



Blockchain

A **blockchain** is a distributed ledger with growing lists of records (*blocks*) that are securely linked together via cryptographic hashes.^{[1][2][3][4]} Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree, where data nodes are represented by leaves). Since each block contains information about the previous block, they effectively form a *chain* (compare linked list data structure), with each additional block linking to the ones before it. Consequently, blockchain transactions are irreversible in that, once they are recorded, the data in any given block cannot be altered retroactively without altering all subsequent blocks.

Blockchains are typically managed by a peer-to-peer (P2P) computer network for use as a public distributed ledger, where nodes collectively adhere to a consensus algorithm protocol to add and validate new transaction blocks. Although blockchain records are not unalterable, since blockchain forks are possible, blockchains may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance.^[5]

A blockchain was created by a person (or group of people) using the name (or pseudonym) Satoshi Nakamoto in 2008 to serve as the public distributed ledger for bitcoin cryptocurrency transactions, based on previous work by Stuart Haber, W. Scott Stornetta, and Dave Bayer.^[6] The implementation of the blockchain within bitcoin made it the first digital currency to solve the double-spending problem without the need for a trusted authority or central server. The bitcoin design has inspired other applications^{[3][2]} and blockchains that are readable by the public and are widely used by cryptocurrencies. The blockchain may be considered a type of payment rail.^[7]

Private blockchains have been proposed for business use. *Computerworld* called the marketing of such privatized blockchains without a proper security model "snake oil";^[8] however, others have argued that permissioned blockchains, if carefully designed, may be more decentralized and therefore more secure in practice than permissionless ones.^{[4][9]}

History

Cryptographer David Chaum first proposed a blockchain-like protocol in his 1982 dissertation "Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups".^[10] Further work on a cryptographically secured chain of blocks was described in 1991 by Stuart Haber and W. Scott Stornetta.^{[4][11]} They wanted to implement a system wherein document timestamps could not be tampered with. In 1992, Haber, Stornetta, and Dave Bayer incorporated Merkle trees into the design, which improved its efficiency by allowing several document certificates to be collected into one block.^{[4][12]} Under their company Surety, their document certificate hashes have been published in *The New York Times* every week since 1995.^[13]

The first decentralized blockchain was conceptualized by a person (or group of people) known as Satoshi Nakamoto in 2008. Nakamoto improved the design in an important way using a Hashcash-like method to timestamp blocks without requiring them to be signed by a trusted party and introducing a difficulty

parameter to stabilize the rate at which blocks are added to the chain.^[4] The design was implemented the following year by Nakamoto as a core component of the cryptocurrency bitcoin, where it serves as the public ledger for all transactions on the network.^[3]

In August 2014, the bitcoin blockchain file size, containing records of all transactions that have occurred on the network, reached 20 GB (gigabytes).^[14] In January 2015, the size had grown to almost 30 GB, and from January 2016 to January 2017, the bitcoin blockchain grew from 50 GB to 100 GB in size. The ledger size had exceeded 200 GB by early 2020.^[15]

The words *block* and *chain* were used separately in Satoshi Nakamoto's original paper, but were eventually popularized as a single word, *blockchain*, by 2016.^[16]

According to Accenture, an application of the diffusion of innovations theory suggests that blockchains attained a 13.5% adoption rate within financial services in 2016, therefore reaching the early adopters' phase.^[17] Industry trade groups joined to create the Global Blockchain Forum in 2016, an initiative of the Chamber of Digital Commerce.

In May 2018, Gartner found that only 1% of CIOs indicated any kind of blockchain adoption within their organisations, and only 8% of CIOs were in the short-term "planning or [looking at] active experimentation with blockchain".^[18] For the year 2019 Gartner reported 5% of CIOs believed blockchain technology was a 'game-changer' for their business.^[19]

Structure and design

A blockchain is a decentralized, distributed, and often public, digital ledger consisting of records called *blocks* that are used to record transactions across many computers so that any involved block cannot be altered retroactively, without the alteration of all subsequent blocks.^{[3][20]} This allows the participants to verify and audit transactions independently and relatively inexpensively.^[21] A blockchain database is managed autonomously using a peer-to-peer network and a distributed timestamping server. They are authenticated by mass collaboration powered by collective self-interests.^[22] Such a design facilitates robust workflow where participants' uncertainty regarding data security is marginal. The use of a blockchain removes the characteristic of infinite reproducibility from a digital asset. It confirms that each unit of value was transferred only once, solving the long-standing problem of double-spending. A blockchain has been described as a *value-exchange protocol*.^[23] A blockchain can maintain title rights because, when properly set up to detail the exchange agreement, it provides a record that compels offer and acceptance.

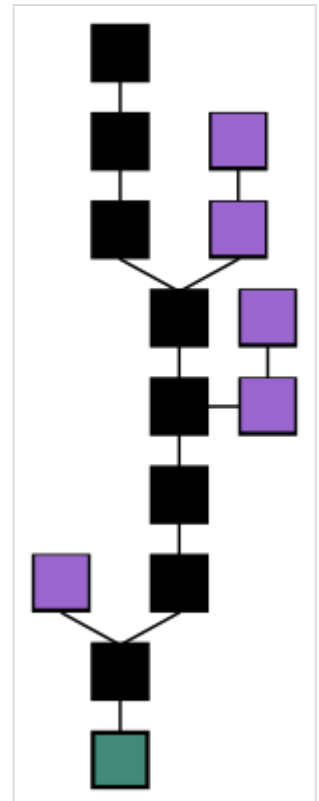
Logically, a blockchain can be seen as consisting of several layers:^[24]

- infrastructure (hardware)
- networking (node discovery, information propagation^[25] and verification)
- consensus (proof of work, proof of stake)
- data (blocks, transactions)
- application (smart contracts/decentralized applications, if applicable)

