksekusi secara berurutan. Jika beberapa alur eksekusi perlu disinkronkan, antarmuka peristiwa perlu dipanggil untuk membuat, melepaskan, merekam, dan menentukan dependensi peristiwa sinkron, untuk memastikan bahwa beberapa alur eksekusi dapat diselesaikan secara sinkron dan hasil akhir model dapat dikeluarkan. Selain menetapkan tugas atau dependensi antar alur eksekusi, antarmuka peristiwa juga dapat digunakan sebagai penanda waktu untuk merekam urutan eksekusi.

Dalam proses eksekusi, antarmuka kontrol eksekusi juga digunakan. Runtime Engine menyelesaikan pemuatan fungsi kernel dan distribusi tugas replikasi asinkron memori melalui antarmuka kontrol eksekusi dan Mailbox.



Gambar 6.17 Konteks Runtime



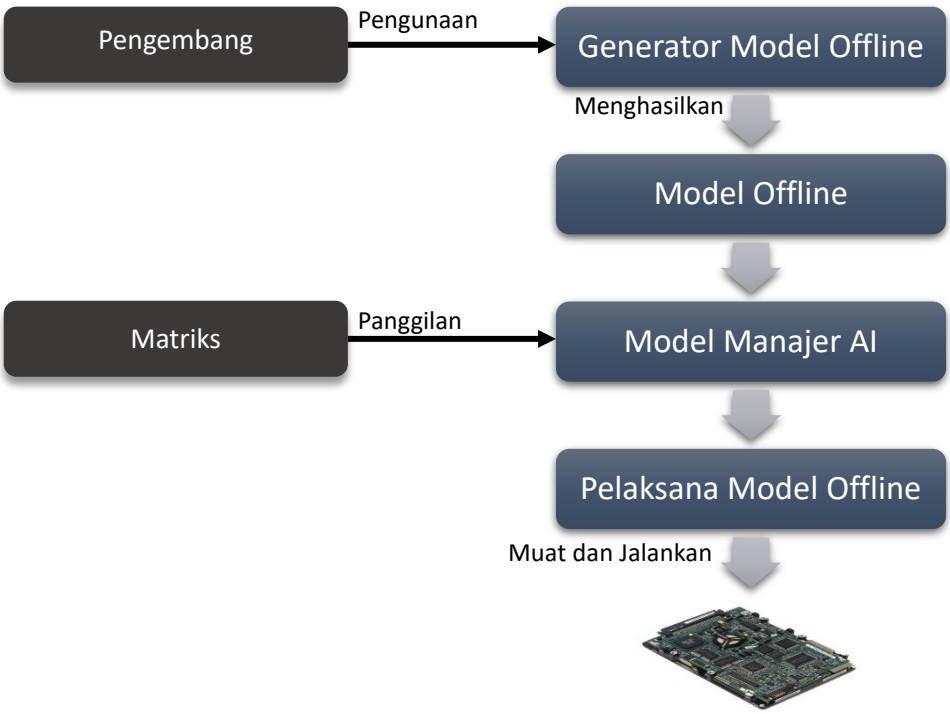
Gambar 6.18 Berbagai antarmuka pemanggilan yang disediakan oleh Runtime

5. Kerangka Kerja

(a) Garis Besar Fungsional Kerangka Kerja

Kerangka kerja bekerja dengan Tensor Boost Engine untuk menghasilkan model luring yang dapat dieksekusi untuk jaringan saraf tiruan. Sebelum eksekusi jaringan saraf tiruan, Kerangka kerja terhubung erat dengan prosesor AI Ascend untuk menghasilkan model luring berkinerja tinggi dengan pencocokan perangkat keras, sementara Matriks dan Runtime dihubungkan untuk menciptakan fusi mendalam antara model luring dan prosesor AI Ascend. Dalam eksekusi jaringan saraf tiruan, Kerangka kerja menggabungkan Matriks, Runtime, Penjadwal Tugas, dan sumber daya perangkat keras yang mendasarinya, dengan integrasi model luring, data, dan Arsitektur Da Vinci, untuk mengoptimalkan proses eksekusi dan mendapatkan keluaran aplikasi dari jaringan saraf tiruan.

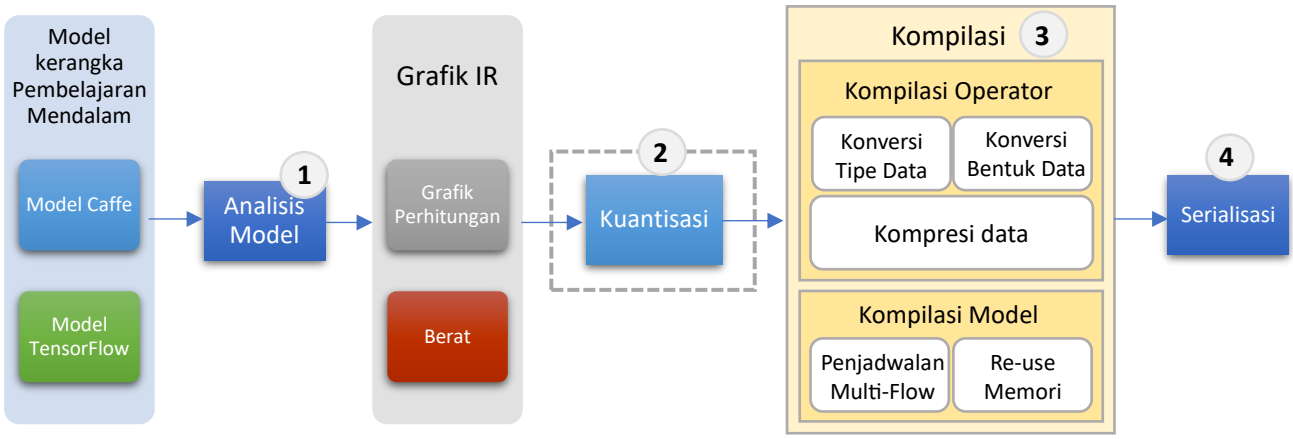
Kerangka kerja terdiri dari tiga bagian: Generator Model Luring (OMG), Eksekutor Model Luring (OME), dan Manajer Model AI, seperti yang ditunjukkan pada Gambar 6.19. Pengembang menggunakan Generator Model Luring untuk menghasilkan model luring dan menyimpannya dengan ekstensi ".om". Kemudian, Matrix dalam tumpukan perangkat lunak memanggil AI Model Manager dalam Framework, memulai Offline Model Executor, memuat model offline ke prosesor Ascend AI, dan akhirnya menyelesaikan eksekusi model offline di seluruh tumpukan perangkat lunak. Dari awal model offline, hingga pemuatan ke perangkat keras prosesor Ascend AI, hingga fungsi akhirnya berjalan, Offline Framework selalu memainkan peran administratif.



Gambar 6.19 Kerangka kerja fungsional model luring

(b) Model Offline yang Dihasilkan oleh Offline Model Generator

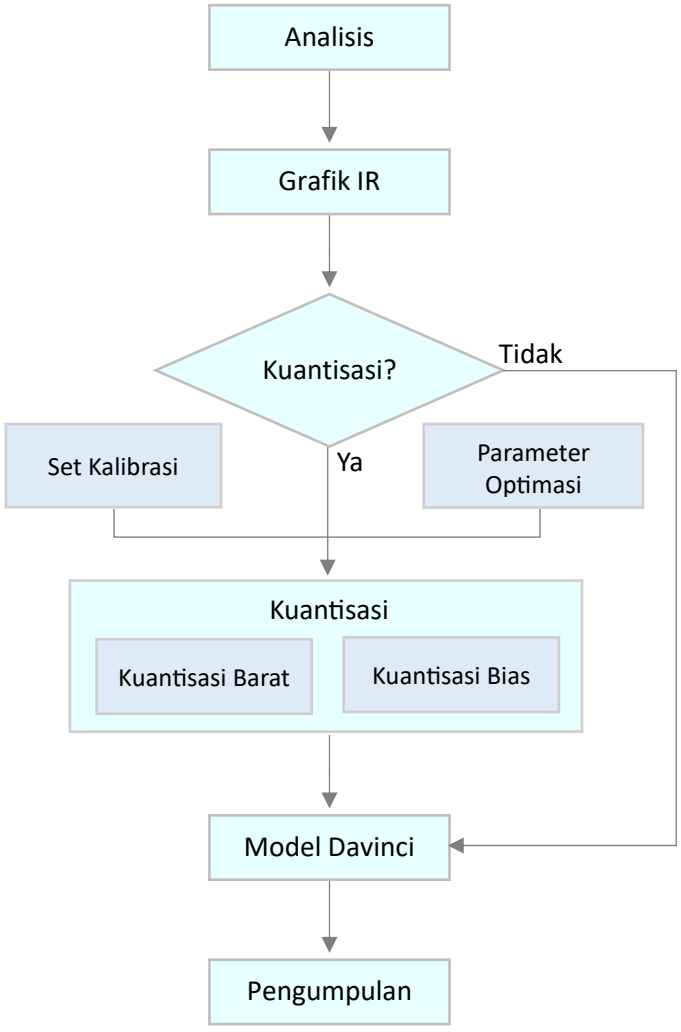
Mengambil contoh Convolutional Neural Network (CNN), model jaringan yang sesuai dibangun di bawah kerangka kerja pembelajaran mendalam, dan data asli dilatih dengan baik. Kemudian, Offline Model Generator digunakan untuk melakukan optimasi penjadwalan operator, penataan ulang dan kompresi data bobot, optimasi memori, dll., dan akhirnya model offline yang dioptimalkan dihasilkan. Offline Model Generator terutama digunakan untuk menghasilkan model offline yang dapat dieksekusi secara efisien pada prosesor Ascend AI. Prinsip kerja Offline Model Generator ditunjukkan pada Gambar 6.20. Setelah menerima model asli, Offline Model Generator melakukan proses model Convolutional Neural Network dalam empat langkah: analisis model, kuantisasi, kompilasi, dan serialisasi.



Gambar 6.20 Prinsip kerja generator model luring

- **Analisis Model**: Dalam proses analisis, Offline Model Generator mendukung analisis model jaringan asli di bawah kerangka kerja yang berbeda, mengekstrak struktur jaringan dan parameter bobot dari model asli, dan kemudian struktur jaringan didefinisikan ulang oleh graf antara terpadu (graf IR) melalui representasi graf. Grafik IR terdiri dari simpul komputasi dan simpul data. Simpul komputasi terdiri dari operator TBE dengan fungsi yang berbeda-beda, sementara simpul data menerima data tensor yang berbeda untuk menyediakan berbagai jenis data masukan bagi seluruh jaringan. Grafik IR terdiri dari grafik kalkulasi dan bobot, yang mencakup semua informasi dari model asli. Grafik IR membangun jembatan antara berbagai kerangka kerja pembelajaran mendalam dan tumpukan perangkat lunak Ascend AI, yang membuat model jaringan saraf tiruan yang dibangun oleh kerangka kerja eksternal dengan mudah diubah menjadi model offline yang didukung oleh prosesor Ascend AI.
- **Kuantisasi**: Kuantisasi mengacu pada kuantisasi bit rendah dari data presisi tinggi, sehingga menghemat ruang memori jaringan, mengurangi penundaan transmisi, dan meningkatkan efisiensi operasi. Proses kuantisasi ditunjukkan pada Gambar 6.21.
  Grafik IR dihasilkan setelah analisis selesai. Jika model masih perlu dikuantifikasi, model tersebut dapat dikuantifikasi oleh alat kuantisasi otomatis berdasarkan struktur

dan bobot grafik IR. Dalam operator, bobot dan bias dapat dikuantisasi. Dalam proses pembuatan model offline, bobot dan bias yang telah dikuantisasi akan disimpan dalam model offline. Dalam perhitungan penalaran, bobot dan bias yang telah dikuantisasi dapat digunakan untuk menghitung data masukan, dan set kalibrasi digunakan untuk melatih parameter kuantisasi dalam proses kuantisasi guna memastikan akurasi.
Jika kuantisasi tidak diperlukan, model offline dikompilasi secara langsung. Kuantisasi dibagi menjadi dua jenis, yaitu kuantisasi offset dan kuantisasi non-offset, yang membutuhkan dua parameter: Skala dan Offset. Dalam proses kuantisasi data, ketika metode kuantisasi ditetapkan sebagai kuantisasi non-offset, data akan menggunakan kuantisasi non-offset untuk menghitung Skala data terkuantisasi; jika metode kuantisasi ditetapkan sebagai kuantisasi offset data, data akan menggunakan kuantisasi offset untuk menghitung Skala dan Offset data keluaran.

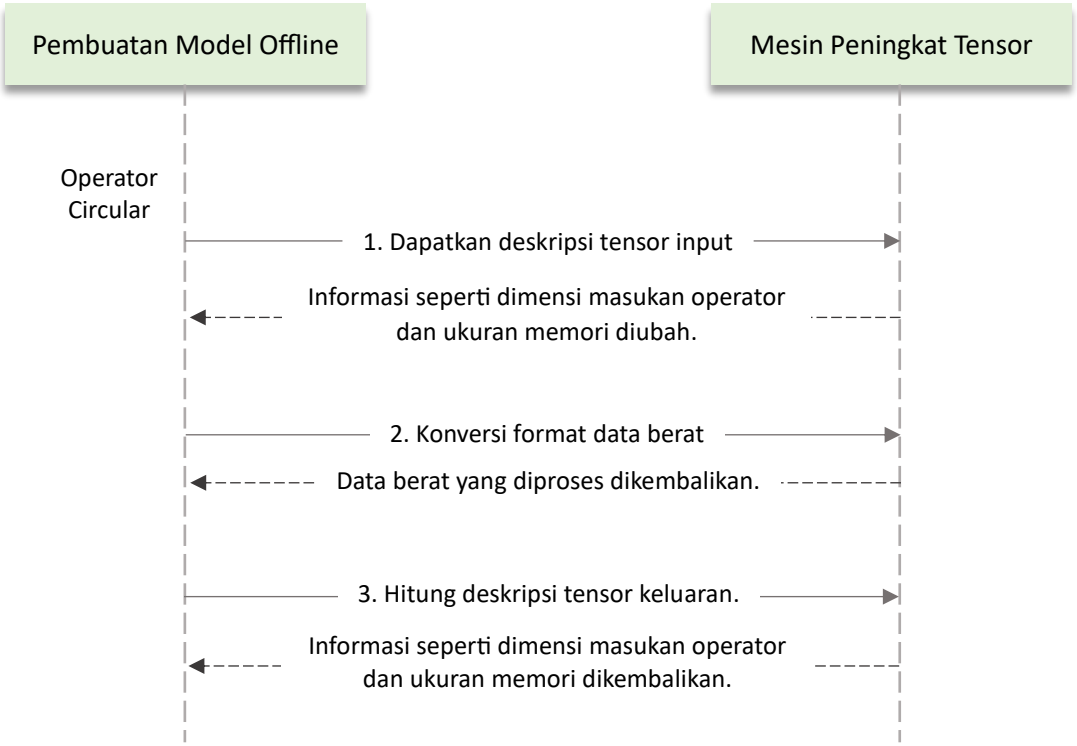


Gambar 6.21 Proses kuantisasi

Dalam proses kuantisasi bobot, karena akurasi kuantisasi bobot yang tinggi, kuantisasi non-offset selalu digunakan. Misalnya, menurut algoritma kuantisasi, kuantisasi tipe INT8 pada berkas bobot dapat menghasilkan bobot dan Skala INT8. Dalam proses

kuantisasi offset, menurut Skala bobot dan tanggal, data offset tipe FP32 dapat dikuantisasi menjadi keluaran data tipe INT32. Ketika terdapat persyaratan yang lebih tinggi untuk ukuran dan performa model, seseorang dapat memilih untuk melakukan operasi kuantisasi. Dalam proses pembuatan model offline, kuantisasi akan mengonversi data presisi tinggi menjadi data bit rendah, sehingga model offline akhir menjadi ringan demi menghemat ruang memori jaringan, meminimalkan penundaan transmisi, dan meningkatkan efisiensi operasi. Dalam proses kuantisasi, ukuran penyimpanan model sangat dipengaruhi oleh parameter, sehingga Offline Model Generator berfokus pada kuantisasi dengan parameter seperti operator konvolusi, operator terhubung penuh, dan Convolution Depthwise.

- **Kompilasi**: Setelah kuantisasi model, model perlu dikompilasi. Kompilasi dibagi menjadi dua bagian: kompilasi operator dan kompilasi model. Kompilasi operator menyediakan implementasi spesifik operator, dan kompilasi model mengagregasi model operator untuk menghasilkan struktur model luring.
  - ✓ *Kompilasi Operator*: Kompilasi Operator digunakan untuk menghasilkan operator, terutama untuk menghasilkan struktur luring spesifik operator. Pembangkitan Operator dibagi menjadi tiga proses: deskripsi tensor masukan, konversi data bobot, dan deskripsi tensor keluaran. Dalam proses deskripsi tensor masukan, dimensi masukan, ukuran memori, dan informasi lain dari setiap operator dihitung, dan bentuk data masukan operator didefinisikan dalam Pembangkit Model Luring. Dalam konversi data bobot, parameter bobot yang digunakan oleh operator diproses berdasarkan format data (seperti konversi FP32 ke FP16), konversi bentuk (seperti penataan ulang fraktal), kompresi data, dan sebagainya. Dalam deskripsi tensor keluaran, dimensi keluaran, ukuran memori, dan informasi lain dari operator dihitung. Proses pembangkitan operator ditunjukkan pada Gambar 6.22. Dalam proses pembangkitan operator, bentuk data keluaran perlu dianalisis, ditentukan, dan dideskripsikan melalui antarmuka pustaka akselerator operator TBE. Konversi format data juga dapat dilakukan melalui antarmuka pustaka akselerasi operator TBE.



Gambar 6.22 Proses pembangkitan operator

Pembuat Model Offline menerima grafik IR yang dihasilkan oleh jaringan saraf tiruan dan mendeskripsikan setiap simpul dalam grafik IR, menganalisis masukan dan keluaran setiap operator satu per satu. Pembuat Model Offline menganalisis sumber data masukan operator saat ini untuk mendapatkan jenis operator yang terhubung langsung dengan operator saat ini di lapisan atas, kemudian memasuki pustaka operator melalui antarmuka pustaka akselerasi operator TBE untuk menemukan deskripsi data keluaran operator sumber. Kemudian, informasi data keluaran operator sumber dikembalikan ke Pembuat Model Offline sebagai deskripsi tensor masukan spesifik operator saat ini. Oleh karena itu, deskripsi data masukan operator saat ini dapat diperoleh dengan mengetahui informasi keluaran operator sumber.

Jika simpul dalam graf IR bukan operator melainkan simpul data, deskripsi tensor masukan tidak diperlukan. Jika operator memiliki data bobot, seperti operator konvolusi dan operator terhubung penuh, deskripsi dan pemrosesan data bobot diperlukan. Jika tipe data bobot masukan adalah FP32, data tersebut perlu dikonversi ke tipe FP16 oleh Offline Model Generator agar memenuhi persyaratan tipe data AI Core. Setelah konversi tipe, Offline Model Generator memanggil antarmuka ccTransFilter untuk menyusun ulang data bobot, sehingga bentuk masukan bobot dapat memenuhi persyaratan format AI Core.

Setelah mendapatkan bobot format tetap, Offline Model Generator memanggil antarmuka ccCompressWeight yang disediakan oleh TBE untuk mengompresi dan mengoptimalkan bobot, sehingga mengurangi ruang memori bobot dan membuat

model lebih ringan. Setelah konversi data bobot selesai, data bobot yang memenuhi persyaratan perhitungan dikembalikan ke Offline Model Generator.

Setelah konversi data bobot selesai, Offline Model Generator juga perlu mendeskripsikan informasi data keluaran operator dan menentukan bentuk tensor keluaran. Untuk operator kompleks tingkat tinggi, seperti operator konvolusi dan operator pooling, Offline Model Generator dapat langsung memperoleh informasi tensor keluaran operator dengan menggabungkan informasi tensor masukan dan informasi bobot operator melalui antarmuka kalkulasi yang disediakan oleh pustaka akselerasi operator TBE. Jika operator sederhana tingkat rendah, seperti operator penjumlahan, informasi tensor keluaran dapat ditentukan langsung oleh informasi tensor masukan operator, dan akhirnya disimpan dalam Offline Model Generator.

Berdasarkan proses operasi di atas, Offline Model Generator menelusuri semua operator dalam Grafik IR jaringan, langkah-langkah pembangkitan operator dieksekusi secara melingkar. Tensor masukan-keluaran dan data bobot semua operator dijelaskan untuk melengkapi representasi struktur luring operator, yang menyediakan model operator untuk langkah pembangkitan model berikutnya.

- ✓ *Kompilasi Model*: Setelah pembangkitan operator selesai dalam proses kompilasi, Offline Model Generator juga perlu membangkitkan model untuk memperoleh struktur luring model. Generator Model Offline memperoleh Grafik IR, melakukan analisis penjadwalan konkuren pada operator, membagi beberapa node Grafik IR menjadi alur eksekusi, sehingga diperoleh beberapa alur eksekusi yang terdiri dari operator dan input data, yang dapat dianggap sebagai urutan eksekusi operator. Untuk node tanpa interdependensi, node tersebut dialokasikan langsung ke alur eksekusi yang berbeda. Jika node dalam alur eksekusi yang berbeda memiliki dependensi, sinkronisasi antara beberapa alur eksekusi dilakukan melalui antarmuka sinkronisasi rtEvent. Jika terdapat surplus sumber daya komputasi di AI Core, pemisahan alur multi-eksekusi dapat menyediakan penjadwalan multi-aliran untuk AI Core, sehingga dapat meningkatkan kinerja komputasi model jaringan. Namun, jika terdapat banyak tugas pemrosesan paralel di AI Core, hal ini akan memperburuk tingkat preemption sumber daya dan memperburuk kinerja eksekusi.

  Secara default, alur eksekusi tunggal diadopsi untuk memproses jaringan, yang dapat mencegah risiko pemblokiran akibat eksekusi beberapa tugas secara konkuren. Pada saat yang sama, berdasarkan hubungan eksekusi spesifik dari urutan eksekusi beberapa operator, Offline Model Generator dapat secara independen melakukan optimasi fusi operator dan optimasi multiplexing memori. Berdasarkan informasi memori input dan output operator, multiplexing memori komputasi dilakukan, dan informasi multiplexing terkait ditulis ke dalam model dan deskripsi operator untuk menghasilkan model offline yang efisien.

Operasi optimasi dapat mengalokasikan kembali sumber daya komputasi selama eksekusi beberapa operator untuk meminimalkan penggunaan memori. Pada saat yang sama, alokasi dan pelepasan memori yang sering juga dapat dihindari selama operasi, sehingga eksekusi beberapa operator dengan penggunaan memori minimum dan frekuensi migrasi data terendah dapat diimplementasikan, dengan kinerja yang lebih baik dan kebutuhan sumber daya perangkat keras yang lebih rendah.

- ✓ *Serialisasi:* Model luring yang telah dikompilasi disimpan dalam memori dan perlu diserialisasi. Proses serialisasi terutama menyediakan fungsi tanda tangan dan enkripsi pada berkas model untuk lebih lanjut mengenkapsulasi dan melindungi integritas model luring. Setelah proses serialisasi selesai, model luring dapat dikeluarkan dari memori ke berkas eksternal yang dapat dipanggil dan dieksekusi oleh chip prosesor Ascend AI jarak jauh.

6. Pra-pemrosesan Visi Digital

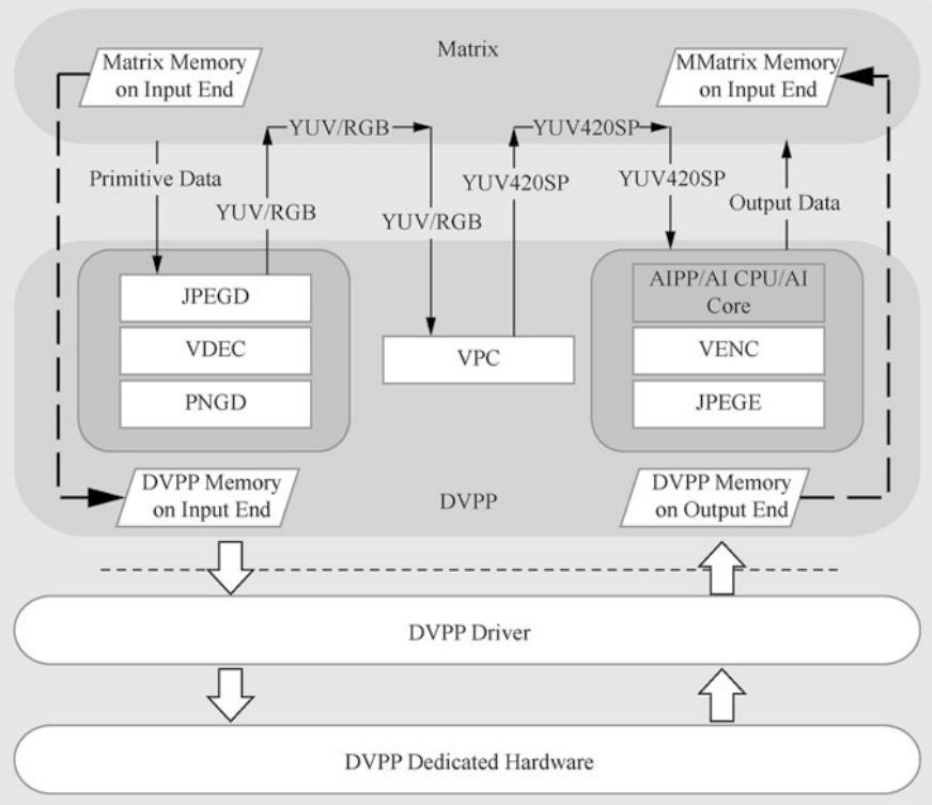
Modul Pra-pemrosesan Visi Digital (DVPP), sebagai modul codec dan konversi gambar dalam tumpukan perangkat lunak Ascend AI, memainkan fungsi tambahan pra-pemrosesan untuk jaringan saraf tiruan. Ketika data video atau gambar dari memori sistem dan jaringan masuk ke sumber daya komputasi prosesor Ascend AI untuk perhitungan, modul DVPP perlu dipanggil untuk konversi format sebelum pemrosesan jaringan saraf tiruan berikutnya jika data tidak memenuhi format input, resolusi, dan persyaratan lain yang ditentukan oleh Arsitektur Da Vinci.

(a) Arsitektur Fungsi DVPP

Terdapat enam modul untuk DVPP, yaitu Video Decoding (VDEC), Video Encoding (VENC), JPEG Decoding (JPEGD), JPEG Encoding (JPEG), PNG Decoding (PNGD), dan Visual Pre-processing Core (VPC).

1) VDEC menyediakan fungsi decoding video H.264/H.265, yang dapat mendekode aliran video input dan menghasilkan gambar. Fungsi ini sering digunakan dalam pra-pemrosesan adegan seperti pengenalan video.
2) VNEC menyediakan fungsi pengkodean video output. Untuk data output VPC atau data input format YUV asli, VNEC dapat menghasilkan pengkodean menjadi video H.264/H.265, yang mudah diputar dan ditampilkan langsung.
3) JPEG dapat mendekode gambar berformat JPEG, mengonversi gambar input JPEG asli menjadi data YUV, dan melakukan pra-proses data input penalaran jaringan saraf tiruan.
4) Setelah pemrosesan gambar JPEG selesai, JPEG perlu digunakan untuk mengembalikan data yang telah diproses ke format JPEG. Modul JPEG sebagian besar digunakan untuk pasca-pemrosesan data keluaran inferensi jaringan saraf tiruan.
5) Ketika format gambar masukan adalah PNG, PNGD perlu dipanggil untuk melakukan dekode, agar Modul PNGD dapat mengeluarkan gambar PNG dalam format RGB untuk penalaran dan perhitungan oleh prosesor Ascend AI.

6) VPC menyediakan fungsi lain dari pemrosesan gambar dan video, seperti konversi format (misalnya konversi format YUV/RGB ke format YUV420), penskalaan ukuran, pemotongan, dan sebagainya.
7) Alur eksekusi modul pemrosesan visi digital ditunjukkan pada Gambar 6.23, yang perlu diselesaikan dengan kerja sama dari Matriks, DVPP, driver DVPP, dan perangkat keras khusus DVPP.



Gambar 6.23 Alur eksekusi modul pemrosesan visi digital

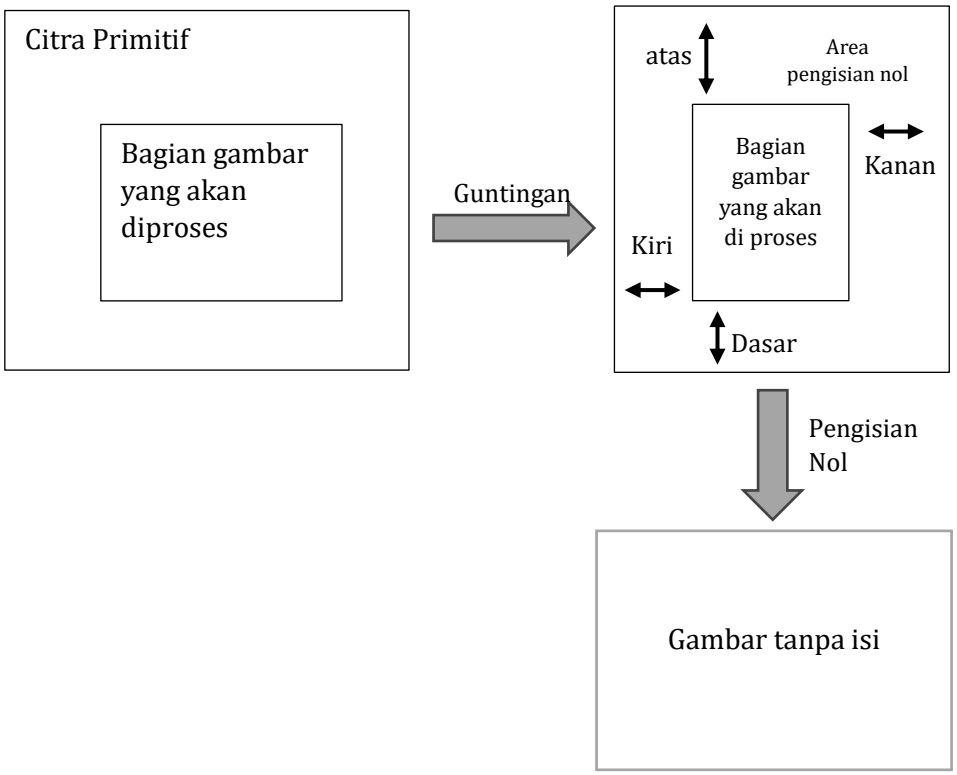
8) Matrix berada di tingkat teratas kerangka kerja, yang bertanggung jawab untuk menjadwalkan modul fungsi dalam DVPP guna memproses dan mengelola alur data.
9) DVPP terletak di lapisan tengah dan atas arsitektur fungsional, yang menyediakan antarmuka pemrograman bagi Matrix untuk memanggil modul pemrosesan grafis video. Melalui antarmuka ini, parameter yang relevan dari modul enkode dan dekode serta modul pra-pemrosesan visual dapat dikonfigurasi.
10) Driver DVPP terletak di lapisan tengah dan bawah arsitektur fungsional, yang merupakan modul perangkat keras terdekat dengan DVPP. Driver ini terutama bertanggung jawab atas manajemen perangkat, manajemen mesin, dan driver grup modul mesin. Driver akan mengalokasikan mesin perangkat keras DVPP yang sesuai sesuai dengan tugas yang diberikan oleh DVPP. Pada saat yang sama, driver ini juga membaca dan menulis register dalam modul perangkat keras untuk menyelesaikan beberapa pekerjaan inisialisasi perangkat keras lainnya.

11) Lapisan terbawah adalah sumber daya komputasi perangkat keras yang sesungguhnya, yaitu grup modul DVPP, yang merupakan akselerator khusus yang independen dari modul lain dalam prosesor Ascend AI. Akselerator ini dirancang khusus untuk tugas-tugas pengodean, dekode, dan pra-pemrosesan terkait gambar dan video.

(b) Mekanisme Pra-pemrosesan DVPP.

Ketika data masukan memasuki mesin data, setelah mesin menemukan bahwa format data tidak memenuhi persyaratan pemrosesan AI Core berikutnya, DVPP dapat dimulai untuk pra-pemrosesan data. Mengambil pra-pemrosesan gambar sebagai contoh untuk menggambarkan keseluruhan proses pra-pemrosesan.

- Pertama, Matrix memindahkan data dari Memori ke Buffer DVPP untuk di-cache.
- Sesuai dengan format data spesifik, mesin pra-pemrosesan menyelesaikan konfigurasi parameter dan transmisi data melalui antarmuka pemrograman yang disediakan oleh DVPP.
- Setelah antarmuka pemrograman dimulai, DVPP mentransfer parameter konfigurasi dan data mentah ke driver, dan driver DVPP memanggil PNG atau JPEG untuk inisialisasi dan distribusi tugas.
- PNG atau JPEG dari perangkat keras khusus DVPP memulai operasi aktual untuk menyelesaikan decoding gambar guna memperoleh data berformat YUV atau RGB untuk memenuhi kebutuhan pemrosesan selanjutnya.
- Setelah dekode, Matrix terus memanggil VPC dengan mekanisme yang sama untuk mengonversi citra lebih lanjut ke format YUV420SP. Karena format YUV420SP memiliki efisiensi penyimpanan data yang lebih tinggi dan menggunakan bandwidth yang lebih sedikit, format ini dapat mengirimkan lebih banyak data dalam bandwidth yang sama untuk memenuhi kebutuhan throughput komputasi AI Core yang tinggi. Pada saat yang sama, DVPP juga dapat menyelesaikan pemotongan dan penskalaan citra. Gambar 6.24 menunjukkan operasi pemotongan dan pengisian nol yang umum untuk mengubah ukuran citra.



Gambar 6.24 Alur data pra-pemrosesan gambar

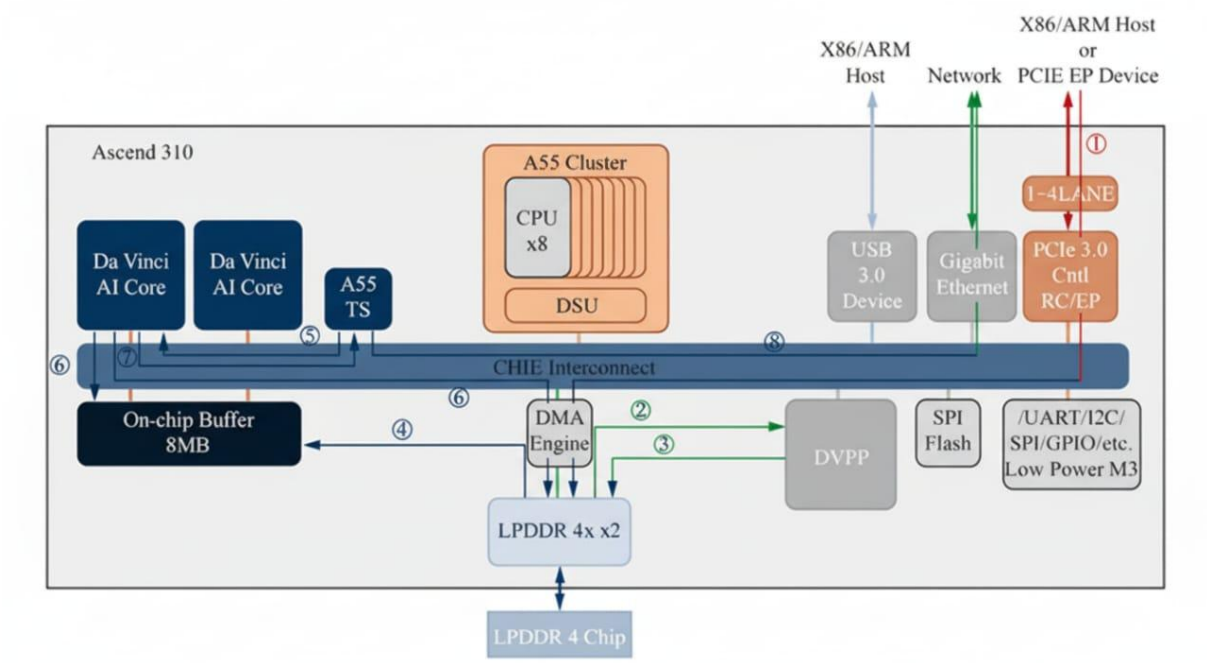
VPC mengambil bagian dari citra primitif yang akan diproses, lalu melakukan pengisian nol pada bagian ini, untuk mempertahankan informasi fitur tepi dalam proses perhitungan CVNN. Operasi pengisian nol membutuhkan empat ukuran pengisian, yaitu atas, bawah, kiri, dan kanan. Tepi citra diperluas di area pengisian. Akhirnya, citra yang telah diisi dapat dihitung secara langsung.

- Setelah serangkaian pra-pemrosesan, data citra dapat diproses dengan dua cara berikut.
  - ✓ Data gambar dapat diproses lebih lanjut oleh AIPP (Pra-pemrosesan AI) sesuai dengan kebutuhan model (Opsional. Jika data keluaran DVPP memenuhi persyaratan gambar, data tersebut tidak perlu diproses oleh AIPP), dan kemudian data gambar yang memenuhi persyaratan akan masuk ke AI Core di bawah kendali AI CPU untuk kalkulasi jaringan saraf tiruan yang diperlukan.
  - ✓ Data gambar keluaran dikodekan dalam JPEG, dan pasca-pemrosesan selesai. Data tersebut dimasukkan ke dalam buffer DVPP. Terakhir, Matrix mengambil data untuk operasi selanjutnya. Pada saat yang sama, sumber daya komputasi DVPP dilepaskan dan cache dipulihkan.

Dalam keseluruhan proses pra-pemrosesan, Matrix menyelesaikan pemanggilan fungsi berbagai modul. Sebagai modul penyedia data yang disesuaikan, DVPP menggunakan metode pemrosesan heterogen atau khusus untuk mengonversi data gambar dengan cepat, menyediakan sumber data yang memadai untuk AI Core, sehingga memenuhi kebutuhan data dalam jumlah besar dan bandwidth yang besar dalam komputasi jaringan saraf tiruan.

Alur Data Prosesor AI Ascend

Mengambil contoh aplikasi penalaran pengenalan wajah, alur data Prosesor AI Ascend (Ascend 310) diperkenalkan. Pertama, data dikumpulkan dan diproses oleh Kamera, kemudian data dinalar, dan akhirnya hasil pengenalan wajah dikeluarkan, seperti yang ditunjukkan pada Gambar 6.25.

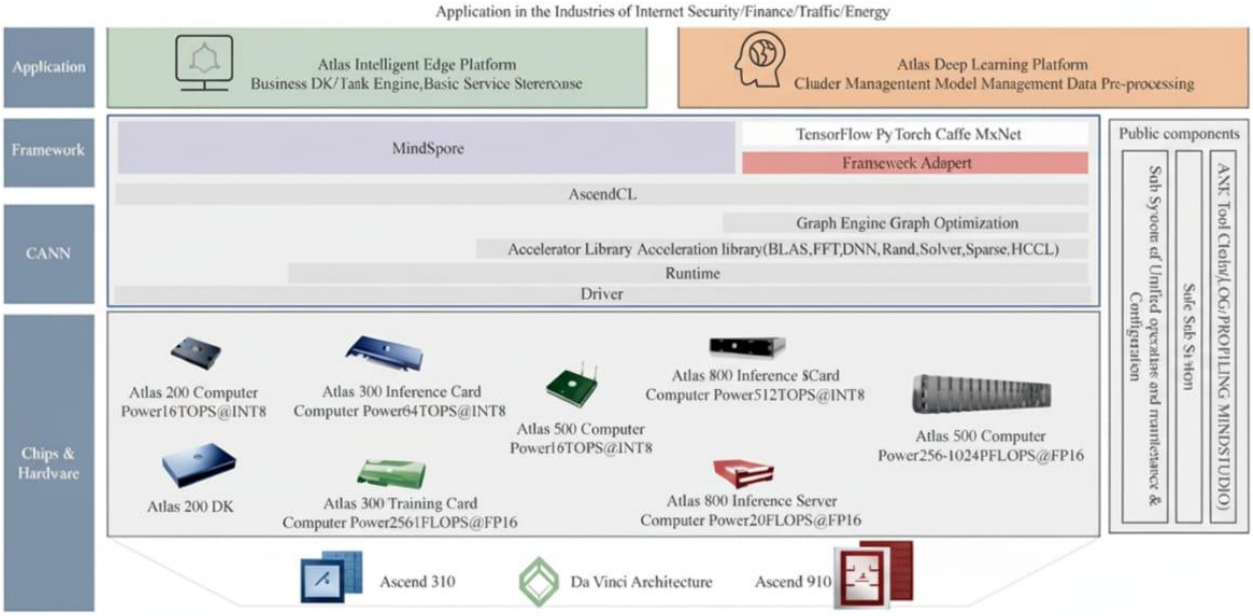


Gambar 6.25 Aliran data prosesor Ascend AI (Ascend 310)

1. Data dikumpulkan dan diproses oleh Kamera.
   - Langkah 1: Aliran video terkompresi dilewatkan dari Kamera, dan data disimpan dalam DDR melalui kanal PCIe.
   - Langkah 2: DVPP membaca aliran video terkompresi ke dalam cache.
   - Langkah 3: Setelah pra-pemrosesan, DVPP menulis frame yang telah didekompresi ke dalam Memori DDR.
2. Penalaran pada data.
   - Langkah 4: Penjadwal Tugas mengirimkan perintah ke Akses Memori Langsung (DMA) untuk memuat sumber daya AI dari DDR ke Buffer On-chip.
   - Langkah 5: Penjadwal Tugas mengonfigurasi Inti AI untuk menjalankan tugas.
   - Langkah 6: Ketika AI Core berfungsi, ia membaca grafik fitur dan bobot, lalu menuliskan hasilnya ke DDR atau On-chip Buffer.
3. Hasil pengenalan wajah dikeluarkan.
   - Langkah 7: Setelah AI Core menyelesaikan pemrosesan, ia mengirimkan sinyal ke Penjadwal Tugas. Kemudian, Penjadwal Tugas memeriksa hasilnya dan menetapkan tugas lain jika perlu, lalu kembali ke langkah 4.
   - Langkah 8: Setelah tugas AI terakhir selesai, Penjadwal Tugas melaporkan hasilnya ke Host.