Blocks

Blocks hold batches of valid transactions that are hashed and encoded into a Merkle tree.^[3] Each block includes the cryptographic hash of the prior block in the blockchain, linking the two. The linked blocks form a chain.^[3] This iterative process confirms the integrity of the previous block, all the way back to the initial block, which is known as the *genesis block* (Block 0).^{[26][27]} To assure the integrity of a block and the data contained in it, the block is usually digitally signed.^[28]

Sometimes separate blocks can be produced concurrently, creating a temporary fork. In addition to a secure hash-based history, any blockchain has a specified algorithm for scoring different versions of the history so that one with a higher score can be selected over others. Blocks not selected for inclusion in the chain are called orphan blocks.^[27] Peers supporting the database have different versions of the history from time to time. They keep only the highest-scoring version of the database known to them. Whenever a peer receives a higher-scoring version (usually the old version with a single new block added) they extend or overwrite their own database and retransmit the improvement to their peers. There is never an absolute guarantee that any particular entry will remain in the best version of history forever. Blockchains are typically built to add the score of new blocks onto old blocks and are given incentives to extend with new blocks rather than overwrite old blocks. Therefore, the probability of an entry becoming superseded decreases exponentially^[29] as more blocks are built on top of it, eventually becoming very low.^{[3][30]:ch. 08[31]} For example, bitcoin uses a proof-of-work system, where the chain with the most cumulative proof-of-work is considered the valid one by the network. There are a number of methods that can be used to demonstrate a sufficient level of computation. Within a blockchain the computation is carried out redundantly rather than in the traditional segregated and parallel manner.^[32]



were recovered after negotiations and ransom payment. Alternatively, to prevent a permanent split, a majority of nodes using the new software may return to the old rules, as was the case of bitcoin split on 12 March 2013.^[34]

A more recent hard-fork example is of Bitcoin in 2017, which resulted in a split creating Bitcoin Cash.^[35] The network split was mainly due to a disagreement in how to increase the transactions per second to accommodate for demand.^[36]

Decentralization

By storing data across its peer-to-peer network, the blockchain eliminates some risks that come with data being held centrally.^[3] The decentralized blockchain may use ad hoc message passing and distributed networking.^[37]

In a so-called "51% attack" a central entity gains control of more than half of a network and can then manipulate that specific blockchain record at will, allowing double-spending.^[38]

Blockchain security methods include the use of public-key cryptography.^{[39]:5} A *public key* (a long, random-looking string of numbers) is an address on the blockchain. Value tokens sent across the network are recorded as belonging to that address. A *private key* is like a password that gives its owner access to their digital assets or the means to otherwise interact with the various capabilities that blockchains now support. Data stored on the blockchain is generally considered incorruptible.^[3]

Every node in a decentralized system has a copy of the blockchain. Data quality is maintained by massive database replication^[40] and computational trust. No centralized "official" copy exists and no user is "trusted" more than any other.^[39] Transactions are broadcast to the network using the software. Messages are delivered on a best-effort basis. Early blockchains rely on energy-intensive mining nodes to validate transactions,^[27] add them to the block they are building, and then broadcast the completed block to other nodes.^{[30]:ch. 08} Blockchains use various time-stamping schemes, such as proof-of-work, to serialize changes.^[41] Later consensus methods include proof of stake.^[27] The growth of a decentralized blockchain is accompanied by the risk of centralization because the computer resources required to process larger amounts of data become more expensive.^[42]

Finality

Finality is the level of confidence that the well-formed block recently appended to the blockchain will not be revoked in the future (is "finalized") and thus can be trusted. Most distributed blockchain protocols, whether proof of work or proof of stake, cannot guarantee the finality of a freshly committed block, and instead rely on "probabilistic finality": as the block goes deeper into a blockchain, it is less likely to be altered or reverted by a newly found consensus.^[43]

Byzantine fault tolerance-based proof-of-stake protocols purport to provide so called "absolute finality": a randomly chosen validator proposes a block, the rest of validators vote on it, and, if a supermajority decision approves it, the block is irreversibly committed into the blockchain.^[43] A modification of this

method, an "economic finality", is used in practical protocols, like the Casper protocol used in Ethereum: validators which sign two different blocks at the same position in the blockchain are subject to "slashing", where their leveraged stake is forfeited.^[43]

Openness

Open blockchains are more user-friendly than some traditional ownership records, which, while open to the public, still require physical access to view. Because all early blockchains were permissionless, controversy has arisen over the blockchain definition. An issue in this ongoing debate is whether a private system with verifiers tasked and authorized (permissioned) by a central authority should be considered a blockchain.^{[44][45][46][47][48]} Proponents of permissioned or private chains argue that the term "blockchain" may be applied to any **data structure** that batches data into time-stamped blocks. These blockchains serve as a distributed version of multiversion concurrency control (MVCC) in databases.^[49] Just as MVCC prevents two transactions from concurrently modifying a single object in a database, blockchains prevent two transactions from spending the same single output in a blockchain.^{[50]:30–31} Opponents say that permissioned systems resemble traditional corporate databases, not supporting decentralized data verification, and that such systems are not hardened against operator tampering and revision.^{[44][46]} Nikolai Hampton of *Computerworld* said that "many in-house blockchain solutions will be nothing more than cumbersome databases," and "without a clear security model, proprietary blockchains should be eyed with suspicion."^{[8][51]}

Permissionless (public) blockchain

An advantage to an open, permissionless, or public, blockchain network is that guarding against bad actors is not required and no access control is needed.^[29] This means that applications can be added to the network without the approval or trust of others, using the blockchain as a transport layer.^[29]

Bitcoin and other cryptocurrencies currently secure their blockchain by requiring new entries to include proof of work. To prolong the blockchain, bitcoin uses Hashcash puzzles. While Hashcash was designed in 1997 by Adam Back, the original idea was first proposed by Cynthia Dwork and Moni Naor and Eli Ponyatovski in their 1992 paper "Pricing via Processing or Combatting Junk Mail".

In 2016, venture capital investment for blockchain-related projects was weakening in the USA but increasing in China.^[52] Bitcoin and many other cryptocurrencies use open (public) blockchains. As of April 2018, bitcoin has the highest market capitalization.

Permissioned (private) blockchain

Permissioned blockchains use an access control layer to govern who has access to the network.^[53] It has been argued that permissioned blockchains can guarantee a certain level of decentralization, if carefully designed, as opposed to permissionless blockchains, which are often centralized in practice.^[9]

Disadvantages of permissioned blockchain

Nikolai Hampton argued in *Computerworld* that "There is also no need for a '51 percent' attack on a private blockchain, as the private blockchain (most likely) already controls 100 percent of all block creation resources. If you could attack or damage the blockchain creation tools on a private corporate server, you could effectively control 100 percent of their network and alter transactions however you

wished."^[8] This has a set of particularly profound adverse implications during a financial crisis or debt crisis like the financial crisis of 2007–08, where politically powerful actors may make decisions that favor some groups at the expense of others,^[54] and "the bitcoin blockchain is protected by the massive group mining effort. It's unlikely that any private blockchain will try to protect records using gigawatts of computing power — it's time-consuming and expensive."^[8] He also said, "Within a private blockchain there is also no 'race'; there's no incentive to use more power or discover blocks faster than competitors. This means that many in-house blockchain solutions will be nothing more than cumbersome databases."^[8]

Blockchain analysis

The analysis of public blockchains has become increasingly important with the popularity of bitcoin, Ethereum, litecoin and other cryptocurrencies.^[55] A blockchain, if it is public, provides anyone who wants access to observe and analyse the chain data, given one has the know-how. The process of understanding and accessing the flow of crypto has been an issue for many cryptocurrencies, crypto exchanges and banks.^{[56][57]} The reason for this is accusations of blockchain-enabled cryptocurrencies enabling illicit dark market trading of drugs, weapons, money laundering, etc.^[58] A common belief has been that cryptocurrency is private and untraceable, thus leading many actors to use it for illegal purposes. This is changing now that specialised tech companies provide blockchain tracking services, making crypto exchanges, law-enforcement and banks more aware of what is happening with crypto funds and fiat-crypto exchanges. The development, some argue, has led criminals to prioritise the use of new cryptos such as Monero.^{[59][60][61]}

Standardisation

In April 2016, Standards Australia submitted a proposal to the International Organization for Standardization to consider developing standards to support blockchain technology. This proposal resulted in the creation of ISO Technical Committee 307, Blockchain and Distributed Ledger Technologies.^[62] The technical committee has working groups relating to blockchain terminology, reference architecture, security and privacy, identity, smart contracts, governance and interoperability for blockchain and DLT, as well as standards specific to industry sectors and generic government requirements.^[63] More than 50 countries are participating in the standardization process together with external liaisons such as the Society for Worldwide Interbank Financial Telecommunication (SWIFT), the European Commission, the International Federation of Surveyors, the International Telecommunication Union (ITU) and the United Nations Economic Commission for Europe (UNECE).^[63]

Many other national standards bodies and open standards bodies are also working on blockchain standards.^[64] These include the National Institute of Standards and Technology^[65] (NIST), the European Committee for Electrotechnical Standardization^[66] (CENELEC), the Institute of Electrical and Electronics Engineers^[67] (IEEE), the Organization for the Advancement of Structured Information Standards (OASIS), and some individual participants in the Internet Engineering Task Force^[68] (IETF).

Centralized blockchain

Although most of blockchain implementation are decentralized and distributed, Oracle launched a centralized blockchain table feature in Oracle 21c database. The Blockchain Table in Oracle 21c database is a centralized blockchain which provide immutable feature. Compared to decentralized blockchains,

centralized blockchains normally can provide a higher throughput and lower latency of transactions than consensus-based distributed blockchains.^{[69][70]}

Types

Currently, there are at least four types of blockchain networks — public blockchains, private blockchains, consortium blockchains and hybrid blockchains.

Public blockchains

A public blockchain has absolutely no access restrictions. Anyone with an Internet connection can send transactions to it as well as become a validator (i.e., participate in the execution of a consensus protocol).^[71] Usually, such networks offer economic incentives for those who secure them and utilize some type of a proof-of-stake or proof-of-work algorithm.

Some of the largest, most known public blockchains are the bitcoin blockchain and the Ethereum blockchain.

Private blockchains

A private blockchain is permissioned.^[53] One cannot join it unless invited by the network administrators. Participant and validator access is restricted. To distinguish between open blockchains and other peer-to-peer decentralized database applications that are not open ad-hoc compute clusters, the terminology Distributed Ledger (DLT) is normally used for private blockchains.

Hybrid blockchains

A hybrid blockchain has a combination of centralized and decentralized features.^[72] The exact workings of the chain can vary based on which portions of centralization and decentralization are used.

Sidechains

A sidechain is a designation for a blockchain ledger that runs in parallel to a primary blockchain.^{[73][74]} Entries from the primary blockchain (where said entries typically represent digital assets) can be linked to and from the sidechain; this allows the sidechain to otherwise operate independently of the primary blockchain (e.g., by using an alternate means of record keeping, alternate consensus algorithm, etc.).^[75]

Consortium blockchain

A consortium blockchain is a type of blockchain that combines elements of both public and private blockchains. In a consortium blockchain, a group of organizations come together to create and operate the blockchain, rather than a single entity. The consortium members jointly manage the blockchain network and are responsible for validating transactions. Consortium blockchains are permissioned, meaning that only certain individuals or organizations are allowed to participate in the network. This allows for greater control over who can access the blockchain and helps to ensure that sensitive information is kept confidential.