6.3 SOLUSI KOMPUTASI ATLAS AI

Berbasis prosesor AI Huawei Ascend, solusi komputasi AI Huawei Atlas membangun Solusi Infrastruktur AI yang komprehensif, berorientasi pada end, edge, dan cloud, melalui berbagai bentuk produk seperti modul, papan, stasiun kecil, server, dan kluster. Bagian ini terutama memperkenalkan produk-produk terkait dari solusi komputasi AI Huawei Atlas, termasuk penalaran dan pelatihan.

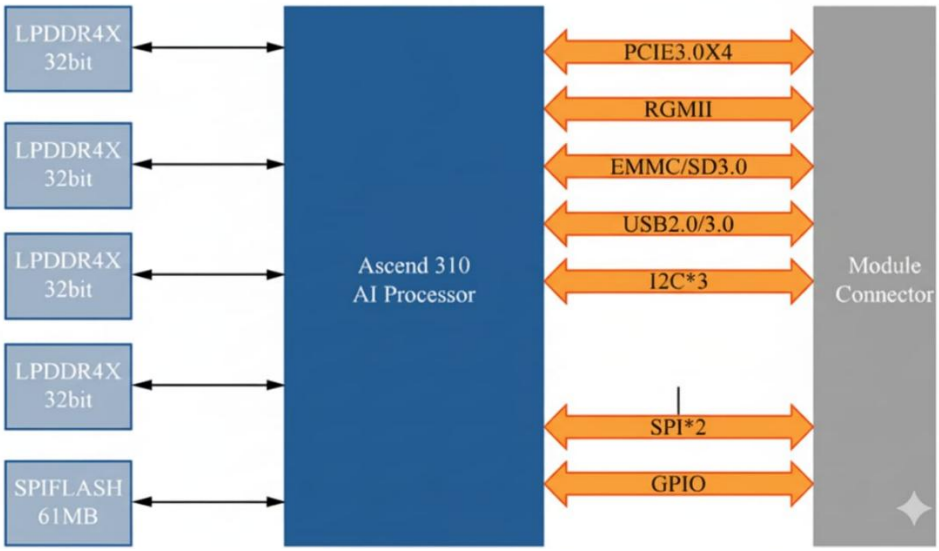


Gambar 6.26 Lanskap solusi komputasi Atlas AI

Produk penalaran terutama mencakup modul akselerasi AI Atlas 200, Atlas 200 DK, kartu penalaran Atlas 300I, stasiun cerdas Atlas 500, dan server penalaran Atlas 800, yang semuanya menggunakan prosesor AI Ascend 310. Produk pelatihan terutama mencakup kartu pelatihan Atlas 300T, server pelatihan Atlas 800, dan kluster AI Atlas 900, yang semuanya menggunakan prosesor AI Ascend 910. Lanskap solusi komputasi AI Atlas ditunjukkan pada Gambar 6.26.

1. Modul Akselerasi AI Atlas 200

Modul Akselerasi AI Atlas 200 adalah modul komputasi cerdas AI dengan kinerja tinggi dan konsumsi daya rendah. Modul ini dapat digunakan pada kamera, UAV (kendaraan udara tak berawak), robot, dan perangkat lain seukuran setengah kartu kredit, dengan konsumsi daya 9,5 W, dan mendukung analisis video HD 16 kanal secara real-time. Atlas 200 terintegrasi dengan prosesor AI Ascend 310, yang dapat mewujudkan analisis dan inferensi gambar, video, dan data lainnya. Modul ini dapat digunakan secara luas dalam pemantauan cerdas, robot, UAV, server video, dan aplikasi lainnya. Diagram blok sistem Atlas 200 ditunjukkan pada Gambar 6.27.



Gambar 6.27 Diagram blok sistem Atlas 200

Berikut adalah karakteristik kinerja Atlas 200.

(a) Didukung oleh prosesor AI Huawei Ascend 310, Atlas 200 dapat memberikan daya komputasi perkalian dan penjumlahan 16TOPS INT8 atau 8TOPS FP16.

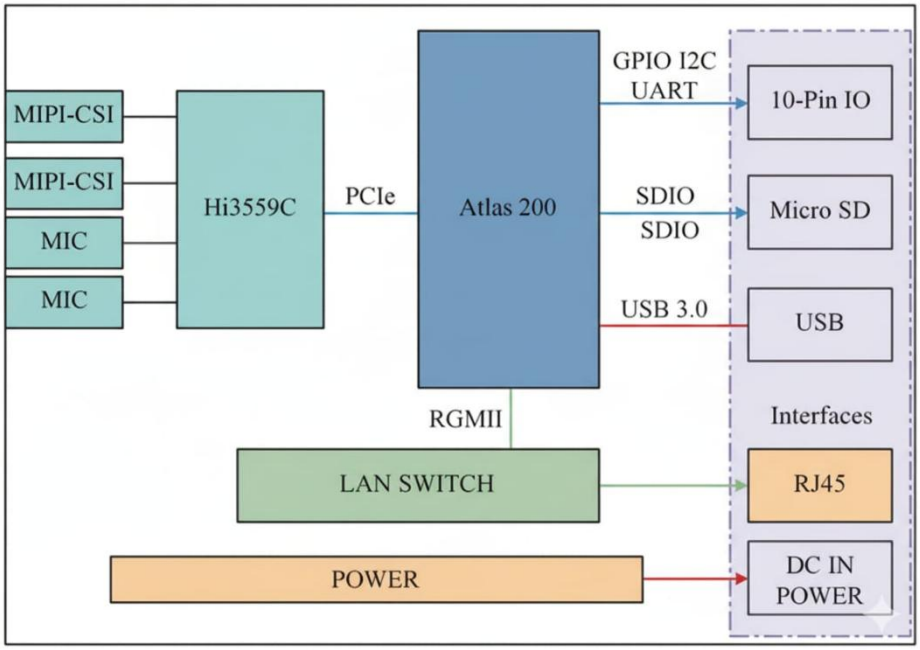
(b) Dengan antarmuka yang kaya, Atlas 200 mendukung PCIe3.0x4, RGMII, USB2.0/USB3.0, I2C, SPI, UART, dan lain-lain.

(c) Atlas 200 dapat mengakses video hingga 16 kanal dengan resolusi 1080p 30fps.

(d) Atlas 200 mendukung berbagai spesifikasi codec video H.264 dan H.265, yang dapat memenuhi berbagai kebutuhan pemrosesan video pengguna.

2. Atlas 200 DK

Atlas 200 Developer Kit (Atlas 200 DK) adalah produk yang inti dari modul akselerasi Atlas 200 AI. Atlas 200 DK dapat membantu pengembang aplikasi AI untuk cepat terbiasa dengan lingkungan pengembangan. Fungsi utamanya adalah untuk membuka fungsi inti prosesor Ascend 310 AI melalui antarmuka periferal pada papan, sehingga pengguna dapat dengan cepat dan mudah mengakses daya pemrosesan prosesor Ascend 310 AI yang tangguh. Atlas 200 DK terutama mencakup modul akselerasi Atlas 200 AI, chip antarmuka gambar/audio (Hi3559C), dan LAN Switch. Arsitektur sistem ditunjukkan pada Gambar 6.28.



Gambar 6.28 Arsitektur sistem Atlas 200 DK

Karakteristik kinerja Atlas 200 DK adalah sebagai berikut.

(a) Atlas 200 DK menyediakan daya komputasi puncak 16TOPS (INT8).
(b) Atlas 200 DK mendukung dua masukan kamera, dua pemrosesan gambar ISP, dan standar teknis HDR10 rentang dinamis tinggi.
(c) Atlas 200 DK, yang dipadukan dengan daya komputasi yang kuat, mendukung Ethernet 1000 MB, menyediakan akses internet berkecepatan tinggi.
(d) Atlas 200 DK menyediakan antarmuka ekstensi 40 pin universal (dicadangkan) untuk memfasilitasi desain prototipe produk.
(e) Atlas 200 DK mendukung masukan daya DC rentang lebar dari 5 hingga 28 V.

Spesifikasi produk Atlas 200 DK ditunjukkan pada Tabel 6.1. Keunggulan Atlas 200 DK: Bagi pengembang, lingkungan pengembangan dapat dibangun menggunakan komputer laptop, karena biaya lingkungan lokal yang independen sangat rendah, dan multifungsi serta multi-antarmukanya dapat memenuhi persyaratan dasar. Bagi peneliti, mode kolaboratif pengembangan lokal plus pelatihan cloud diadopsi untuk membangun lingkungan tersebut. Huawei Cloud dan Atlas 200 DK mengadopsi serangkaian tumpukan protokol, pelatihan cloud, dan penerapan lokal tanpa modifikasi apa pun. Bagi wirausahawan, Atlas 200 DK menyediakan prototipe tingkat kode (Demo), berdasarkan arsitektur referensi, dan memodifikasi 10% kode untuk melengkapi fungsi algoritma. Interaksi antar pengembang dan komunitas serta migrasi produk komersial yang lancar dapat tercapai.

Tabel 6.1 Spesifikasi produk Atlas 200 DK

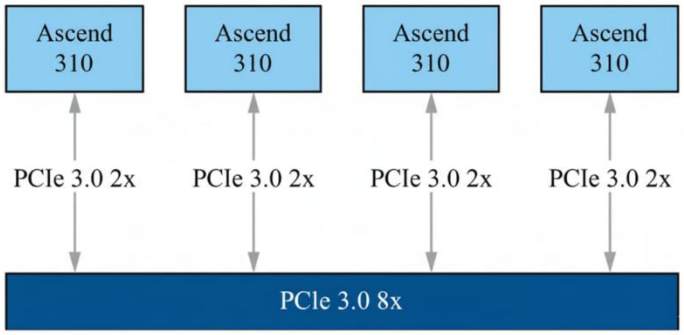
Item	Spesifikasi
Prosesor AI	2 Inti AI DaVinci CPU: 8-core A55, maksimum 1,6 GHz

Daya Komputasi AI	Daya komputasi perkalian dan penjumlahan: 8 TFLOPS/FP16, 16 TOPS/INT8
Memori	LPDDR4X, 128 bit Kapasitas: 8 GB / 4 GB Kecepatan antarmuka: 3200 Mbit/s
Penyimpanan	1 Kartu Micro SD, mendukung SD3.0 Kecepatan maksimum yang didukung: SDR50 Kapasitas maksimum: 2 TB
Antarmuka Jaringan	1 GE RJ45
Antarmuka USB	1 Antarmuka USB3.0 Type-C, hanya digunakan sebagai *Slave Device*, kompatibel dengan USB2.0
Antarmuka Lainnya	1 Konektor IO 40 pin 2 Konektor MIPI 22 pin 2 Mikrofon *on-board*
Daya	5 V ~ 28 V DC, Pengaturan bawaan: Adaptor 12 V 3A
Dimensi	137,8 mm × 93,0 mm × 32,9 mm
Konsumsi Daya	20 W
Berat	234 g
Suhu Operasional	0 °C ~ 35 °C (32 °F ~ 95 °F)
Suhu Penyimpanan	0 °C ~ 85 °C (32 °F ~ 185 °F)

3. Kartu Inferensi Atlas 300I

Kartu Inferensi Huawei Atlas 300I adalah kartu akselerator AI inferensi video 64 kanal dengan kepadatan tertinggi di industri, termasuk model 3000 dan 3010, yaitu kartu akselerator AI Huawei Atlas 300 model 3000 dan kartu akselerator AI Huawei Atlas 300 model 3010. Perbedaan antara kedua model ini terutama terletak pada perbedaan arsitektur (seperti x86, ARM, dll.). Di sini, hanya kartu akselerator AI Huawei Atlas 300 (Model 3000) yang diperkenalkan.

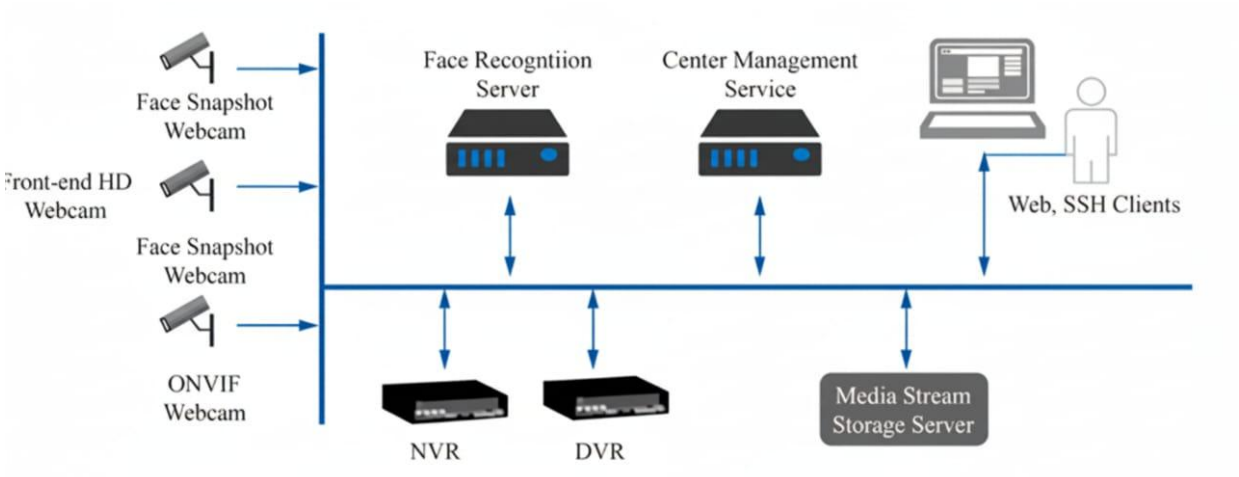
Kartu akselerator AI Huawei Atlas 300 (Model 3000) dirancang dan dikembangkan berdasarkan prosesor AI Ascend 310. Kartu ini mengadopsi empat kartu PCI HHHL dari prosesor AI Ascend 310, dan bekerja sama dengan peralatan utama (seperti Server Huawei Taishan) untuk mencapai inferensi yang cepat dan efisien, seperti klasifikasi citra, deteksi target, dll. Arsitektur sistem kartu akselerator AI Huawei Atlas 300 (Model 3000) ditunjukkan pada Gambar 6.29. Kartu akselerator AI Atlas 300 (Model 3000) digunakan dalam analisis video, OCR, pengenalan suara, pemasaran presisi, analisis citra medis, dan berbagai bidang lainnya.



Gambar 6.29 Arsitektur sistem kartu akselerator AI Huawei Atlas 300 (Model 3000)

Kartu akselerator AI Atlas 300 (Model 3000) umumnya digunakan dalam sistem pengenalan wajah. Sistem ini terutama menggunakan algoritma deteksi wajah, algoritma pelacakan wajah, algoritma penilaian kualitas wajah, dan algoritma pengenalan kontras wajah berkecepatan tinggi untuk mencapai pemodelan snapshot wajah secara real-time, alarm kontras daftar hitam secara real-time, pengambilan latar belakang wajah, dan fungsi lainnya.

Arsitektur sistem pengenalan wajah ditunjukkan pada Gambar 6.30. Komponen utamanya meliputi webcam HD front-end atau mesin snapshot wajah, server memori aliran media (opsional), server analisis kecerdasan wajah, server pencarian perbandingan wajah, server manajemen pusat, perangkat lunak manajemen klien, dll. Kartu akselerator AI Atlas 300 (Model 3000) digunakan dalam server analisis kecerdasan wajah, terutama untuk mencapai dekode/pra-pemrosesan video, deteksi wajah, penyelarasan (koreksi) wajah, serta ekstraksi dan inferensi fitur wajah.



Gambar 6.30 Arsitektur sistem pengenalan wajah

Spesifikasi produk kartu akselerator AI Atlas 300 (Model 3000) ditunjukkan pada Tabel 6.2. Fitur utama kartu akselerator AI Atlas 300 (Model 3000): mendukung antarmuka standar PCIe 3.0 × 16 HHHL dengan tinggi dan panjang setengah (slot tunggal); dengan konsumsi daya maksimum 67 W; mendukung fungsi pemantauan konsumsi daya dan manajemen out-of-band; mendukung kompresi dan dekompresi video perangkat keras H.264 dan H.265.

Tabel 6.2 Spesifikasi produk Kartu Akselerator AI Atlas 300 (Model 3000)

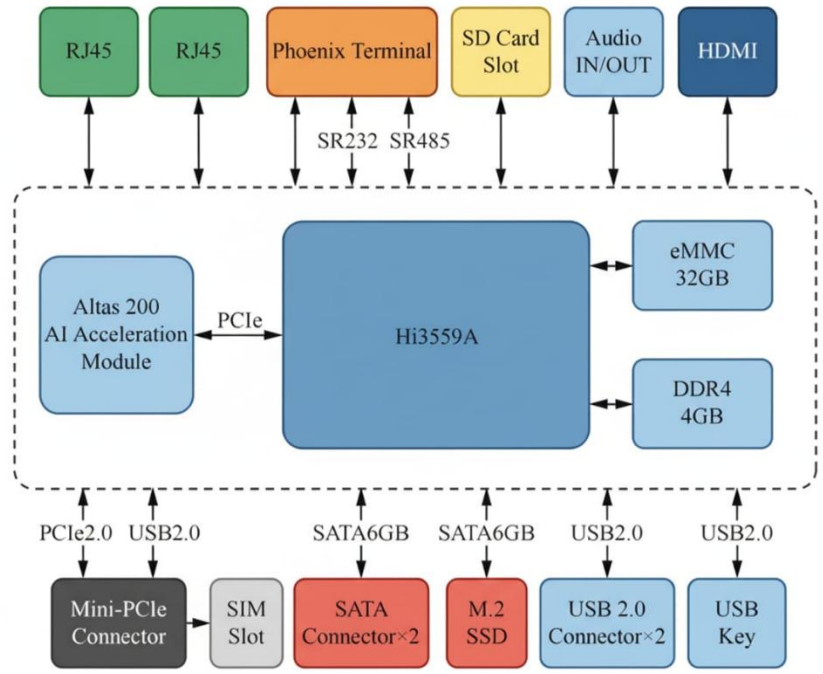
Item	Spesifikasi
Bentuk (Form)	Kartu PCIe dengan tinggi setengah dan panjang setengah
Memori	LPDDR4 × 32 GB, 3200 Mbit/s
Daya Komputasi AI	64 TOPS INT8
Codec	Mendukung *Hardware Decoding* H.264, 64-kanal 1080P 30FPS (2-kanal 3840 × 2160 60FPS) Mendukung *Hardware Decoding* H.265, 64-kanal 1080P 30FPS (2-kanal 3840 × 2160 60FPS) Mendukung *Hardware Decoding* H.264, 4-kanal 1080P 30FPS Mendukung *Hardware Decoding* H.265, 4-kanal 1080P 30FPS *JPEG Decoding* 4 × 1080P 256FPS, *Encoding* 4 × 1080P 64FPS *PNG Decoding* 4 × 1080P 48FPS
Antarmuka PCIe	PCIe Gen3.0, kompatibel dengan 2.0/1.0 ×16 *Lanes*, kompatibel dengan ×8/×4/×2/×1
Konsumsi Daya	67 W
Dimensi	169,5 mm × 68,9 mm
Berat	319 g
Suhu Operasional	0 °C ~ 55 °C (32 °F ~ 131 °F)

4. Stasiun Edge Cerdas Atlas 500

Atlas 500 Intelligent Edge Station terbagi menjadi dua model, Model 3000 dan Model 3010, yang dirancang untuk berbagai arsitektur CPU. Berikut beberapa fitur umum Atlas 500 Intelligent Edge Station. Perangkat ini merupakan perangkat edge ringan dari Huawei untuk berbagai skenario aplikasi edge, yang dicirikan oleh performa komputasi super, penyimpanan berkapasitas besar, konfigurasi fleksibel, ukuran kecil, rentang suhu yang luas, kesesuaian yang kuat dengan lingkungan sekitar, perawatan dan pengelolaan yang mudah, dll.

Atlas 500 Intelligent Edge Station adalah produk komputasi edge yang tangguh yang dapat melakukan pemrosesan real-time pada perangkat edge. Satu Atlas 500 Intelligent Edge Station dapat menyediakan daya pemrosesan 16 TOPS INT8 dengan konsumsi daya yang sangat rendah. Atlas 500 Intelligent Edge Station mengintegrasikan antarmuka data nirkabel Wi-Fi dan LTE untuk menyediakan solusi akses jaringan dan transmisi data yang fleksibel. Atlas 500 Intelligent Edge Station adalah pelopor dalam industri yang menerapkan teknologi pendinginan semikonduktor Thermo-electric Cooling (TEC) pada produk komputasi tepi skala besar, yang adaptif terhadap kondisi lingkungan yang keras. Arsitektur logis Atlas 500 Intelligent Edge Station ditunjukkan pada Gambar 6.31. Atlas 500 Intelligent Edge Station

menghadirkan kemudahan penggunaan adegan tepi dan analisis video 16 kanal serta kapasitas penyimpanan.



Gambar 6.31 Arsitektur logis Atlas 500 Intelligent Edge Station

(a) Kegunaan edge scene terutama mencakup aspek-aspek berikut.

- ✓ Real-time: Memberikan respons real-time dalam proses pemrosesan data.
- ✓ Bandwidth rendah: Hanya informasi yang diperlukan yang dikirim ke Cloud.
- ✓ Perlindungan privasi: Pelanggan dapat memutuskan informasi apa yang ingin mereka kirim ke Cloud dan menyimpannya secara lokal. Semua informasi yang dikirim ke Cloud dapat dienkripsi.
- ✓ Mendukung mesin kontainer standar, algoritma pihak ketiga, dan penerapan cepat aplikasi.

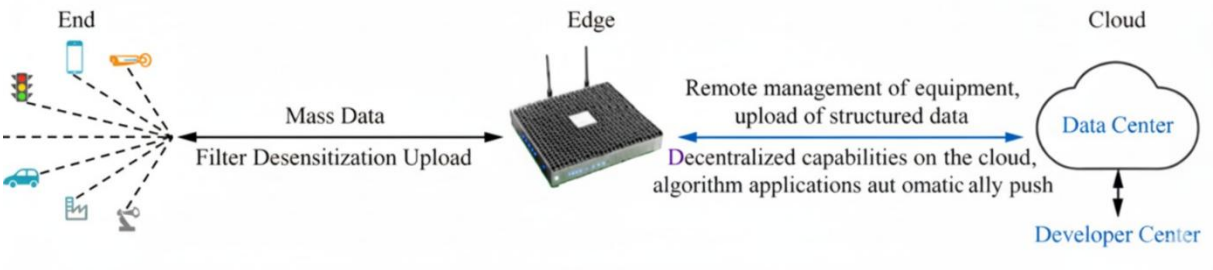
(b) Analisis video dan kapasitas penyimpanan 16 kanal terutama mencakup aspek-aspek berikut.

- ✓ Mendukung kemampuan analisis video 16 kanal (dekode 1080P maksimum 16 kanal, daya komputasi INT8 16T).
- ✓ Mendukung kapasitas memori 12 TB, dan video 16-saluran 1080P@4 MB Data Rate dapat di-cache selama 7 hari sementara video 8-saluran 1080P@4 MB Data Rate dapat di-cache selama 30 hari. Atlas 500 Intelligent Edge Station terutama diterapkan dalam pemantauan video cerdas, analisis, penyimpanan data dan pemandangan lainnya, termasuk kota yang aman, transportasi cerdas, komunitas cerdas, pemantauan lingkungan, manufaktur cerdas, perawatan cerdas, ritel swalayan, bangunan cerdas, dll. Dapat digunakan secara luas di berbagai perangkat edge dan ruang komputer pusat

untuk memenuhi kebutuhan keamanan publik, komunitas, taman, pusat perbelanjaan, supermarket dan area lingkungan kompleks lainnya Gunakan, seperti yang ditunjukkan pada Gambar 6.32.

Gambar 6.32 Skenario aplikasi Stasiun Tepi Cerdas ATLAS 500

Dalam skenario aplikasi ini, arsitektur umum Atlas 500 Intelligent Edge Station adalah sebagai berikut: Terminal, yang terhubung ke IPC (Kamera IP) atau perangkat front-end lainnya melalui nirkabel atau kabel; Edge, yang mencapai ekstraksi informasi nilai, penyimpanan dan pengunggahan; Cloud, yang melaluinya model pusat data di-push, dikelola, dikembangkan, dan diterapkan, seperti yang ditunjukkan pada Gambar 6.33.



Gambar 6.33 Arsitektur tipikal Stasiun Tepi Cerdas ATLAS 500

Spesifikasi produk Atlas 500 Intelligent Edge Station ditunjukkan pada Tabel 6.3.

Tabel 6.3 Spesifikasi produk Atlas 500 Intelligent Edge Station

Item	Spesifikasi
Prosesor AI	Dilengkapi dengan 1 Modul Akselerasi AI Atlas 200, menyediakan daya komputasi 16 TOPS INT8; mendukung dekode video HD 16 saluran
Internet	2 × 100 MB/1000 MB Port Ethernet Adaptif

Modul Nirkabel	Dukungan opsional untuk modul 3 GB/4 GB atau modul Wi-Fi, alternatif dengan antena ganda
Tampilan	Antarmuka HDMI satu saluran
Audio	Masukan dan keluaran satu saluran, konektor audio 3,5 mm, keluaran dua saluran
Daya	DC 12 V, adaptor daya eksternal
Suhu	−40 ~ 70 °C, tergantung pada konfigurasi

5. Server Inferensi Atlas 800

Server Inferensi Atlas 800 terbagi menjadi dua model: Model 3000 dan Model 3010.

(a) Server Inferensi Atlas 800 (Model 3000)

Server Inferensi Atlas 800 (Model 3000) adalah server inferensi pusat data berbasis prosesor Huawei Kunpeng 920. Server ini dapat mendukung delapan kartu akselerator AI Atlas 300 (Model 3000), memberikan kemampuan penalaran waktu nyata yang canggih, dan banyak digunakan dalam skenario inferensi AI. Server ini berorientasi pada internet, penyimpanan terdistribusi, komputasi awan, data besar, bisnis perusahaan, dan bidang lainnya, dengan keunggulan komputasi berkinerja tinggi, penyimpanan berkapasitas besar, konsumsi energi rendah, pengelolaan mudah, dan penerapan yang mudah, dan sebagainya.

Karakteristik kinerja Server Inferensi Atlas 800 (Model 3000) adalah sebagai berikut.

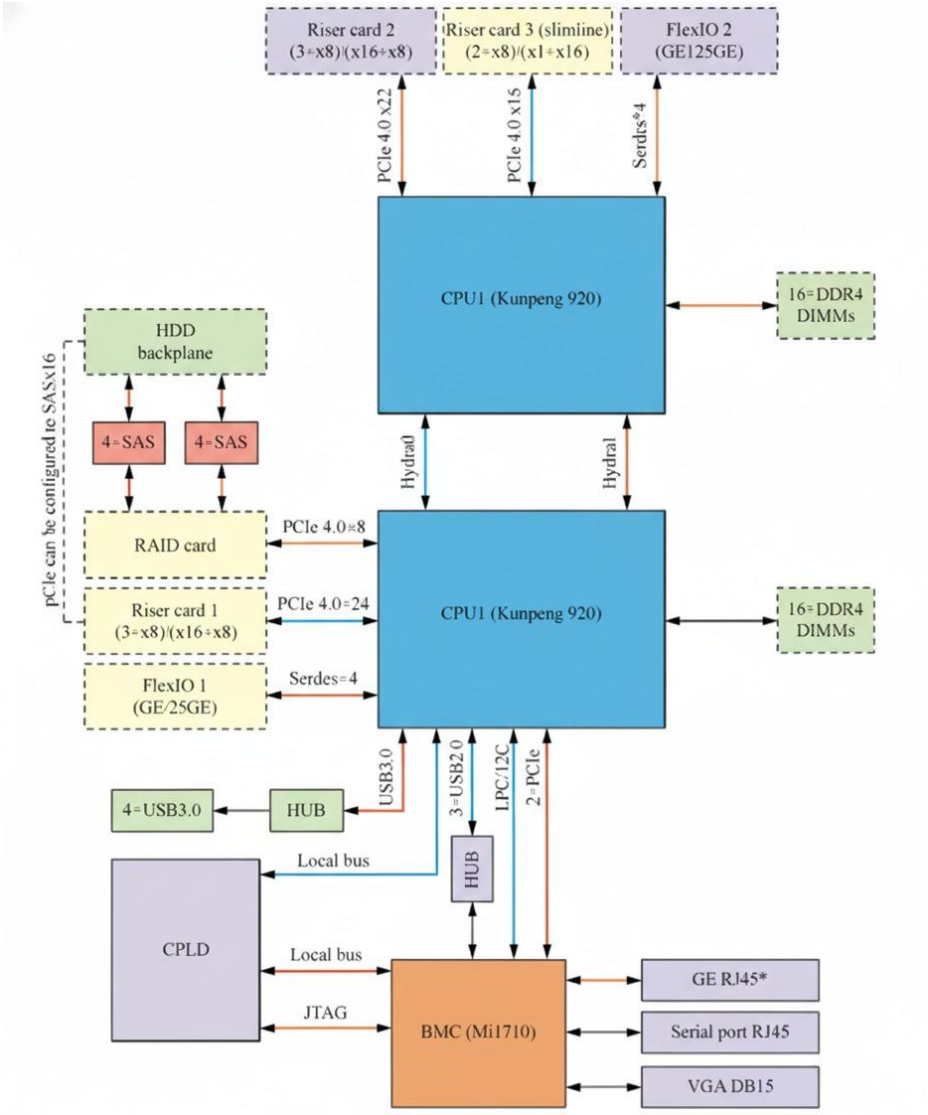
- Server Inferensi Atlas 800 (Model 3000) mendukung prosesor Kunpeng 920 multi-inti 64-bit berkinerja tinggi yang dikembangkan oleh Huawei dan ditujukan untuk bidang server. Server ini mengintegrasikan DDR4, PCIe4.0, 25GE, 10GE, GE, dan antarmuka lainnya, serta menyediakan fungsi SOC yang lengkap.
  - Server ini dapat mendukung kartu akselerator AI Atlas 300 (Model 3000), yang memberikan kemampuan penalaran waktu nyata yang andal.
  - Server ini dapat mendukung 64 inti dan frekuensi 3,0 GHz. Server ini juga mendukung berbagai model nomor dan frekuensi inti.
  - Server ini kompatibel dengan fitur arsitektur ARMv8-A, mendukung ekstensi ARMv8.1 dan ARMv8.2.
  - Inti server ini merupakan inti TaiShan 64-bit yang dikembangkan sendiri.

Karakteristik kinerja Atlas 800 Inference Server (Model 3000) adalah sebagai berikut.

- Atlas 800 Inference Server (Model 3000) mendukung prosesor Kunpeng 920 multi-core 64-bit berkinerja tinggi yang dikembangkan oleh Huawei dan ditujukan untuk bidang server. Prosesor ini mengintegrasikan DDR4, PCIe4.0, 25GE, 10GE, GE, dan antarmuka lainnya, serta menyediakan fungsi SOC yang lengkap.
  - ✓ Mendukung sebagian besar kartu akselerator AI Atlas 300 (Model 3000), yang memberikan kemampuan penalaran real-time yang andal.
  - ✓ Mendukung sebagian besar 64-core dan frekuensi 3,0 GHz. Server ini juga mendukung berbagai model jumlah dan frekuensi core.

- ✓ Kompatibel dengan fitur arsitektur ARMv8-A, mendukung ekstensi ARMv8.1 dan ARMv8.2.
- ✓ Core merupakan inti TaiShan 64-bit yang dikembangkan sendiri. – Setiap inti mengintegrasikan 64 KB L1 ICache, 64 KB L1 DCache, dan 512 KB L2 DCache.
- ✓ Dapat mendukung kapasitas cache L3 sebesar 45,5 MB ~ 46 Mb.
- ✓ Mendukung superskalar, panjang variabel, dan pipeline yang tidak berurutan.
- ✓ Mendukung koreksi kesalahan ECC 1 bit dan pelaporan kesalahan ECC 2 bit.
- ✓ Mendukung antarmuka Hydra antar-chip berkecepatan tinggi dengan kecepatan kanal maksimum 30 Gbit/dtk.
- ✓ Mendukung delapan pengontrol DDR.
- ✓ Mendukung hingga delapan port Ethernet fisik.
- ✓ Mendukung tiga pengontrol PCIe, GNE4 (16 Gbit/dtk), dan kompatibilitas ke bawah.
- ✓ Mendukung mesin pemeliharaan IMU untuk mengumpulkan status CPU.

- ❖ Satu Server Inferensi Atlas 800 (Model 3000) mendukung dua prosesor dan maksimum 128 inti, yang memaksimalkan eksekusi aplikasi multi-thread secara bersamaan.
- ❖ Atlas 800 Inference Server paling banyak dapat mendukung 32 buah memori DDR4 ECC 2933 MHZ, sedangkan memori dapat mendukung RDIMM dengan maksimum 4096 GB.



Gambar 6.34 Arsitektur logis Atlas 800 Inference Server (Model 3000)

Arsitektur logis Atlas 800 Inference Server (Model 3000) ditunjukkan pada Gambar 6.34, dan fitur-fiturnya adalah sebagai berikut.

- ✓ Mendukung dua prosesor Kunpeng 920 yang dikembangkan sendiri oleh Huawei, dan masing-masing prosesor mendukung 16 DIMM DDR4.
- ✓ CPU1 dan CPU2 saling terhubung melalui dua bus Hydra, dan kecepatan transmisi dapat mencapai maksimum 30 Gbit/dtk.
- ✓ Mendukung dua jenis kartu antarmuka Ethernet fleksibel, termasuk 4 × GE dan 4 × 25GE, melalui antarmuka Serdes berkecepatan tinggi CPU.
- ✓ Kartu kontrol RAID terhubung ke CPU1 melalui bus PCIe, dan terhubung ke backplane hard disk melalui kabel sinyal SAS. Platform ini dapat mendukung berbagai spesifikasi penyimpanan lokal melalui berbagai backplane hard disk.
- ✓ Baseboard Manager Controller (BMC) menggunakan Hi1710, sebuah chip manajemen yang dikembangkan oleh Huawei. Platform ini dapat digunakan untuk antarmuka

manajemen seperti Video Graphic Array (VGA), port jaringan manajemen, dan port serial debugging.

Atlas 800 Inference Server (Model 3000) adalah platform inferensi yang efisien berbasis prosesor Kunpeng. Spesifikasi produk ditunjukkan pada Tabel 6.4.

Tabel 6.4 Spesifikasi produk Atlas 800 Inference Server (Model 3000)

Item	Spesifikasi
Formulir	Server AI 2U
Model Prosesor	• Mendukung 2 kanal Prosesor Kunpeng 920, tersedia dalam 3 konfigurasi 64 inti, 48 inti, dan 32 inti, dengan frekuensi 2,6 GHz; • 2 Hydra Interconnected Links, kecepatan maksimum satu tautan adalah 30 gbit/dtk; • Volume L3 Cache adalah 45,5 MB ~ 46 MB; • Daya TDP Desain Termal CPU adalah 138 W ~ 195 W
Kartu Akselerator AI	Mendukung maksimal 8 Kartu Akselerator AI Atlas 300
Slot Memori	• Maksimum 32 slot memori DDR4, mendukung RDIMM; • Kecepatan memori maksimum 2933 MT/s; • Proteksi memori yang mendukung fungsi ECC, SEC/DED, SDDC, dan Patrol Scrubbing • Chip memori tunggal yang mendukung 16 GB/32 GB/64 GB/128 GB
Penyimpanan Lokal	• Konfigurasi Hard Disk 25 × 2,5 inci; • Konfigurasi Hard Disk 12 × 3,5 inci; • Konfigurasi Hard Disk 8 × 2,5 SAS/SATA + 12 × 2,5 NVMe inci
Tingkat RAID yang Didukung	RAID 0, RAID 1, RAID 5, RAID 6, RAID 10, RAID 50, RAID 60; Perlindungan Kegagalan Daya Super Kapasitor
Kartu IO Fleksibel	Papan tunggal ini mendukung maksimal dua kartu IO fleksibel. Satu kartu IO fleksibel menyediakan antarmuka jaringan berikut: • 4 Antarmuka Listrik GE, Mendukung Fungsi PXE; • 4 Antarmuka Optik 25GE/10GE, Mendukung Fungsi PXE
Ekstensi PCIe	Hingga 9 antarmuka PCIe4.0 PCIe didukung, satu di antaranya khusus untuk kartu RAID, dan 8 lainnya adalah slot ekstensi PCIe standar. Spesifikasi slot ekstensi PCIe 4.0 standar adalah sebagai berikut. Modul IO 1 dan modul IO 2 mendukung spesifikasi PCIe berikut: • Mendukung dua slot standar PCIe4.0 × 16 dengan tinggi dan panjang penuh (sinyalnya PCIe4.0 × 8) dan satu slot

	standar dengan tinggi dan setengah panjang (sinyalnya PCIe4.0 × 8) • Mendukung satu slot standar PCIe4.0 × 16 dengan tinggi dan panjang penuh, dan satu slot standar dengan tinggi dan setengah panjang penuh (sinyalnya PCIe4.0 × 8) • Modul IO 3 mendukung spesifikasi berikut. • Mendukung dua slot standar PCIe4.0 × 16 dengan tinggi dan panjang setengahnya (sinyalnya adalah PCIe4.0 × 8); • Mendukung satu slot standar PCIe4.0 × 16 dengan tinggi dan panjang setengahnya; Slot ekstensi PCIe mendukung kartu memori PCI SSD, yang dikembangkan secara independen oleh Huawei, sehingga meningkatkan kinerja I/O dalam pencarian, cache, pengunduhan, dan bidang aplikasi lainnya secara signifikan; Melalui kartu Riser khusus, slot PCI dapat mendukung Atlas 300 AI Accelerator (Model 3000), yang dikembangkan secara independen oleh Huawei, yang menjalankan tugas inferensi, pengenalan gambar, dan pemrosesan dengan cepat dan efisien.
Daya	2 modul daya AC 1500 W atau 2000 W hot-swap, mendukung redundansi 1 + 1
Catu Daya	Mendukung 100 V ~ 240 V AC, 240 V DC
Kipas	Mendukung 4 modul kipas hot-swap dan redundansi N + 1
Suhu	5 ◦C ~ 40 ◦C
Dimensi (Lebar × Dalam × Tinggi)	447 mm × 790 mm × 86,1 mm

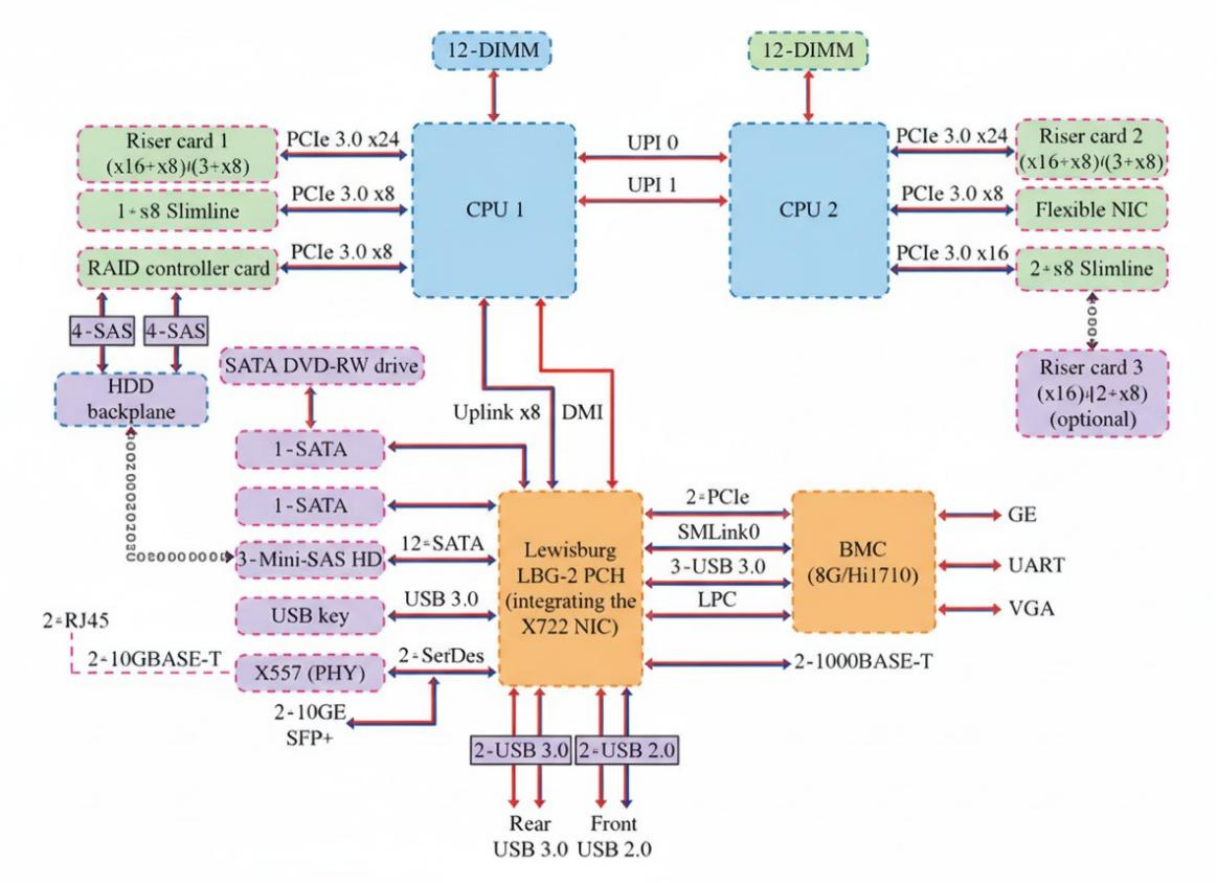
(b) Atlas 800 Inference Server (Model 3010)

Atlas 800 Inference Server (Model 3010) adalah platform inferensi berbasis prosesor Intel, yang banyak digunakan dalam skenario inferensi AI. Platform ini mendukung hingga tujuh akselerator AI Atlas 300 atau Nvidia T4 dan hingga 448 kanal analisis video HD secara real-time. Server Inferensi Atlas 800 (Model 3010) memiliki banyak keunggulan seperti konsumsi daya rendah, ekstensibilitas tinggi, keandalan tinggi, manajemen dan penerapan yang mudah, dll.