Consortium blockchains are commonly used in industries where multiple organizations need to collaborate on a common goal, such as supply chain management or financial services. One advantage of consortium blockchains is that they can be more efficient and scalable than public blockchains, as the number of nodes required to validate transactions is typically smaller. Additionally, consortium blockchains can provide greater security and reliability than private blockchains, as the consortium members work together to maintain the network. Some examples of consortium blockchains include Quorum and Hyperledger.^[76]

Uses

Blockchain technology can be integrated into multiple areas. The primary use of blockchains is as a distributed ledger for cryptocurrencies such as bitcoin; there were also a few other operational products that had matured from proof of concept by late 2016.^[52] As of 2016, some businesses have been testing the technology and conducting low-level implementation to gauge blockchain's effects on organizational efficiency in their back office.^[77]

In 2019, it was estimated that around \$2.9 billion were invested in blockchain technology, which represents an 89% increase from the year prior. Additionally, the International Data Corp estimated that corporate investment into blockchain technology would reach \$12.4 billion by 2022.^[78] Furthermore, According to PricewaterhouseCoopers (PwC), the second-largest professional services network in the world, blockchain technology has the potential to generate an annual business value of more than \$3 trillion by 2030. PwC's estimate is further augmented by a 2018 study that they have conducted, in which PwC surveyed 600 business executives and determined that 84% have at least some exposure to utilizing blockchain technology, which indicates a significant demand and interest in blockchain technology.^[79]

In 2019, the BBC World Service radio and podcast series *Fifty Things That Made the Modern Economy* identified blockchain as a technology that would have far-reaching consequences for economics and society. The economist and *Financial Times* journalist and broadcaster Tim Harford discussed why the underlying technology might have much wider applications and the challenges that needed to be overcome.^[80] His first broadcast was on June 29, 2019.

The number of blockchain wallets quadrupled to 40 million between 2016 and 2020.^[81]

A paper published in 2022 discussed the potential use of blockchain technology in sustainable management.^[82]

Cryptocurrencies

Most cryptocurrencies use blockchain technology to record transactions. For example, the bitcoin network and Ethereum network are both based on blockchain.

The criminal enterprise Silk Road, which operated on Tor, utilized cryptocurrency for payments, some of which the US federal government seized through research on the blockchain and forfeiture.^[83]

Governments have mixed policies on the legality of their citizens or banks owning cryptocurrencies. China implements blockchain technology in several industries including a national digital currency which launched in 2020.^[84] To strengthen their respective currencies, Western governments including the

European Union and the United States have initiated similar projects.^[85]

Smart contracts

Blockchain-based smart contracts are contracts that can be partially or fully executed or enforced without human interaction.^[86] One of the main objectives of a smart contract is automated escrow. A key feature of smart contracts is that they do not need a trusted third party (such as a trustee) to act as an intermediary between contracting entities — the blockchain network executes the contract on its own. This may reduce friction between entities when transferring value and could subsequently open the door to a higher level of transaction automation.^[87] An IMF staff discussion from 2018 reported that smart contracts based on blockchain technology might reduce moral hazards and optimize the use of contracts in general, but "no viable smart contract systems have yet emerged." Due to the lack of widespread use, their legal status was unclear.^{[88][89]}

Financial services

According to *Reason*, many banks have expressed interest in implementing distributed ledgers for use in banking and are cooperating with companies creating private blockchains;^{[90][91][92]} according to a September 2016 IBM study, it is occurring faster than expected.^[93]

Banks are interested in this technology not least because it has the potential to speed up back office settlement systems.^[94] Moreover, as the blockchain industry has reached early maturity institutional appreciation has grown that it is, practically speaking, the infrastructure of a whole new financial industry, with all the implications which that entails.^[95]

Banks such as UBS are opening new research labs dedicated to blockchain technology in order to explore how blockchain can be used in financial services to increase efficiency and reduce costs.^{[96][97]}

Berenberg, a German bank, believes that blockchain is an "overhyped technology" that has had a large number of "proofs of concept", but still has major challenges, and very few success stories.^[98]

The blockchain has also given rise to initial coin offerings (ICOs) as well as a new category of digital asset called security token offerings (STOs), also sometimes referred to as digital security offerings (DSOs).^[99] STO/DSOs may be conducted privately or on public, regulated stock exchange and are used to tokenize traditional assets such as company shares as well as more innovative ones like intellectual property, real estate,^[100] art, or individual products. A number of companies are active in this space providing services for compliant tokenization, private STOs, and public STOs.

Games

Blockchain technology, such as cryptocurrencies and non-fungible tokens (NFTs), has been used in video games for monetization. Many live-service games offer in-game customization options, such as character skins or other in-game items, which the players can earn and trade with other players using in-game currency. Some games also allow for trading of virtual items using real-world currency, but this may be illegal in some countries where video games are seen as akin to gambling, and has led to gray market issues such as skin gambling, and thus publishers typically have shied away from allowing players to earn real-world funds from games.^[101] Blockchain games typically allow players to trade these in-game items for cryptocurrency, which can then be exchanged for money.^[102]

The first known game to use blockchain technologies was CryptoKitties, launched in November 2017, where the player would purchase NFTs with Ethereum cryptocurrency, each NFT consisting of a virtual pet that the player could breed with others to create offspring with combined traits as new NFTs.^{[103][102]} The game made headlines in December 2017 when one virtual pet sold for more than US\$100,000.^[104] CryptoKitties also illustrated scalability problems for games on Ethereum when it created significant congestion on the Ethereum network in early 2018 with approximately 30% of all Ethereum transactions being for the game.^{[105][106]}

By the early 2020s, there had not been a breakout success in video games using blockchain, as these games tend to focus on using blockchain for speculation instead of more traditional forms of gameplay, which offers limited appeal to most players. Such games also represent a high risk to investors as their revenues can be difficult to predict.^[102] However, limited successes of some games, such as Axie Infinity during the COVID-19 pandemic, and corporate plans towards metaverse content, refueled interest in the area of GameFi, a term describing the intersection of video games and financing typically backed by blockchain currency, in the second half of 2021.^[107] Several major publishers, including Ubisoft, Electronic Arts, and Take Two Interactive, have stated that blockchain and NFT-based games are under serious consideration for their companies in the future.^[108]

In October 2021, Valve Corporation banned blockchain games, including those using cryptocurrency and NFTs, from being hosted on its Steam digital storefront service, which is widely used for personal computer gaming, claiming that this was an extension of their policy banning games that offered in-game items with real-world value. Valve's prior history with gambling, specifically skin gambling, was speculated to be a factor in the decision to ban blockchain games.^[109] Journalists and players responded positively to Valve's decision as blockchain and NFT games have a reputation for scams and fraud among most PC gamers,^{[101][109]} and Epic Games, which runs the Epic Games Store in competition to Steam, said that they would be open to accepted blockchain games in the wake of Valve's refusal.^[110]

Supply chain

There have been several different efforts to employ blockchains in supply chain management.