Arsitektur logis Server Inferensi Atlas 800 (Model 3010) ditunjukkan pada Gambar 6.35. Fitur-fitur Server Inferensi Atlas 800 (Model 3010) adalah sebagai berikut.

- Mendukung satu atau dua prosesor Intel Extreme Extensible.
- Mendukung 24 memori.
- Prosesor-prosesor tersebut terhubung melalui dua bus UltraPath Interconnect (UPI), dan laju transmisi dapat mencapai maksimum 10,4 GT/s.

- Prosesor terhubung dengan tiga kartu PCIe Riser melalui bus PCIe, dan mendukung berbagai spesifikasi slot PCIe melalui berbagai kartu PCIe Riser.
- Kartu Pengontrol Raid terhubung dengan CPU1 melalui bus PCIe, dan terhubung dengan backplane hard disk melalui kabel sinyal SAS. Kartu ini dapat mendukung berbagai spesifikasi penyimpanan lokal melalui berbagai backplane hard disk.
- Hub Pengontrol Platform (PCH) Lbg-2 digunakan untuk mendukung dua antarmuka berikut.
  - ✓ Mendukung dua antarmuka optik 10GE on-board atau dua antarmuka listrik 10GE on-board melalui X557 (PHY).
  - ✓ Mendukung dua antarmuka GE on-board.
  - ✓ Chip manajemen Hi1710 digunakan untuk mendukung VGA keluar, port jaringan manajemen, port serial debugging, dan antarmuka manajemen lainnya.



Gambar 6.35 Arsitektur logis Atlas 800 Inference Server (Model 3010)

Atlas 800 Inference Server (Model 3010) adalah platform inferensi fleksibel berbasis prosesor Intel, dan spesifikasi produknya ditunjukkan pada Tabel 6.5.

Tabel 6.5 Spesifikasi produk Atlas 800 Inference Server (Model 3010)

Item	Spesifikasi
Formulir	Prosesor AI 2U

Prosesor	Prosesor Intel® Xeon® SP Skylake atau Cascade Lake 1/2 inci, dengan daya maksimum 205 W
Akselerator AI	Hingga 7 akselerator AI Atlas 300 atau NVIDIA T4
Memori	24 slot memori DDR4, dengan kecepatan memori maksimum 2933 MT/s
Penyimpanan Lokal	Konfigurasi hard disk yang didukung: • Konfigurasi Hard Disk 8 × 2,5 inci; • Konfigurasi Hard Disk 12 × 3,5 inci; • Konfigurasi Hard Disk 20 × 2,5 inci; • Konfigurasi Hard Disk 24 × 2,5 inci; • Konfigurasi Hard Disk 25 × 2,5 inci. Penyimpanan Flash yang Didukung: • SSD M.2 Ganda
Level RAID yang Didukung	Dukungan opsional untuk RAID0, RAID1, RAID10, RAID1E, RAID5, RAID50, RAID6, RAI0, perlindungan superkapasitor cache, migrasi tingkat RAID, roaming disk, diagnosis mandiri, pengaturan jarak jauh Web, dan fungsi lainnya.
Internet	Kartu plug-in on-board: dua antarmuka 10GE dan dua antarmuka GE Kartu Fleksibel: Dapat dilengkapi secara opsional dengan 2 × GE atau 4 × GE atau 2 × 10GE atau 2 × 25GE atau 1/2 antarmuka FDR IB 56G
Ekstensi PCIe	Hingga 10 slot ekstensi PCIe3.0, termasuk satu kartu ekstensi PCIe untuk kartu RAID dan satu kartu LOM fleksibel
Kipas	4 kipas hot-swappable, redundansi N+1 didukung
Daya	2 catu daya redundan hot-swappable dapat dikonfigurasi untuk mendukung redundansi 1 + 1. Spesifikasi opsionalnya adalah sebagai berikut (Catatan 1): Catu daya platinum AC 550 W, catu daya platinum/titanium AC 900 W, catu daya platinum AC 1500 W; Catu daya DC tegangan tinggi 1500 W 380 V; Catu daya DC 1200 W - 48 V ~ -60 V
Suhu Operasional	5 ◦C ~ 45 ◦C
Dimensi (Lebar × Dalam × Tinggi)	Dimensi kotak hard disk 3,5 inci: 86,1 mm × 447 mm × 748 mm Dimensi kotak hard disk 2,5 inci: 86,1 mm × 447 mm × 708 mm

Atlas untuk Akselerasi Pelatihan AI

1. Kartu Pelatihan Atlas 300T: Kartu Pelatihan AI Tercanggih

Kartu Pelatihan Huawei Atlas 300T, atau Huawei Atlas 300 Accelerator (Model 9000), dirancang dan dikembangkan berdasarkan prosesor AI Ascend 910. Kartu ini menyediakan

daya komputasi AI FP16 hingga 256TOPS untuk skenario pelatihan pusat data, menjadikannya akselerator AI terkuat di industri. Kartu ini dapat digunakan secara luas di berbagai server umum di pusat data, memberikan solusi AI berkinerja tinggi, efisiensi tinggi, dan TCO rendah kepada pelanggan.

Berbasis prosesor AI Ascend 910, Huawei Atlas 300 Accelerator (Model 9000) memiliki beberapa fitur berikut.

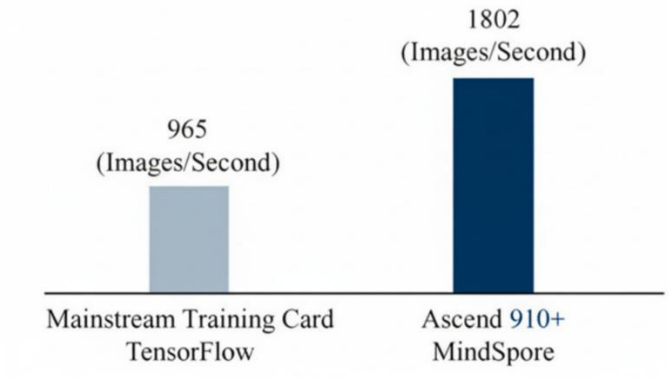
(a) Mendukung antarmuka standar PCIe 4.0 × 16 full-height dan 3/4 panjang (slot ganda).
(b) Konsumsi daya maksimum adalah 350 W.
(c) Mendukung pemantauan konsumsi daya dan manajemen out-of-band.
(d) Mendukung kompresi atau dekompresi video perangkat keras H.264/H.265.
(e) Mendukung kerangka kerja pelatihan Huawei MindSpore dan TensorFlow.
(f) Mendukung OS Linux pada platform x86.
(g) Mendukung OS Linux pada platform ARM.

Spesifikasi produk Huawei Atlas 300 Accelerator (Model 9000) ditunjukkan pada Tabel 6.6.

Tabel 6.6 Spesifikasi produk Huawei Atlas 300 Accelerator (Model 9000)

Item	Spesifikasi
Formulir	PCIe standar tinggi penuh dan panjang 3/4 inci
Memori	HBM 32 GB bawaan + Memori Besar Dua Tingkat 16 GB
Daya Komputasi AI	256T FLOPSa@FP16
Antarmuka PCIe	PCIe 4.0 × 16

Daya komputasi kartu tunggal Atlas 300 (Model 9000) meningkat dua kali lipat, dan penundaan sinkronisasi gradien berkurang 70%.



Gambar 6.36 Perbandingan kecepatan antara Huawei Ascend 910 + MindSpore dan lainnya

Gambar 6.36 menunjukkan perbandingan kecepatan antara kerangka kerja kartu pelatihan utama + TensorFlow dan Huawei Ascend 910 + MindSpore, dengan menggunakan ResNet 50 V1.5 untuk membandingkan hasil pada dataset ImageNet 2012 melalui "Ukuran batch optimal masing-masing". Terlihat bahwa kecepatan pelatihan Huawei Ascend 910 + MindSpore jauh lebih cepat daripada yang lain.

2. Atlas 800 Training Server: Server Pelatihan AI Tercanggih

Atlas 800 Training Server (Model 9000) terutama digunakan dalam adegan pelatihan AI dengan performa super untuk membangun platform komputasi AI dengan efisiensi tinggi dan konsumsi daya rendah untuk adegan pelatihan. Server ini mendukung beberapa Akselerator atau modul Akselerasi Atlas 300, yang beradaptasi dengan berbagai adegan analisis gambar video.

Server ini terutama digunakan dalam analisis video, pelatihan pembelajaran mendalam, dan pelatihan lainnya. Atlas 800 Training Server (Model 9000) berbasis prosesor Ascend 910. Kepadatan daya komputasinya meningkat 2,5 kali lipat, kapasitas dekode perangkat keras meningkat 25 kali lipat, dan rasio efisiensi energi meningkat 1,8 kali lipat. Atlas 800 Training Server (Model 9000) memiliki kepadatan daya komputasi terkuat, hingga daya komputasi super 2P FLOPS@FP16/4U. Atlas 800 Training Server (Model 9000) memiliki konfigurasi fleksibel dan dapat beradaptasi dengan berbagai beban. Server ini mendukung konfigurasi fleksibel kombinasi SAS/SATA/NVMe/M.2 SSD. Server ini juga mendukung kartu jaringan on-board dan kartu IO fleksibel, sehingga menyediakan beragam antarmuka jaringan. Spesifikasi produk Atlas 800 Training Server (Model 9000) ditunjukkan pada Tabel 6.7.

Tabel 6.7 Spesifikasi produk Atlas 800 Training Server (Model 9000)

Item	Spesifikasi
Formulir	Server AI 4U
Prosesor	4 prosesor Kunpeng 920
Daya Komputasi AI	2P FLOPS@FP16
Kemampuan Codec	32 dekoder perangkat keras terintegrasi Dapat diproses secara paralel dengan pelatihan
Pembuangan Panas	Dua jenis pembuangan panas didukung, pendinginan udara dan pendinginan cair
Konsumsi Daya	2P FLOPS/5,6 KW

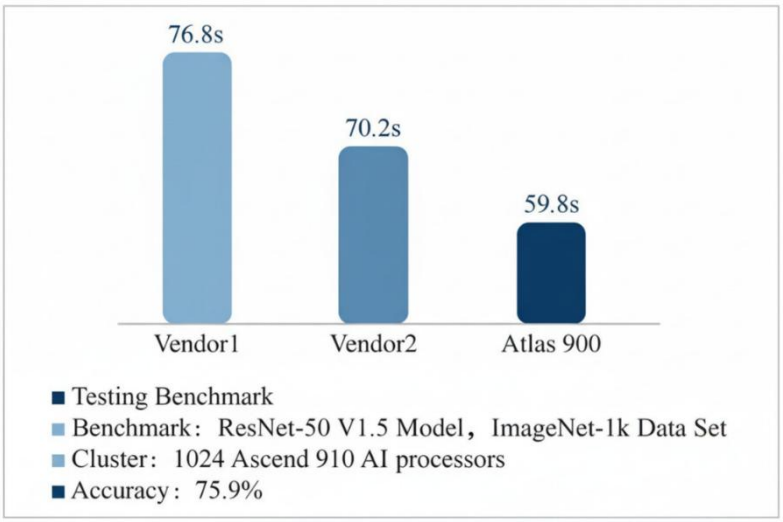
3. Atlas 900 AI Cluster: Cluster Pelatihan AI Tercepat di Dunia

Atlas 900 AI Cluster merepresentasikan daya komputasi puncak di dunia, yang terdiri dari ribuan prosesor AI Ascend 910. Melalui pustaka komunikasi klaster Huawei dan platform penjadwalan pekerjaan, klaster ini mengintegrasikan tiga antarmuka berkecepatan tinggi, yaitu HCCS, PCIe4.0, dan 100g RoCE, untuk sepenuhnya memaksimalkan kinerja prosesor AI Ascend 910 yang tangguh. Total daya komputasi mencapai 256p ~ 1024p FLOPS @FP16, yang setara dengan daya komputasi 500.000 PC.

Berdasarkan pengukuran aktual, Atlas 900 AI Cluster dapat menyelesaikan pelatihan berdasarkan model ResNet-50 dalam 60 detik, 15% lebih cepat daripada peringkat kedua, seperti yang ditunjukkan pada Gambar 6.37. Hal ini memungkinkan para peneliti untuk melatih model AI dengan gambar dan suara lebih cepat, sehingga manusia dapat lebih efisien menjelajahi misteri alam semesta, memprediksi cuaca, mengeksplorasi minyak, dan mempercepat proses komersial kendaraan otomatis.

Berikut adalah fitur-fitur utama Atlas 900 AI Cluster.

(a) Pemimpin industri daya komputasi: 256–1024 PFLOPS@FP16, ribuan prosesor AI Ascend 910 saling terhubung, memberikan kinerja ResNet-50@ImageNet tercepat di industri.



Gambar 6.37 Perbandingan kecepatan Atlas 900 AI Cluster dan klaster pelatihan lainnya

(b) Jaringan klaster terbaik: Tiga jenis antarmuka berkecepatan tinggi, HCCS, PCIe, dan RoCE 100 GB, diintegrasikan untuk mengintegrasikan pustaka komunikasi, topologi, dan jaringan dengan penundaan rendah secara vertikal dengan linearitas lebih dari 80%.

(c) Sistem pembuangan panas super: Sistem pendingin cair campuran 50 KW kabinet tunggal mendukung pendinginan cair lebih dari 95%, PUE kurang dari 1,1, menghemat ruang hingga 79%.

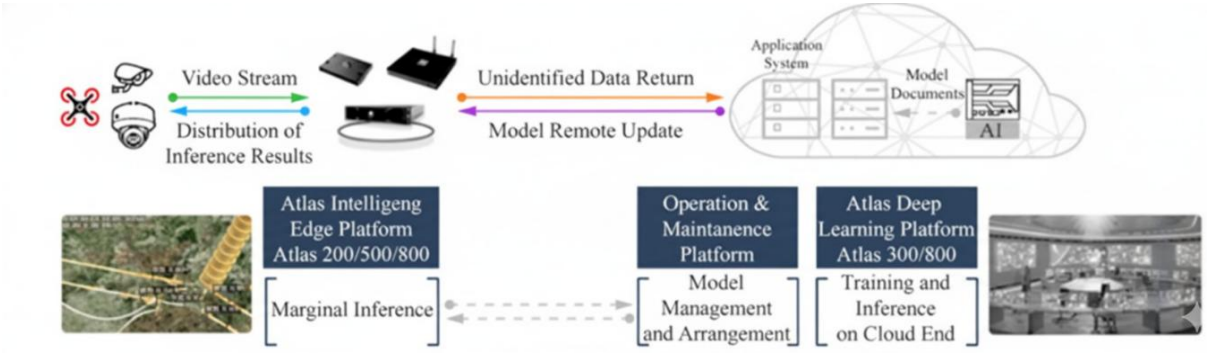
Untuk memungkinkan semua industri mendapatkan daya komputasi super, Huawei akan menerapkan Atlas 900 AI Cluster ke Cloud, meluncurkan layanan klaster Huawei Cloud EI, dan membuka aplikasinya untuk lembaga penelitian ilmiah dan universitas global dengan harga yang sangat kompetitif.

Kolaborasi Atlas Device-Edge-Cloud

Solusi komputasi AI Huawei Atlas memiliki tiga keunggulan dibandingkan solusi industri umum, yaitu pengembangan terpadu, operasi dan pemeliharaan terpadu, dan peningkatan keamanan. Umumnya, arsitektur pengembangan yang berbeda digunakan di sisi tepi dan sisi tengah industri, sehingga model memerlukan pengembangan sekunder karena tidak dapat mengalir dengan bebas. Huawei Atlas, berdasarkan arsitektur pengembangan terpadu dari Da Vinci Architecture dan Compute Architecture for Neural Networks (CANN), dapat digunakan di terminal, tepi dan cloud untuk pengembangan primer.

Umumnya, tidak ada alat manajemen operasi dan pemeliharaan di industri ini, sementara hanya API yang terbuka, sehingga pelanggan perlu mengembangkannya sendiri. Namun, FusionDirector dari Huawei Atlas dapat mengelola hingga 50.000 node untuk

mencapai manajemen terpadu perangkat pusat dan tepi, dan baik model push maupun peningkatan perangkat dapat dilakukan dari jarak jauh. Industri ini umumnya tidak memiliki mesin enkripsi dan dekripsi dan tidak mengenkripsi model, sementara Huawei Atlas mengenkripsi keamanan saluran transmisi dan model untuk perlindungan ganda. Kolaborasi AtlasDevice-Edge-Cloud, pelatihan sisi pusat yang konsisten, dan pembaruan mode jarak jauh ditunjukkan pada Gambar 6.38.



Gambar 6.38 Kolaborasi AtlasDevice-Edge-Cloud

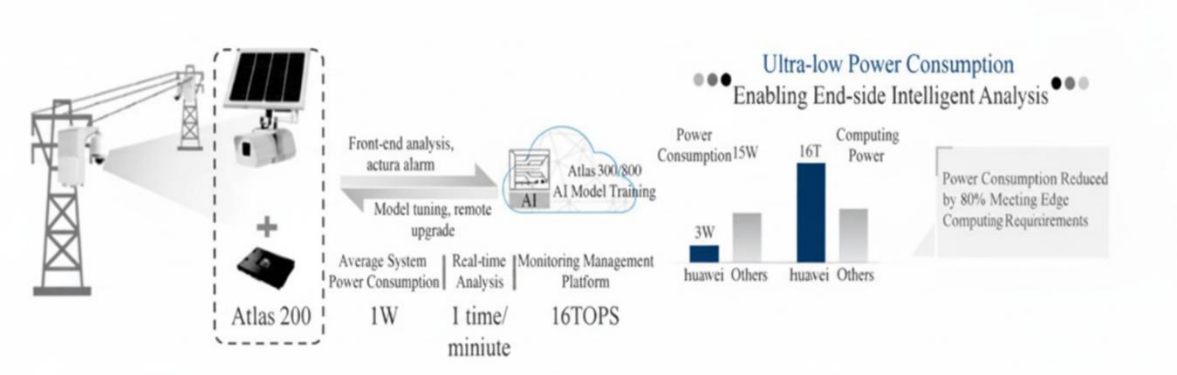
6.4 IMPLEMENTASI INDUSTRI ATLAS

Bagian ini terutama memperkenalkan skenario aplikasi industri solusi komputasi Atlas AI, seperti aplikasi di bidang ketenagalistrikan, keuangan, manufaktur, transportasi, superkomputer, dan bidang lainnya.

Ketenagalistrikan: Solusi Jaringan Cerdas TIK Terpadu

Dengan meningkatnya ketergantungan masyarakat modern terhadap listrik, cara pemanfaatan daya tradisional yang ekstensif dan tidak efisien tidak dapat memenuhi permintaan saat ini, sehingga dibutuhkan pasokan daya yang lebih efisien dan terjangkau. Bagaimana mencapai jaringan listrik yang andal, ekonomis, efisien, dan ramah lingkungan merupakan tantangan terbesar bagi industri ketenagalistrikan.

Dengan mengandalkan teknologi TIK terdepannya, Huawei, bersama para mitranya, telah meluncurkan serangkaian lengkap solusi bisnis cerdas yang mencakup semua aspek termasuk pembangkitan, transmisi, transformasi, distribusi, dan pemanfaatan daya. Sistem ketenagalistrikan tradisional terintegrasi secara mendalam dengan komputasi awan, data besar, Internet of Things, dan teknologi seluler untuk mencapai persepsi, interkoneksi, dan kecerdasan bisnis yang komprehensif dari berbagai terminal daya. Sebagai contoh, inspeksi patroli tanpa awak cerdas pionir (lihat Gambar 6.39) menggantikan inspeksi patroli manual tradisional, yang meningkatkan efisiensi operasional hingga lima kali lipat dan mengurangi biaya sistem hingga 30%. Kamera depan sistem inspeksi patroli tanpa awak cerdas ini dilengkapi dengan modul akselerasi Huawei Atlas 200, yang dapat menganalisis masalah dengan cepat dan mengirimkan alarm.



Gambar 6.39 Inspeksi patroli tanpa awak yang cerdas

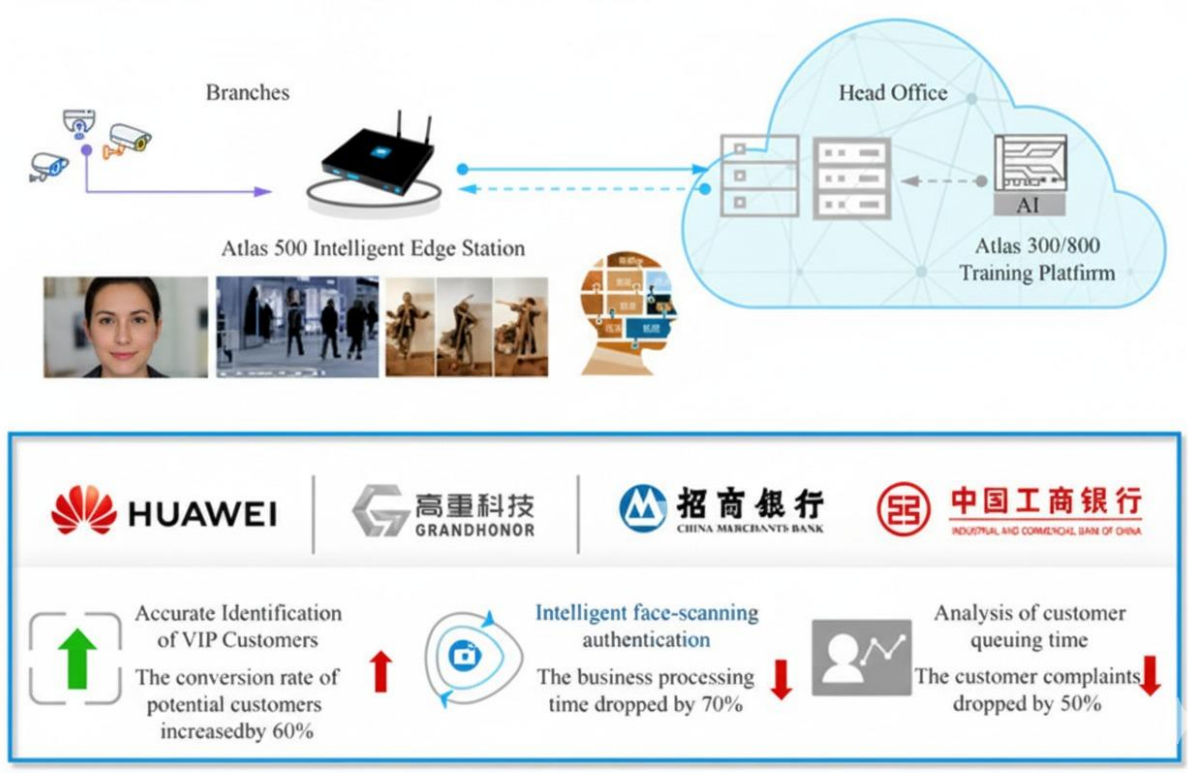
Pada saat yang sama, platform pemantauan dan manajemen jarak jauh yang dilengkapi dengan kartu pelatihan AI Atlas 300 atau server AI Atlas 800, yang digunakan untuk model pelatihan, dapat mewujudkan operasi peningkatan model dari jarak jauh.

Keuangan Cerdas: Transformasi Digital Holistik

Teknologi keuangan dan layanan keuangan digital telah menjadi bagian integral dari gaya hidup masyarakat, yang tidak terbatas pada pembayaran, tetapi juga investasi, simpanan, dan pinjaman. Salah satu solusi komputasi AI Huawei Atlas untuk industri keuangan adalah gerai perbankan pintar, yang mengadopsi solusi akses canggih, keamanan, dan teknologi all-in-one untuk membantu nasabah membangun generasi baru gerai perbankan pintar. Solusi komputasi AI Huawei Atlas menggunakan AI untuk mengubah keuangan, membantu gerai perbankan melakukan transformasi cerdas. Melalui identifikasi nasabah VIP yang akurat, tingkat konversi calon nasabah meningkat sebesar 60%; Melalui otentikasi pemindaian wajah yang cerdas, waktu pemrosesan bisnis berkurang sebesar 70%; Melalui analisis waktu antre pelanggan, keluhan pelanggan turun hingga 50%, seperti yang ditunjukkan pada Gambar 6.40.

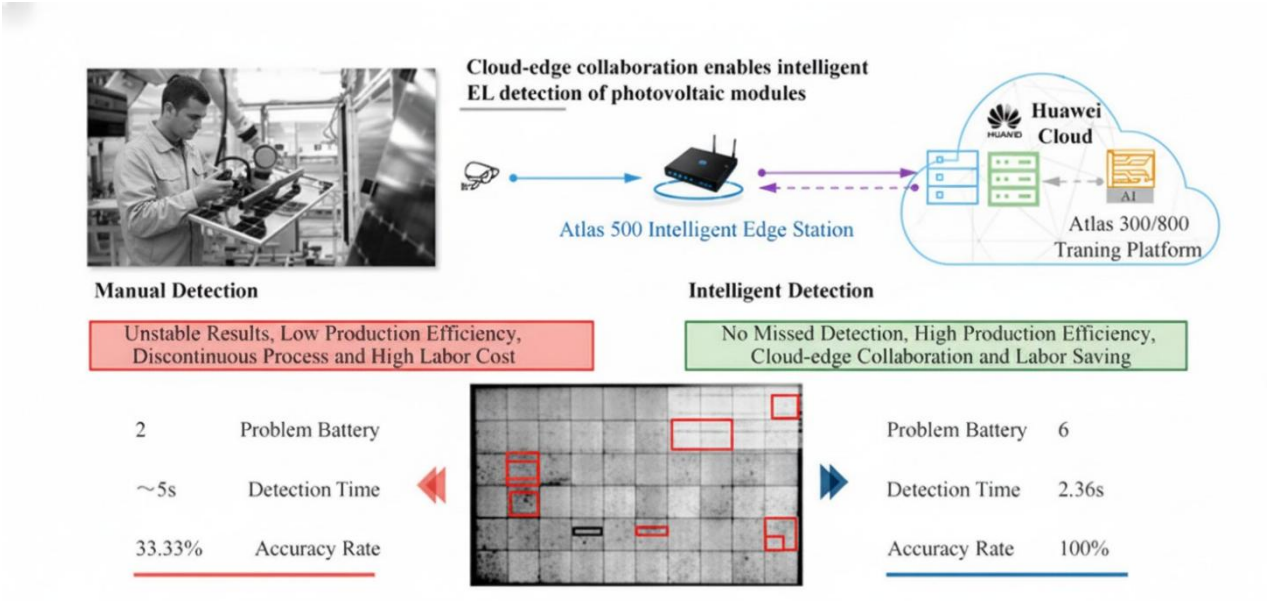
Manufaktur Cerdas: Integrasi Digital Mesin dan Pikiran

Di era industri 4.0, integrasi mendalam antara teknologi informasi generasi baru dan industri manufaktur telah membawa perubahan industri yang luas. Kustomisasi massal, desain kolaboratif global, pabrik cerdas berbasis Cyber Physical System (CPS), dan *Internet of Vehicles* (IoT) sedang membentuk kembali rantai nilai industri, membentuk mode produksi baru, bentuk industri, model bisnis, dan titik pertumbuhan ekonomi. Berbasis komputasi awan, big data, *Internet of Things* (IoT), dan teknologi lainnya, Huawei bekerja sama dengan mitra global untuk membantu pelanggan di industri manufaktur membentuk kembali rantai nilai industri manufaktur, berinovasi dalam model bisnis, dan mencapai penciptaan nilai baru.



Gambar 6.40 Keuangan cerdas transformasi cerdas cabang bank

Solusi komputasi AI Huawei Atlas membantu meningkatkan kecerdasan lini produksi, menggunakan teknologi visi mesin untuk deteksi cerdas, alih-alih deteksi manual tradisional. "Hasil yang tidak stabil, efisiensi produksi yang rendah, proses yang terputus-putus, dan biaya tenaga kerja yang tinggi" dari deteksi manual diubah menjadi "tanpa deteksi yang terlewat, efisiensi produksi yang tinggi, kolaborasi cloud-edge, dan penghematan tenaga kerja" dari deteksi cerdas, seperti yang ditunjukkan pada Gambar 6.41.

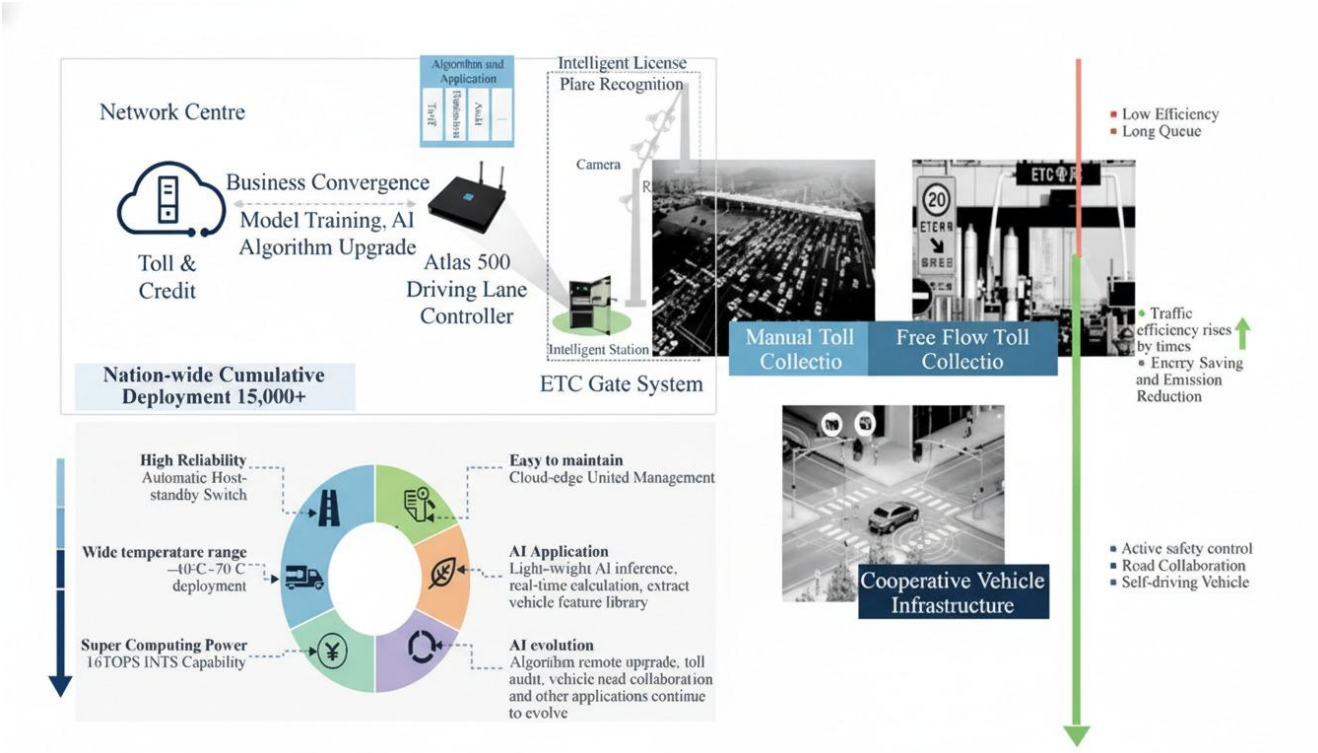


Gambar 6.41 Kolaborasi cloud-edge, deteksi cerdas

Transportasi Cerdas: Perjalanan Lebih Mudah dan Logistik yang Lebih Baik

Dengan percepatan globalisasi dan urbanisasi, permintaan transportasi terus meningkat dari hari ke hari, yang mendorong permintaan pembangunan sistem transportasi terintegrasi modern yang ramah lingkungan, aman, efisien, dan lancar. Berpegang pada konsep "Perjalanan Lebih Mudah dan Logistik yang Lebih Baik", Huawei berkomitmen untuk menyediakan solusi inovatif bagi pelanggan seperti kereta api digital, angkutan kereta api perkotaan digital, dan bandara pintar.

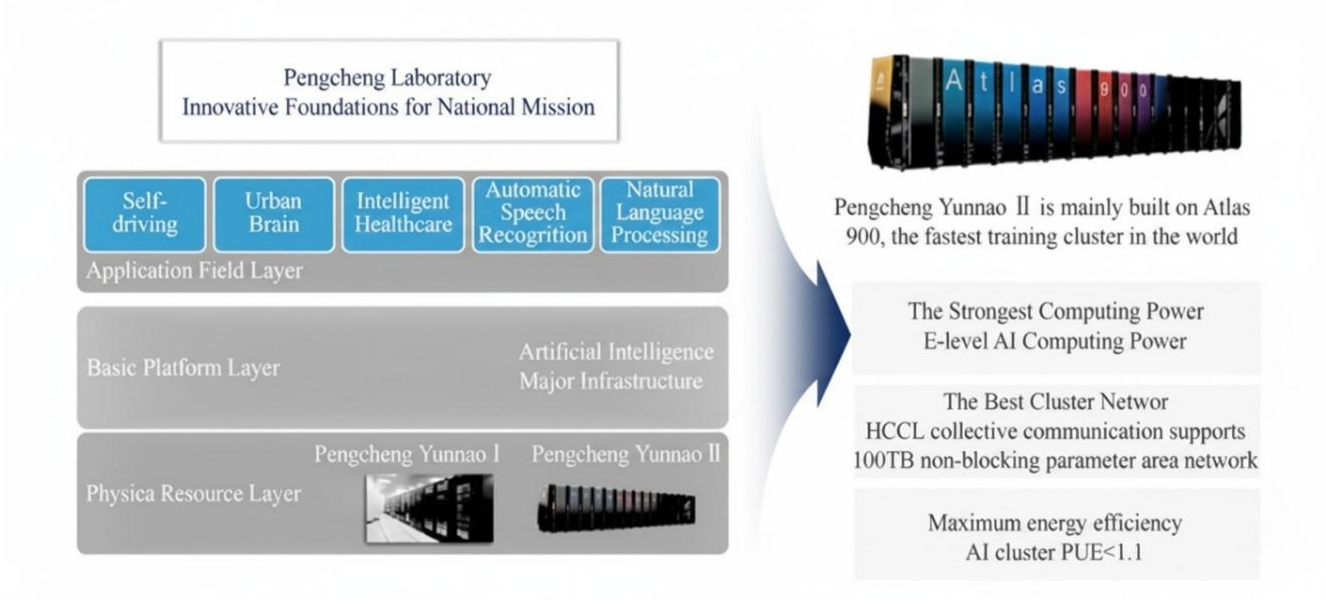
Melalui teknologi TIK baru seperti komputasi awan, big data, Internet of Things, jaringan tangkas, BYOD, eLTE, dan GSM-R, Huawei meningkatkan tingkat informasi industri dan membantu pelanggan industri meningkatkan layanan transportasi, memastikan perjalanan yang lebih mudah, logistik yang lebih baik, serta lalu lintas yang lebih lancar dan aman. Solusi komputasi AI Huawei Atlas membantu meningkatkan jaringan jalan tol nasional, sehingga efisiensi lalu lintas meningkat lima kali lipat melalui infrastruktur kendaraan kooperatif, seperti yang ditunjukkan pada Gambar 6.42.



Gambar 6.42 Infrastruktur kendaraan kooperatif, peningkatan efisiensi lalu lintas

Super Komputer: Membangun Platform AI Tingkat Negara

Pengcheng Yunnao II terutama dibangun di atas Atlas 900, klaster pelatihan tercepat di dunia, dengan daya komputasi terkuat (daya komputasi AI tingkat E), jaringan klaster terbaik (komunikasi kolektif HCCL mendukung jaringan area parameter non-blocking 100 TB), dan efisiensi energi tertinggi (PUE klaster AI < 1,1).



Gambar 6.43 Laboratorium Pengcheng

Atlas membantu Pengcheng Yunnao II mendirikan Laboratorium Pengcheng, yang merupakan platform inovasi dasar untuk mewujudkan misi nasional, seperti yang ditunjukkan pada Gambar 6.43.

6.5 KESIMPULAN

Bab ini berfokus pada prosesor AI Huawei Ascend dan solusi komputasi AI Atlas. Pertama, struktur perangkat keras dan perangkat lunak prosesor AI Ascend diperkenalkan, kemudian produk inferensi dan produk pelatihan terkait dari solusi komputasi AI Atlas diperkenalkan. Terakhir, skenario aplikasi industri Atlas diperkenalkan.

Latihan Soal

1. Sebagai dua prosesor untuk komputasi AI, apa perbedaan antara CPU dan GPU?
2. Arsitektur Da Vinci dikembangkan secara khusus untuk meningkatkan daya komputasi AI. Arsitektur ini bukan hanya mesin komputasi AI Ascend, tetapi juga inti dari prosesor AI Ascend. Apa tiga komponen utama Arsitektur Da Vinci?
3. Apa tiga jenis sumber daya komputasi dasar yang terdapat dalam unit komputasi Arsitektur Da Vinci?
4. Tumpukan perangkat lunak prosesor AI Ascend terutama dibagi menjadi empat tingkat dan satu rantai alat bantu. Apa saja keempat tingkat tersebut? Kemampuan bantu apa saja yang disediakan oleh rantai alat tersebut?
5. Alur perangkat lunak jaringan saraf tiruan prosesor AI Ascend merupakan jembatan antara kerangka kerja pembelajaran mendalam dan prosesor AI Ascend. Alur ini menyediakan jalan pintas bagi jaringan saraf tiruan untuk bertransformasi dari model asli, ke representasi grafik komputasi menengah, dan kemudian ke model luring independen. Alur perangkat lunak jaringan saraf tiruan prosesor AI Ascend terutama

menyelesaikan pembuatan, pemuatan, dan eksekusi model luring aplikasi jaringan saraf tiruan. Apa saja modul fungsi utama yang terkandung di dalamnya?

6. Prosesor AI Ascend dibagi menjadi dua jenis, yaitu Ascend 310 dan Ascend 910, yang keduanya berbasis Arsitektur Da Vinci. Namun, keduanya berbeda dalam hal presisi, konsumsi daya, dan proses. Apa perbedaan dalam bidang aplikasinya?
7. Produk-produk terkait dari solusi komputasi AI Atlas terutama mencakup inferensi dan pelatihan. Apa saja produk penalaran dan pelatihan?
8. Ilustrasikan skenario penerapan solusi komputasi Atlas AI.

BAB 7
PLATFORM TERBUKA HUAWEI ARTIFICIAL INTELLIGENCE (HIAI)

Bab ini merupakan pengantar HUAWEI HiAI, sebuah platform kapabilitas AI terbuka untuk perangkat pintar. Pertama, Arsitektur Platform HUAWEI HiAI, sebuah arsitektur terbuka tiga lapis berbasis "Layanan, Mesin, dan Fondasi" yang menyediakan kapabilitas pada chip, aplikasi, dan layanan, serta tiga submodulnya, diperkenalkan. Kemudian, pengembangan Aplikasi berbasis HUAWEI HiAI dan beberapa solusi HUAWEI HiAI juga diperkenalkan secara singkat.

7.1 PENGANTAR PLATFORM HUAWEI HIAI

Saat ini, sebagian besar konsumen terpapar pada Aplikasi AI seperti asisten suara, fotografi AI, dan penghias gambar, yang skenario aplikasinya relatif tunggal dan terbatas. Faktanya, dengan evolusi AI on-engine menjadi AI terdistribusi dan pembagian sumber daya serta daya komputasi di antara beberapa terminal, skenario aplikasi AI on-engine akan sangat diperluas, yang selanjutnya akan memberdayakan pengembang untuk mencapai inovasi yang lebih cerdas dan menghadirkan pengalaman luar biasa bagi konsumen. Berdasarkan latar belakang di atas, Huawei meluncurkan HUAWEI HiAI 3.0. Evolusi platform HUAWEI HiAI telah mengalami perkembangan versi 1.0 untuk perangkat tunggal, versi 2.0 untuk multiperangkat, dan versi 3.0 untuk skenario terdistribusi saat ini, seperti yang ditunjukkan pada Gambar 7.1.

Gambar 7.1 Evolusi platform HUAWEI HiAI

AWEI HiAI 3.0 resmi dirilis pada Konferensi Pengembang Software Green Alliance pada 19 November 2019, menandai bahwa AI on-engine secara resmi bergerak menuju AI terdistribusi. HUAWEI HiAI 3.0 akan menghadirkan pengalaman hidup cerdas yang luar biasa bagi pengguna. HUAWEI HiAI 3.0 menyediakan layanan satu akses dan operasi adaptif multiterminal. Pengguna dapat menikmati layanan praktis seperti asisten suara dan HiBoard di ponsel, tablet, layar pintar, speaker pintar, dan perangkat lainnya, sehingga layanan yang

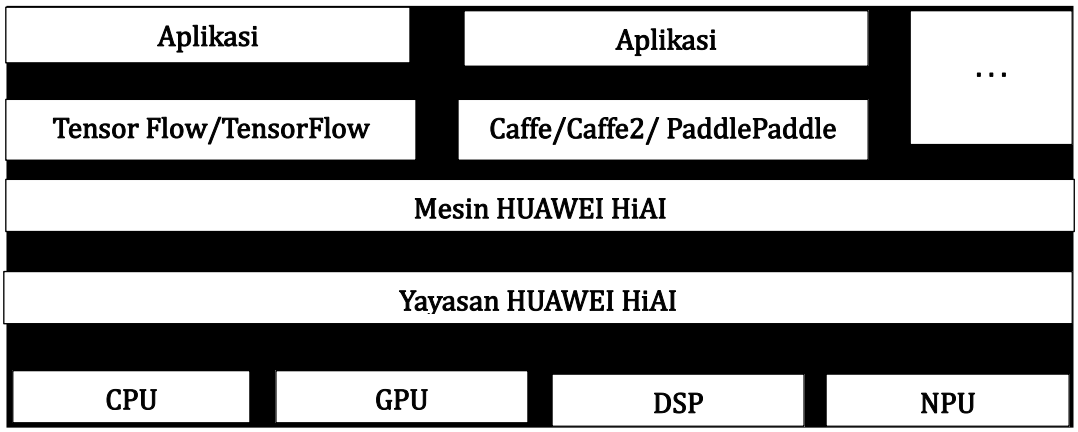
sama dapat diterapkan di perangkat yang berbeda. Berikut adalah dua contoh kasus: pelatihan privat dan pengalaman berkendara.

- *Kasus 1*: Pelatihan privat. HUAWEI HiAI 3.0 membuka Computer Vision (CV) terdistribusi dan Pengenalan Ucapan Otomatis (ASR), yang dirancang untuk memungkinkan pengguna berolahraga di rumah seefektif dipandu oleh pelatih pribadi di pusat kebugaran. Distributed Computer Vision dapat mengenali titik-titik kunci tubuh manusia 3D sehingga pengguna dapat menangkap berbagai sudut postur gerak secara real-time melalui beberapa kamera di rumah, dan memperbaiki postur melalui beberapa tampilan layar. Selain itu, Pengenalan Ucapan Otomatis membantu pengguna mengontrol ritme gerak mereka dan selanjutnya membantu mereka menikmati pelatihan pribadi di rumah.
- *Kasus 2*: Pengalaman berkendara. Dikombinasikan dengan teknologi terdistribusi, HUAWEI HiAI 3.0 memungkinkan pengguna untuk menghubungkan ponsel pintar dengan mobil, sehingga deteksi keselamatan perilaku berkendara pengguna dilakukan melalui kamera di dalam mobil, dan daya komputasi chip AI memberikan pengingat keselamatan untuk perilaku berbahaya seperti mengemudi dalam kondisi kelelahan. Melalui lingkungan jaringan di dalam mobil dan operasi data lokal dengan penundaan yang lebih rendah, pengemudi dapat lebih melindungi diri dari kecelakaan.

Arsitektur Platform HUAWEI HiAI

Platform HUAWEI HiAI membangun ekosistem tiga lapis yang terdiri dari "Layanan, Mesin, dan Fondasi". Layanan mendukung kerangka kerja antarmuka utama yang kaya, sementara Mesin menyediakan berbagai API bisnis fungsional tingkat atas yang dapat berjalan secara efisien di perangkat seluler, dan Fondasi secara fleksibel menjadwalkan sumber daya heterogen untuk memenuhi kebutuhan pengembang guna mempercepat kalkulasi model jaringan saraf tiruan dan kalkulasi operator.

Selain itu, HUAWEI HiAI menyediakan rantai alat yang sistematis, dokumen yang lengkap, beragam API, dan kode sumber yang mudah digunakan, yang memungkinkan pengembangan aplikasi yang cepat. Arsitektur platform komputasi seluler HUAWEI HiAI ditunjukkan pada Gambar 7.2.



Gambar 7.2 Arsitektur platform komputasi seluler HUAWEI HiAI

HUAWEI HiAI adalah platform komputasi AI berorientasi terminal seluler. Dibandingkan dengan AI on-service, AI on-engine memiliki tiga keunggulan inti: keamanan yang lebih baik, biaya yang lebih rendah, dan penundaan yang lebih sedikit. HUAWEI HiAI membangun tiga lapis Ekologi AI: kapabilitas layanan terbuka, kapabilitas aplikasi terbuka, dan kapabilitas chip terbuka. Platform terbuka tiga lapis "Layanan, Engine, dan Fondasi" menghadirkan pengalaman yang lebih luar biasa bagi pengguna dan pengembang. Setiap lapis memiliki fitur-fitur sebagai berikut.

- Layanan: Buat sekali, gunakan kembali berkali-kali.
- Engine: Terdistribusi, skenario lengkap.
- Fondasi: Daya komputasi yang lebih besar, lebih banyak operator; lebih banyak frame, model yang lebih kecil.

Ekologi AI tiga lapis HiAI ditunjukkan pada Gambar 7.3.

Lapisan	Komponen	Kemampuan Terbuka	Deskripsi
Service (Layanan)	HUAWEI HiAI Service	Kemampuan Layanan Terbuka (Open Service Capability)	Menyediakan layanan secara tepat waktu dan sesuai dengan kebutuhan pengguna, serta membantu layanan untuk secara aktif menemukan pengguna.
Engine (Mesin)	HUAWEI HiAI Engine	Kemampuan Aplikasi Terbuka (Open Application Capability)	Memudahkan integrasi berbagai kemampuan AI dengan aplikasi, sehingga aplikasi menjadi lebih cerdas dan kuat.
Foundation (Fondasi)	HUAWEI HiAI Foundation	Kemampuan Chip Terbuka (Open Chip Capability)	Memungkinkan transformasi dan migrasi cepat dari model yang ada, serta memperoleh performa terbaik dengan bantuan penjadwalan heterogen dan akselerasi NPU.

Gambar 7.3 Ekologi AI tiga lapis HiAI

HUAWEI HiAI memungkinkan Aplikasi untuk memiliki nilai-nilai berikut: waktu nyata, ketepatan waktu, stabilitas, keamanan, dan biaya. Fitur terbesar dari platform HUAWEI HiAI 3.0 adalah AI yang memungkinkan skenario penuh terdistribusi. Arsitektur HUAWEI HiAI terdiri dari tiga lapisan: Layanan, Mesin, dan Fondasi. Submodul yang sesuai dari Perangkat adalah Layanan, yang terutama untuk membuka kemampuan layanan. Ini akan mendorong layanan tepat waktu dan sesuai dengan kebutuhan pengguna, sehingga layanan dapat secara aktif menemukan pengguna. Apa yang dibawanya kepada pengguna adalah untuk membuat sekali dan menggunakan kembali berkali-kali.

Submodul yang sesuai dengan Mesin disebut HiAI Engine, yang terutama menyediakan API untuk membuka kemampuan aplikasi AI. Ini dapat dengan mudah mengintegrasikan berbagai kemampuan AI dengan Aplikasi, membuat Aplikasi lebih cerdas dan kuat. Melalui

mesin HiAI, Anda dapat memanggil berbagai algoritma di platform HiAI dan mengintegrasikannya di Aplikasi. Misalnya, jika Anda ingin mencapai pengenalan gambar, pengenalan karakter, pengenalan wajah, pengenalan ucapan, pemahaman bahasa alami, Anda dapat langsung memanggil API di HiAI Engine. Mesin HiAI dapat digunakan dalam skenario terdistribusi dan penuh. Fondasi adalah batch chip, terutama didasarkan pada chip Kirin Huawei, dengan kemampuan chip terbuka.

Sub-modul yang berkaitan dengan Foundation disebut HiAI Foundation, yang utamanya bertanggung jawab untuk menyediakan operator. HiAI Foundation dapat dengan cepat mentransformasi dan memigrasikan model yang ada, serta mencapai performa terbaik dengan bantuan penjadwalan heterogen dan akselerasi NPU. Chip ini menyediakan lebih banyak operator, daya komputasi yang lebih besar, dan lebih banyak kerangka kerja untuk menyederhanakan model. Jika Anda ingin memigrasikan beberapa aplikasi AI yang telah dikembangkan secara lokal ke perangkat terminal, Anda dapat menggunakan HiAI Foundation untuk mentransformasi model agar sesuai dengan perangkat terminal. Ketiga sub-modul tersebut dijelaskan di bawah ini.

HUAWEI HiAI Foundation

HiAI Foundation AP adalah Pustaka Komputasi Kecerdasan Buatan (AI) dalam platform komputasi seluler, yang dirancang untuk pengembang aplikasi kecerdasan buatan. API ini memungkinkan pengembang untuk menulis aplikasi kecerdasan buatan yang berjalan di perangkat seluler dengan mudah dan efisien. Fitur-fiturnya adalah sebagai berikut:

1. Berdasarkan peningkatan kinerja dan presisi tinggi chip Kirin yang berkelanjutan, AP ini memberikan kinerja AI yang lebih baik dengan daya komputasi yang lebih besar.
2. Jumlah operator yang didukung melebihi 300, yang merupakan jumlah terbesar di industri, dan lebih banyak kerangka kerja yang didukung, sehingga fleksibilitas dan kompatibilitasnya telah ditingkatkan secara signifikan.
3. Chip Honghu, chip Kirin, chip Kamera AI, dan chip full scene memungkinkan lebih banyak kemampuan AI pada perangkat.

API HiAI Foundation akan dirilis sebagai berkas biner terpadu. Rangkaian API ini bertujuan untuk mempercepat kalkulasi jaringan neural melalui platform komputasi heterogen HiAI. Saat ini, API ini hanya mendukung SoC Kirin.

Dengan menggunakan API HiAI Foundation, pengembang dapat berfokus pada pengembangan aplikasi AI baru, alih-alih optimasi kinerja komputasi. API HiAI Foundation terintegrasi pada chip SoC Kirin, yang menyediakan lingkungan kerja dan alat debugging berbasis perangkat seluler bagi pengembang. Pengembang dapat menjalankan model jaringan neural di perangkat seluler dan memanggil API HiAI Foundation untuk mempercepat komputasi. API HiAI Foundation tidak perlu diinstal karena mendukung integrasi, pengembangan, dan verifikasi yang relevan dengan menggunakan citra bawaan perangkat seluler.

Dua fungsi utama berikut disediakan oleh API HiAI Foundation bagi pengembang aplikasi AI.

1. Menyediakan API bisnis AI umum yang dapat berjalan secara efisien di perangkat seluler.
2. Menyediakan API akselerasi yang independen dari perangkat keras prosesor, sehingga produsen dan pengembang aplikasi dapat mempercepat komputasi model dan komputasi operator pada sistem akselerasi heterogen HiAI.

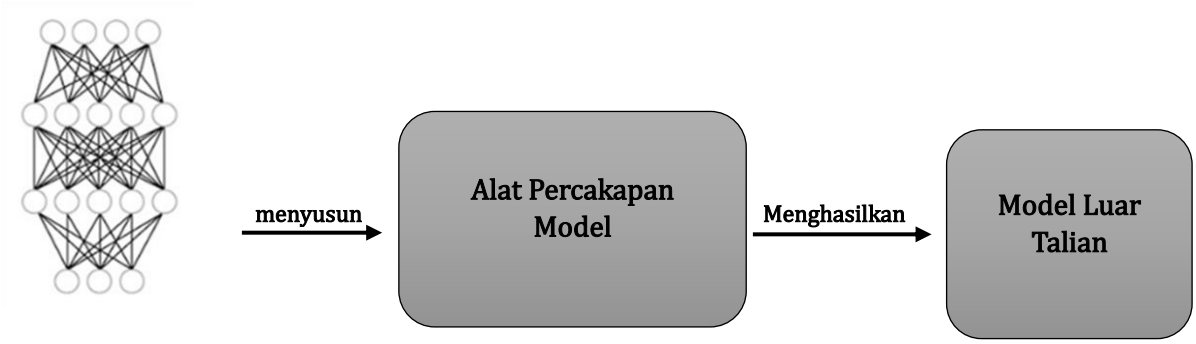
Fungsi dasar berikut didukung oleh API HiAI Foundation.

1. Mendukung antarmuka manajemen model AI seperti kompilasi model, pemuatan model, pengoperasian model, dan penghancuran model.
2. Mendukung antarmuka komputasi operator dasar, termasuk konvolusi, penggabungan, dan tautan penuh.

HiAI Foundation mendukung set instruksi AI khusus untuk operasi model jaringan neural, sehingga lebih banyak operator jaringan neural dapat dieksekusi secara efisien dan paralel dengan siklus clock terkecil.

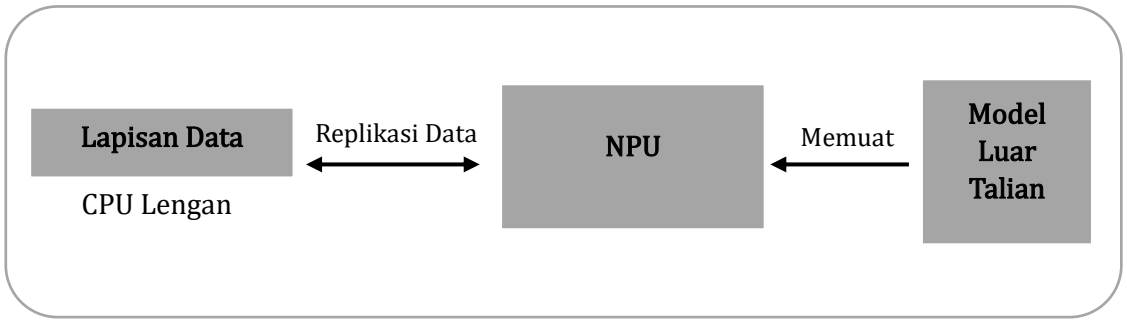
HiAI Foundation melakukan kompilasi luring (offline) berbagai operator jaringan neural, seperti konvolusi, penggabungan, aktivasi, dan tautan penuh ke dalam urutan instruksi AI khusus NPU melalui berbagai alat. Pada saat yang sama, HiAI Foundation menata ulang data dan bobot, serta menggabungkan instruksi dan data untuk menghasilkan model eksekusi luring. Saat mengompilasi secara luring (offline), operator yang dapat digabungkan antara lapisan depan dan belakang (konvolusi, fungsi aktivasi Relu, pooling) dapat digabungkan antar lapisan. Metode ini mengurangi bandwidth baca-tulis DDR dan meningkatkan kinerja.

HiAI Foundation menyusun ulang data yang relevan (Batch, Channel, Height, Width) dalam model jaringan saraf tiruan secara efisien, terutama data kanal dari grafik karakteristik. Selama operasi konvolusi, efisiensi perhitungan yang terkait dengan kanal meningkat pesat. HiAI Foundation mendukung akselerasi model sparse. Dengan asumsi tidak ada kehilangan akurasi perhitungan, bobot ditetapkan ke nol dan optimasi sparse dilakukan. NPU melewati operasi perkalian dan penjumlahan dengan koefisien nol, yang sangat meningkatkan efisiensi perhitungan dan mengurangi bandwidth. Gambar 7.4 menunjukkan bahwa model jaringan saraf tiruan yang telah dilatih dihasilkan oleh alat kompiler, yang dieksekusi secara efisien di HiAI Foundation, dan disimpan sebagai berkas biner model luring (offline).



Gambar 7.4 Model jaringan saraf tiruan yang dikompilasi menjadi model luring

Model jaringan saraf tiruan standar, seperti Caffe, dikompilasi dan diubah menjadi model luring. Tujuan kompilasi adalah untuk mengoptimalkan konfigurasi jaringan dan menghasilkan berkas objek yang dioptimalkan, yaitu model luring. Model luring disimpan secara serial di disk. Ketika jaringan saraf tiruan digunakan untuk kalkulasi maju, berkas objek yang dioptimalkan langsung digunakan untuk kalkulasi, sehingga kecepatannya lebih tinggi. Gambar 7.5 menunjukkan bahwa dalam kalkulasi model luring, model luring dimuat dari berkas, dan data masukan pengguna (seperti gambar) disalin ke NPU HiAI untuk kalkulasi. Dalam proses kalkulasi, setiap inferensi hanya perlu mengimpor dan mengekspor data pengguna dari DDR ke NPU satu kali.



Gambar 7.5 Perhitungan pemuatan model luring

HUAWEI HiAI Foundation mendukung beragam kerangka kerja platform cerdas, termasuk Caffe dan TensorFlow, dan berbagai kerangka kerja platform cerdas digunakan. Pihak ketiga perlu menunjukkan kerangka kerja platform cerdas spesifik yang akan digunakan dalam perhitungan ini di antarmuka, dan antarmuka serta parameter lainnya tidak perlu dimodifikasi.

HUAWEI HiAI Foundation mendukung sebagian besar model dan operator jaringan saraf tiruan, serta terus mengoptimalkan dan menyempurnakannya.

HUAWEI HiAI Engine

HUAWEI HiAI Engine, sebagai platform terbuka dengan kapabilitas aplikasi, dengan mudah mengintegrasikan beragam kapabilitas AI dengan Aplikasi, menjadikan Aplikasi lebih cerdas dan bertenaga. HUAWEI HiAI Engine 3.0 menambahkan beberapa kapabilitas pengenalan API berdasarkan yang sebelumnya, sehingga jumlah API yang mendasarinya melebihi 40. HUAWEI HiAI Engine tidak hanya memungkinkan pengguna untuk langsung memanggil API yang ada, tetapi juga membantu pengembang untuk fokus pada pengembangan bisnis. Untuk mencapai pengenalan gambar, pemrosesan suara, dan fungsi lainnya, pengguna hanya perlu memasukkan API terintegrasi ke dalam aplikasi. Selain itu, HUAWEI HiAI 3.0 mendistribusikan API seperti visi komputer dan pengenalan suara, yang membantu pengembang mengembangkan pengalaman hidup yang lebih cerdas dan menyeluruh.

Mesin aplikasi terbuka HiAI meliputi mesin CV, mesin ASR, mesin NLU, dan lain-lain. Berdasarkan hasil survei permintaan pengembang terhadap kapabilitas HiAI, lebih dari 60% responden berfokus pada CV, ASR, dan NLU.

1. *Computer Vision (CV)* menampilkan kemampuan komputer untuk mensimulasikan sistem visual manusia guna memahami lingkungan sekitar, yang terdiri dari penilaian, pengenalan, dan pemahaman ruang. Ini mencakup resolusi gambar super, pengenalan wajah, pengenalan objek, dan sebagainya.
2. *Automatic Speech Recognition (ASR)* menunjukkan kemampuan mengubah suara manusia menjadi teks untuk analisis dan pemahaman lebih lanjut oleh komputer. Ini mencakup pengenalan suara, konversi suara, dan sebagainya.
3. *Natural Language Understanding (NLU)* menunjukkan kemampuan komputer untuk memahami suara atau teks manusia, berkomunikasi atau bertindak secara alami dalam kombinasi dengan ASR. Ini mencakup segmentasi kata, pengenalan entitas teks, analisis bias sentimen, penerjemahan mesin, dan sebagainya.

Skenario aplikasi dan mesin terbuka HUAWEI HiAI Engine ditunjukkan pada Tabel 7.1.

Tabel 7.1 Skenario aplikasi dan mesin terbuka HUAWEI HiAI

Video pendek, Video langsung	Platform sosial	AR	Fotografi, pasca-pemrosesan gambar	Belanja	Terjemahan Pemrosesan Kata
Pengenalan Wajah Pengenalan Isyarat Segmentasi Potret Pengenalan Postur Stilisasi Video Kontrol Ucapan Kontrol Kedalaman Bidang Cerdas Pemandangan Gambar Pengenalan	Klasifikasi Foto Pengenalan Gambar Resolusi Super Gambar Informasi Sensitif Identifikasi	Situasional Kesadaran Kontrol Ucapan Estimasi Kedalaman Estimasi Sinar	Mode Rias Wajah Intensifikasi Gambar Penilaian Estetika Pembuatan Album Foto Kamera Kontrol Ucapan Kamera Kontrol Gerakan	Pemindaian Kode QR Layanan dan Rekomendasi Langsung Identifikasi Kartu Identitas Identifikasi Kartu Bank Belanja Pembacaan Gambar	Terjemahan Foto OCR Segmentasi Kata Pengenalan Entitas Bernama Pengenalan Emosi Teks Teks Cerdas Balasan Teks Super Gambar Resolusi
CV, ASR	CV, NLU	ASR, CV	CV	CV	NLU, CV, ASR

Layanan HUAWEI HiAI

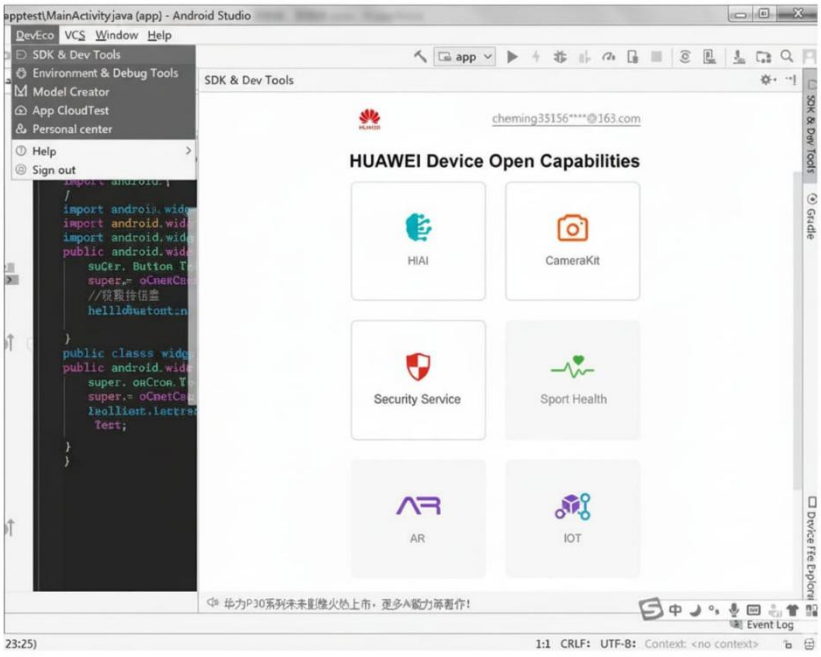
HUAWEI HiAI Service API mencapai distribusi cerdas terminal Pan, sehingga pengembang hanya perlu mengakses layanan sekali sebelum menggunakannya kembali di ponsel, komputer tablet, dan terminal lainnya untuk menyelesaikan distribusi secara efisien. HiAI Service API dapat merekomendasikan aplikasi atau layanan AI kepada pengguna secara

tepat waktu dan tepat, sehingga pengguna dengan cepat mendapatkan apa yang mereka butuhkan dalam layanan massal. Di saat yang sama, aplikasi AI juga menghubungkan pengguna secara akurat.

Dengan bantuan HiAI Service API, setiap fungsi atau konten dalam aplikasi dipecah menjadi layanan atom individual untuk proses push. HiAI Service API memiliki fungsi distribusi presisi multi-adegan dan multi-entri. HiAI Service API digunakan di berbagai portal seperti asisten cerdas HiBoard, pencarian global, HiVoice, HiTouch, dan HiVison untuk merekomendasikan dan menampilkan aplikasi yang relevan sesuai dengan kebiasaan pengguna atau konten pencarian, instruksi suara, dan operasi lainnya, sehingga aplikasi yang sesuai dapat menjangkau pengguna secara lebih cerdas dan akurat. API Layanan HiAI menghubungkan manusia dan layanan secara cerdas, mewujudkan peningkatan pengalaman dari "manusia mencari layanan" menjadi "layanan mencari manusia".

7.2 PENGEMBANGAN APLIKASI BERBASIS PLATFORM HUAWEI HIAI

HUAWEI HiAI juga menyediakan IDE, sebuah alat pengembangan untuk integrasi cepat kapabilitas HiAI, yang bertujuan membantu pengembang menggunakan kapabilitas terbuka Huawei EMUI dengan cepat, nyaman, dan efisien. Berbasis ekstensi fungsi Android Studio (disediakan dalam bentuk plug-in), IDE mendukung HiAI Engine, HiAI Foundation (analisis model AI, transformasi model AI, pembuatan kelas bisnis, pasar model AI), dll. IDE mendukung operasi drag, integrasi yang cepat dan efisien, serta menyediakan layanan mesin nyata jarak jauh gratis (3000+ mesin nyata AI, debugging jarak jauh satu klik 7x24 jam).

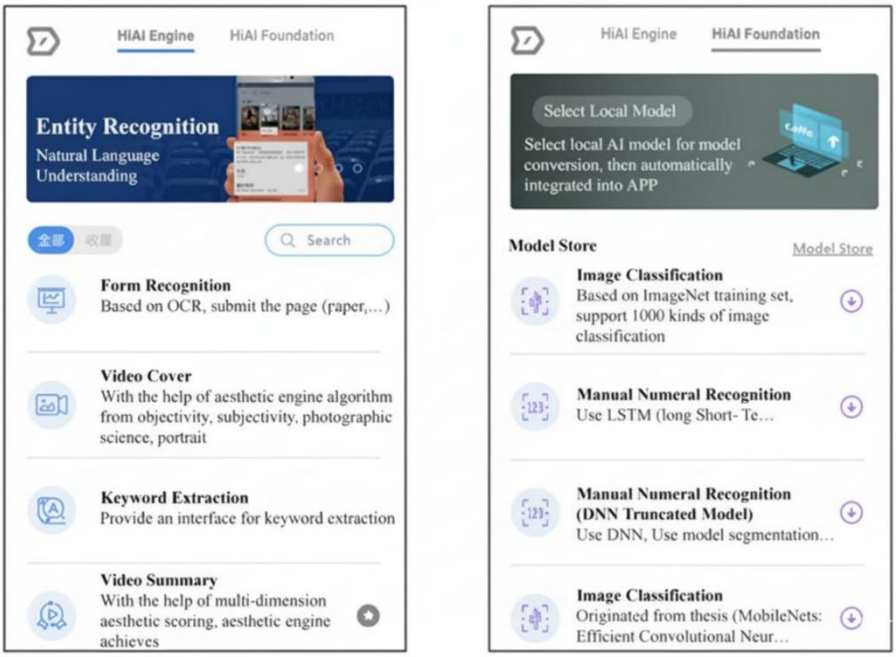


Gambar 7.6 HiAI IDE terintegrasi dengan android studio

IDE mendukung sistem operasi seperti Android Studio 2.3.X dan yang lebih baru, Windows 7, Windows 10, Mac OS 10.12/10.13. Jika sistem operasi tidak memenuhi persyaratan, hal ini

hanya memengaruhi fungsi konversi model lokal AI. IDE memilih fungsi yang sesuai berdasarkan skenario aktual: HUAWEI HiAI Engine dipilih saat menggunakan EMUI AI API, sementara HUAWEI HiAI Foundation dipilih untuk mengonversi model TensorFlow/Caffe menjadi model HUAWEI HiAI, lalu mengintegrasikan model ke dalam Aplikasi. Layanan HUAWEI HiAI digunakan sebagai penyedia layanan untuk Aplikasi biasa.

HiAI terintegrasi sempurna dengan Android Studio, artinya, HiAI digunakan sebagai plug-in untuk Android Studio, seperti yang ditunjukkan pada Gambar 7.6. Plug-in platform HiAI menyediakan fungsi HiAI Engine dan HiAI Foundation. HiAI Engine terutama menyediakan API yang terintegrasi dengan Aplikasi, yang dapat dipanggil secara langsung. HiAI Foundation mengintegrasikan model yang telah dilatih, yang dapat diunduh dan digunakan secara langsung, seperti yang ditunjukkan pada Gambar 7.7.

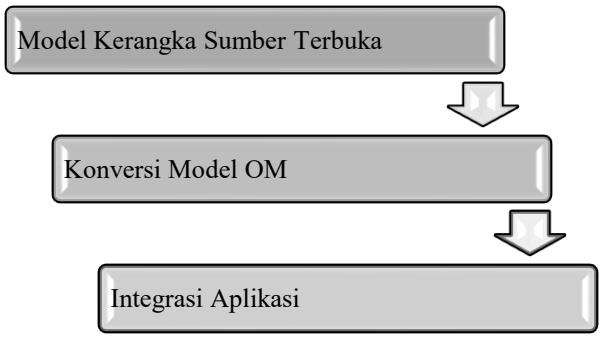


Gambar 7.7 Fungsi HiAI yang terintegrasi dengan Android Studio

Setelah pengembangan APP selesai, akan memasuki fase debugging mesin nyata. Huawei menyediakan rangkaian lengkap layanan debugging mesin nyata jarak jauh Huawei bagi para pengembang. Pengembang dapat mengakses mesin nyata laboratorium terminal jarak jauh Huawei dengan sekali klik, melakukan kendali jarak jauh waktu nyata dan debugging satu langkah, dengan profil dan log yang disediakan. Beberapa model Huawei yang didukung oleh HiAI ditunjukkan pada Gambar 7.8.

Gambar 7.8 Model Huawei yang didukung oleh HiAI

Langkah-langkah integrasi Aplikasi ke dalam HiAI DDK adalah sebagai berikut: Pertama, kita mendapatkan model kerangka kerja seperti Caffe/TensorFlow yang telah dilatih, lalu menggunakan alat konversi model OMG yang disediakan untuk mengonversi model kerangka kerja sumber terbuka asli menjadi model OM, yang berisi fungsi kuantisasi 8-bit, yang sesuai untuk platform DaVinci. Terakhir, integrasi Aplikasi dilakukan, yang meliputi prapemrosesan model, inferensi model, dan bagian lainnya, seperti yang ditunjukkan pada Gambar 7.9.



Gambar 7.9 Prosedur Integrasi Aplikasi ke HiAI DDK

Proses integrasi Aplikasi adalah sebagai berikut.

1. Langkah 1: Buat proyek.
   (a) Buat proyek Android Studio dan centang opsi "Sertakan dukungan C++".
   (b) C++ Standard pilih C++ 11, centang opsi "Dukungan Pengecualian (-fexceptions)", centang opsi "Dukungan Informasi Jenis Runtime (-frtti)".
2. Langkah 2: Kompilasi JNI.
   (a) Realisasikan JNI, tulis Dokumen Android.mk.
   (b) Tulis File Application.mk, salin sdk ke pustaka sumber daya.
   (c) Pada File build.gradle, tentukan NDK untuk mengompilasi file C++.
3. Langkah 3: Integrasi Model.
   (a) Prapemrosesan Model: prapemrosesan model lapisan aplikasi, prapemrosesan model lapisan JNI.

(b) Inferensi Model.

7.3 BAGIAN DARI SOLUSI HUAWEI HIAI

HUAWEI HiAI Membantu Tuna Rungu dan Bisu

Anak-anak tuna rungu tidak dapat menikmati waktu luang yang normal karena keterbatasan fisik mereka. Mereka tidak dapat mendengar sapaan dari keluarga dan teman-teman mereka. Bagi mereka, dunia terasa sunyi dan sepi. Menurut statistik, terdapat sekitar 32 juta anak tuna rungu di dunia. Mereka tidak dapat mendengar suara yang indah maupun mengungkapkan isi hati mereka, sehingga cara mereka berkomunikasi dengan dunia penuh dengan hambatan.

Realitas pahitnya adalah 90% orang tua dari anak-anak tuna rungu memiliki kemampuan berbahasa. Namun, 78% dari mereka tidak dapat berkomunikasi dengan anak-anak mereka. Anak-anak tuna rungu memiliki kesulitan besar dalam pembelajaran bahasa dan membaca. Bahasa adalah dasar dari mendengarkan, berbicara, membaca, dan menulis, dan mendengarkan adalah satu-satunya cara untuk belajar bahasa.

Misalnya, ketika menemukan kata baru, anak-anak normal dapat memahami artinya dengan mendengarkan penjelasan orang dewasa, dan kemudian mereka dapat menguasainya melalui mendengarkan, berbicara, membaca, dan menulis secara terus-menerus. Namun, anak-anak tuna rungu tidak dapat melakukannya, karena pembelajaran bahasa mereka dilakukan melalui bahasa isyarat. Tanpa bantuan guru bahasa isyarat profesional, mereka tidak dapat berkomunikasi dengan orang biasa.

Gambar 7.10 Huawei HiAI menganimasikan teks

Oleh karena itu, bersama dengan Uni Eropa untuk Tuna Rungu (sebuah organisasi nirlaba), Penguin Group, dan Aardman (ahli animasi), Huawei telah mengembangkan aplikasi StorysSign. Dengan bantuan beberapa kemampuan platform Huawei HiAI seperti Pengenalan Gambar dan Pengenalan Karakter Optik (OCR), jika Anda menggunakan ponsel untuk menghadap teks pada buku, Huawei HiAI akan langsung menganimasikan teks tersebut. Saudari Xingxing menggunakan bahasa isyarat untuk mengekspresikan teks pada buku, seperti yang ditunjukkan pada Gambar 7.10.

HUAWEI HiAI Meningkatkan Efek Visual Aplikasi Tes Mengemudi Yuanbei

Tes mengemudi Yuanbei adalah aplikasi pembelajaran mengemudi yang dirancang khusus untuk pemula. Aplikasi ini menyediakan layanan tes mengemudi bergambar, termasuk pendaftaran sekolah mengemudi, pemesanan pembelajaran, dan simulasi tes mengemudi.

Aplikasi ini berkomitmen untuk membangun platform tes mengemudi terpadu yang nyaman dan praktis. Simulasi tes mengemudi, salah satu fitur utama tes mengemudi Yuanbei, menggabungkan grafis, video, suara, dan bentuk lain dari paket instalasi bawaan. Aplikasi ini secara efektif membantu pelajar dengan cepat memahami konten dan spesifikasi tes, sehingga mereka dapat lulus tes mengemudi dengan cepat. Ujian simulasi berisi sejumlah besar gambar untuk membantu peserta didik berlatih, namun latihan tersebut mungkin terpengaruh karena beberapa gambar berkualitas rendah ditampilkan dengan buruk dan kurang jelas pada telepon seluler biasa.

Pada sebagian besar perangkat, program pengoptimalan gambar simulasi uji mengemudi bergantung pada internet. Oleh karena itu, jika sinyal jaringan lemah atau tidak ada jaringan, peningkatan kualitas gambar akan terhambat secara signifikan. HUAWEI HiAI mengadopsi pengurangan noise cerdas dan amplifikasi resolusi 9x, yang secara signifikan meningkatkan kualitas gambar, menghadirkan detail gambar yang lebih jernih, dan meningkatkan pengalaman visual pengguna secara komprehensif.

Mengandalkan model pembelajaran on-device HUAWEI HiAI, uji mengemudi Yuanbei mencapai pengoptimalan dan amplifikasi gambar on-device, dan gambar yang sama ditampilkan lebih jelas pada model NPU Huawei. Sementara itu, tanpa ketergantungan pada jaringan, pengguna tetap dapat melihat gambar amplifikasi berkualitas tinggi saat jaringan tidak stabil atau terputus, seperti yang ditunjukkan pada Gambar 7.11.

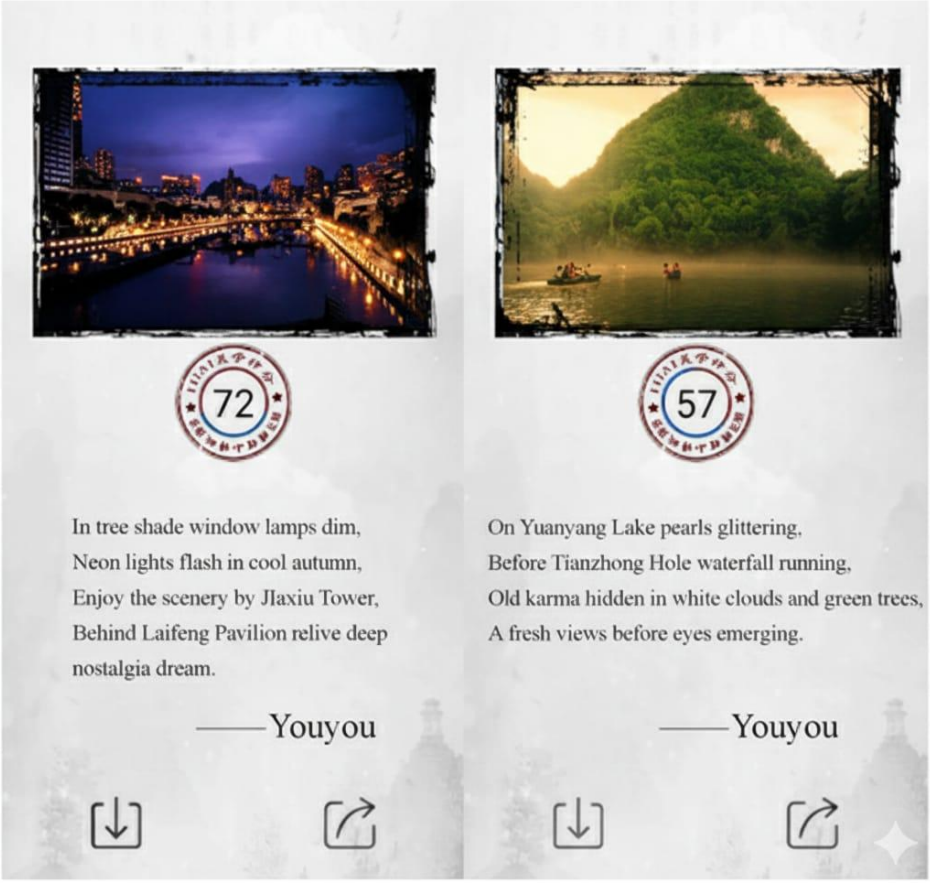
Gambar 7.11 HUAWEI HiAI meningkatkan efek visual aplikasi uji mengemudi Yuanbei

HUAWEI HiAI Memberdayakan Perjalanan Ctrip

Aplikasi seluler Ctrip menyediakan layanan perjalanan yang komprehensif bagi pengguna, termasuk reservasi hotel, tiket pesawat, tiket kereta api, panduan perjalanan, diskon tiket masuk, asuransi perjalanan, dll. Selama perjalanan, pengguna sering mengambil

banyak foto, berharap dapat mengabadikan pemandangan indah sebagai kenang-kenangan. Namun, karena kurangnya pengetahuan fotografi profesional, kebanyakan orang tidak dapat menilai secara akurat apakah foto tersebut bagus atau tidak, dan mereka selalu ragu apakah foto tersebut telah menghasilkan efek terbaik. Di saat yang sama, gambar yang diambil pengguna kurang jernih, dan efek renderingnya buruk. Oleh karena itu, meningkatkan kualitas gambar menjadi daya tarik banyak pengguna.