- **Precious commodities mining** — Blockchain technology has been used for tracking the origins of gemstones and other precious commodities. In 2016, *The Wall Street Journal* reported that the blockchain technology company Everledger was partnering with IBM's blockchain-based tracking service to trace the origin of diamonds to ensure that they were ethically mined.^[111] As of 2019, the Diamond Trading Company (DTC) has been involved in building a diamond trading supply chain product called Tracer.^[112]
- **Food supply** — As of 2018, Walmart and IBM were running a trial to use a blockchain-backed system for supply chain monitoring for lettuce and spinach – all nodes of the blockchain were administered by Walmart and located on the IBM cloud.^[113]
- **Fashion industry** — There is an opaque relationship between brands, distributors, and customers in the fashion industry, which prevents the sustainable and stable development of the fashion industry. Blockchain could make this information transparent, assisting sustainable development of the industry.^[114]
- **Motor vehicles** — Mercedes-Benz and partner Icertis developed a blockchain prototype used to facilitate consistent documentation of contracts along the supply chain so that the ethical standards and contractual obligations required of its direct suppliers can be passed on to second tier suppliers and beyond.^{[115][116]} In another project, the company uses

blockchain technology to track the emissions of climate-relevant gases and the amount of secondary material along the supply chain for its battery cell manufacturers.^[117]

Domain names

There are several different efforts to offer domain name services via the blockchain. These domain names can be controlled by the use of a private key, which purports to allow for uncensorable websites. This would also bypass a registrar's ability to suppress domains used for fraud, abuse, or illegal content.^[118]

Namecoin is a cryptocurrency that supports the ".bit" top-level domain (TLD). Namecoin was forked from bitcoin in 2011. The .bit TLD is not sanctioned by ICANN, instead requiring an alternative DNS root.^[118] As of 2015, .bit was used by 28 websites, out of 120,000 registered names.^[119] Namecoin was dropped by OpenNIC in 2019, due to malware and potential other legal issues.^[120] Other blockchain alternatives to ICANN include *The Handshake Network*,^[119] *EmerDNS*, and *Unstoppable Domains*.^[118]

Specific TLDs include ".eth", ".luxe", and ".kred", which are associated with the Ethereum blockchain through the Ethereum Name Service (ENS). The .kred TLD also acts as an alternative to conventional cryptocurrency wallet addresses as a convenience for transferring cryptocurrency.^[121]

Other uses

Blockchain technology can be used to create a permanent, public, transparent ledger system for compiling data on sales, tracking digital use and payments to content creators, such as wireless users^[122] or musicians.^[123] The Gartner 2019 CIO Survey reported 2% of higher education respondents had launched blockchain projects and another 18% were planning academic projects in the next 24 months.^[124] In 2017, IBM partnered with ASCAP and PRS for Music to adopt blockchain technology in music distribution.^[125] Imogen Heap's Mycelia service has also been proposed as a blockchain-based alternative "that gives artists more control over how their songs and associated data circulate among fans and other musicians."^{[126][127]}

New distribution methods are available for the insurance industry such as peer-to-peer insurance, parametric insurance and microinsurance following the adoption of blockchain.^{[128][129]} The sharing economy and IoT are also set to benefit from blockchains because they involve many collaborating peers.^[130] The use of blockchain in libraries is being studied with a grant from the U.S. Institute of Museum and Library Services.^[131]

Other blockchain designs include Hyperledger, a collaborative effort from the Linux Foundation to support blockchain-based distributed ledgers, with projects under this initiative including Hyperledger Burrow (by Monax) and Hyperledger Fabric (spearheaded by IBM).^{[132][133][134]} Another is Quorum, a permissioned private blockchain by JPMorgan Chase with private storage, used for contract applications.^[135]

Oracle introduced a blockchain table feature in its Oracle 21c database.^{[69][70]}

Blockchain is also being used in peer-to-peer energy trading.^{[136][137][138]}

Lightweight blockchains, or simplified blockchains, are more suitable for internet of things (IoT) applications than conventional blockchains.^[139] One experiment suggested that a lightweight blockchain-based network could accommodate up to 1.34 million authentication processes every second, which could be sufficient for resource-constrained IoT networks.^[140]

Blockchain could be used in detecting counterfeits by associating unique identifiers to products, documents and shipments, and storing records associated with transactions that cannot be forged or altered.^{[141][142]} It is however argued that blockchain technology needs to be supplemented with technologies that provide a strong binding between physical objects and blockchain systems,^[143] as well as provisions for content creator verification *ala* KYC standards.^[144] The EUIPO established an Anti-Counterfeiting Blockathon Forum, with the objective of "defining, piloting and implementing" an anti-counterfeiting infrastructure at the European level.^{[145][146]} The Dutch Standardisation organisation NEN uses blockchain together with QR Codes to authenticate certificates.^[147]

Beijing and Shanghai are among the cities designated by China to trial blockchain applications as January 30, 2022.^[148] In Chinese legal proceedings, blockchain technology was first accepted as a method for authenticating internet evidence by the Hangzhou Internet Court in 2019 and has since been accepted by other Chinese courts.^{[149]:123–125}

Blockchain interoperability

With the increasing number of blockchain systems appearing, even only those that support cryptocurrencies, blockchain interoperability is becoming a topic of major importance. The objective is to support transferring assets from one blockchain system to another blockchain system. Wegner^[150] stated that "interoperability is the ability of two or more software components to cooperate despite differences in language, interface, and execution platform". The objective of blockchain interoperability is therefore to support such cooperation among blockchain systems, despite those kinds of differences.