Dengan mengakses kemampuan penilaian estetika HUAWEI HiAI Engine, aplikasi ini secara otomatis mengintegrasikan faktor-faktor teknis seperti defokus dan jitter pada gambar, serta nuansa estetika subjektif seperti kemiringan, warna, dan komposisi, untuk mengevaluasi dan menilai gambar. Pengguna dapat dengan cepat memahami kualitas foto melalui tingkat penilaian, mengatasi keraguan mereka, dan menyesuaikannya, untuk mengambil pemandangan terindah. Selain itu, dengan bantuan HUAWEI HiAI, aplikasi ini juga mewujudkan fungsi pengaktifan suara dan penulisan puisi dalam sekali klik, yang memberikan banyak kemudahan bagi pengguna, seperti yang ditunjukkan pada Gambar 7.12.



Gambar 7.12 HUAWEI HiAI memberdayakan Ctrip dengan penulisan puisi dalam sekali klik

HUAWEI HiAI Memberdayakan Deteksi dan Koreksi Kesalahan Dokumen WPS

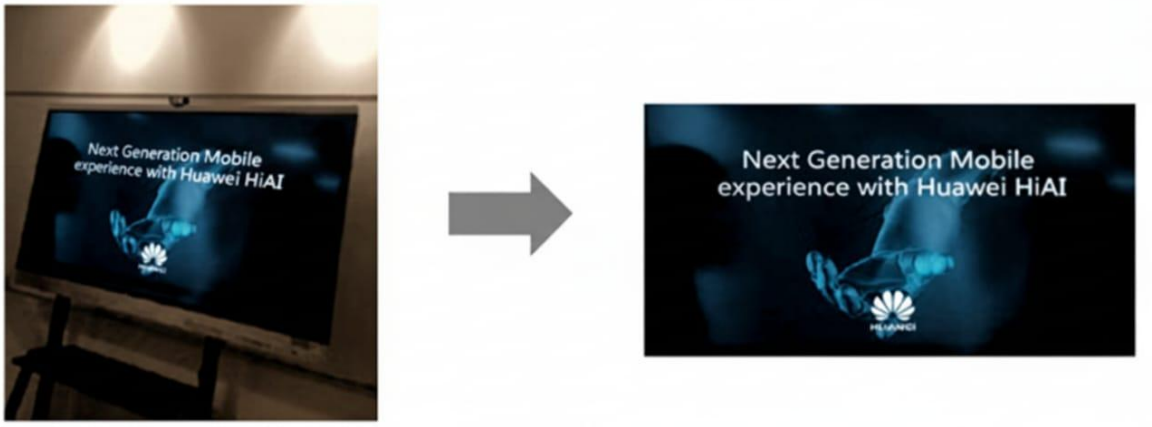
Aplikasi WPS adalah perangkat lunak perkantoran yang dapat mengedit dan melihat dokumen perkantoran umum seperti teks, formulir, dan presentasi. Selain itu, pengguna dapat menggunakan ruang cloud dan templat dokumen gratis. Dengan meningkatnya penggunaan perangkat seluler, ponsel semakin banyak digunakan untuk mengedit dokumen,

mengirim, dan menerima email. Namun, tanpa bantuan keyboard atau mouse, hanya dengan sentuhan jari di layar untuk menyelesaikan operasi, efisiensi penggunaan ponsel di kantor sangatlah rendah. Misalnya, ketika menghadiri kelas, rapat, atau pelatihan, kita melihat poin-poin penting dan "inti" pada presentasi, kita akan segera mengambil foto dan merekamnya. Namun, foto-foto tersebut seringkali perlu dipotong di komputer sebelum diurutkan ke dalam presentasi, yang cukup merepotkan dan memakan waktu.

1. Gangguan Lingkungan: Selain presentasi, terdapat gangguan lingkungan seperti layar, dinding, meja, dan kursi pada gambar yang diambil oleh pengguna, yang perlu dipotong.
2. Gambar dokumen terdistorsi: Ketika sudut pengambilan gambar tidak langsung menghadap dokumen, gambar dokumen akan terdistorsi hingga berbagai tingkat, dan gambar yang diregangkan atau dikompresi akan memengaruhi penggunaan selanjutnya.
3. Gambar buram: Karena keterbatasan cahaya, jarak, dan faktor lainnya, gambar yang diambil pengguna mungkin buram, yang selanjutnya akan memengaruhi persepsi dan pengenalan informasi.
4. Konten yang tidak dapat diedit: Banyak pengguna ingin mengedit atau mengubah konten presentasi saat melihat gambar presentasi yang diambil. Namun, konten gambar tidak dapat diedit secara langsung.

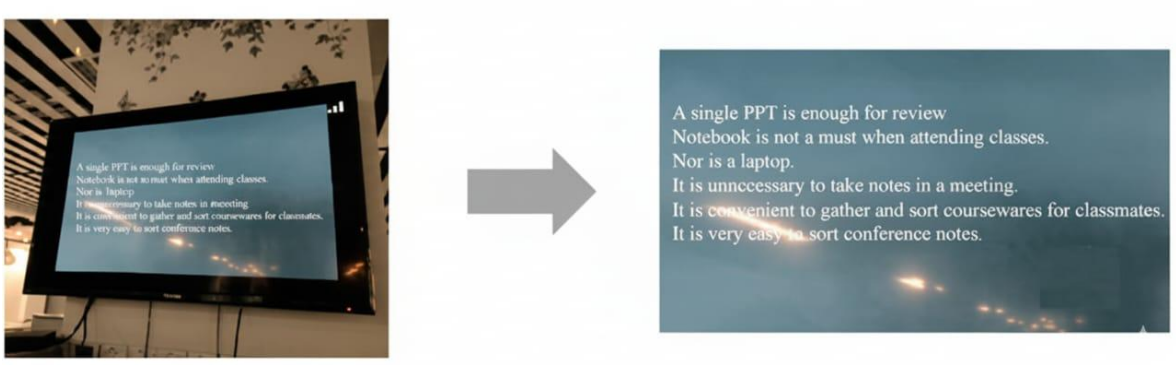
WPS dapat dengan mudah mengatasi masalah di atas dengan mengakses ekologi HUAWEI HiAI dan kinerja prosesor Huawei Kirin 970 yang bertenaga. Hanya membutuhkan waktu 3 detik untuk menghasilkan presentasi dari beberapa gambar dengan satu klik.

1. Persepsi dokumen untuk identifikasi otomatis area valid dokumen: Dengan mengakses kemampuan deteksi dan koreksi dokumen HUAWEI HiAI Engine, WPS secara akurat mendeteksi area di mana dokumen berada, dan secara otomatis memotong area di sekitarnya seperti layar, dinding, meja, dan kursi, seperti yang ditunjukkan pada Gambar 7.13.



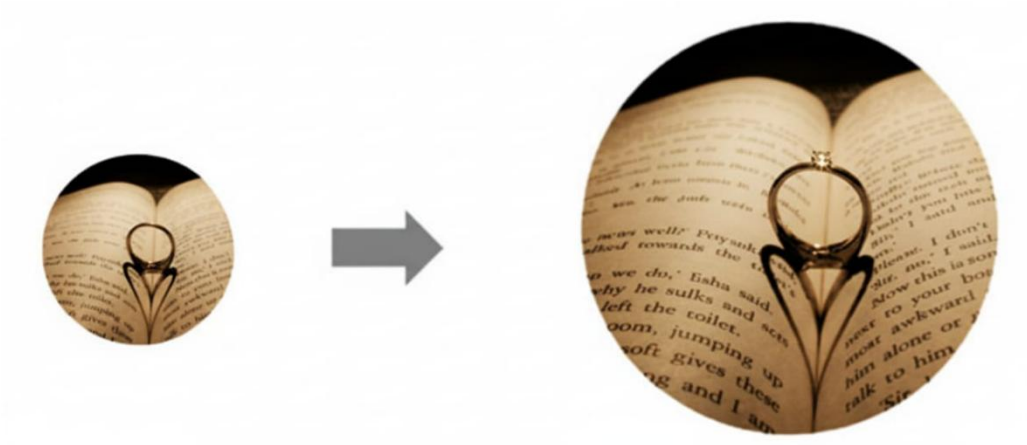
Gambar 7.13 Persepsi dokumen WPS

2. Koreksi dokumen untuk penyesuaian cepat ke tampilan tengah: Ini merupakan peningkatan tambahan dalam proses pembuatan ulang dokumen, yang secara otomatis menyesuaikan sudut kamera langsung ke dokumen, dengan sudut koreksi maksimum 45◦, seperti yang ditunjukkan pada Gambar 7.14.



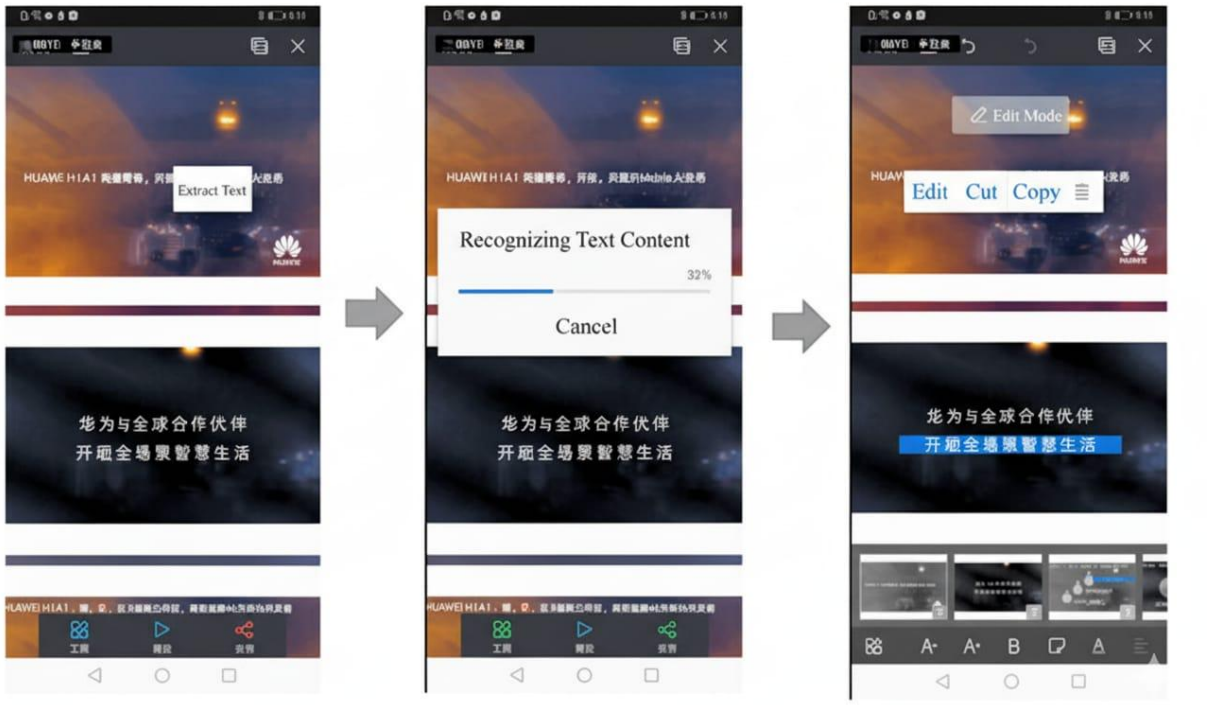
Gambar 7.14 Koreksi dokumen WPS

3. Resolusi super teks untuk teks yang lebih jelas dalam dokumen: HUAWEI HiAI memperbesar gambar yang berisi konten teks hingga sembilan kali resolusi (tiga kali tinggi dan lebar), sehingga kualitas gambar meningkat secara signifikan dan teks terbaca, seperti yang ditunjukkan pada Gambar 7.15.



Gambar 7.15 Resolusi super teks

4. Pengenalan OCR untuk pengeditan bebas konten teks dalam gambar: Dengan mengakses OCR umum, WPS secara otomatis mengenali dan mengekstrak informasi teks dalam gambar, sehingga memungkinkan modifikasi, pemotongan, penyalinan, dan penghapusan konten teks dalam gambar presentasi secara bebas, seperti yang ditunjukkan pada Gambar 7.16.



Gambar 7.16 Pengenalan OCR WPS

Untuk solusi lebih lanjut, silakan kunjungi situs web resmi HiAI.

7.4 KESIMPULAN

Bab ini terutama memperkenalkan struktur ekologi tiga lapis platform HUAWEI HiAI: HUAWEI HiAI Foundation, HUAWEI HiAI Engine, dan HUAWEI HiAI Service API. Kapabilitas yang relevan dari setiap lapis, dan beberapa solusi HiAI juga diperkenalkan. Akhirnya, untuk menghubungkan para pengembang secara menyeluruh, HUAWEI HiAI telah mengadopsi kegiatan-kegiatan berikut untuk mendorong inovasi dan mencapai saling menguntungkan secara ekologis.

1. HUAWEI HiAI telah menyelenggarakan kegiatan komunikasi mendalam berikut untuk koneksi luring.
   (a) Salon HUAWEI Developer Day.
   (b) Kelas terbuka HUAWEI HiAI.
   (c) Pertemuan teknis khusus HUAWEI HiAI.
2. HUAWEI HiAI telah menyelenggarakan kegiatan-kegiatan berikut senilai $1 miliar untuk memacu inovasi di seluruh lingkungan.
   (a) Inovasi terbuka kapabilitas terminal.
   (b) Inovasi layanan digital di seluruh lingkungan.
   (c) Ko-konstruksi ekologis layanan Cloud.
3. HUAWEI HiAI telah menyelenggarakan kontes inovasi berikut.
   (a) Kontes inovasi aplikasi AI.
   (b) Kontes kreativitas aplikasi masa depan.

(c) Kontes inovasi aplikasi AR.

Huawei percaya bahwa AI dapat meningkatkan kualitas hidup pengguna. Baik di back-end maupun terminal, AI dapat menembus imajinasi dan menghadirkan kemudahan yang belum pernah ada sebelumnya bagi pengguna. Namun, semua ini perlu memiliki skenario aplikasi praktis, yang memungkinkan lebih banyak perusahaan dan pengembang untuk berpartisipasi sehingga pengguna dapat memperoleh peningkatan pengalaman yang substansial. Huawei bersedia bekerja sama dengan lebih banyak talenta dan perusahaan, untuk bersama-sama mempromosikan implementasi kecerdasan industri berbasis platform HUAWEI HiAI 3.0.

Latihan Soal

1. HUAWEI HiAI 3.0 resmi dirilis pada Konferensi Pengembang Software Green Alliance pada 19 November 2019, menandai bahwa AI on-engine secara resmi bergerak menuju terdistribusi, yang akan menghadirkan pengalaman hidup cerdas dengan latar belakang terbaik. Apa ekologi AI tiga lapis dari HUAWEI HiAI? 2. Lapisan HUAWEI HiAI mana yang dapat mengkompilasi model jaringan saraf standar menjadi model offline?
2. Lapisan HUAWEI HiAI mana yang dapat dengan mudah mengintegrasikan berbagai kemampuan AI dengan Aplikasi untuk menjadikan Aplikasi lebih cerdas dan canggih?
3. Alat apa yang dapat diintegrasikan dengan sempurna dengan HiAI?
4. Bagaimana proses integrasi Aplikasi?

BAB 8
PLATFORM APLIKASI HUAWEI CLOUD ENTERPRISE INTELLIGENCE

Bab ini terutama memperkenalkan Huawei CLOUD Enterprise Intelligence (EI), termasuk keluarga layanan Huawei CLOUD EI. Bab ini berfokus pada platform Huawei ModelArts dan solusi Huawei EI.

8.1 KELUARGA LAYANAN HUAWEI CLOUD EI

Keluarga layanan Huawei CLOUD EI terdiri dari big data EI, platform dasar EI, bot percakapan, pemrosesan bahasa alami (NLP), interaksi ucapan, analisis ucapan, pengenalan gambar, peninjauan konten, pencarian gambar, pengenalan wajah, Pengenalan Karakter Optik (OCR), dan agen EI, seperti yang ditunjukkan pada Gambar 8.1.

Gambar 8.1 Rangkaian layanan Huawei CLOUD EI

1. Big data EI menyediakan layanan seperti akses data, migrasi data CLOUD, komputasi streaming waktu nyata, MapReduce, Data Lake Insight, penyimpanan tabel.
2. Platform dasar EI menyediakan layanan seperti platform ModelArts, pembelajaran mendalam, pembelajaran mesin, HiLens, layanan mesin grafik, dan akses video.
3. Bot percakapan menyediakan QABot cerdas, TaskBot, bot inspeksi kualitas cerdas, dan layanan bot konvensional yang dapat disesuaikan.
4. Pemrosesan bahasa alami menyediakan dasar-dasar pemrosesan bahasa alami, peninjauan konten—pemahaman teks dan bahasa, pembangkitan bahasa, Kustomisasi NLP, penerjemahan mesin.
5. Interaksi ucapan menyediakan pengenalan ucapan, sintesis ucapan, dan transkripsi ucapan waktu nyata.

6. Analisis video menyediakan analisis konten video, penyuntingan video, deteksi kualitas video, dan penandaan video.
7. Pengenalan gambar menyediakan layanan penandaan gambar dan pengenalan selebritas.
8. Peninjauan konten menyediakan peninjauan teks, gambar, dan video.
9. Pencarian gambar menunjukkan pencarian gambar dengan gambar, membantu pelanggan mencari gambar yang sama atau serupa dari pustaka gambar yang ditentukan.
10. Pengenalan wajah menyediakan pengenalan wajah dan analisis tubuh.
11. OCR menyediakan pengenalan karakter untuk kelas umum, kelas sertifikat, kelas tagihan, kelas industri, dan kelas templat khusus.
12. Agen EI terdiri dari agen AI transportasi, agen AI industri, agen AI taman, agen AI jaringan, agen AI otomotif, mesin AI medis, dan agen AI geografis.

Agen EI Huawei CLOUD

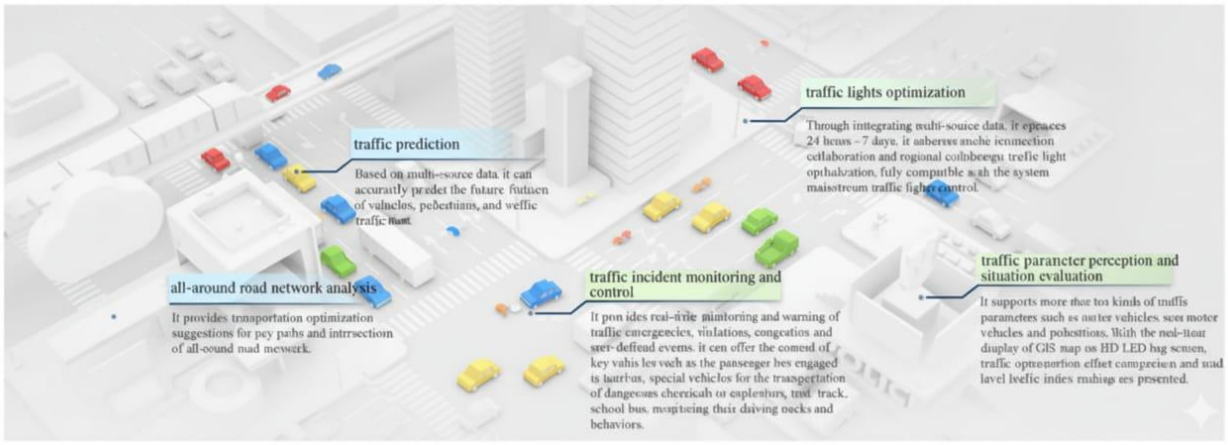
Agen EI mengintegrasikan teknologi AI ke dalam skenario aplikasi untuk semua lapisan masyarakat. Dikombinasikan dengan berbagai teknologi, agen ini menggali nilai data secara mendalam dan memanfaatkan teknologi AI secara maksimal, sehingga solusi berbasis skenario dikembangkan untuk meningkatkan efisiensi dan pengalaman pengguna. Agen EI terdiri dari mesin AI transportasi, mesin AI industri, mesin AI taman, dan mesin AI jaringan, seperti yang ditunjukkan pada Gambar 8.2. Selain itu, Huawei juga meluncurkan mesin AI kendaraan, mesin AI medis, dan mesin AI geografis.

Transportasi AI Mesin | Mesin AI industri | Mesin Taman AI | Jaringan AI Mesin

Gambar 8.2 Agen EI

1. Mesin AI Transportasi

Mesin AI transportasi mewujudkan produk dan solusi seperti analisis jaringan jalan menyeluruh, prediksi lalu lintas, pemantauan dan pengendalian insiden lalu lintas, optimalisasi lampu lalu lintas, persepsi parameter lalu lintas, dan evaluasi situasi untuk memastikan perjalanan yang efisien, ramah lingkungan, dan aman. Mesin AI transportasi ditunjukkan pada Gambar 8.3.



Gambar 8.3 Mesin AI Transportasi

Mesin AI transportasi memiliki keunggulan sebagai berikut.

(a) Mewujudkan penggalian data yang komprehensif dan mendalam, dengan internet dan big data transportasi yang terintegrasi penuh, serta nilai big data yang digali secara mendalam.

(b) Menyediakan kolaborasi menyeluruh dan kolaborasi pejalan kaki-kendaraan, yang memaksimalkan arus lalu lintas di seluruh wilayah dan meminimalkan waktu tunggu kendaraan di wilayah tersebut. AI ini juga mengoordinasikan permintaan lalu lintas kendaraan dan pejalan kaki, sehingga mewujudkan lalu lintas kendaraan dan pejalan kaki yang tertib.

(c) Menyediakan penjadwalan lampu lalu lintas waktu nyata, yang merupakan yang pertama di industri yang mencapai formulasi standar untuk antarmuka komunikasi yang aman antara agen AI transportasi dan platform kontrol lampu lalu lintas.

(d) Dapat secara akurat memprediksi permintaan jalur lalu lintas sehingga dapat merencanakan rute terlebih dahulu.

Mesin AI transportasi memiliki karakteristik sebagai berikut.

(a) Penuh waktu: Mewujudkan persepsi insiden lalu lintas di seluruh area dan penuh waktu selama 7 × 24 jam.

(b) Kecerdasan: Mencapai optimasi lampu lalu lintas regional.

(c) Kelengkapan: Dapat mengidentifikasi titik kemacetan utama dan jalur kemacetan utama, serta melakukan analisis penyebaran kemacetan.

(d) Prediksi: Memprediksi kepadatan kerumunan, memperoleh keteraturan lalu lintas migrasi kerumunan.

(e) Akurasi: Mencapai kontrol kondisi lalu lintas yang komprehensif dan akurat dalam 7×24 jam.

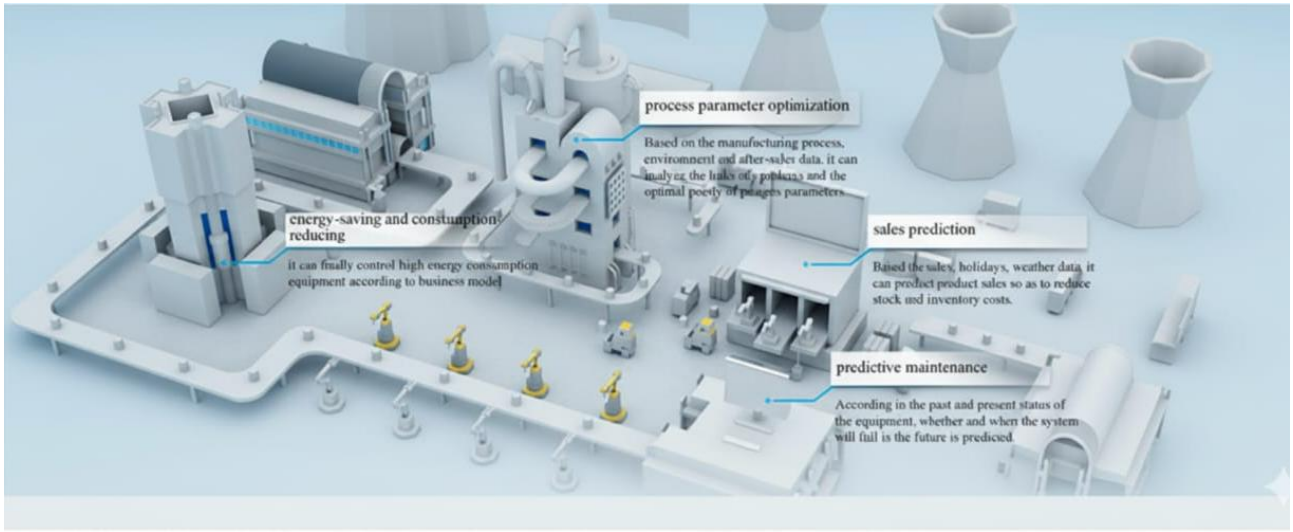
(f) Kenyamanan: Mewujudkan penjadwalan lampu lalu lintas secara real-time dan pengaturan lalu lintas sesuai permintaan.

(g) Visualitas: Menampilkan situasi lalu lintas langsung di layar besar.

(h) Kehalusan: Mewujudkan kontrol kendaraan utama dan manajemen yang cermat.

2. Mesin AI Industri

Mengandalkan big data dan kecerdasan buatan, mesin AI industri menyediakan layanan rantai penuh di bidang desain, produksi, logistik, penjualan, dan layanan. Mesin ini menggali nilai data, membantu perusahaan untuk memimpin dengan teknologi baru. Mesin AI industri ditunjukkan pada Gambar 8.4.



Gambar 8.4 Mesin AI Industri

Mesin AI industri menyebabkan tiga perubahan besar dalam industri yang ada.

(a) Transformasi dari pengalaman buatan ke kecerdasan data: Berdasarkan analisis penambangan data, pengalaman baru dalam peningkatan efisiensi dan kualitas produk dapat diperoleh dari data.

(b) Transisi dari digital ke cerdas: Kemampuan analisis cerdas telah menjadi kekuatan pendorong baru digitalisasi perusahaan.

(c) Transisi dari manufaktur produk ke inovasi produk: Kolaborasi data dari desain produk hingga penjualan di perusahaan, serta kolaborasi data antara hulu dan hilir rantai industri, menghasilkan keunggulan kompetitif baru.

Penerapan mesin AI industri adalah sebagai berikut.

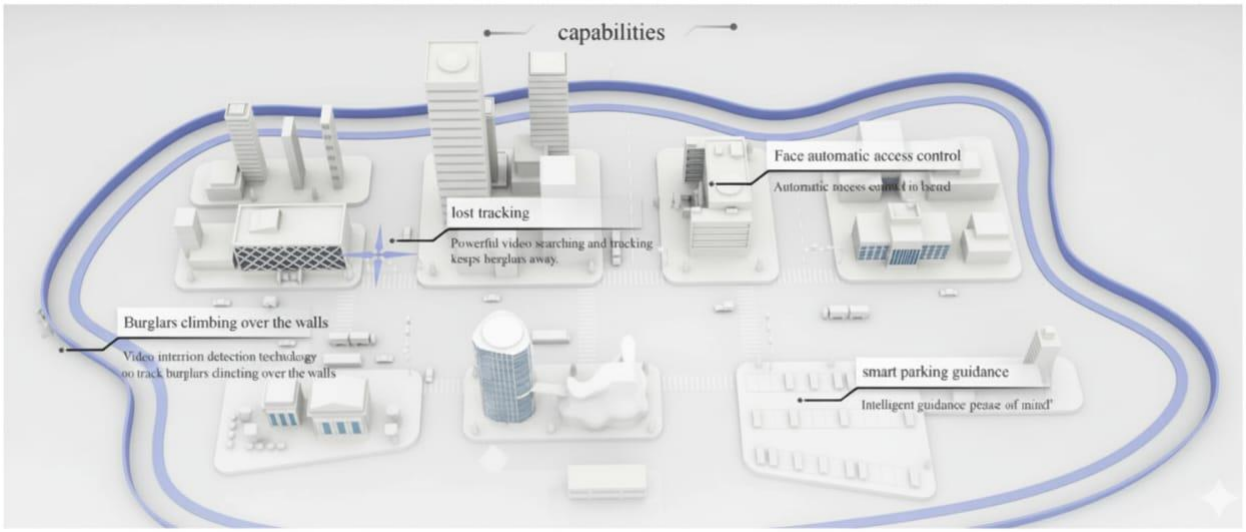
(a) Optimalisasi dan peningkatan kualitas produk: Berdasarkan umpan balik pelanggan, analisis komentar internet, analisis pesaing, catatan pemeliharaan, dan data historis purnajual, analisis terklasifikasi dilakukan untuk menemukan masalah utama produk, sehingga dapat memandu peningkatan produk baru dan promosi kualitas produk.

(b) Pemeliharaan peralatan cerdas: Berdasarkan status sistem di masa lalu dan saat ini, pemeliharaan prediktif dilakukan melalui metode inferensi prediktif seperti prediksi deret waktu, prediksi jaringan saraf tiruan, dan analisis regresi. Sistem ini dapat memprediksi apakah sistem akan gagal di masa mendatang, kapan akan gagal, dan jenis kegagalannya, sehingga dapat meningkatkan efisiensi operasi dan pemeliharaan layanan, mengurangi waktu henti peralatan yang tidak direncanakan, dan menghemat biaya tenaga kerja untuk layanan di lokasi.

(c) Estimasi material produksi: Berdasarkan data material historis, material yang dibutuhkan untuk produksi diprediksi secara akurat sehingga dapat mengurangi siklus penyimpanan dan meningkatkan efisiensi. Optimasi algoritma mendalam didasarkan pada model algoritma deret waktu industri, yang dikombinasikan dengan optimasi mendalam rantai pasokan Huawei.

3. Mesin Park AI

Mesin Park AI menerapkan kecerdasan buatan untuk manajemen dan pemantauan kawasan industri, kawasan perumahan, dan kawasan komersial, sehingga menyediakan lingkungan yang nyaman dan efisien melalui teknologi seperti analisis video dan penambangan data. Mesin Park AI ditunjukkan pada Gambar 8.5. Mesin AI parkir menghadirkan tiga perubahan berikut.



Gambar 8.5 Mesin Park AI

(a) Dari pertahanan manual menjadi pertahanan cerdas: Keamanan cerdas berbasis kecerdasan buatan dapat meringankan beban petugas keamanan.

(b) Dari penggesekan kartu menjadi pemindaian wajah: Pemindaian wajah secara otomatis mencatat waktu masuk, sehingga tidak perlu lagi membawa kartu masuk.

(c) Dari kekhawatiran menjadi kepastian: Dengan kemampuan pelacakan dan analisis barang hilang yang canggih, kecerdasan buatan membuat karyawan dan pemilik properti merasa lebih nyaman.

Aplikasi mesin AI parkir adalah sebagai berikut.

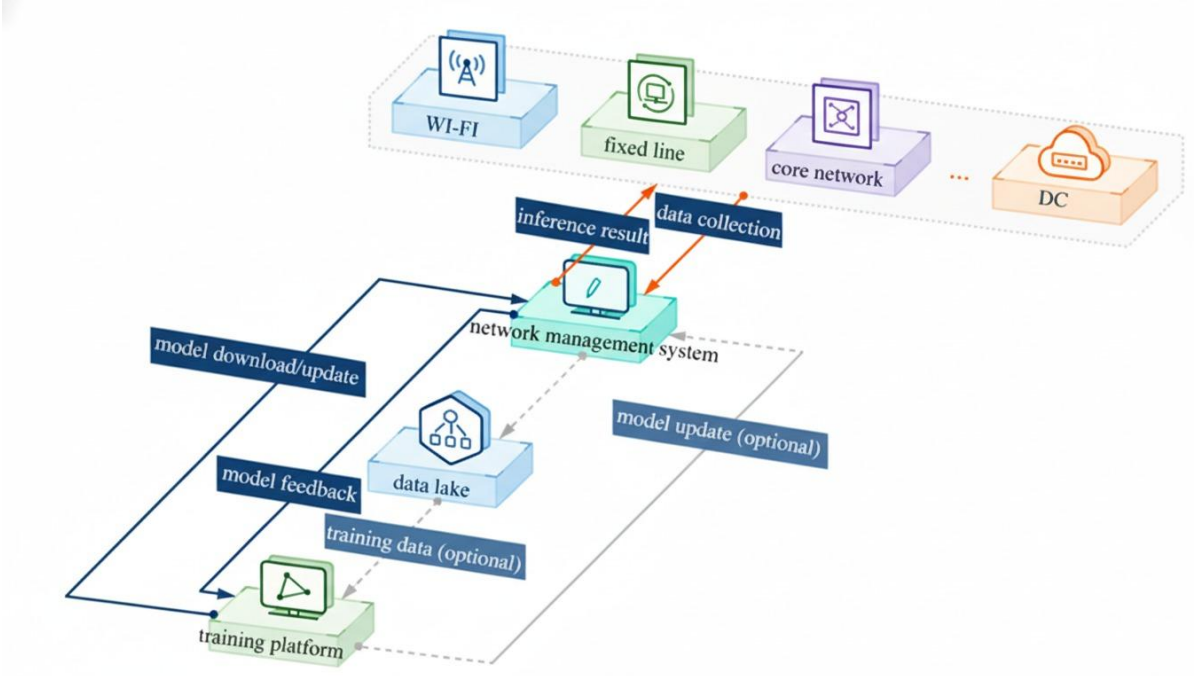
(a) Kontrol pintu masuk taman: Teknologi pengenalan wajah dapat mengidentifikasi identitas pengunjung secara akurat dan dengan cepat memberikan hasilnya. Oleh karena itu, throughput tinggi dari kontrol pintu masuk dan manajemen taman otomatis dapat dicapai.

(b) Pemantauan Keamanan: Melalui teknologi cerdas seperti deteksi intrusi, deteksi orang berkeliaran, deteksi objek terlantar, area dapat dipantau untuk memastikan keamanannya.

(c) Parkir cerdas: Melalui pengenalan plat nomor kendaraan dan lintasan lintasan, layanan seperti kontrol akses kendaraan, kontrol jalur berkendara, manajemen parkir ilegal, dan manajemen ruang parkir dapat diwujudkan.

4. Mesin AI Jaringan

Mesin AI Jaringan (NAIE) memperkenalkan AI ke dalam bidang jaringan untuk memecahkan masalah prediksi, pengulangan, dan kompleksitas bisnis jaringan. Hal ini meningkatkan pemanfaatan sumber daya jaringan, efisiensi operasi dan pemeliharaan, efisiensi energi, dan pengalaman bisnis, sehingga memungkinkan terwujudnya jaringan penggerak otomatis. Mesin AI Jaringan ditunjukkan pada Gambar 8.6.



Gambar 8.6 Mesin AI jaringan

Mesin AI Jaringan memiliki nilai komersial sebagai berikut.

(a) Tingkat pemanfaatan sumber daya telah ditingkatkan. AI diperkenalkan untuk memprediksi lalu lintas jaringan, berdasarkan hasil prediksi, sehingga sumber daya jaringan dikelola secara seimbang untuk meningkatkan tingkat pemanfaatan sumber daya jaringan.

(b) Meningkatkan efisiensi operasi dan pemeliharaan. AI diperkenalkan untuk mengurangi banyak pekerjaan berulang, memprediksi kesalahan, dan melakukan pemeliharaan preventif, sehingga dapat meningkatkan efisiensi operasi dan pemeliharaan jaringan.

(c) Meningkatkan efisiensi pemanfaatan energi. Teknologi AI digunakan untuk memprediksi status bisnis secara real-time, secara otomatis melakukan penyesuaian

dinamis konsumsi energi sesuai volume bisnis, sehingga meningkatkan efisiensi pemanfaatan energi.

Keunggulan teknis mesin AI jaringan adalah sebagai berikut.

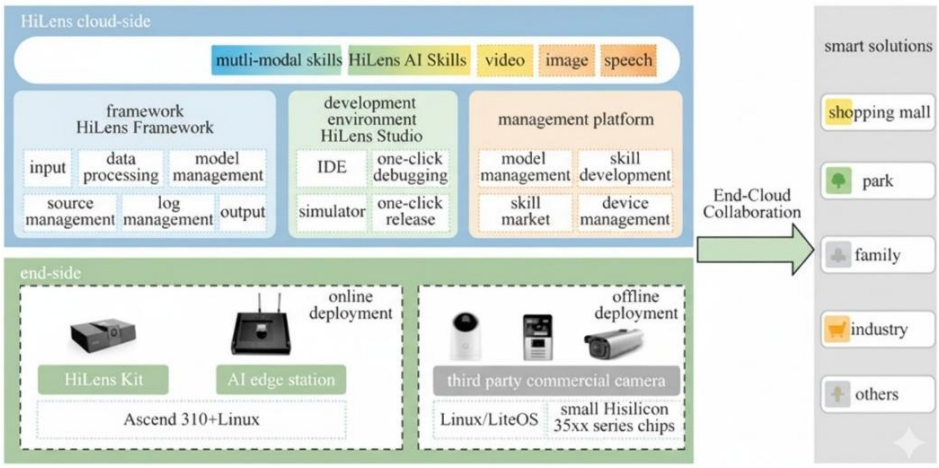
(a) Data memasuki lake dengan aman. Mendukung pengumpulan berbagai jenis data secara cepat seperti parameter jaringan, kinerja, dan alarm ke dalam lake. Di satu sisi, sejumlah besar alat disediakan untuk meningkatkan efisiensi tata kelola data. Pada saat yang sama, isolasi multi-penyewa, enkripsi, dan penyimpanan diterapkan untuk memastikan keamanan seluruh siklus hidup data yang memasuki lake.

(b) Penyematan pengalaman jaringan. Menggunakan lingkungan pengembangan model terpandu dan prasetel beberapa templat pengembangan model AI dalam domain jaringan. Menyediakan berbagai layanan untuk berbagai pengembang seperti layanan pelatihan, layanan pembuatan model, layanan model komunikasi, yang bertujuan untuk membantu pengembang menyelesaikan pengembangan model/aplikasi dengan cepat.

(c) Layanan aplikasi yang kaya. Menyediakan layanan aplikasi untuk berbagai skenario bisnis jaringan seperti akses nirkabel, akses jaringan tetap, beban transmisi, jaringan inti, arus searah (DC), dan energi. Platform ini secara efektif dapat memecahkan masalah spesifik terkait efisiensi operasi dan pemeliharaan, efisiensi konsumsi energi, dan tingkat pemanfaatan sumber daya dalam layanan jaringan.

Platform Dasar EI: Huawei HiLens

Huawei HiLens adalah platform pengembangan dan aplikasi AI multi-moda kolaborasi End-Cloud, yang terdiri dari perangkat komputasi end-side dan platform Cloud. Platform ini menyediakan kerangka kerja pengembangan yang sederhana, lingkungan pengembangan yang siap pakai, pasar keterampilan AI yang kaya, dan platform manajemen Cloud. Platform ini terhubung dengan berbagai perangkat komputasi end-side, mendukung pengembangan aplikasi AI visual dan auditori, penerapan daring aplikasi AI, dan manajemen perangkat masif. HiLens membantu pengguna mengembangkan aplikasi AI multi-moda dan mendistribusikannya ke perangkat end-side untuk mewujudkan solusi cerdas dalam berbagai skenario. HiLens ditunjukkan pada Gambar 8.7.



Gambar 8.7 Kolaborasi End-Cloud Huawei Hilens

Fitur produk HiLens sebagai berikut:

1. Inferensi kolaboratif End-Cloud, yang menyeimbangkan penundaan komputasi rendah dan akurasi tinggi.
2. Analisis data end-side, meminimalkan biaya penyimpanan cloud.
3. Pengembangan keterampilan terpadu, memperpendek siklus pengembangan.
4. Pasar keterampilan menyediakan keahlian yang kaya, pelatihan daring, dan penerapan satu klik.

1. Keunggulan Produk HiLens

(a) Inferensi kolaboratif End-Cloud.