There are already several blockchain interoperability solutions available.^[151] They can be classified into three categories: cryptocurrency interoperability approaches, blockchain engines, and blockchain connectors.

Several individual IETF participants produced the draft of a blockchain interoperability architecture.^[152]

Energy consumption concerns

Some cryptocurrencies use blockchain mining — the peer-to-peer computer computations by which transactions are validated and verified. This requires a large amount of energy. In June 2018, the Bank for International Settlements criticized the use of public proof-of-work blockchains for their high energy consumption.^{[153][154][155]}

Early concern over the high energy consumption was a factor in later blockchains such as Cardano (2017), Solana (2020) and Polkadot (2020) adopting the less energy-intensive proof-of-stake model. Researchers have estimated that bitcoin consumes 100,000 times as much energy as proof-of-stake networks.^{[156][157]}

In 2021, a study by Cambridge University determined that bitcoin (at 121 terawatt-hours per year) used more electricity than Argentina (at 121TWh) and the Netherlands (109TWh).^[158] According to Digiconomist, one bitcoin transaction required 708 kilowatt-hours of electrical energy, the amount an average U.S. household consumed in 24 days.^[159]

In February 2021, U.S. Treasury secretary Janet Yellen called bitcoin "an extremely inefficient way to conduct transactions", saying "the amount of energy consumed in processing those transactions is staggering".^[160] In March 2021, Bill Gates stated that "Bitcoin uses more electricity per transaction than any other method known to mankind", adding "It's not a great climate thing."^[161]

Nicholas Weaver, of the International Computer Science Institute at the University of California, Berkeley, examined blockchain's online security, and the energy efficiency of proof-of-work public blockchains, and in both cases found it grossly inadequate.^{[162][163]} The 31TWh-45TWh of electricity used for bitcoin in 2018 produced 17-23 million tonnes of CO₂.^{[164][165]} By 2022, the University of Cambridge and Digiconomist estimated that the two largest proof-of-work blockchains, bitcoin and Ethereum, together used twice as much electricity in one year as the whole of Sweden, leading to the release of up to 120 million tonnes of CO₂ each year.^[166]

Some cryptocurrency developers are considering moving from the proof-of-work model to the proof-of-stake model.^[167]

Academic research

In October 2014, the MIT Bitcoin Club, with funding from MIT alumni, provided undergraduate students at the Massachusetts Institute of Technology access to \$100 of bitcoin. The adoption rates, as studied by Catalini and Tucker (2016), revealed that when people who typically adopt technologies early are given delayed access, they tend to reject the technology.^[168] Many universities have founded departments focusing on crypto and blockchain, including MIT, in 2017. In the same year, Edinburgh became "one of the first big European universities to launch a blockchain course", according to the *Financial Times*.^[169]



Blockchain panel discussion at the first IEEE Computer Society TechIgnite conference

Adoption decision

Motivations for adopting blockchain technology (an aspect of innovation adoption) have been investigated by researchers. For example, Janssen, et al. provided a framework for analysis,^[170] and Koens & Poll pointed out that adoption could be heavily driven by non-technical factors.^[171] Based on behavioral models, Li^[172] has discussed the differences between adoption at the individual level and organizational levels.


Collaboration

Scholars in business and management have started studying the role of blockchains to support collaboration.^{[173][174]} It has been argued that blockchains can foster both cooperation (i.e., prevention of opportunistic behavior) and coordination (i.e., communication and information sharing). Thanks to reliability, transparency, traceability of records, and information immutability, blockchains facilitate collaboration in a way that differs both from the traditional use of contracts and from relational norms. Contrary to contracts, blockchains do not directly rely on the legal system to enforce agreements.^[175] In addition, contrary to the use of relational norms, blockchains do not require a trust or direct connections between collaborators.

Blockchain and internal audit

The need for internal audits to provide effective oversight of organizational efficiency will require a change in the way that information is accessed in new formats.^[177] Blockchain adoption requires a framework to identify the risk of exposure associated with transactions using blockchain. The Institute of Internal Auditors has identified the need for internal auditors to address this transformational technology. New methods are required to develop audit plans that identify threats and risks. The Internal Audit Foundation study, *Blockchain and Internal Audit*, assesses these factors.^[178] The American Institute of Certified Public Accountants has outlined new roles for auditors as a result of blockchain.^[179]

External videos

 Blockchain Basics & Cryptography (<https://www.youtube.com/watch?v=0UvVOMZqpEA>), Gary Gensler, Massachusetts Institute of Technology, 0:30^[176]

Journals

In September 2015, the first peer-reviewed academic journal dedicated to cryptocurrency and blockchain technology research, *Ledger*, was announced. The inaugural issue was published in December 2016.^[180] The journal covers aspects of mathematics, computer science, engineering, law, economics and philosophy that relate to cryptocurrencies.^{[181][182]} The journal encourages authors to digitally sign a file hash of submitted papers, which are then timestamped into the bitcoin blockchain. Authors are also asked to include a personal bitcoin address on the first page of their papers for non-repudiation purposes.^[183]

See also



Economics portal

- Changelog – a record of all notable changes made to a project
- Checklist – an informational aid used to reduce failure
- Economics of digitization
- List of blockchains
- Privacy and blockchain
- Version control – a record of all changes (mostly of software project) in a form of a graph

References



Deep learning

Deep learning is a subset of machine learning methods that utilize neural networks for representation learning. The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data. The adjective "deep" refers to the use of multiple layers (ranging from three to several hundred or thousands) in the network. Methods used can be either supervised, semi-supervised or unsupervised.^[2]

Some common deep learning network architectures include fully connected networks, deep belief networks, recurrent neural networks, convolutional neural networks, generative adversarial networks, transformers, and neural radiance fields. These