- Kolaborasi End-Cloud dapat bekerja dalam skenario jaringan yang tidak stabil dan menghemat bandwidth pengguna.
- Perangkat end-side dapat bekerja sama dengan perangkat cloud-side untuk memperbarui model secara daring, dan dengan cepat meningkatkan akurasi end-side.
- End-side dapat menganalisis data lokal, sehingga meminimalkan aliran data cloud dan menghemat biaya penyimpanan.

(b) Platform pengembangan keterampilan terpadu.

Dengan optimalisasi kolaboratif perangkat lunak dan perangkat keras, produk HiLens menggunakan kerangka kerja pengembangan keterampilan terpadu, merangkum komponen dasar, dan mendukung model pembelajaran mendalam yang umum.

(c) Desain lintas platform.

- Produk HiLens mendukung prosesor AI Ascend, chip seri Hisilicon 35xx, dan chip arus utama lainnya di pasaran, yang dapat memenuhi kebutuhan skenario pemantauan arus utama.
- Produk HiLens menyediakan konversi model dan optimalisasi algoritma untuk chip end-side.

(d) Pasar keterampilan yang kaya.

- Beragam keterampilan telah tersedia di pasar keterampilan HiLens, seperti deteksi bentuk manusia dan deteksi tangisan. Pengguna dapat memilih keterampilan yang dibutuhkan langsung dari pasar keterampilan dan dengan cepat menerapkannya di sisi pengguna tanpa langkah pengembangan apa pun.
- HiLens telah melakukan banyak optimasi algoritma untuk mengatasi masalah memori kecil dan presisi rendah pada perangkat end-to-end.
- Pengembang juga dapat mengembangkan keterampilan khusus melalui konsol admin HiLens dan bergabung dengan pasar keterampilan.

2. Aplikasi HiLens

(a) Dari perspektif pengguna, Huawei HiLens terutama memiliki tiga jenis pengguna: pengguna biasa, pengembang AI, dan produsen kamera.

- ➲ Pengguna biasa: Pengguna biasa adalah pengguna keterampilan, yang dapat berupa anggota keluarga, pemilik supermarket, petugas parkir, atau manajer lokasi. HiLens Kit dapat mencapai fungsi-fungsi seperti peningkatan keamanan keluarga, penghitungan arus pelanggan, identifikasi atribut kendaraan dan plat

nomor, serta deteksi penggunaan helm pengaman. Pengguna hanya perlu mendaftar ke konsol manajemen HiLens, membeli, atau menyesuaikan keahlian yang sesuai (seperti identifikasi plat nomor, pengenalan helm pengaman, dll.) di pasar keahlian platform. Menginstal HiLens Kit hanya dengan sekali klik dapat memenuhi kebutuhan mereka.

- Pengembang AI: Pengembang AI biasanya adalah teknisi atau mahasiswa yang terlibat dalam pengembangan AI. Mereka dapat dengan mudah menerapkannya ke perangkat untuk melihat cara kerja keahlian tersebut secara langsung jika ingin mendapatkan penghasilan atau pengetahuan tertentu dari AI. Para pengguna ini dapat mengembangkan keahlian AI di konsol manajemen HiLens.
- HiLens mengintegrasikan kerangka kerja HiLens di sisi pengguna, yang merangkum komponen-komponen dasar, menyederhanakan proses pengembangan, dan menyediakan antarmuka API terpadu yang memudahkan pengembang untuk menyelesaikan pengembangan keahlian. Setelah pengembangan keahlian selesai, pengguna dapat menerapkannya ke HiLens Kit dengan sekali klik untuk melihat hasilnya. Pada saat yang sama, keahlian juga dapat dirilis ke pasar keahlian agar dapat dibeli dan digunakan oleh pengguna lain, atau keahlian dapat dibagikan sebagai templat untuk dipelajari oleh pengembang lain.
- Produsen kamera: Produsen produk kamera chip seri Hisilicon 35xx. Meskipun seri kamera ini mungkin memiliki kemampuan AI yang lemah atau bahkan tidak ada, produsen berharap produk mereka akan lebih kompetitif setelah mendapatkan kemampuan AI yang lebih kuat.

(b) Dalam hal skenario aplikasi, Huawei HiLens dapat diterapkan di berbagai bidang seperti pengawasan cerdas rumah, pengawasan cerdas taman, pengawasan cerdas supermarket, dan pengawasan cerdas di dalam kendaraan.

- Pengawasan cerdas rumah: Kamera cerdas rumah dan produsen rumah pintar, yang terintegrasi dengan chip seri Huawei Hisilicon 35xx, serta Kit HiLens berkinerja tinggi yang terintegrasi dengan chip D, dapat digunakan untuk meningkatkan kemampuan analisis cerdas video rumah. Hal ini dapat diterapkan pada skenario berikut:
  - ✓ Deteksi bentuk manusia. Dapat mendeteksi sosok manusia dalam pengawasan keluarga, merekam waktu kemunculannya, atau mengirimkan peringatan ke ponsel setelah terdeteksi dalam jangka waktu tertentu ketika tidak ada orang di rumah.
  - ✓ Deteksi jatuh. Ketika mendeteksi seseorang jatuh, peringatan akan dikirim. Alat ini terutama digunakan untuk perawatan lansia.
  - ✓ Deteksi tangisan. Ketika mendeteksi tangisan bayi, peringatan akan dikirim ke ponsel pengguna. Alat ini digunakan untuk perawatan anak.
  - ✓ Pengenalan kosakata. Alat ini dapat menyesuaikan kata tertentu, seperti "tolong", dan memberikan peringatan ketika kata tersebut terdeteksi.

- ✓ Deteksi atribut wajah. Atribut wajah terdeteksi, termasuk jenis kelamin, usia, wajah tersenyum, yang dapat digunakan untuk keamanan pintu, penyaringan video, dan sebagainya.
- ✓ Album waktu. Klip video anak-anak yang terdeteksi digabungkan menjadi album waktu untuk mencatat pertumbuhan anak-anak.

➲ Pengawasan cerdas taman: Melalui konsol manajemen HiLens, kemampuan AI didistribusikan ke stasiun kecil cerdas dengan chip Ascend terintegrasi, sehingga perangkat edge dapat menangani data tertentu. Alat ini dapat diterapkan pada skenario berikut.

- ✓ Gerbang pengenalan wajah. Berdasarkan teknologi pengenalan wajah, pengenalan wajah dapat direalisasikan untuk gerbang masuk dan keluar taman.
- ✓ Identifikasi plat nomor/kendaraan. Di pintu masuk dan keluar area parkir atau garasi, identifikasi plat nomor dan jenis kendaraan dapat dilakukan untuk mendapatkan sertifikasi otorisasi plat nomor dan jenis kendaraan.
- ✓ Deteksi helm pengaman. Pekerja yang tidak mengenakan helm pengaman akan terdeteksi melalui pengawasan video dan peringatan akan diberikan pada peralatan yang ditentukan.
- ✓ Pemulihan jalur. Wajah orang atau kendaraan yang sama yang diidentifikasi oleh beberapa kamera dianalisis untuk memulihkan jalur pejalan kaki atau kendaraan.
- ✓ Pengambilan wajah. Dalam pengawasan, pengenalan wajah digunakan untuk mengidentifikasi wajah yang ditentukan, yang dapat digunakan untuk pengenalan daftar hitam.
- ✓ Deteksi suara abnormal. Ketika suara abnormal seperti pecahan kaca dan suara ledakan terdeteksi, peringatan harus dilaporkan.
- ✓ Deteksi intrusi. Peringatan akan dikirim ketika bentuk manusia terdeteksi di area pengawasan yang ditentukan.

➲ Pengawasan cerdas pusat perbelanjaan: Perangkat terminal yang berlaku untuk pusat perbelanjaan meliputi HiLens Kit, stasiun tepi cerdas, dan kamera komersial. Supermarket kecil dapat mengintegrasikan HiLens Kit, yang mendukung 4–5 saluran analisis video. Dengan ukurannya yang kecil, perangkat ini dapat ditempatkan di dalam ruangan. Dapat diterapkan pada skenario berikut:

- ✓ Statistik arus pelanggan. Melalui pengawasan toko dan supermarket, statistik arus pelanggan yang cerdas di pintu masuk dan keluar dapat diwujudkan, yang dapat digunakan untuk menganalisis perubahan arus pelanggan pada periode yang berbeda.
- ✓ Identifikasi VIP. Melalui pengenalan wajah, pelanggan VIP dapat diidentifikasi secara akurat untuk membantu merumuskan strategi pemasaran.
- ✓ Statistik pelanggan baru dan lama. Melalui pengenalan wajah, jumlah pelanggan baru dan lama dapat dihitung.

- ✓ Peta panas penghitungan pejalan kaki. Melalui analisis peta panas penghitungan pejalan kaki, kepadatan kerumunan dapat diidentifikasi, yang bermanfaat untuk analisis popularitas komoditas.

➲ Kendaraan Cerdas: Peralatan kendaraan cerdas berbasis sistem Android, yang dapat mewujudkan analisis cerdas secara real-time terhadap kondisi internal dan eksternal. Ini berlaku untuk skenario seperti deteksi perilaku mengemudi, pengawasan "dua jenis bus penumpang dan satu jenis truk bahan kimia berbahaya". Ini dapat diterapkan pada skenario berikut.

- ✓ Pengenalan wajah. Dengan mengidentifikasi apakah wajah pengemudi cocok dengan perpustakaan foto pemilik yang telah tersimpan, otoritas pengemudi dikonfirmasi.
- ✓ Mengemudi dalam kondisi kelelahan. Pengawasan real-time terhadap kondisi mengemudi pengemudi dapat dideteksi, dan peringatan cerdas akan dikirimkan jika kelelahan mengemudi terdeteksi.
- ✓ Analisis postur. Mendeteksi perilaku pengemudi yang terganggu, seperti menelepon, minum air, melihat-lihat, dan merokok.
- ✓ Deteksi kendaraan dan pejalan kaki. Dapat digunakan untuk mendeteksi pejalan kaki di area buta.

Platform Dasar EI: Layanan Mesin Grafik

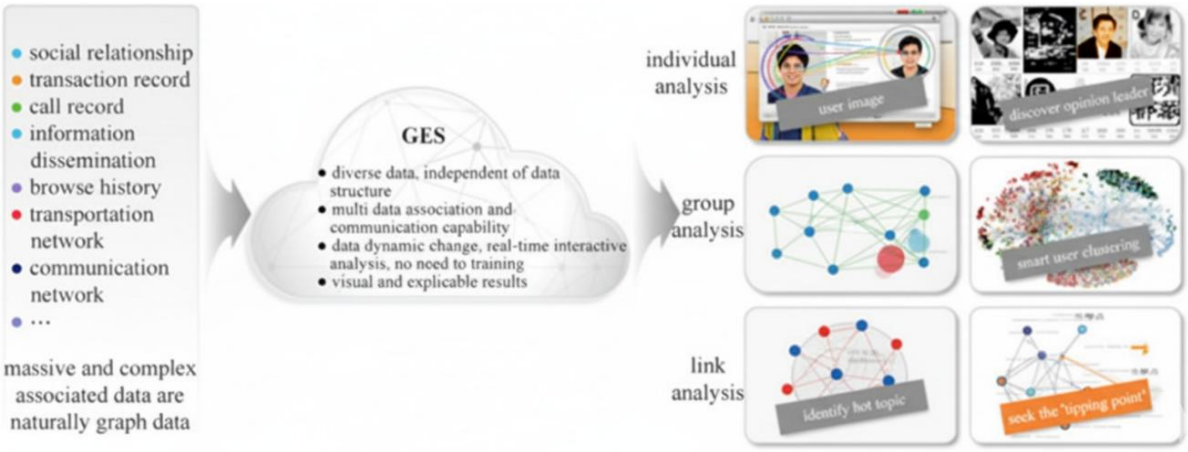
Huawei *Graph Engine Service* (GES) adalah mesin grafik asli terdistribusi komersial pertama dengan hak kekayaan intelektual independen di Tiongkok. Layanan ini untuk kueri dan analisis data struktur grafik berdasarkan "relasi". Mengadopsi EYWA, mesin grafik berkinerja tinggi yang dikembangkan oleh Huawei, sebagai intinya, GES memiliki sejumlah paten independen. Layanan ini banyak digunakan dalam skenario dengan data relasional yang kaya seperti aplikasi sosial, analisis hubungan perusahaan, distribusi logistik, perencanaan rute bus antar-jemput, grafik pengetahuan perusahaan, pengendalian risiko, rekomendasi, opini publik, dan pencegahan penipuan.

Data terkait yang masif dan kompleks seperti hubungan sosial, catatan transaksi, dan jaringan transportasi secara alami merupakan data grafik, sementara Huawei GES adalah layanan untuk penyimpanan, kueri, dan analisis data terstruktur grafik berdasarkan relasi. GES memainkan peran penting dalam berbagai skenario seperti aplikasi sosial, analisis hubungan perusahaan, distribusi logistik, perencanaan rute bus antar-jemput, grafik pengetahuan perusahaan, dan pengendalian risiko.

Dalam analisis individual, GES melakukan analisis potret pengguna pada masing-masing node berdasarkan jumlah dan karakteristik tetangganya. GES juga dapat menggali dan mengidentifikasi pemimpin opini berdasarkan karakteristik dan kepentingan node tersebut. Misalnya, dengan mempertimbangkan faktor kuantitas, semakin banyak perhatian yang diterima pengguna dari orang lain, semakin penting pengguna tersebut. Di sisi lain, faktor transfer kualitas dipertimbangkan berdasarkan karakteristik transfer pada grafik, kualitas penggemar ditransfer ke yang bersangkutan. Ketika yang bersangkutan adalah penggemar berkualitas tinggi, kualitas yang bersangkutan meningkat.

Dalam analisis grup, dengan algoritma propagasi label dan algoritma penemuan komunitas, GES membagi node dengan karakteristik serupa ke dalam satu kelas, sehingga dapat diterapkan pada skenario klasifikasi node seperti rekomendasi teman, rekomendasi grup, dan pengelompokan pengguna. Misalnya, dalam lingkaran sosial, jika dua orang memiliki teman yang sama, mereka akan memiliki kemungkinan untuk menjadi teman di masa mendatang. Semakin banyak teman yang sama, semakin kuat hubungan mereka. Hal ini memungkinkan rekomendasi teman berdasarkan jumlah teman bersama.

Dalam analisis tautan, GES dapat menggunakan algoritma analisis tautan dan algoritma prediksi hubungan untuk memprediksi dan mengidentifikasi topik hangat, sehingga dapat menemukan "titik kritis", seperti yang ditunjukkan pada Gambar 8.8. Dapat dilihat bahwa skenario aplikasi GES di dunia nyata sangat kaya dan luas. Akan ada lebih banyak industri dan skenario aplikasi di masa mendatang yang layak untuk dieksplorasi secara mendalam.



Gambar 8.8 Layanan mesin grafik

Keunggulan produk GES adalah sebagai berikut.

1. Skala besar: GES menyediakan organisasi data yang efisien, yang dapat melakukan kueri dan menganalisis data 10 miliar node dan 100 miliar edge secara lebih efektif.
2. Performa tinggi: GES menyediakan mesin komputasi grafik terdistribusi yang dioptimalkan secara mendalam, yang memungkinkan pengguna memperoleh kemampuan kueri waktu nyata dengan konkurensi tinggi, tingkat kedua, dan multi-hop.
3. Integrasi kueri dan analisis: GES menyediakan beragam algoritma analisis grafik, yang mencakup beragam kemampuan analisis untuk bisnis seperti analisis hubungan, perencanaan rute, dan pemasaran presisi.4. Kemudahan penggunaan: GES menyediakan antarmuka analisis visual yang mudah digunakan dan terpandu, dan apa yang Anda lihat adalah apa yang Anda dapatkan; GES mendukung bahasa kueri Gremlin, yang kompatibel dengan kebiasaan pengguna.

Fungsi-fungsi yang disediakan oleh GES adalah sebagai berikut.

1. **Algoritma Rich Domain**: GES mendukung berbagai algoritma seperti PageRank, K-core, jalur terpendek, algoritma propagasi label, penghitungan segitiga, dan prediksi interaksi.
2. **Analisis grafis visual**: GES menyediakan lingkungan eksplorasi terpandu yang mendukung visualisasi hasil kueri.
3. **API Analisis Kueri**: GES menyediakan API untuk kueri grafik, statistik indeks grafik, kueri Gremlin, algoritma grafik, manajemen grafik, manajemen cadangan, dll.
4. **Kompatibel dengan Ecology sumber terbuka**: GES kompatibel dengan Apache TinkerPop Gremlin 3.3.0.
5. **Manajemen Grafik**: GES menyediakan layanan mesin grafik seperti ikhtisar, manajemen grafik, cadangan grafik, dan manajemen metadata.

Pengantar Layanan Lain yang Disediakan oleh Keluarga EI

1. BOT Percakapan

Layanan BOT Percakapan (CBS) terdiri dari QABot, TaskBot, Bot Inspeksi Kualitas Cerdas, dan Bot Konvensional yang Disesuaikan. BOT percakapan ditunjukkan pada Gambar 8.9.

Gambar 8.9 Bot percakapan

(a) QABot: Dapat membantu perusahaan dengan cepat membangun, merilis, dan mengelola sistem robot tanya jawab cerdas.

(b) Taskbot: Dapat secara akurat memahami maksud percakapan dan mengekstrak informasi penting. Dapat digunakan untuk lalu lintas telepon cerdas dan perangkat keras cerdas.

(c) Bot inspeksi kualitas cerdas menggunakan algoritma bahasa alami dan aturan yang ditentukan pengguna untuk menganalisis percakapan antara agen layanan pelanggan

dan pelanggan dalam skenario pusat panggilan, membantu perusahaan meningkatkan kualitas layanan dan kepuasan pelanggan.

(d) Bot percakapan yang dikustomisasi dapat membangun bot AI dengan berbagai kemampuan sesuai kebutuhan pelanggan, termasuk QA basis pengetahuan dan grafik pengetahuan, percakapan berbasis tugas, pemahaman bacaan, pembuatan teks otomatis, multimodalitas, dan melayani pelanggan di berbagai industri.

Skenario Aplikasi	Menjawab pertanyaan dengan cerdas	Analisis opini publik	Identifikasi iklan	Komputasi pengetahuan	Pembuatan konten	Terjemahan
Melayani	Dasar – dasar pemrosesan bahasa alami	Pemahaman bahasa	Generasi bahasa	Terjemahan mesin		
Keterampilan pemrosesan bahasa alami	Segmentasi kata	Kesamaan teks	Kesamaan teks	Penguraian ketergantungan	Terjemahan dokumen	
	Analisis sentimen	Pemahaman niat	Klasifikasi teks	Pembuatan teks	Teks abstrak	Terjemahan mesin

Gambar 8.10 Layanan pemrosesan bahasa alami

2. Pemrosesan Bahasa Alami

Pemrosesan bahasa alami (NLP) menyediakan layanan terkait yang dibutuhkan robot untuk mewujudkan pemahaman semantik, yang terdiri dari empat sublayanan: dasar-dasar NLP, pemahaman bahasa, pembangkitan bahasa, dan penerjemahan mesin. Layanan NLP ditunjukkan pada Gambar 8.10. Dasar-Dasar Pemrosesan Bahasa Alami menyediakan API terkait bahasa alami bagi pengguna, termasuk segmentasi kata, pengenalan entitas bernama, ekstraksi kata kunci, kesamaan teks, yang dapat diterapkan pada skenario seperti tanya jawab cerdas, bot percakapan, analisis opini publik, rekomendasi konten, evaluasi dan analisis e-commerce.

Pemahaman bahasa menyediakan API terkait pemahaman bahasa bagi pengguna, seperti analisis sentimen, ekstraksi opini, klasifikasi teks, pemahaman intensi, dll., yang dapat diterapkan pada skenario seperti penambangan komentar, analisis opini publik, asisten cerdas, dan bot percakapan. Berdasarkan model bahasa tingkat lanjut, pembangkitan bahasa menghasilkan teks yang dapat dibaca sesuai dengan informasi masukan, termasuk teks, data, atau gambar. Dapat diterapkan pada skenario interaksi manusia-komputer seperti tanya jawab dan percakapan cerdas, ringkasan berita, dan pembuatan laporan.

Kustomisasi NLP bertujuan untuk membangun model pemrosesan bahasa alami yang unik sesuai dengan kebutuhan spesifik pelanggan, seperti model klasifikasi otomatis dokumen hukum yang dapat disesuaikan, model pembuatan laporan medis otomatis yang dapat disesuaikan, dan model analisis opini publik yang dapat disesuaikan untuk bidang tertentu, yang bertujuan untuk memberikan daya saing yang unik bagi aplikasi perusahaan.

3. Interaksi Ucapan

Pengenalan satu kalimat/pengenalan ucapan singkat

Transkripsi ucapan waktu nyata

Pengenalan berkas rekaman

Buku audio

Gambar 8.11 Interaksi Ucapan

Interaksi ucapan terdiri dari pengenalan ucapan, sintesis ucapan, dan transkripsi suara waktu nyata (real-time), seperti yang ditunjukkan pada Gambar 8.11.

Aplikasi utama pengenalan ucapan adalah sebagai berikut.

(a) Pencarian ucapan: Konten pencarian dimasukkan langsung melalui ucapan, sehingga pencarian lebih efisien. Pengenalan ucapan mendukung pencarian ucapan dalam berbagai skenario, seperti navigasi peta, pencarian web, dan sebagainya.

(b) Interaksi manusia-komputer: Melalui pengaktifan ucapan dan pengenalan ucapan, perintah ucapan dikirim ke perangkat terminal dan pengoperasian perangkat secara waktu nyata diimplementasikan untuk meningkatkan pengalaman interaksi manusia-komputer.

Aplikasi sintesis ucapan adalah sebagai berikut.

(a) Navigasi ucapan: Sintesis ucapan dapat mengonversi data navigasi internal menjadi materi ucapan, menyediakan layanan navigasi ucapan yang akurat bagi pengguna. Dengan kemampuan kustomisasi personalisasi sintesis ucapan, layanan ini menyediakan layanan navigasi ucapan yang kaya.

(b) Buku audio: Sintesis ucapan dapat mengubah konten teks buku, majalah, dan berita menjadi suara manusia yang realistis, sehingga orang-orang dapat sepenuhnya membebaskan mata mereka sambil mendapatkan informasi dan bersenang-senang dalam skenario seperti naik kereta bawah tanah, mengemudi, atau latihan fisik.

(c) Tindak lanjut telepon: Dalam sistem layanan pelanggan, konten tindak lanjut diubah menjadi suara manusia melalui sintesis ucapan, dan pengalaman pengguna ditingkatkan melalui komunikasi langsung dengan pelanggan melalui ucapan.

(d) Pendidikan cerdas: Sintesis ucapan dapat mensintesis konten teks dalam buku menjadi ucapan. Pengucapan yang mendekati orang sungguhan dapat mensimulasikan adegan pengajaran langsung, sehingga mewujudkan pembacaan nyaring dan pembacaan teks utama, membantu siswa lebih memahami dan menguasai konten pengajaran.

Aplikasi transkripsi ucapan waktu nyata adalah sebagai berikut.

(a) Subtitel langsung: Transkripsi ucapan waktu nyata mengubah audio dalam video langsung atau siaran langsung menjadi subtitel secara waktu nyata, memberikan pengalaman menonton yang lebih efisien bagi audiens dan memfasilitasi pemantauan konten.

(b) Perekaman konferensi secara real-time: Transkripsi ucapan secara real-time mengubah audio dalam video atau telekonferensi menjadi teks secara real-time, yang dapat memeriksa, mengubah, dan mengambil konten konferensi yang ditranskripsi secara real-time, sehingga meningkatkan efisiensi konferensi.

(c) Entri teks instan: Transkripsi ucapan secara real-time pada aplikasi seluler dapat digunakan untuk merekam dan menyediakan teks yang ditranskripsi secara real-time, seperti metode input ucapan. Transkripsi ini praktis untuk pasca-pemrosesan dan pengarsipan konten, menghemat tenaga kerja dan waktu perekaman, sehingga efisiensi konversi sangat ditingkatkan.

4. Analisis Video

Analisis video menyediakan layanan seperti analisis konten video, penyuntingan video, dan penandaan video.

Aplikasi analisis konten video adalah sebagai berikut.

(a) Manajemen pemantauan: Analisis konten video melakukan analisis secara real-time pada semua video di pusat perbelanjaan atau taman, untuk mengekstrak episode-episode penting, seperti pemantauan gudang, masalah kepatuhan kasir, penyumbatan pintu darurat. Analisis ini juga melakukan deteksi penyusup di area dengan keamanan tinggi, deteksi orang yang berkeliaran, deteksi objek terlantar, dll.; Pencegahan kehilangan yang cerdas dapat dilakukan, seperti pengawasan potret, deteksi pencurian, dll.

(b) Analisis pejalan kaki taman: Melalui analisis waktu nyata terhadap pejalan kaki aktif di taman, analisis konten video mengidentifikasi dan melacak orang-orang berisiko tinggi setelah mengonfigurasi daftar hitam pejalan kaki, dan mengirimkan peringatan. Analisis ini menghitung arus pejalan kaki di persimpangan utama untuk merumuskan strategi pengelolaan taman.

(c) Analisis karakter video: Melalui analisis tokoh publik dalam video media, analisis konten video secara akurat mengidentifikasi tokoh politik, bintang film, dan selebritas lain dalam video tersebut.

(d) Pengenalan gerakan: Analisis konten video mendeteksi dan mengenali gerakan dalam video setelah menganalisis informasi bingkai depan dan belakang, informasi gerakan aliran optik, dan informasi konten adegan.

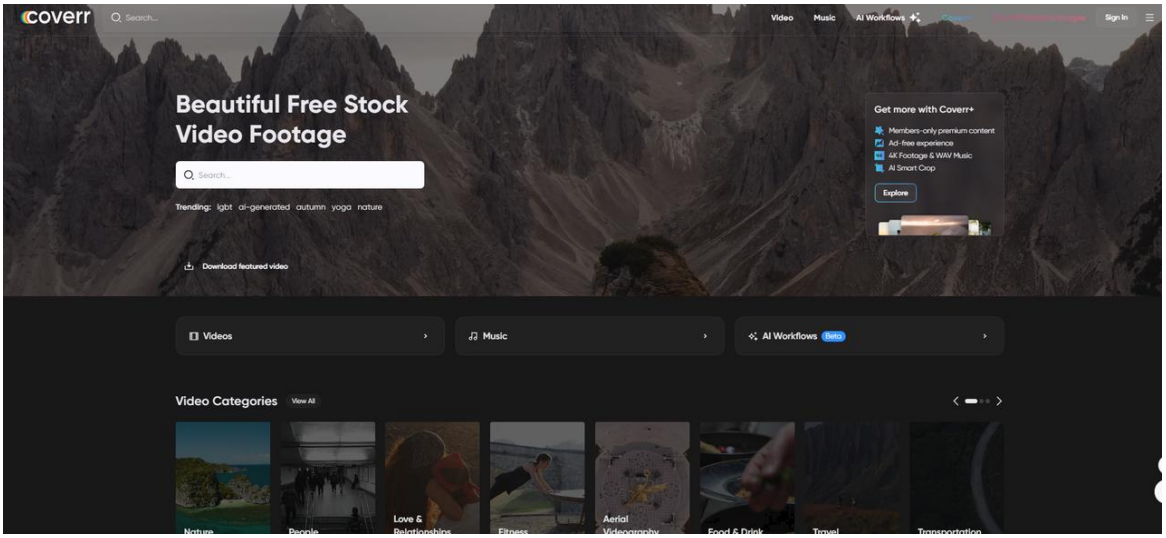
Aplikasi penyuntingan video adalah sebagai berikut.

(a) Ekstraksi klip sorotan: Berdasarkan relevansi konten dan sorotan video, penyuntingan video mengekstrak segmen adegan untuk membuat ringkasan video.

(b) Pemecahan Video Berita: Penyuntingan video membagi seluruh berita menjadi segmen-segmen berita dengan tema berbeda berdasarkan analisis karakter, adegan, suara, dan pengenalan karakter dalam berita.

Aplikasi penandaan video adalah sebagai berikut.

(a) Pencarian Video: Berdasarkan analisis klasifikasi adegan video, pengenalan orang, pengenalan suara, dan pengenalan karakter, penandaan video membentuk tag klasifikasi hierarkis untuk mendukung pencarian video yang akurat dan efisien serta meningkatkan pengalaman pencarian, seperti yang ditunjukkan pada Gambar 8.12.



Gambar 8.12 Pencarian Video

(b) Rekomendasi Video: Berdasarkan analisis klasifikasi adegan, pengenalan orang, pengenalan suara, dan OCR, penandaan video membentuk tag klasifikasi hierarkis untuk rekomendasi video yang dipersonalisasi.

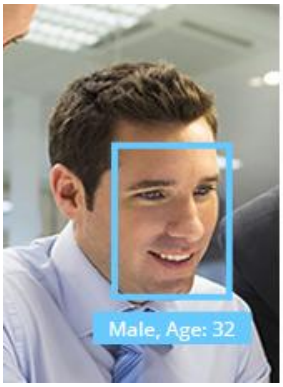
5. Pengenalan Gambar

Pengenalan gambar, berbasis teknologi pembelajaran mendalam, dapat mengidentifikasi konten visual dalam gambar secara akurat, menyediakan puluhan ribu objek, adegan, dan tag konsep. Teknologi ini memiliki kemampuan deteksi target dan pengenalan atribut, sehingga membantu pelanggan mengidentifikasi dan memahami konten gambar secara akurat. Pengenalan gambar menyediakan fungsi-fungsi seperti analisis pemandangan,

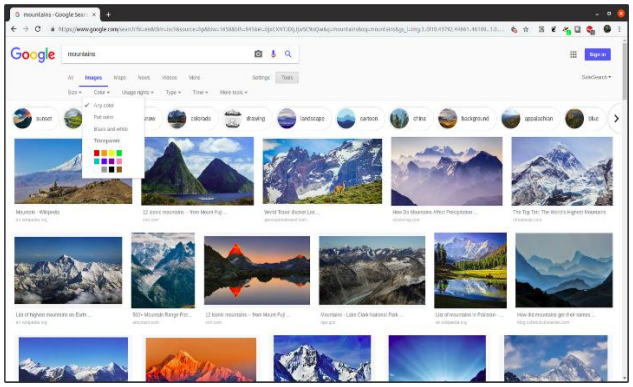
album foto cerdas, deteksi target, dan pencarian gambar, seperti yang ditunjukkan pada Gambar 8.13.

Analisis Pemandangan

Album Foto Pintar



Deteksi Target



Pencarian Gambar

Gambar 8.13 Aplikasi Pengenalan Gambar

(a) Analisis pemandangan: Kurangnya penandaan konten menyebabkan rendahnya efisiensi pengambilan. Penandaan gambar dapat mengidentifikasi konten gambar secara akurat, meningkatkan efisiensi dan akurasi pengambilan, sehingga rekomendasi yang dipersonalisasi, pengambilan konten, dan distribusi menjadi lebih efektif.

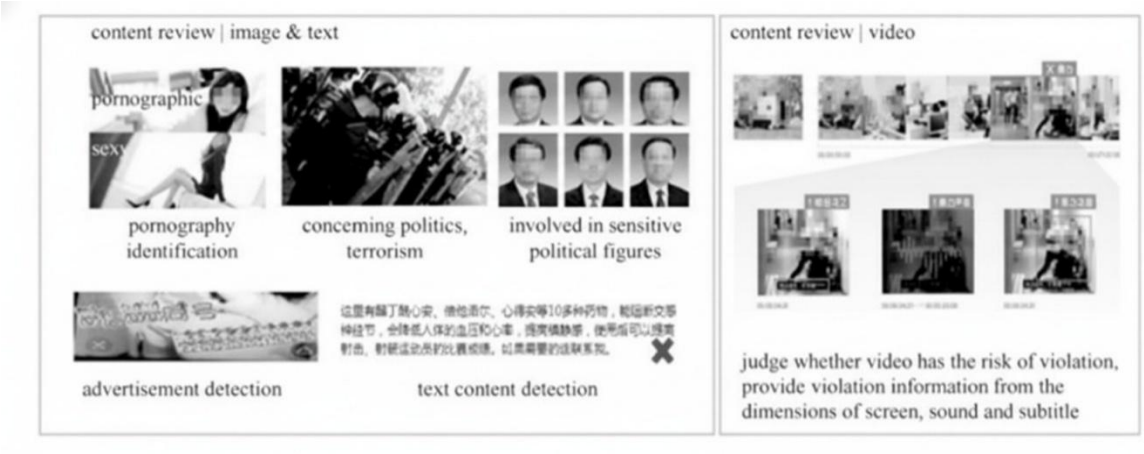
(b) Album foto pintar: Berdasarkan puluhan ribu tag yang diidentifikasi dari gambar, album foto pintar dapat dikustomisasi berdasarkan kategori, seperti "tanaman", "makanan", "pekerjaan", yang memudahkan pengelolaan pengguna.

(c) Deteksi Target: Di lokasi konstruksi, berdasarkan pengenalan gambar yang dikustomisasi, sistem deteksi target dapat memantau secara real-time apakah staf di lokasi mengenakan helm pengaman, untuk mengurangi risiko keselamatan.

(d) Pencarian Gambar: Pencarian basis data gambar yang besar cukup merepotkan. Teknologi pencarian gambar, berdasarkan tag gambar, dapat dengan cepat mencari gambar yang diinginkan, baik pengguna memasukkan kata kunci maupun gambar.

6. Tinjauan Konten

Tinjauan konten terdiri dari tinjauan teks, tinjauan gambar, dan tinjauan video. Berdasarkan teknologi deteksi teks, gambar, dan video terdepan, sistem ini dapat secara otomatis mendeteksi konten yang berkaitan dengan pornografi, iklan, terorisme, dan politik, sehingga membantu pelanggan mengurangi risiko pelanggaran bisnis. Tinjauan konten ditunjukkan pada Gambar 8.14.



Gambar 8.14 Tinjauan Konten

Tinjauan konten mencakup aplikasi-aplikasi berikut.

(a) Identifikasi pornografi: Tinjauan konten dapat menilai tingkat pornografi suatu gambar, memberikan tiga skor keyakinan: pornografi, seksi, dan normal.
(b) Deteksi terorisme: Tinjauan konten dapat dengan cepat mendeteksi apakah gambar tersebut mengandung konten yang berkaitan dengan api, senjata api, pisau, pertumpahan darah, bendera terorisme, dll.
(c) Tokoh sensitif yang terlibat dalam politik: Tinjauan konten dapat menilai apakah konten tersebut melibatkan tokoh politik sensitif.
(d) Deteksi konten teks: Tinjauan konten dapat mendeteksi apakah konten teks tersebut berkaitan dengan pornografi, politik, iklan, pelecehan, penambahan air, dan barang selundupan.
(e) Tinjauan Video: Tinjauan konten dapat menilai apakah video berisiko melanggar sehingga dapat memberikan informasi pelanggaran dari dimensi layar, suara, dan subtitel.

7. Pencarian Gambar

Pencarian gambar adalah pencarian gambar dengan gambar. Berdasarkan pembelajaran mendalam dan teknologi pengenalan gambar, pencarian ini menggunakan vektorisasi fitur dan kemampuan pencarian untuk membantu pengguna mencari gambar yang sama atau serupa dari pustaka yang ditentukan.

Aplikasi pencarian gambar adalah sebagai berikut.

(a) Pencarian gambar komoditas: Pencarian gambar dapat mencari gambar yang diambil oleh pengguna dari pustaka komoditas. Dengan pencarian gambar serupa, komoditas

yang sama atau serupa akan dikirimkan kepada pengguna, untuk menjual atau merekomendasikan komoditas terkait, seperti yang ditunjukkan pada Gambar 8.15.

Gambar 8.15 Pencarian Komoditas

(b) Pencarian hak cipta gambar: Hak cipta gambar merupakan aset penting bagi situs web fotografi dan desain. Pencarian gambar dapat dengan cepat menemukan gambar yang melanggar hukum dari basis data gambar yang besar untuk melindungi hak-hak situs web sumber gambar.

8. Pengenalan Wajah

Pengenalan wajah dapat dengan cepat mengidentifikasi wajah dalam gambar, menganalisis informasi penting, dan memperoleh atribut wajah, sehingga perbandingan dan pengambilan wajah yang akurat dapat dicapai. Aplikasi pengenalan wajah adalah sebagai berikut.

(a) Autentikasi Identitas: Identifikasi dan perbandingan wajah dapat digunakan untuk autentikasi identitas, yang cocok untuk area autentikasi seperti bandara dan bea cukai.

(b) Absensi Elektronik: Identifikasi dan perbandingan wajah dapat diterapkan untuk absensi elektronik karyawan di perusahaan serta pemantauan keamanan.

(c) Analisis Trajektori: Pencarian wajah dapat mengambil N citra wajah dan tingkat kemiripannya yang paling mirip dengan wajah input dalam basis data citra. Berdasarkan informasi waktu, tempat, dan perilaku dari citra yang dikembalikan, hal ini dapat membantu pelanggan melakukan analisis trajektori.

(d) Analisis Alur Pelanggan: Analisis alur pelanggan sangat bermanfaat bagi pusat perbelanjaan. Berdasarkan teknologi pengenalan wajah, perbandingan, dan pencarian, analisis ini dapat menganalisis informasi pelanggan seperti usia dan jenis kelamin secara akurat, sehingga dapat membedakan pelanggan baru dan lama, sehingga membantu pelanggan dalam pemasaran yang efisien. Analisis alur pelanggan ditunjukkan pada Gambar 8.16.

Gambar 8.16 Analisis alur pelanggan

9. Pengenalan Karakter Optik

Pengenalan Karakter Optik (OCR) adalah pengenalan teks dalam gambar atau salinan pindaian menjadi teks yang dapat diedit. OCR dapat menggantikan input manual untuk meningkatkan efisiensi bisnis. OCR mendukung pengenalan karakter untuk berbagai skenario seperti KTP, SIM, faktur, dokumen bea cukai Inggris, formulir umum, dan karakter umum lainnya, seperti yang ditunjukkan pada Gambar 8.17. OCR mendukung pengenalan karakter untuk kelas umum, kelas sertifikat, kelas tagihan, kelas industri, dan kelas templat khusus.

Autentikasi Pengguna

Audit Penggantian Keuangan

Keuangan dan Asuransi

Dokumen Bea Cukai Elektronik

Gambar 8.17 Pengenalan karakter optik

OCR umum mendukung pengenalan otomatis informasi teks pada gambar dengan format apa pun, seperti formulir, dokumen, dan gambar jaringan. OCR dapat menganalisis berbagai tata letak dan formulir, sehingga dapat dengan cepat mewujudkan elektronisasi berbagai dokumen.

Penerapan OCR umum adalah sebagai berikut.

(a) Pengarsipan elektronik dokumen dan laporan historis perusahaan: OCR dapat mengidentifikasi informasi karakter dalam dokumen dan laporan serta membuat berkas elektronik, sehingga memudahkan pengambilan data dengan cepat.

(b) Pengisian otomatis informasi pengirim pengiriman ekspres: OCR dapat mengidentifikasi informasi kontak pada gambar dan mengisi formulir pengiriman ekspres secara otomatis, sehingga mengurangi input manual.

(c) Peningkatan efisiensi penanganan kontrak: OCR dapat mengidentifikasi informasi terstruktur secara otomatis dan mengekstrak area tanda tangan dan stempel, yang membantu audit cepat.

(d) Dokumen kepabeanan elektronik: Karena banyak perusahaan memiliki bisnis di luar negeri, OCR umum dapat mewujudkan struktur otomatis dan elektronisasi data dokumen kepabeanan, sehingga meningkatkan efisiensi dan akurasi entri informasi.

Pengenalan karakter sertifikat mendukung identifikasi otomatis informasi yang valid dan ekstraksi terstruktur bidang-bidang penting pada KTP, SIM, STNK, dan paspor.

Penerapan pengenalan karakter sertifikat adalah sebagai berikut.

(a) Autentikasi Cepat: Dapat menyelesaikan autentikasi nama asli untuk pembukaan rekening ponsel dan proses lainnya dengan cepat, sehingga mengurangi biaya verifikasi identitas pengguna.

(b) Entri Informasi Otomatis: Dapat mengidentifikasi informasi penting dalam sertifikat sehingga menghemat entri manual dan meningkatkan efisiensi.

(c) Verifikasi Informasi Identitas: Dapat memverifikasi apakah pengguna adalah pemegang sertifikat asli.

Pengenalan Karakter Jenis Tagihan mendukung pengenalan otomatis dan ekstraksi terstruktur informasi valid pada berbagai faktur dan formulir, seperti faktur PPN, faktur penjualan kendaraan bermotor, faktur medis, dll.

Aplikasi pengenalan karakter tagihan adalah sebagai berikut.

(a) Entri Otomatis Informasi Dokumen Penggantian: Dapat mengidentifikasi informasi penting dalam faktur dengan cepat dan secara efektif mempersingkat waktu penggantian.

(b) Entri Otomatis Informasi Dokumen: Dapat memasukkan informasi penting dalam faktur penjualan kendaraan bermotor dan kontrak dengan cepat, sehingga meningkatkan efisiensi pemrosesan pinjaman kendaraan.

(c) Asuransi Kesehatan: Dapat secara otomatis mengidentifikasi bidang-bidang penting seperti detail obat, usia, dan jenis kelamin pada dokumen medis sebelum

memasukkannya ke dalam sistem. Dikombinasikan dengan OCR kartu identitas dan kartu bank, sistem ini dapat menyelesaikan proses klaim asuransi dengan cepat.

Pengenalan karakter jenis industri mendukung ekstraksi dan pengenalan informasi terstruktur dari berbagai gambar spesifik industri, seperti lembar logistik dan dokumen tes medis, yang membantu meningkatkan efisiensi otomatisasi industri.

Aplikasi pengenalan karakter jenis industri adalah sebagai berikut.

(a) Pengisian otomatis informasi pengirim untuk pengiriman ekspres: Sistem ini dapat mengidentifikasi informasi kontak pada gambar sebelum mengisi formulir pengiriman ekspres secara otomatis sehingga meminimalkan input manual.

(b) Asuransi kesehatan: Sistem ini dapat secara otomatis mengidentifikasi bidang-bidang penting seperti detail obat, usia, dan jenis kelamin pada dokumen medis, lalu memasukkannya ke dalam sistem. Dikombinasikan dengan OCR kartu identitas dan kartu bank, sistem ini dapat menyelesaikan proses klaim asuransi dengan cepat.

Pengenalan karakter jenis templat yang disesuaikan mendukung templat pengenalan yang ditentukan pengguna. Sistem ini dapat menentukan bidang-bidang penting yang akan dikenali, sehingga mewujudkan pengenalan otomatis dan ekstraksi struktural gambar berformat spesifik pengguna.

(a) Identifikasi berbagai sertifikat: Untuk gambar kartu dengan berbagai format, dapat digunakan untuk membuat templat guna mewujudkan identifikasi dan ekstraksi bidang-bidang kunci secara otomatis.

(b) Pengenalan berbagai tagihan: Untuk berbagai gambar tagihan, dapat digunakan untuk membuat templat guna mewujudkan pengenalan dan ekstraksi bidang-bidang kunci secara otomatis.

8.2 MODELARTS

Sebagai platform dasar EI dalam keluarga layanan EI, ModelArts adalah platform pengembangan terpadu bagi para pengembang AI. Platform ini menyediakan prapemrosesan data masif dan anotasi semi-otomatis, pelatihan terdistribusi skala besar, pembuatan model otomatis, dan kemampuan penerapan sesuai permintaan untuk model End, Edge, dan Cloud, membantu pengguna membuat dan menerapkan model dengan cepat serta mengelola alur kerja AI siklus penuh.

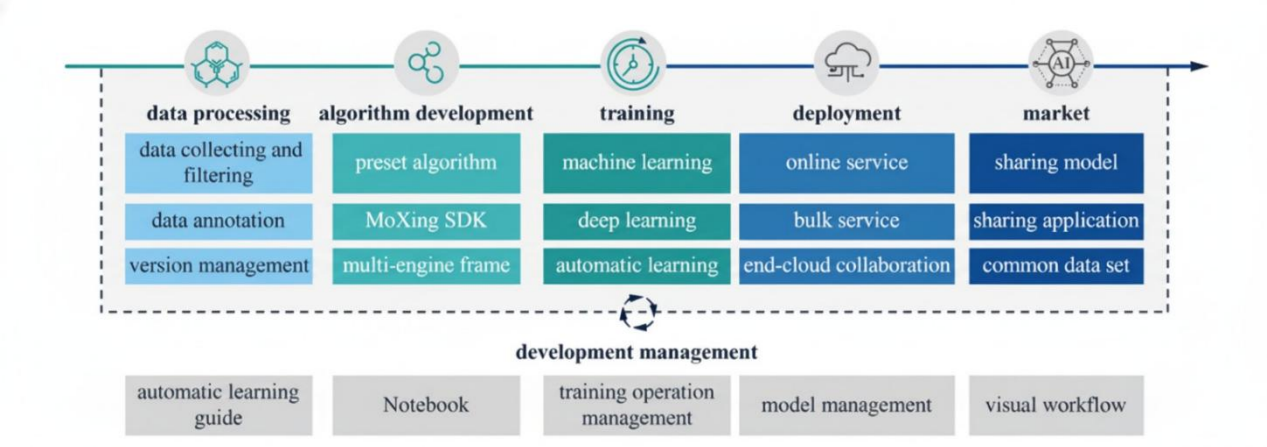
"Terpadu" berarti semua aspek pengembangan AI, termasuk pemrosesan data, pengembangan algoritma, pelatihan model, dan penerapan model, dapat diselesaikan di ModelArts. Secara teknis, ModelArts mendukung berbagai sumber daya komputasi heterogen, sehingga para pengembang dapat memilih untuk menggunakannya secara fleksibel sesuai kebutuhan mereka, terlepas dari teknologi yang mendasarinya. Pada saat yang sama, ModelArts mendukung kerangka kerja pengembangan AI sumber terbuka arus utama seperti TensorFlow dan MXNet, serta kerangka kerja algoritma yang dikembangkan sendiri agar sesuai dengan kebiasaan penggunaan para pengembang.

Bertujuan untuk membuat pengembangan AI lebih mudah dan nyaman, ModelArts menyediakan proses yang praktis dan mudah digunakan bagi para pengembang AI. Misalnya,

pengembang yang berorientasi bisnis dapat menggunakan proses pembelajaran otomatis untuk membangun aplikasi AI dengan cepat tanpa berfokus pada model atau pengodean; pemula AI dapat menggunakan algoritma yang telah ditetapkan untuk membangun aplikasi AI tanpa terlalu memikirkan pengembangan model; dilengkapi dengan beragam lingkungan pengembangan, proses operasi, dan mode oleh ModelArts, insinyur AI dapat dengan mudah mengodekan ekspansi dan membangun model dan aplikasi dengan cepat.

Fungsi ModelArts

ModelArts memungkinkan pengembang untuk menyelesaikan semua tugas sekaligus, mulai dari persiapan data hingga pengembangan algoritma dan pelatihan model, hingga akhirnya menerapkan model dan mengintegrasikannya ke dalam lingkungan produksi. Ikhtisar fungsi ModelArts ditunjukkan pada Gambar 8.18.



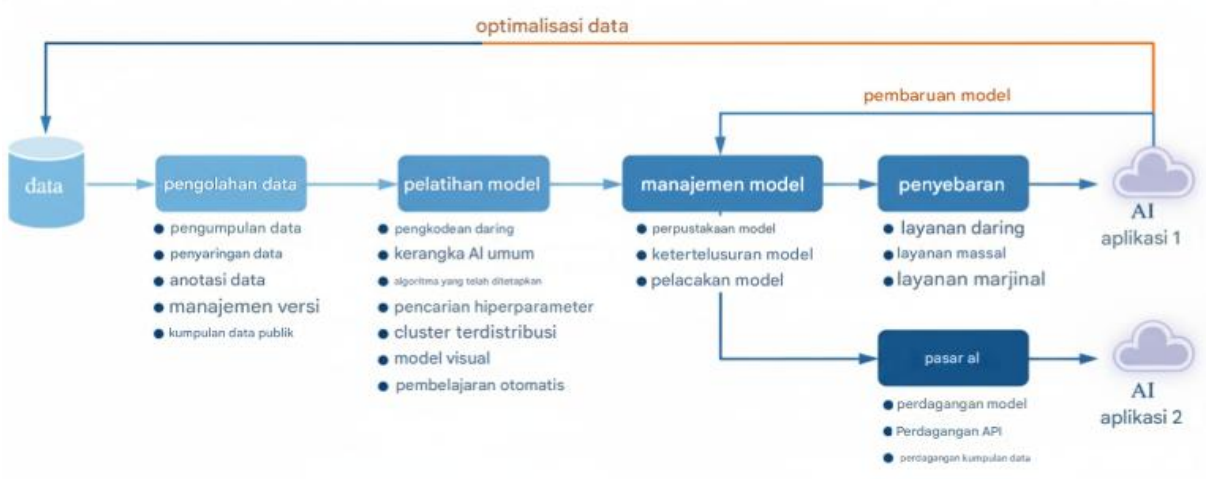
Gambar 8.18 Ikhtisar fungsi ModelArts

Fitur ModelArts adalah sebagai berikut.

1. Tata kelola data: ModelArts mendukung pemrosesan data seperti pemfilteran dan anotasi data, menyediakan manajemen versi set data, terutama set data besar untuk pembelajaran mendalam, sehingga hasil pelatihan dapat direproduksi.
2. Pelatihan yang sangat "cepat" dan "sederhana": Kerangka kerja pembelajaran mendalam Moxing, yang dikembangkan oleh ModelArts, lebih efisien dan mudah digunakan, sehingga meningkatkan kecepatan pelatihan secara signifikan.
3. Penerapan multi-skenario end, edge, dan cloud: ModelArts mendukung penerapan model ke berbagai lingkungan produksi, yang dapat diterapkan sebagai penalaran online cloud dan penalaran batch, atau diterapkan langsung ke end dan edge.
4. Pembelajaran otomatis: ModelArts mendukung berbagai kemampuan pembelajaran otomatis. Melalui model pelatihan "pembelajaran otomatis", pengguna dapat menyelesaikan pemodelan otomatis dan penerapan sekali klik tanpa menulis kode.
5. Alur kerja visual: ModelArts menggunakan GES untuk mengelola metadata proses pengembangan dan secara otomatis memvisualisasikan hubungan antara alur kerja dan evolusi versi, sehingga mewujudkan ketertelusuran model.
6. Pasar AI: ModelArts telah menetapkan algoritme dan set data umum, yang mendukung berbagi model di dalam perusahaan atau publik.

Struktur Produk dan Aplikasi ModelArts

Sebagai platform pengembangan terpadu, ModelArts mendukung seluruh proses pengembangan pengembang, mulai dari data hingga aplikasi AI, termasuk pemrosesan data, pelatihan model, manajemen model, penerapan model, dan operasi lainnya. ModelArts juga menyediakan fungsi pasar AI, dan dapat berbagi model dengan pengembang lain di pasar. Struktur produk ModelArts ditunjukkan pada Gambar 8.19.



Gambar 8.19 Struktur produk ModelArts

ModelArts mendukung seluruh proses pengembangan, mulai dari persiapan data hingga penerapan model AI, dan berbagai skenario aplikasi AI, sebagaimana dijelaskan di bawah ini.

1. Pengenalan gambar: Dapat mengidentifikasi informasi klasifikasi objek dalam gambar secara akurat, seperti identifikasi hewan, pengenalan logo merek, identifikasi kendaraan, dll.
2. Analisis video: Dapat menganalisis informasi penting dalam video secara akurat, seperti pengenalan wajah dan pengenalan fitur kendaraan.
3. Pengenalan ucapan: Memungkinkan mesin memahami sinyal ucapan, membantu dalam pemrosesan informasi ucapan, yang dapat diterapkan pada QA layanan pelanggan cerdas, asisten cerdas, dll.
4. Rekomendasi produk: Dapat memberikan rekomendasi bisnis yang dipersonalisasi untuk pelanggan sesuai dengan karakteristik dan perilaku mereka.
5. Deteksi anomali: Dalam pengoperasian peralatan jaringan, dapat menggunakan sistem deteksi jaringan otomatis untuk melakukan analisis waktu nyata (real-time) sesuai dengan situasi lalu lintas sehingga dapat memprediksi lalu lintas mencurigakan atau peralatan yang mungkin mengalami kegagalan.
6. Ke depannya, akan terus memperkuat keunggulannya dalam peningkatan data, kecepatan pelatihan model, dan pembelajaran supervisi yang lemah, yang selanjutnya akan meningkatkan efisiensi pengembangan model AI.

Keunggulan Produk ModelArts

Keunggulan produk ModelArts tercermin dalam empat aspek berikut.

1. One-stop: Siap pakai. ModelArts mencakup seluruh proses pengembangan AI, termasuk fungsi pemrosesan data, pengembangan model, pelatihan, manajemen, dan penerapan. Satu atau lebih fungsi ini dapat digunakan secara fleksibel.
2. Mudah digunakan: ModelArts menawarkan beragam model prasetel, dan model sumber terbuka dapat digunakan kapan pun diinginkan; superparameter model dioptimalkan secara otomatis, yang sederhana dan cepat; pengembangan tanpa kode mudah dioperasikan untuk melatih model Anda sendiri; penerapan model satu klik ke ujung, tepi, dan cloud didukung.
3. Performa tinggi: Kerangka kerja pembelajaran mendalam Moxing yang dikembangkan oleh ModelArts meningkatkan efisiensi pengembangan algoritma dan kecepatan pelatihan; optimalisasi pemanfaatan GPU dalam penalaran model mendalam dapat mempercepat penalaran daring cloud. Hasilnya, model yang berjalan pada chip Ascend dapat dihasilkan untuk mewujudkan inferensi sisi ujung yang efisien. 4. Fleksibilitas: Mendukung berbagai kerangka kerja sumber terbuka arus utama (TensorFlow, Spark)_ MLlib, dll.). Mendukung GPU arus utama dan chip ascend yang dikembangkan sendiri. Mendukung penggunaan sumber daya eksklusif secara eksklusif. Mendukung mirror image yang ditentukan pengguna untuk memenuhi kebutuhan kerangka kerja dan operator yang ditentukan pengguna.

Selain itu, ModelArts memiliki keunggulan sebagai berikut.

1. Tingkat perusahaan: Mendukung prapemrosesan data masif dan manajemen versi. Mendukung penerapan model multi-scene dari end, edge, dan cloud, untuk mewujudkan manajemen visual dari seluruh proses pengembangan AI. ModelArts juga menyediakan platform berbagi AI, membantu perusahaan membangun ekologi AI internal dan eksternal.
2. Intelektualisasi: Mendukung desain model otomatis, yang dapat melatih model secara otomatis sesuai dengan lingkungan penerapan dan persyaratan kecepatan penalaran. ModelArts juga mendukung pemodelan otomatis klasifikasi gambar dan scene deteksi objek, serta rekayasa fitur otomatis dan pemodelan otomatis data terstruktur.
3. Efisiensi persiapan data meningkat 100 kali lipat: Dilengkapi kerangka kerja data AI bawaan, yang meningkatkan efisiensi persiapan data melalui kombinasi pra-anotasi otomatis dan anotasi set kasus sulit.
4. Pengurangan konsumsi waktu pelatihan model yang signifikan: Menyediakan kerangka kerja terdistribusi berkinerja tinggi Moxing yang dikembangkan oleh Huawei, mengadopsi teknologi inti seperti cascade hybrid parallel, kompresi gradien, dan akselerasi konvolusi, untuk mengurangi konsumsi waktu pelatihan model secara signifikan.
5. Model dapat di-deploy hingga ke end, edge, dan cloud hanya dengan satu klik.
6. Deployment model AI: Menyediakan edge reasoning, online reasoning, dan batch reasoning.
7. Mempercepat proses pengembangan AI dengan metode AI—Pembelajaran Otomatis: Menyediakan panduan UI dan pelatihan adaptif.

8. Menciptakan manajemen proses secara menyeluruh: Mewujudkan visualisasi otomatis proses pengembangan, titik henti pelatihan ulang, dan perbandingan hasil pelatihan yang mudah.
9. Berbagi AI, membantu pengembang mewujudkan penggunaan kembali sumber daya AI: Mewujudkan berbagi intra-perusahaan untuk meningkatkan efisiensi.

Pendekatan untuk Mengunjungi ModelArts

Platform layanan cloud Huawei menyediakan platform manajemen layanan berbasis web, yaitu konsol manajemen dan mode manajemen Antarmuka Pemrograman Aplikasi (API) berdasarkan permintaan HTTPS. ModelArts dapat diakses dengan tiga cara berikut.

1. Mode Konsol Manajemen

ModelArts menyediakan konsol manajemen yang sederhana dan mudah digunakan, mencakup fungsi-fungsi seperti pembelajaran otomatis, manajemen data, lingkungan pengembangan, pelatihan model, manajemen model, penerapan daring, pasar AI, yang dapat menyelesaikan pengembangan AI secara menyeluruh di konsol manajemen. Untuk menggunakan konsol manajemen ModelArts, Anda perlu mendaftarkan akun cloud Huawei terlebih dahulu. Setelah mendaftarkan akun cloud Huawei, Anda dapat mengeklik tautan "EI Enterprise Intelligence! AI Services! EI Basic Platform! AI Development Platform ModelArts" di halaman beranda Huawei Cloud, dan mengeklik tombol "masuk ke konsol" di halaman yang muncul untuk langsung masuk ke konsol manajemen.

2. Mode SDK

Jika Anda perlu mengintegrasikan ModelArts ke dalam sistem pihak ketiga untuk pengembangan sekunder, Anda dapat memilih untuk memanggil ModelArts SDK. ModelArts SDK adalah enkapsulasi Python dari REST API yang disediakan oleh layanan ModelArts, yang menyederhanakan pekerjaan pengembangan pengguna. Untuk operasi spesifik pemanggilan ModelArts SDK dan deskripsi detail SDK, silakan merujuk ke dokumen bantuan produk "Referensi SDK" di situs web resmi ModelArts. Selain itu, saat menulis kode di Notebook konsol manajemen, Anda dapat langsung memanggil ModelArts SDK.

3. Mode API

ModelArts dapat diintegrasikan ke dalam sistem pihak ketiga untuk pengembangan sekunder. ModelArts juga dapat diakses dengan memanggil API ModelArts. Untuk detail pengoperasian dan deskripsi API, silakan lihat dokumen bantuan produk "Ikhtisar API" di situs web resmi ModelArts.

Cara Menggunakan ModelArts

ModelArts adalah platform pengembangan terpadu untuk pengembang AI. Melalui seluruh proses manajemen pengembangan AI, ModelArts membantu pengembang membuat model AI secara cerdas dan efisien serta menerapkannya ke end, edge, dan cloud hanya dengan satu klik. ModelArts tidak hanya mendukung fungsi pembelajaran otomatis, tetapi juga telah menyiapkan berbagai model terlatih, mengintegrasikan Jupyter Notebook untuk menyediakan lingkungan pengembangan kode daring. Pola penggunaan ModelArts yang berbeda dapat dipilih sesuai dengan kelompok pengguna yang berbeda.

Bagi pengembang bisnis tanpa pengalaman pengembangan AI, ModelArts menyediakan fungsi pembelajaran otomatis, yang dapat membangun model AI tanpa fondasi. Pengembang tidak perlu berfokus pada detail pengembangan seperti pengembangan model dan penyesuaian parameter. Hanya tiga langkah (anotasi data, pelatihan otomatis, penerapan daring) yang diperlukan untuk menyelesaikan proyek pengembangan AI. Dokumen bantuan produk "praktik terbaik" di situs web resmi ModelArts menyediakan contoh "Temukan Yunbao" (Yunbao adalah maskot Huawei Cloud), yang digunakan untuk membantu pengembang bisnis dengan cepat memahami proses penggunaan pembelajaran otomatis ModelArts. Contoh ini merupakan proyek skenario "deteksi objek". Melalui set data gambar Yunbao yang telah ditetapkan, model deteksi dilatih dan dibuat secara otomatis, dan model yang dihasilkan disebarkan sebagai layanan daring. Setelah penyebaran, pengguna dapat mengidentifikasi apakah gambar input mengandung Yunbao melalui layanan daring.

Bagi pemula AI dengan dasar-dasar AI tertentu, ModelArts menyediakan algoritma yang telah ditetapkan berdasarkan mesin utama di industri. Pembelajar tidak perlu memperhatikan proses pengembangan model. Mereka langsung menggunakan algoritma yang telah ditetapkan untuk melatih data yang ada dan dengan cepat menyebarkannya sebagai layanan. Algoritma yang telah ditetapkan yang disediakan oleh ModelArts di pasar AI dapat digunakan untuk deteksi objek, klasifikasi gambar, dan klasifikasi teks. Dokumen bantuan produk "praktik terbaik" di situs web resmi ModelArts memberikan contoh penerapan klasifikasi citra bunga, yang membantu pemula AI dengan cepat terbiasa dengan proses penggunaan algoritma prasetel ModelArts untuk membangun model. Contoh ini menggunakan set data citra bunga prasetel untuk memberi anotasi pada data citra yang ada, lalu menggunakan prasetel RESNET_v1_50. Akhirnya, model tersebut diterapkan sebagai layanan daring. Setelah penerapan, pengguna dapat mengidentifikasi spesies bunga dari citra input melalui layanan daring tersebut.

Bagi insinyur AI yang terbiasa dengan penulisan dan penelusuran kesalahan kode, ModelArts menyediakan kemampuan manajemen terpadu, yang memungkinkan insinyur AI menyelesaikan seluruh proses AI dalam satu tahap, mulai dari persiapan data, pengembangan model, pelatihan model, hingga penerapan model. ModelArts kompatibel dengan mesin-mesin utama di industri dan kebiasaan pengguna. Selain itu, ModelArts juga menyediakan kerangka kerja pembelajaran mendalam MoXing yang dikembangkan sendiri untuk meningkatkan efisiensi pengembangan dan kecepatan pelatihan algoritma.

Dokumen bantuan produk "Praktik Terbaik" di situs web resmi ModelArts memberikan contoh penggunaan MXNet dan NoteBook untuk menerapkan pengenalan gambar digital tulisan tangan, yang membantu para insinyur AI dengan cepat memahami seluruh proses pengembangan AI ModelArts. MNIST adalah kumpulan data pengenalan angka tulisan tangan, yang sering digunakan sebagai contoh pembelajaran mendalam. Contoh ini akan menggunakan antarmuka asli MXNet atau skrip pelatihan model NoteBook (disediakan secara default oleh ModelArts) untuk kumpulan data MNIST, dan menerapkan model sebagai layanan daring. Setelah penerapan, pengguna dapat mengidentifikasi angka yang dimasukkan dalam gambar melalui layanan daring.

8.3 SOLUSI HUAWEI CLOUD EI

Bab ini terutama memperkenalkan kasus aplikasi dan solusi Huawei Cloud EI.

Layanan OCR yang Memungkinkan Penggantian Biaya Otomatis Seluruh Proses

Layanan OCR Huawei Cloud dapat diterapkan pada skenario penggantian biaya keuangan. Bahasa Indonesia: Ini secara otomatis mengekstrak informasi kunci dari tagihan, membantu karyawan secara otomatis mengisi formulir penggantian. Sementara itu, dikombinasikan dengan Robotic Process Automation (RPA) untuk itu dapat sangat meningkatkan efisiensi kerja penggantian keuangan.

Pengenalan OCR Tagihan Huawei Cloud mendukung pengenalan OCR dari berbagai tagihan seperti faktur PPN, faktur taksi, tiket kereta api, lembar rencana perjalanan, tiket belanja. Itu dapat memperbaiki kemiringan dan distorsi gambar, secara efektif menghilangkan dampak segel pada pengenalan karakter sehingga dapat meningkatkan akurasi pengenalan. Dalam penggantian keuangan, sangat umum untuk memiliki beberapa tagihan dalam satu gambar. Umumnya, layanan OCR hanya dapat mengidentifikasi satu jenis tagihan. Misalnya, layanan faktur PPN hanya dapat mengidentifikasi satu faktur PPN.

Namun, layanan OCR Huawei Cloud, layanan klasifikasi dan identifikasi cerdas online, mendukung berbagai format faktur dan segmentasi kartu. Itu dapat mengenali satu gambar dari beberapa tiket, satu gambar dari beberapa kartu, kartu dan tiket campuran, dan mewujudkan total penagihan. Dikombinasikan dengan setiap layanan OCR, solusi ini dapat mengenali berbagai jenis faktur dan kartu, termasuk namun tidak terbatas pada tiket pesawat, tiket kereta api, faktur medis, SIM, kartu bank, kartu identitas, paspor, izin usaha, dll.

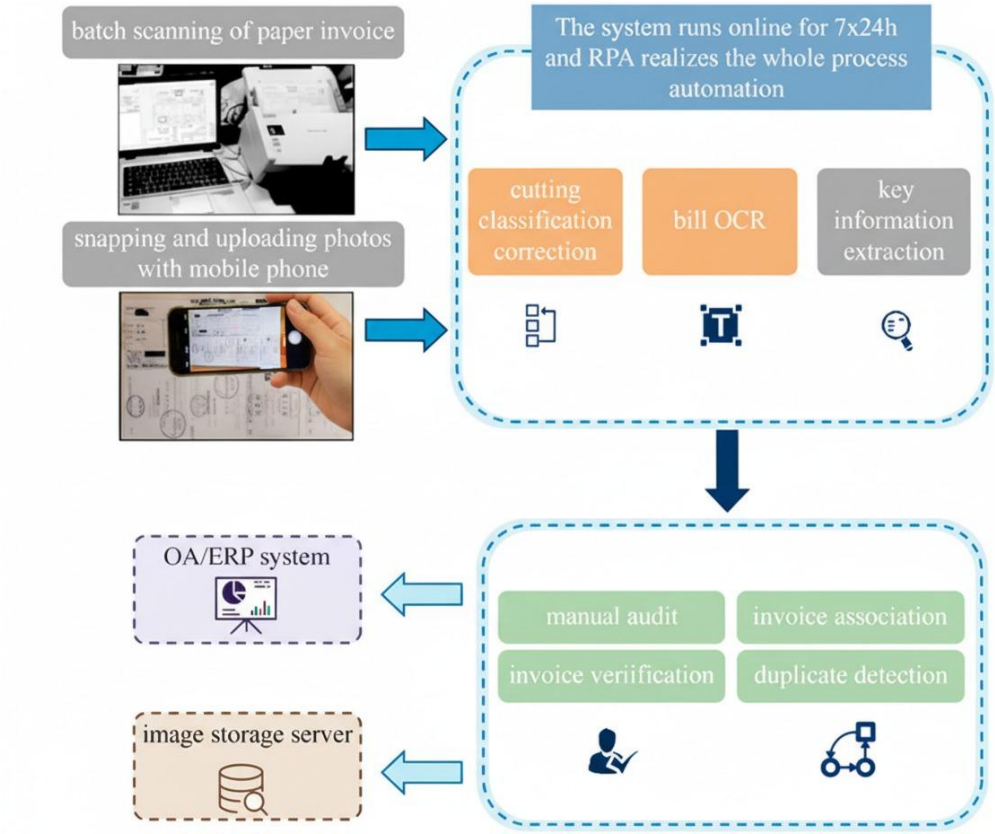
Petugas keuangan perlu memasukkan informasi faktur secara manual ke dalam sistem setelah menerima sejumlah faktur keuangan. Bahkan jika Anda menggunakan layanan OCR Huawei Cloud, Anda perlu mengambil foto setiap faktur keuangan dan mengunggahnya ke komputer atau server. Huawei Cloud dapat menyediakan solusi pengenalan OCR pemindaian batch, yang hanya membutuhkan pemindai dan PC untuk memindai faktur secara batch melalui pemindai untuk menghasilkan gambar berwarna. Solusi ini dapat secara otomatis memanggil layanan OCR Huawei Cloud secara batch, untuk menyelesaikan proses ekstraksi informasi faktur dengan cepat, dan membandingkan hasil pengenalan secara visual. Solusi ini juga dapat mengekspor hasil identifikasi ke Excel atau sistem keuangan secara batch, sehingga sangat menyederhanakan proses entri data.

Solusi ini memiliki karakteristik sebagai berikut:

1. Berbagai metode akses: pemindai koneksi otomatis, akuisisi gambar batch; kamera tinggi, akuisisi gambar foto ponsel.
2. Mode penerapan fleksibel: mendukung cloud publik, HCS, all-in-one, dan mode penerapan lainnya, serta menyatukan antarmuka API standar.
3. Berlaku untuk semua jenis faktur: PPN umum/khusus/elektronik/ETC/voucher, tarif taksi/tiket kereta api/daftar rencana perjalanan/faktur kuota/tol, dll.
4. Mendukung satu gambar untuk beberapa faktur: klasifikasi dan pengenalan otomatis beberapa faktur.

5. Perbandingan visual: pengembalian informasi lokasi, konversi format Excel, mudah untuk statistik dan analisis.

Solusi penggantian faktur ditunjukkan pada Gambar 8.20. Keunggulan solusi ini adalah sebagai berikut: meningkatkan efisiensi dan mengurangi biaya, mengoptimalkan operasional, menyederhanakan proses, dan meningkatkan kepatuhan.

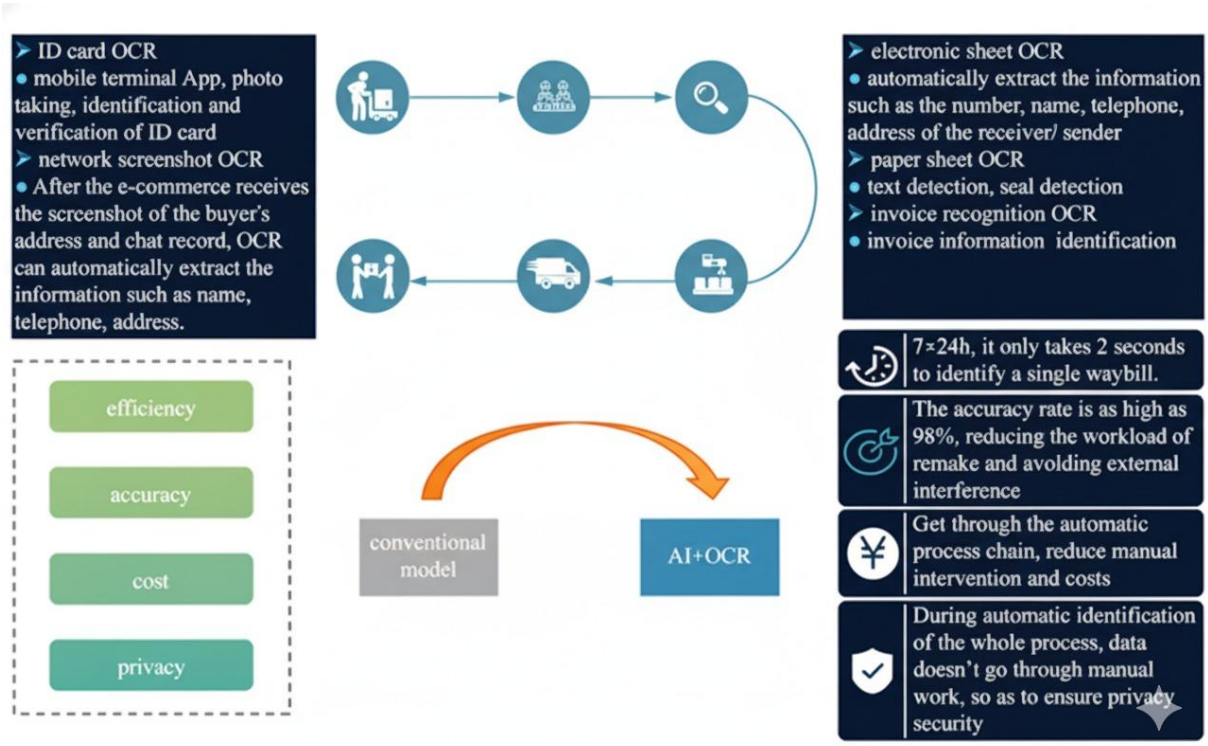


Gambar 8.20 Solusi penggantian faktur

OCR Mendukung Logistik Cerdas

Kurir dapat mengambil gambar kartu identitas melalui terminal seluler (seperti Aplikasi seluler) saat mengambil barang. Dengan layanan identifikasi Huawei Cloud ID, informasi identitas dikenali secara otomatis. Saat mengisi informasi ekspres, Anda dapat menyelesaikan entri otomatis informasi ekspres dengan mengunggah tangkapan layar alamat, tangkapan layar rekaman obrolan, dan gambar lainnya. OCR dapat mengekstrak informasi seperti nama, telepon, dan alamat secara otomatis.

Dalam proses pengiriman ekspres, OCR juga dapat mengekstrak informasi waybill untuk menyelesaikan penyortiran otomatis pengiriman ekspres, menilai kelengkapan informasi pada lembar muka ekspres. Layanan OCR Huawei Cloud mendukung pengenalan OCR untuk gambar kompleks dari berbagai sudut, pencahayaan tidak merata, dan gambar tidak lengkap. OCR memiliki tingkat pengenalan yang tinggi dan stabilitas yang baik, yang dapat mengurangi biaya tenaga kerja secara signifikan dan meningkatkan pengalaman pengguna. Solusi logistik cerdas ditunjukkan pada Gambar 8.21.



Gambar 8.21 Solusi logistik cerdas

Bot Percakapan

Biasanya, robot dengan satu fungsi tidak dapat menyelesaikan semua masalah dalam skenario bisnis pelanggan. Dengan mengintegrasikan beberapa robot dengan fungsi yang berbeda, solusi gabungan bot percakapan tercipta, yang disajikan sebagai satu antarmuka layanan. Pelanggan hanya dapat memanggil satu antarmuka untuk menyelesaikan berbagai masalah bisnis. Karakteristik fungsional setiap robot adalah sebagai berikut.

1. Skenario QABot Cerdas yang Berlaku

(a) QABot Cerdas dapat menyelesaikan berbagai jenis masalah umum seperti konsultasi, pencarian bantuan di bidang TI, e-commerce, keuangan, dan pemerintahan. Dalam skenario ini, pengguna sering berkonsultasi atau mencari bantuan.

(b) QABot Cerdas memiliki cadangan pengetahuan, dengan basis pengetahuan QA, FAQ atau dokumen serupa, serta perintah kerja dan data QA layanan pelanggan.

2. Skenario TaskBot yang Berlaku

(a) TaskBot memiliki tugas percakapan yang jelas. TaskBot dapat secara fleksibel mengonfigurasi proses percakapan (interaksi multi-putaran) sesuai dengan skenario bisnis aktual. Setelah memuat templat skrip, TaskBot melakukan beberapa putaran dialog dengan pelanggan berdasarkan ucapan atau teks dalam adegan yang sesuai, memahami dan merekam keinginan pelanggan secara bersamaan.

(b) Bot Keluar: TaskBot jenis ini dapat menyelesaikan berbagai tugas seperti kunjungan balasan untuk kepuasan bisnis, verifikasi informasi pengguna, penunjukan rekrutmen,

pemberitahuan pengiriman ekspres, promosi penjualan, dan penyaringan pelanggan berkualitas tinggi.

(c) Layanan Pelanggan: TaskBot jenis ini dapat menyelesaikan berbagai tugas seperti reservasi hotel, reservasi tiket pesawat, dan aktivasi kartu kredit.

(d) Perangkat Keras Cerdas: TaskBot jenis ini dapat digunakan di berbagai bidang seperti asisten suara dan rumah pintar.

3. Skenario yang Berlaku untuk Knowledge Graph QABot

(a) Sistem pengetahuan yang kompleks.

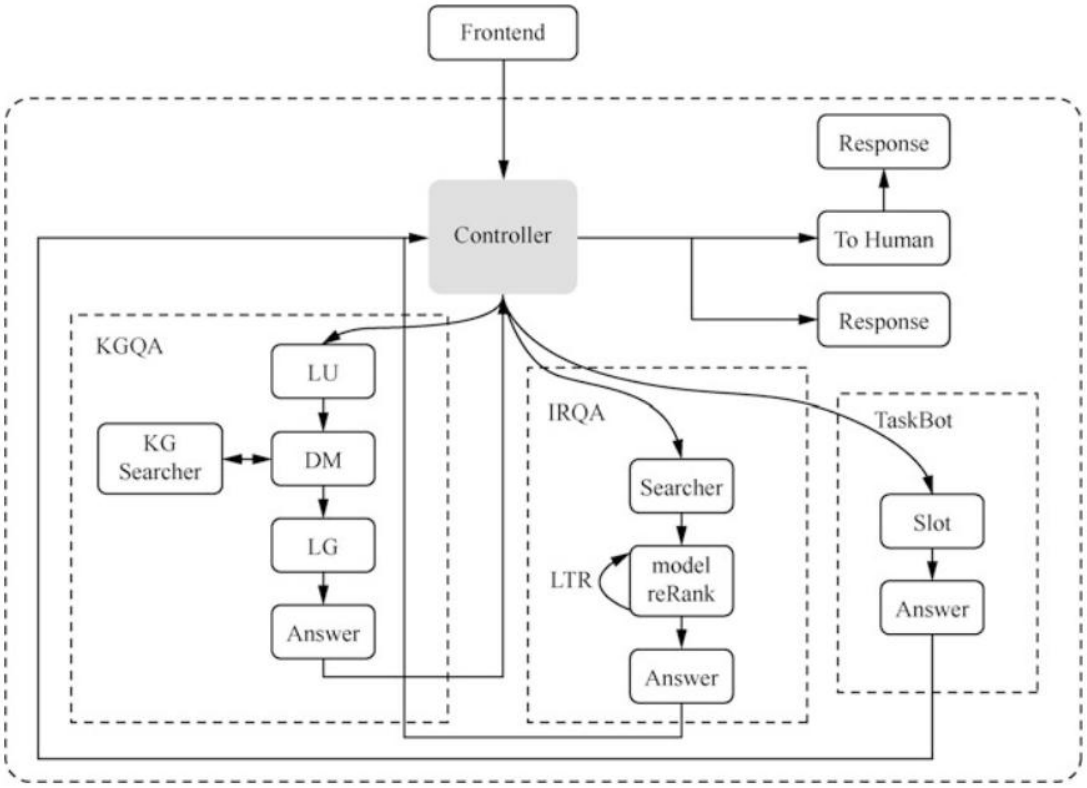
(b) Jawaban yang membutuhkan inferensi logis.

(c) Beberapa putaran interaksi.

(d) Masalah faktual yang melibatkan nilai atribut entitas atau hubungan antar entitas yang tidak dapat diselesaikan dengan enumerasi.

Fitur-fitur robot dialog adalah sebagai berikut.

(a) Integrasi cerdas multi-robot, lebih komprehensif: sejumlah robot memiliki keunggulan, pembelajaran mandiri, dan pengoptimalan mandiri, sehingga solusi terbaik dapat direkomendasikan kepada pelanggan.

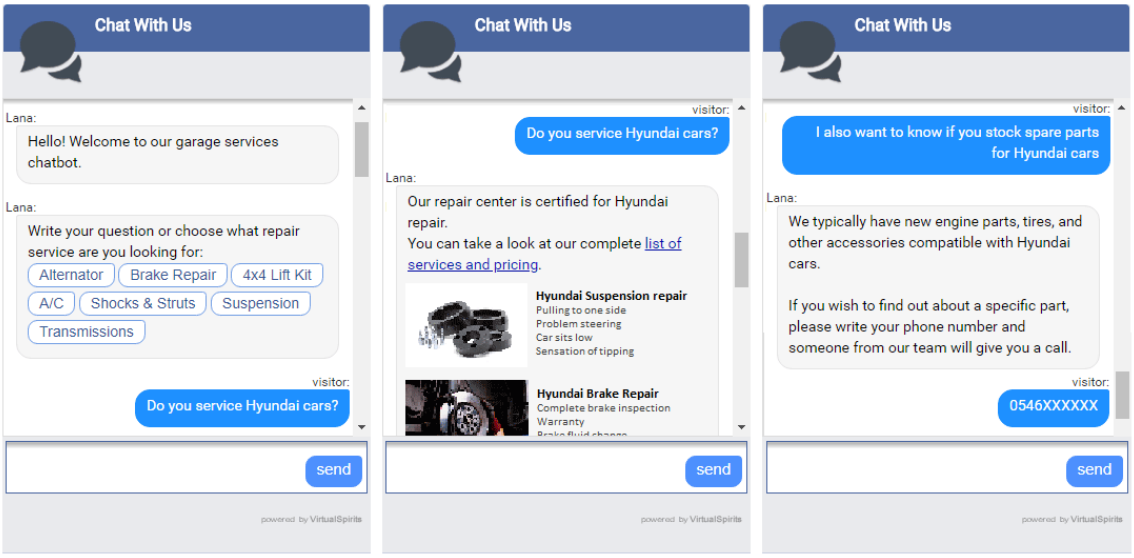


Gambar 8.22 Arsitektur bot percakapan

(b) Beberapa putaran panduan cerdas, pemahaman yang lebih baik: beberapa putaran dialog, interaksi alami, dapat mengidentifikasi maksud pengguna secara akurat, sehingga semantik potensial pengguna dapat dipahami.