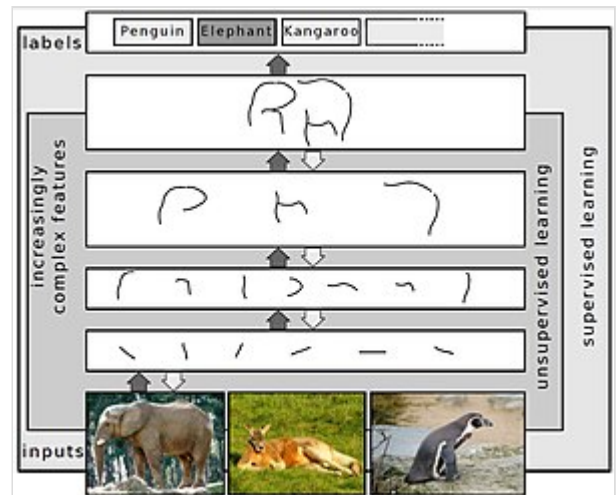
architectures have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.^{[3][4][5]}

Early forms of neural networks were inspired by information processing and distributed communication nodes in biological systems, particularly the human brain. However, current neural networks do not intend to model the brain function of organisms, and are generally seen as low-quality models for that purpose.^[6]

Overview

Most modern deep learning models are based on multi-layered neural networks such as convolutional neural networks and transformers, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines.^[7]

Fundamentally, deep learning refers to a class of machine learning algorithms in which a hierarchy of layers is used to transform input data into a slightly more abstract and composite representation. For example, in an image recognition model, the raw input may be an image (represented as a tensor of pixels). The first representational layer may attempt to identify basic shapes such as lines and circles, the second layer may compose and encode arrangements of edges, the third layer may encode a nose and eyes, and the fourth layer may recognize that the image contains a face.



Representing images on multiple layers of abstraction in deep learning^[1]

Importantly, a deep learning process can learn which features to optimally place at which level *on its own*. Prior to deep learning, machine learning techniques often involved hand-crafted feature engineering to transform the data into a more suitable representation for a classification algorithm to operate on. In the deep learning approach, features are not hand-crafted and the model discovers useful feature representations from the data automatically. This does not eliminate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.^{[8][2]}

The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial *credit assignment path* (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited.^[9] No universally agreed-upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth higher than two. CAP of depth two has been shown to be a universal approximator in the sense that it can emulate any function.^[10] Beyond that, more layers do not add to the function approximator ability of the network. Deep models (CAP > two) are able to extract better features than shallow models and hence, extra layers help in learning the features effectively.

Deep learning architectures can be constructed with a greedy layer-by-layer method.^[11] Deep learning helps to disentangle these abstractions and pick out which features improve performance.^[8]

Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabeled data are more abundant than the labeled data. Examples of deep structures that can be trained in an unsupervised manner are deep belief networks.^{[8][12]}

The term *Deep Learning* was introduced to the machine learning community by Rina Dechter in 1986,^[13] and to artificial neural networks by Igor Aizenberg and colleagues in 2000, in the context of Boolean threshold neurons.^{[14][15]} Although the history of its appearance is apparently more complicated.^[16]

Interpretations

Deep neural networks are generally interpreted in terms of the universal approximation theorem^{[17][18][19][20][21]} or probabilistic inference.^{[22][23][8][9][24]}

The classic universal approximation theorem concerns the capacity of feedforward neural networks with a single hidden layer of finite size to approximate continuous functions.^{[17][18][19][20]} In 1989, the first proof was published by George Cybenko for sigmoid activation functions^[17] and was generalised to feed-forward multi-layer architectures in 1991 by Kurt Hornik.^[18] Recent work also showed that universal approximation also holds for non-bounded activation functions such as Kunihiko Fukushima's rectified linear unit.^{[25][26]}

The universal approximation theorem for deep neural networks concerns the capacity of networks with bounded width but the depth is allowed to grow. Lu et al.^[21] proved that if the width of a deep neural network with ReLU activation is strictly larger than the input dimension, then the network can

approximate any Lebesgue integrable function; if the width is smaller or equal to the input dimension, then a deep neural network is not a universal approximator.

The probabilistic interpretation^[24] derives from the field of machine learning. It features inference,^{[23][7][8][9][12][24]} as well as the optimization concepts of training and testing, related to fitting and generalization, respectively. More specifically, the probabilistic interpretation considers the activation nonlinearity as a cumulative distribution function.^[24] The probabilistic interpretation led to the introduction of dropout as regularizer in neural networks. The probabilistic interpretation was introduced by researchers including Hopfield, Widrow and Narendra and popularized in surveys such as the one by Bishop.^[27]

History

Before 1980

There are two types of artificial neural network (ANN): feedforward neural network (FNN) or multilayer perceptron (MLP) and recurrent neural networks (RNN). RNNs have cycles in their connectivity structure, FNNs don't. In the 1920s, Wilhelm Lenz and Ernst Ising created the Ising model^{[28][29]} which is essentially a non-learning RNN architecture consisting of neuron-like threshold elements. In 1972, Shun'ichi Amari made this architecture adaptive.^{[30][31]} His learning RNN was republished by John Hopfield in 1982.^[32] Other early recurrent neural networks were published by Kaoru Nakano in 1971.^{[33][34]} Already in 1948, Alan Turing produced work on "Intelligent Machinery" that was not published in his lifetime,^[35] containing "ideas related to artificial evolution and learning RNNs."^[31]

Frank Rosenblatt (1958)^[36] proposed the perceptron, an MLP with 3 layers: an input layer, a hidden layer with randomized weights that did not learn, and an output layer. He later published a 1962 book that also introduced variants and computer experiments, including a version with four-layer perceptrons "with adaptive preterminal networks" where the last two layers have learned weights (here he credits H. D. Block and B. W. Knight).^[37]section 16 The book cites an earlier network by R. D. Joseph (1960)^[38] "functionally equivalent to a variation of" this four-layer system (the book mentions Joseph over 30 times). Should Joseph therefore be considered the originator of proper adaptive multilayer perceptrons with learning hidden units? Unfortunately, the learning algorithm was not a functional one, and fell into oblivion.