(c) Grafik pengetahuan, lebih cerdas: model bahasa domain umum + grafik pengetahuan domain; pembaruan dinamis konten peta; robot berbasis grafik yang lebih cerdas. Arsitektur bot percakapan ditunjukkan pada Gambar 8.22.

QABot cerdas berbasis grafik pengetahuan dapat melakukan tanya jawab pengetahuan yang akurat. Misalnya, bot percakapan kendaraan dapat diterapkan untuk menanyakan harga dan konfigurasi model kendaraan tertentu. Bot ini dapat merekomendasikan kendaraan berdasarkan harga dan jenis level. Bot ini juga dapat melakukan perbandingan kendaraan, dan menawarkan informasi terkait seperti teks, tabel, gambar. Bot percakapan kendaraan ditunjukkan pada Gambar 8.23.



Gambar 8.23 Bot percakapan kendaraan

Kasus Tanya Jawab Cerdas Perusahaan di Distrik Tertentu

Sistem tanya jawab cerdas perusahaan di sebuah distrik di Shenzhen menyediakan respons otomatis bagi robot bisnis yang relevan. Pertanyaan yang tidak langsung dijawab oleh robot akan direkam secara otomatis, dan kemudian jawaban manual lanjutan akan diteruskan kepada penanya. Sistem ini menyediakan solusi loop tertutup yang lengkap untuk masalah yang belum terpecahkan, yang dapat mewujudkan proses optimasi berkelanjutan dari pencatatan masalah yang belum terpecahkan, pembentukan pengetahuan loop tertutup buatan, anotasi model, dan optimasi, sehingga menjadikan robot lebih cerdas. Sistem tanya jawab cerdas perusahaan ditunjukkan pada Gambar 8.24.

Gambar 8.24 Sistem QA cerdas perusahaan

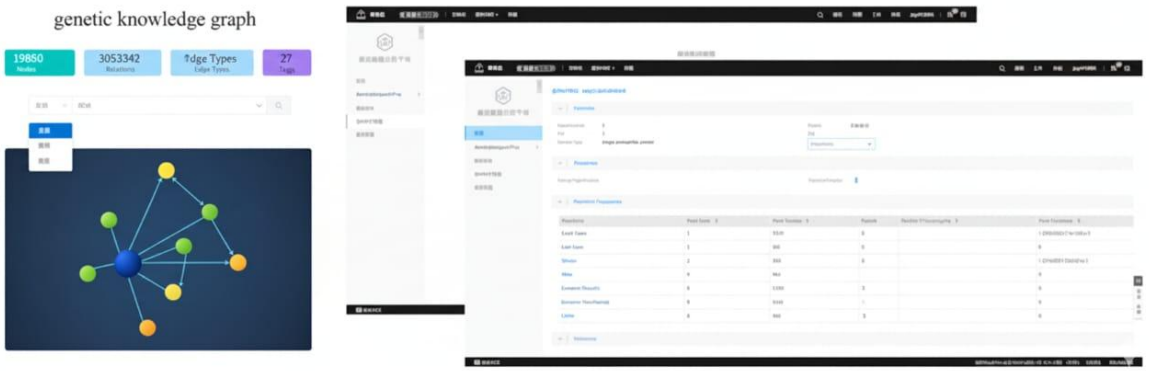
Bisnis yang terkait dengan sistem QA cerdas perusahaan terutama mencakup tiga kategori berikut.

1. Konsultasi kebijakan (perubahan kebijakan yang sering terjadi).
2. Masalah perusahaan di ruang kantor (lebih dari 500 item).
3. Banding (berbagai jenis).

Kasus dalam Grafik Pengetahuan Genetik

Grafik pengetahuan genetik mencakup berbagai jenis entitas, seperti gen, mutasi, penyakit, obat, dll., serta hubungan kompleks antara gen dan mutasi, variasi dan penyakit, serta penyakit dan obat. Berdasarkan grafik ini, fungsi-fungsi berikut dapat direalisasikan.

1. Kueri entitas: Berdasarkan grafik pengetahuan genetik, informasi suatu entitas (gen, mutasi, penyakit, obat) dapat dicari dengan cepat.
2. Diagnosis tambahan: Berdasarkan hasil pengujian genetik, kemungkinan variasi atau penyakit dapat disimpulkan melalui grafik untuk memberikan saran diagnosis dan pengobatan serta merekomendasikan obat.
3. Laporan pengujian gen: Berdasarkan data terstruktur atau semi-terstruktur entitas gen dan pengetahuan asosiasinya dengan variasi dan penyakit, laporan pengujian gen yang dapat dibaca akan dibuat secara otomatis.



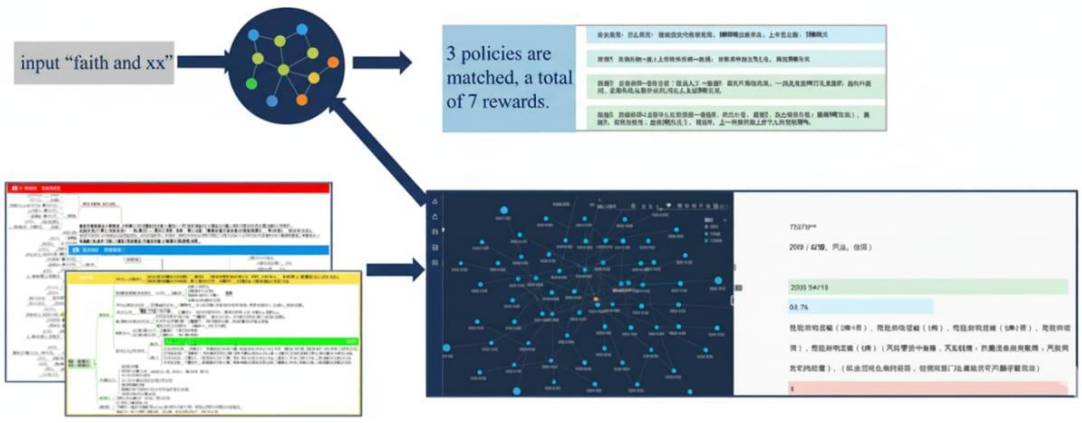
Gambar 8.25 Grafik pengetahuan genetik

Grafik pengetahuan genetik ditunjukkan pada Gambar 8.25.

Kueri Kebijakan Berdasarkan Grafik Pengetahuan

Pemerintah negara bagian sering mengeluarkan beberapa kebijakan insentif kepada perusahaan, seperti kebijakan pengurangan pajak dan rabat pajak. Isi kebijakan tersebut sangat profesional sehingga orang awam sulit memahaminya dan membutuhkan interpretasi profesional. Terdapat banyak jenis kebijakan dan kategori penghargaan. Terdapat lebih dari 300 kondisi yang dapat dikenali oleh perusahaan melalui kebijakan, dan terdapat hubungan logis antara kondisi-kondisi dalam kebijakan yang sama, seperti dan, atau, dan tidak. Oleh karena itu, sangat sulit bagi perusahaan untuk segera mendapatkan kebijakan yang dapat mereka nikmati.

Melalui pembuatan peta pengetahuan kebijakan, berbagai macam insentif kebijakan dan kondisi identifikasi dapat diperoleh. Selain itu, kita dapat membangun peta pengetahuan informasi perusahaan. Terakhir, kita hanya perlu memasukkan nama perusahaan, dan secara otomatis mendapatkan nilai dari berbagai informasi (kondisi identifikasi) perusahaan dari peta perusahaan, seperti jenis, jumlah pajak, skala, dan kondisi identifikasi lainnya. Berdasarkan kondisi identifikasi ini, kita dapat memperoleh semua kebijakan dan penghargaan yang dapat dinikmati perusahaan. Kueri kebijakan berdasarkan peta pengetahuan ditunjukkan pada Gambar 8.26.

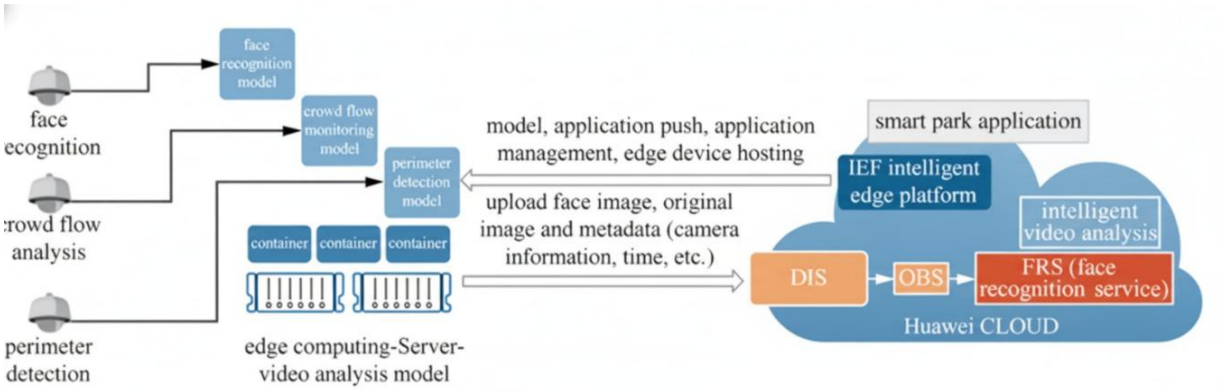


Gambar 8.26 Kueri kebijakan berdasarkan grafik pengetahuan

Kasus di Taman Pintar

Tian'an Cloud Valley terletak di Kota Sains dan Teknologi Banxuegang, pusat kota Shenzhen, seluas 760.000 meter persegi, dengan total luas 2,89 juta meter persegi. Proyek ini berfokus pada teknologi informasi generasi baru seperti komputasi awan dan internet seluler, serta industri-industri terkemuka seperti penelitian dan pengembangan robot dan perangkat cerdas. Bersamaan dengan itu, industri jasa modern dan industri jasa produktif yang relevan dikembangkan di sekitarnya. Untuk memenuhi kebutuhan industri-industri terkemuka, Tian'an Cloud Valley menyediakan ruang terbuka dan bersama serta pembangunan lingkungan cerdas, sehingga menciptakan ekosistem kota industri pintar yang sepenuhnya terhubung dengan perusahaan dan talenta. Proyek ini mengadopsi skema analisis video kolaborasi Edge-cloud. Model analisis video seperti pengenalan wajah, pengenalan kendaraan, dan deteksi intrusi didistribusikan ke server inferensi GPU lokal di taman. Setelah analisis aliran video real-time selesai secara lokal, hasil analisis dapat diunggah ke cloud atau disimpan ke docking sistem aplikasi lokal.

Dengan mengadopsi skema analisis video kolaborasi Edge-cloud, taman ini mewujudkan analisis cerdas terhadap video pengawasan, persepsi penyusup secara real-time, arus orang yang besar, dan kejadian abnormal lainnya, sehingga dapat mengurangi biaya tenaga kerja taman. Pada saat yang sama, kamera IPC yang ada di taman dapat diubah menjadi kamera cerdas melalui kolaborasi edge cloud, yang sangat melindungi aset stok pengguna. Taman pintar ditunjukkan pada Gambar 8.27.



Gambar 8.27 Taman pintar

Sisi ujung adalah kamera IPC definisi tinggi biasa, sementara sisi tepi mengadopsi server GPU perangkat keras. Daya saing dan nilai analisis video edge adalah sebagai berikut.

1. Nilai bisnis: Taman melakukan analisis cerdas terhadap video pengawasan, dengan deteksi real-time terhadap kejadian abnormal seperti penyusupan, arus orang yang besar, sehingga dapat mengurangi biaya tenaga kerja taman.
2. Kolaborasi edge-cloud: Edge menerapkan manajemen siklus hidup, dengan peningkatan yang lancar.
3. Pelatihan model cloud: Model secara otomatis melakukan pelatihan, dengan skalabilitas algoritma yang baik, dan mudah diperbarui.

4. Kompatibilitas yang baik: Kamera IPC yang ada di taman dapat digunakan untuk beralih ke kamera cerdas melalui kolaborasi edge cloud.

Kasus Penghitungan Pejalan Kaki dan Peta Panas

Penghitungan pejalan kaki dan peta panas terutama digunakan untuk mengidentifikasi informasi kerumunan di layar, termasuk informasi jumlah personel dan informasi panas personel regional. Pengaturan waktu yang ditentukan pengguna dan pengaturan interval pengiriman hasil didukung. Hal ini terutama diterapkan pada penghitungan pejalan kaki, penghitungan pengunjung, dan identifikasi panas di area komersial, seperti yang ditunjukkan pada Gambar 8.28.

Peningkatan berikut dapat dicapai dengan menggunakan penghitungan pejalan kaki dan peta panas.

1. Anti-interferensi yang kuat: Mendukung penghitungan pejalan kaki dalam situasi kompleks, seperti wajah atau tubuh yang sebagian tertutup.
2. Skalabilitas tinggi: Mendukung pengiriman statistik penyeberangan pejalan kaki, statistik regional, dan statistik peta panas secara bersamaan.
3. Peningkatan kegunaan: Dapat dihubungkan ke kamera pengawas 1080P biasa.

Pejalan Kaki Regional Peta panas keramaian regional

Gambar 8.28 Penghitungan pejalan kaki dan peta panas

Kasus Pengenalan Kendaraan

Pengenalan kendaraan ditunjukkan pada Gambar 8.29. Dengan pengenalan kendaraan, peningkatan berikut dapat dicapai.

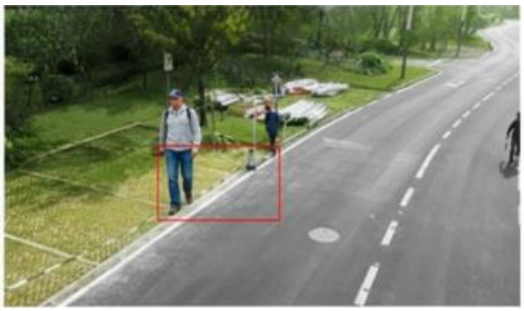
1. Cakupan pemandangan yang komprehensif: Mendukung berbagai skenario seperti jenis kendaraan, warna bodi, dan pengenalan plat nomor dalam berbagai situasi seperti polisi listrik, bayonet, dll.
2. Kemudahan penggunaan yang tinggi: Akses ke kamera pengawas 1080P biasa, dapat mengidentifikasi informasi kendaraan dalam gambar, termasuk informasi plat nomor dan atribut kendaraan. Dapat mengenali jenis kendaraan seperti mobil, kendaraan berukuran sedang, dan juga warna kendaraan, termasuk plat nomor biru dan plat nomor energi baru. Terutama digunakan dalam berbagai skenario seperti manajemen kendaraan parkir, manajemen kendaraan tempat parkir, dan pelacakan kendaraan.

Gambar 8.29 Pengenalan kendaraan

Kasus Identifikasi Intrusi

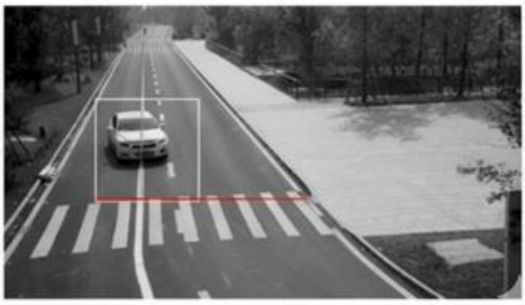
Identifikasi intrusi terutama digunakan untuk mengidentifikasi perilaku intrusi ilegal di layar. Sistem ini mendukung ekstraksi target bergerak di bidang pandang kamera. Ketika target melewati area yang ditentukan, alarm akan berbunyi. Sistem ini juga mendukung pengaturan jumlah minimum orang di area alarm, pengaturan waktu pemicu alarm, dan pengaturan siklus deteksi algoritma. Deteksi intrusi terutama digunakan untuk mengidentifikasi masuk secara ilegal ke area penting, masuk secara ilegal ke area berbahaya, atau pendakian ilegal. Identifikasi intrusi ditunjukkan pada Gambar 8.30.

Identifikasi intrusi manusia melalui jalur



Identifikasi intrusi area manusia

Identifikasi intrusi pendakian manusia



Identifikasi intrusi kendaraan yang melintas

Gambar 8.30 Identifikasi Intrusi

Dengan menggunakan deteksi intrusi, peningkatan berikut dapat dicapai.

1. Fleksibilitas tinggi: Mendukung ukuran target alarm yang fleksibel dan pengaturan kategori.
2. Tingkat alarm palsu rendah: Mendukung alarm intrusi berdasarkan orang/kendaraan, menyaring gangguan dari objek lain.
3. Peningkatan kegunaan: Dapat diakses ke kamera pengawas 1080P biasa.

Platform Komputasi Kognitif CNPC: Identifikasi Reservoir untuk Well Logging

Dengan penyelesaian dan penyempurnaan sistem terintegrasi, CNPC telah mengumpulkan sejumlah besar data terstruktur dan tidak terstruktur. Data terstruktur telah dimanfaatkan dengan baik, tetapi data tidak terstruktur belum sepenuhnya diterapkan. Selain itu, akumulasi pengetahuan yang relevan dan pengalaman para ahli belum sepenuhnya dimanfaatkan, dan kemampuan analisis serta penerapan data yang cerdas belum memadai. Data tidak terstruktur memiliki kapasitas data yang besar, beragam variasi, dan kepadatan nilai yang rendah.

Komputasi kognitif merupakan mode komputasi baru, yang merupakan tahap lanjutan dari pengembangan kecerdasan buatan. Komputasi kognitif mengandung banyak inovasi teknologi di bidang analisis informasi, pemrosesan bahasa alami, dan pembelajaran mesin, yang dapat membantu para pengambil keputusan untuk mendapatkan informasi berharga dari sejumlah besar data tidak terstruktur.

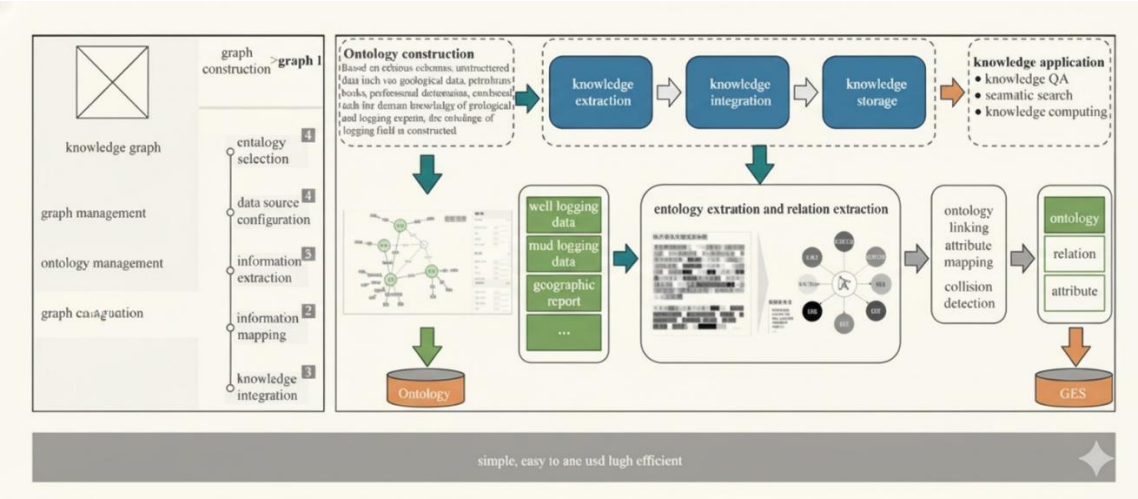
Dengan menggunakan peta pengetahuan cloud Huawei dan teknologi NLP, CNPC telah membangun peta pengetahuan industri minyak dan gas. Berdasarkan peta pengetahuan, ia membangun aplikasi bisnis atas (identifikasi reservoir untuk pencatatan sumur diidentifikasi sebagai salah satu skenario bisnis, dan skenario lainnya mencakup interpretasi cakrawala seismik, prediksi kadar air, diagnosis kondisi kerja, dsb.) Akhirnya, fungsi-fungsi berikut ini terwujud.

1. Agregasi pengetahuan: Peta pengetahuan industri minyak dan gas dapat mempercepat pengetahuan profesional di industri minyak dan gas.
2. Pengurangan biaya dan peningkatan efisiensi: Berdasarkan peta pengetahuan industri minyak dan gas, aplikasi bisnis tingkat atas dapat menyederhanakan proses bisnis dan mempersingkat waktu kerja.
3. Pertumbuhan cadangan dan peningkatan produksi: Berdasarkan peta pengetahuan industri minyak dan gas, aplikasi bisnis tingkat atas dapat memajukan cadangan terbukti dan memastikan keamanan energi.

Solusi identifikasi reservoir untuk pencatatan sumur memiliki fitur-fitur sebagai berikut.

1. Dapat secara fleksibel memodifikasi dan mengintervensi secara manual tautan-tautan kunci seperti ontologi, sumber data, ekstraksi informasi, pemetaan pengetahuan, dan fusi pengetahuan.
2. Penggunaan kembali pengetahuan yang sederhana: Dapat dengan cepat membuat tugas-tugas pipeline baru dan membangun atlas berdasarkan ontologi dan sumber data yang ada.
3. Modifikasi yang fleksibel dan efek sekali klik: Dapat menguji secara berkala dan cepat untuk meningkatkan efisiensi. Terakhir, waktu yang dibutuhkan dapat dipersingkat

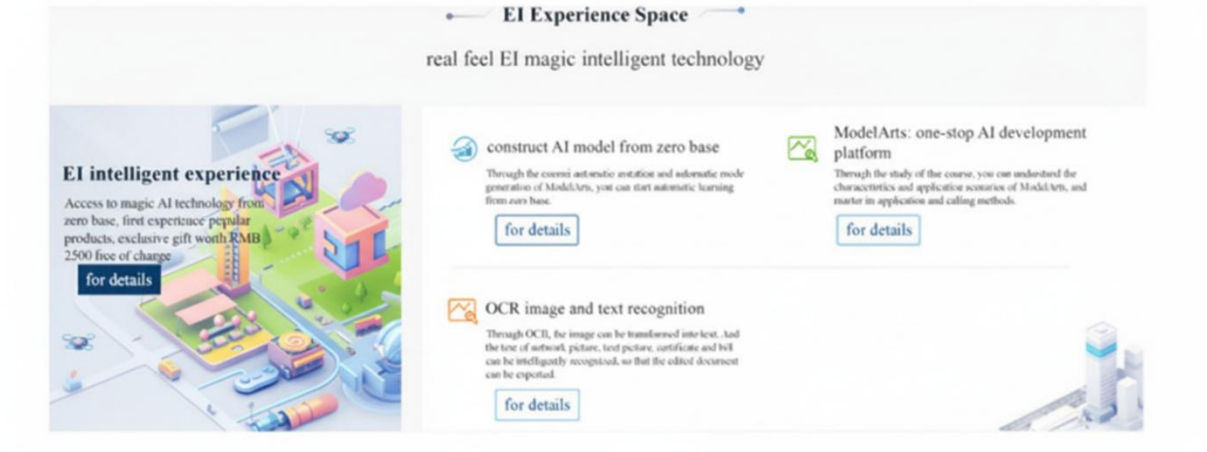
hingga 70% dan tingkat koinsidensi dapat ditingkatkan hingga 5%. Identifikasi reservoir untuk pencatatan sumur ditunjukkan pada Gambar 8.31.



Gambar 8.31 Platform komputasi kognitif CNPC—identifikasi reservoir untuk pencatatan sumur

8.4 KESIMPULAN

Bab ini pertama-tama memperkenalkan ekologi EI Huawei Cloud, dan menjelaskan layanan terkaitnya. Kemudian, bab ini berfokus pada platform dasar EI Huawei ModelArts, dan pengguna dapat mempelajari lebih lanjut tentang layanannya dari eksperimen yang tercantum. Terakhir, kasus-kasus relevan dalam penerapan praktis kecerdasan perusahaan dibahas. Perlu dicatat bahwa Huawei berkomitmen untuk menurunkan ambang batas penerapan AI. Untuk membantu para penggemar AI lebih memahami platform aplikasi EI Huawei Cloud, situs web resmi Huawei Cloud telah menyiapkan ruang pengalaman EI dan kamp pelatihan kursus EI, seperti yang ditunjukkan pada Gambar 8.32 dan 8.33.



Gambar 8.32 Ruang pengalaman EI

Kamp Pelatihan Kursus EI

Belajar dan berlatih secara bersamaan, mainkan huawei cloud EI dengan Mudah

Pembelajaran Mesin *(Machine Learning)*			
	**Pembelajaran Mesin Pengantar 7 Hari.** Kuliah tentang prinsip algoritma dan penggunaan alat membuat pembelajaran mesin menjadi lebih mudah.		**Pembelajaran Mesin Tingkat Lanjut 7 Hari.** Kasus praktis dan praktik model optimasi pembelajaran mesin mendalam.
Data Besar *(Big Data)*			
	**Paket AI Preferensial, Produk AI Populer Rp.150.000 per/Tahun.** Diskon besar tahunan, pengembang menikmati 1 juta panggilan API.		**Kamp Pelatihan Transisi 21 Hari ke Big Data.** Kursus MRS 21 hari membantu Anda memahami peluang industri.
AI			
	**Transisi 21 Hari ke Kamp Latihan AI.** Pembelajaran teknologi AI membantu perusahaan mewujudkan otomatisasi produksi.		**Pengembangan Model AI Jaringan Traktis 14 Hari.** Satu atap menguasai proses pengembangan model AI Jaringan.

Gambar 8.33 Kamp pelatihan kursus EI

Latihan Soal

1. Huawei Cloud EI adalah agen yang memungkinkan kecerdasan perusahaan. Berbasis AI dan teknologi big data, Huawei Cloud EI menyediakan platform yang terbuka, tepercaya, dan cerdas melalui layanan cloud (cloud publik, cloud khusus, dll.). Layanan apa saja yang saat ini tersedia dalam rangkaian layanan Huawei Cloud EI?
2. Dalam rangkaian layanan Huawei Cloud EI, solusi untuk skenario skala besar disebut agen EI. Apa saja itu?
3. Dalam rangkaian layanan Huawei Cloud EI, apa saja platform dasar EI?
4. ModelArts termasuk dalam platform dasar EI dalam rangkaian layanan Huawei Cloud EI. ModelArts merupakan platform pengembangan terpadu untuk pengembang AI. Apa saja fungsinya?
5. Sebagai platform pengembangan AI terpadu, apa saja keunggulan produk ModelArts?

DAFTAR PUSTAKA

Agarwal, P., Swami, S., & Malhotra, S. K. (2024). Artificial intelligence adoption in the post COVID-19 new-normal and role of smart technologies in transforming business: a review. Journal of Science and Technology Policy Management, 15(3), 506-529.

Agrawal, P., Nikhade, P., & Nikhade, P. P. (2022). Artificial intelligence in dentistry: past, present, and future. Cureus, 14(7).

Ahmad, S. F., Rahmat, M. K., Mubarik, M. S., Alam, M. M., & Hyder, S. I. (2021). Artificial intelligence and its role in education. Sustainability, 13(22), 12902.

Alahi, M. E. E., Sukkuea, A., Tina, F. W., Nag, A., Kurdthongmee, W., Suwannarat, K., & Mukhopadhyay, S. C. (2023). Integration of IoT-enabled technologies and artificial intelligence (AI) for smart city scenario: recent advancements and future trends. Sensors, 23(11), 5206.

Alam, A. (2021, November). Possibilities and apprehensions in the landscape of artificial intelligence in education. In 2021 International conference on computational intelligence and computing applications (ICCICA) (pp. 1-8). IEEE.

Alam, A. (2022). Employing adaptive learning and intelligent tutoring robots for virtual classrooms and smart campuses: reforming education in the age of artificial intelligence. In Advanced computing and intelligent technologies: Proceedings of ICACIT 2022 (pp. 395-406). Singapore: Springer Nature Singapore.

Alon, I., Zhang, W., & Lattemann, C. (2021). The case for regulating Huawei. FIIB Business Review, 10(3), 202-204.

Bag, S., Srivastava, G., Bashir, M. M. A., Kumari, S., Giannakis, M., & Chowdhury, A. H. (2022). Journey of customers in this digital era: Understanding the role of artificial intelligence technologies in user engagement and conversion. Benchmarking: An International Journal, 29(7), 2074-2098.

Barsha, S., & Munshi, S. A. (2024). Implementing artificial intelligence in library services: a review of current prospects and challenges of developing countries. Library Hi Tech News, 41(1), 7-10.

Belk, R. W., Belanche, D., & Flavián, C. (2023). Key concepts in artificial intelligence and technologies 4.0 in services. *Service Business*, *17*(1), 1.

Ben Ayed, R., & Hanana, M. (2021). Artificial intelligence to improve the food and agriculture sector. Journal of Food Quality, 2021(1), 5584754.

Boulesnam, M., Kahela, A., Hocini, O., & El-Shobary, N. M. H. (2025). Integración del gemelo digital y resultados de sostenibilidad: evidencia del proceso de transformación de Huawei (2008–2024). Revista Uniandes Episteme, 12(4), 554-571.

Brem, A., Giones, F., & Werle, M. (2021). The AI digital revolution in innovation: A conceptual framework of artificial intelligence technologies for the management of innovation. IEEE Transactions on Engineering Management, 70(2), 770-776.

Budhwar, P., Malik, A., De Silva, M. T., & Thevisuthan, P. (2022). Artificial intelligence–challenges and opportunities for international HRM: a review and research agenda. The InTernaTIonal Journal of human resource managemenT, 33(6), 1065-1097.

Chang, M. B., & Tajudin, N. B. (2023). An Examination Of Global Mobile Telecommunications And Deployment Estimation: A Study Focused On Huawei Telecommunications. Cuestiones de Fisioterapia, 52(3), 588-596.

Cheah, S. L. Y., Yi, E. T. S., & Zhi, J. N. J. (2025). Huawei: Facing the Green Intelligence Dilemma. In Corporate Entrepreneurship and Sustainability (pp. 33-55). Routledge.

Chen, L., Chen, Z., Zhang, Y., Liu, Y., Osman, A. I., Farghali, M., ... & Yap, P. S. (2023). Artificial intelligence-based solutions for climate change: a review. Environmental Chemistry Letters, 21(5), 2525-2557.

Dmitrienko, A., Lipovenko, M., Gostilovich, A., Kolosov, A., & Ming, L. (2023). Development strategies of high-tech companies in China: Huawei and Tencent Estrategias de desarrollo de empresas de alta tecnología en China: Huawei y Tencent.

Dogan, M. E., Goru Dogan, T., & Bozkurt, A. (2023). The use of artificial intelligence (AI) in online learning and distance education processes: A systematic review of empirical studies. Applied sciences, 13(5), 3056.

Fitria, T. N. (2021, December). Artificial intelligence (AI) in education: Using AI tools for teaching and learning process. In Prosiding seminar nasional & call for paper STIE AAS (pp. 134-147).

Gao, P., Li, J., & Liu, S. (2021). An introduction to key technology in artificial intelligence and big data driven e-learning and e-education. Mobile Networks and Applications, 26(5), 2123-2126.

Guo, Y., Wang, H., Zhang, H., Liu, T., Li, L., Liu, L., ... & Lip, G. Y. (2021). Photoplethysmography-based machine learning approaches for atrial fibrillation prediction: a report from the huawei heart study. JACC: Asia, 1(3), 399-408.

He, L., Da, E., & Cheng, M. (2021). Fast development of artificial intelligence on resource constrained devices. ResearchGate. Preprint.

Helo, P., & Hao, Y. (2022). Artificial intelligence in operations management and supply chain management: An exploratory case study. Production planning & control, 33(16), 1573-1590.

Hirsch-Kreinsen, H. (2024). Artificial intelligence: A “promising technology”. AI & society, 39(4), 1641-1652.

Lin, B. (2024). Beyond authoritarianism and liberal democracy: Understanding China’s artificial intelligence impact in Africa. Information, Communication & Society, 27(6), 1126-1141.

Lowe, M., Qin, R., & Mao, X. (2022). A review on machine learning, artificial intelligence, and smart technology in water treatment and monitoring. Water, 14(9), 1384.

Lundvall, B. Å., & Rikap, C. (2022). China's catching-up in artificial intelligence seen as a co-evolution of corporate and national innovation systems. Research Policy, 51(1), 104395.

Lv, Z. (2023). Generative artificial intelligence in the metaverse era. Cognitive Robotics, 3, 208-217.

Lyu, W., & Liu, J. (2021). Artificial Intelligence and emerging digital technologies in the energy sector. Applied energy, 303, 117615.

Mann, M., Kumar, C., Zeng, W. F., & Strauss, M. T. (2021). Artificial intelligence for proteomics and biomarker discovery. Cell systems, 12(8), 759-770.

Mannuru, N. R., Shahriar, S., Teel, Z. A., Wang, T., Lund, B. D., Tijani, S., ... & Vaidya, P. (2025). Artificial intelligence in developing countries: The impact of generative artificial intelligence (AI) technologies for development. Information development, 41(3), 1036-1054.

Mengjiao, Z. H. U., & Yanyong, D. U. (2025). Research on the Status of Artificial Intelligence Reasoning Technology: A Case Study of China and the United States and Their Key Enterprises. 北京科技大学学报 (社会科学版), 41(3), 51-66.

Mogaji, E., Viglia, G., Srivastava, P., & Dwivedi, Y. K. (2024). Is it the end of the technology acceptance model in the era of generative artificial intelligence?. International Journal of Contemporary Hospitality Management, 36(10), 3324-3339.

Moore, G. J. (2023). Huawei, cyber-sovereignty and liberal norms: China’s challenge to the west/democracies. Journal of Chinese Political Science, 28(1), 151-167.

Mukhamediev, R. I., Popova, Y., Kuchin, Y., Zaitseva, E., Kalimoldayev, A., Symagulov, A., ... & Yelis, M. (2022). Review of artificial intelligence and machine learning technologies: classification, restrictions, opportunities and challenges. Mathematics, 10(15), 2552.

Nahr, J. G., Nozari, H., & Sadeghi, M. E. (2021). Green supply chain based on artificial intelligence of things (AIoT). International Journal of Innovation in Management, Economics and Social Sciences, 1(2), 56-63.

Nawaz, H., Maqsood, M., Ghafoor, A. H., Ali, S., Maqsood, A., & Maqsood, A. (2024). Huawei Pakistan Providing Cloud Solutions for Banking Industry: A Data Driven Study. The Asian Bulletin of Big Data Management, 4(1), 89-107.

Peng, Y., Liu, E., Peng, S., Chen, Q., Li, D., & Lian, D. (2022). Using artificial intelligence technology to fight COVID-19: a review. Artificial intelligence review, 55(6), 4941-4977.

Pinto-Coelho, L. (2023). How artificial intelligence is shaping medical imaging technology: a survey of innovations and applications. Bioengineering, 10(12), 1435.

Pratama, M. P., Sampelolo, R., & Lura, H. (2023). Revolutionizing education: harnessing the power of artificial intelligence for personalized learning. Klasikal: Journal of education, language teaching and science, 5(2), 350-357.

Qin, M., Hu, W., Qi, X., & Chang, T. (2024). Do the benefits outweigh the disadvantages? Exploring the role of artificial intelligence in renewable energy. Energy Economics, 131, 107403.

Ribqi, N., & Mazri, T. (2025, April). 5G Networks: A Comparative Analysis of Nokia and Huawei Equipment. In International Conference on Connected Objects and Artificial Intelligence (pp. 280-285). Cham: Springer Nature Switzerland.

Ryan, M., & Burman, S. (2024). The United States–China 'tech war': Decoupling and the case of Huawei. Global policy, 15(2), 355-367.

Sarkar, C., Das, B., Rawat, V. S., Wahlang, J. B., Nongpiur, A., Tiewsoh, I., ... & Sony, H. T. (2023). Artificial intelligence and machine learning technology driven modern drug discovery and development. International Journal of Molecular Sciences, 24(3), 2026.

Shaik, T., Tao, X., Higgins, N., Li, L., Gururajan, R., Zhou, X., & Acharya, U. R. (2023). Remote patient monitoring using artificial intelligence: Current state, applications, and challenges. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 13(2), e1485.

Sharma, L., & Garg, P. K. (Eds.). (2021). Artificial intelligence: technologies, applications, and challenges.

Tagde, P., Tagde, S., Bhattacharya, T., Tagde, P., Chopra, H., Akter, R., ... & Rahman, M. H. (2021). Blockchain and artificial intelligence technology in e-Health. Environmental Science and Pollution Research, 28(38), 52810-52831.

Tapalova, O., & Zhiyenbayeva, N. (2022). Artificial intelligence in education: AIEd for personalised learning pathways. Electronic Journal of e-Learning, 20(5), 639-653.

von Gerich, H., Moen, H., Block, L. J., Chu, C. H., DeForest, H., Hobensack, M., ... & Peltonen, L. M. (2022). Artificial Intelligence-based technologies in nursing: A scoping literature review of the evidence. International journal of nursing studies, 127, 104153.

Vora, L. K., Gholap, A. D., Jetha, K., Thakur, R. R. S., Solanki, H. K., & Chavda, V. P. (2023). Artificial intelligence in pharmaceutical technology and drug delivery design. Pharmaceutics, 15(7), 1916.

Wong, B. K. M., Vengusamy, S., & Bastrygina, T. (2024). Healthcare digital transformation through the adoption of artificial intelligence. In Artificial Intelligence, big data, blockchain and 5G for the digital transformation of the healthcare industry (pp. 87-110). Academic Press.

Yang, C. H. (2022). How artificial intelligence technology affects productivity and employment: Firm-level evidence from Taiwan. Research Policy, 51(6), 104536.

Zhang, Z., Zhang, M., Gong, J., Hu, X., Xiong, H., Zhou, H., & Cao, Z. (2023). LuoJiaAI: A cloud-based artificial intelligence platform for remote sensing image interpretation. Geo-Spatial Information Science, 26(2), 218-241.

Zhou, Y., Shen, M., Cui, X., Shao, Y., Li, L., & Zhang, Y. (2021). Triboelectric nanogenerator based self-powered sensor for artificial intelligence. Nano Energy, 84, 105887.

TEKNOLOGI AI
(Artificial Intelligence)

Dr. Joseph Teguh Santoso, S.Kom, M.Kom.

BIODATA PENULIS

Dr. Joseph Teguh Santoso, M.Kom memiliki Jabatan Akademik Lektor Kepala dan praktisi industri yang berpengalaman. Saat ini menjabat sebagai Rektor Universitas Sains dan Teknologi Komputer (Universitas STEKOM), salah satu universitas terkemuka di Jawa Tengah, Indonesia. Dengan pengalaman lebih dari 13 tahun di dunia bisnis dan praktisi industri di China, beliau membawa perspektif global dan inovasi yang signifikan ke dalam dunia akademis.Sebagai seorang entrepreneur, penulis adalah pencipta TopLoker.com, sebuah platform inovatif yang merevolusi cara mencari dan menawarkan pekerjaan. TopLoker.com adalah portal lowongan bursa kerja terbesar di Indonesia, khusus untuk pendidikan SMA/SMK sederajat. TopLoker.com telah mendapatkan penghargaan sebagai juara 1 Startup4industry 2022 oleh Kementerian Perindustrian Republik Indonesia. Kontribusi Dr. Joseph dalam menyediakan akses pekerjaan yang luas bagi lulusan SMA/SMK telah membantu banyak individu menemukan peluang kerja yang sesuai dengan keahlian mereka. Selain itu, Dr. Joseph Teguh Santoso, M.Kom adalah pendiri dari dua organisasi yaitu (1) organisasi guru/pendidik PTIC (Perkumpulan Teacherpreneur Indonesia Cerdas) yang bertujuan untuk meningkatkan kualitas pendidikan dan kesejahteraan guru/pendidik dengan wawasan entrepreneurship, serta (2) organisasi industri PERKIVI (Perkumpulan Komunitas Industri dan Vokasi Indonesia) yang berfokus pada pengembangan link and match antara industri dan dunia pendidikan. Sebagai Rektor, Dr. Joseph Teguh Santoso, M.Kom memiliki kepemimpinan yang berorientasi pada hasil, dan berkomitmen untuk mendorong kemajuan Universitas Sains dan Teknologi Komputer (Universitas STEKOM). Saat ini Universitas STEKOM telah mengalami transformasi positif dalam peningkatan kualitas pendidikan, perluasan fasilitas, serta penguatan kemitraan Perguruan Tinggi Nasional dan Internasional. Beliau memprioritaskan pengembangan sumber daya manusia dan penelitian, serta memastikan bahwa universitas berada di garis depan dalam inovasi dan teknologi untuk mencapai tujuan akhir, yaitu lulusan yang mampu bekerja dan sukses setelah lulus. Dr. Joseph Teguh Santoso, M.Kom sering diundang sebagai pembicara di berbagai konferensi nasional maupun internasional dan telah menerima berbagai penghargaan atas dedikasinya dalam bidang pendidikan, industri, dan kewirausahaan.

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7227-55-3 (PDF)
9 786347 227553