The first working deep learning algorithm was the Group method of data handling, a method to train arbitrarily deep neural networks, published by Alexey Ivakhnenko and Lapa in 1965. They regarded it as a form of polynomial regression,^[39] or a generalization of Rosenblatt's perceptron.^[40] A 1971 paper described a deep network with eight layers trained by this method,^[41] which is based on layer by layer training through regression analysis. Superfluous hidden units are pruned using a separate validation set. Since the activation functions of the nodes are Kolmogorov-Gabor polynomials, these were also the first deep networks with multiplicative units or "gates."^[31]

The first deep learning multilayer perceptron trained by stochastic gradient descent^[42] was published in 1967 by Shun'ichi Amari.^[43] In computer experiments conducted by Amari's student Saito, a five layer MLP with two modifiable layers learned internal representations to classify non-linearly separable

pattern classes.^[31] Subsequent developments in hardware and hyperparameter tunings have made end-to-end stochastic gradient descent the currently dominant training technique.

In 1969, Kunihiko Fukushima introduced the ReLU (rectified linear unit) activation function.^{[25][31]} The rectifier has become the most popular activation function for deep learning.^[44]

Deep learning architectures for convolutional neural networks (CNNs) with convolutional layers and downsampling layers began with the Neocognitron introduced by Kunihiko Fukushima in 1979, though not trained by backpropagation.^{[45][46]}

Backpropagation is an efficient application of the chain rule derived by Gottfried Wilhelm Leibniz in 1673^[47] to networks of differentiable nodes. The terminology "back-propagating errors" was actually introduced in 1962 by Rosenblatt,^[37] but he did not know how to implement this, although Henry J. Kelley had a continuous precursor of backpropagation in 1960 in the context of control theory.^[48] The modern form of backpropagation was first published in Seppo Linnainmaa's master thesis (1970).^{[49][50][31]} G.M. Ostrovski et al. republished it in 1971.^{[51][52]} Paul Werbos applied backpropagation to neural networks in 1982^[53] (his 1974 PhD thesis, reprinted in a 1994 book,^[54] did not yet describe the algorithm^[52]). In 1986, David E. Rumelhart et al. popularised backpropagation but did not cite the original work.^{[55][56]}

1980s-2000s

The time delay neural network (TDNN) was introduced in 1987 by Alex Waibel to apply CNN to phoneme recognition. It used convolutions, weight sharing, and backpropagation.^{[57][58]} In 1988, Wei Zhang applied a backpropagation-trained CNN to alphabet recognition.^[59] In 1989, Yann LeCun et al. created a CNN called LeNet for recognizing handwritten ZIP codes on mail. Training required 3 days.^[60] In 1990, Wei Zhang implemented a CNN on optical computing hardware.^[61] In 1991, a CNN was applied to medical image object segmentation^[62] and breast cancer detection in mammograms.^[63] LeNet-5 (1998), a 7-level CNN by Yann LeCun et al., that classifies digits, was applied by several banks to recognize hand-written numbers on checks digitized in 32x32 pixel images.^[64]

Recurrent neural networks (RNN)^{[28][30]} were further developed in the 1980s. Recurrence is used for sequence processing, and when a recurrent network is unrolled, it mathematically resembles a deep feedforward layer. Consequently, they have similar properties and issues, and their developments had mutual influences. In RNN, two early influential works were the Jordan network (1986)^[65] and the Elman network (1990),^[66] which applied RNN to study problems in cognitive psychology.

In the 1980s, backpropagation did not work well for deep learning with long credit assignment paths. To overcome this problem, in 1991, Jürgen Schmidhuber proposed a hierarchy of RNNs pre-trained one level at a time by self-supervised learning where each RNN tries to predict its own next input, which is the next unexpected input of the RNN below.^{[67][68]} This "neural history compressor" uses predictive coding to learn internal representations at multiple self-organizing time scales. This can substantially facilitate downstream deep learning. The RNN hierarchy can be *collapsed* into a single RNN, by distilling a higher level *chunker* network into a lower level *automatizer* network.^{[67][68][31]} In 1993, a neural history compressor solved a "Very Deep Learning" task that required more than 1000 subsequent layers in an RNN unfolded in time.^[69] The "P" in ChatGPT refers to such pre-training.

Sepp Hochreiter's diploma thesis (1991)^[70] implemented the neural history compressor,^[67] and identified and analyzed the vanishing gradient problem.^{[70][71]} Hochreiter proposed recurrent residual connections to solve the vanishing gradient problem. This led to the long short-term memory (LSTM), published in 1995.^[72] LSTM can learn "very deep learning" tasks^[9] with long credit assignment paths that require memories of events that happened thousands of discrete time steps before. That LSTM was not yet the modern architecture, which required a "forget gate", introduced in 1999,^[73] which became the standard RNN architecture.

In 1991, Jürgen Schmidhuber also published adversarial neural networks that contest with each other in the form of a zero-sum game, where one network's gain is the other network's loss.^{[74][75]} The first network is a generative model that models a probability distribution over output patterns. The second network learns by gradient descent to predict the reactions of the environment to these patterns. This was called "artificial curiosity". In 2014, this principle was used in generative adversarial networks (GANs).^[76]

During 1985–1995, inspired by statistical mechanics, several architectures and methods were developed by Terry Sejnowski, Peter Dayan, Geoffrey Hinton, etc., including the Boltzmann machine,^[77] restricted Boltzmann machine,^[78] Helmholtz machine,^[79] and the wake-sleep algorithm.^[80] These were designed for unsupervised learning of deep generative models. However, those were more computationally expensive compared to backpropagation. Boltzmann machine learning algorithm, published in 1985, was briefly popular before being eclipsed by the backpropagation algorithm in 1986. (p. 112 ^[81]). A 1988 network became state of the art in protein structure prediction, an early application of deep learning to bioinformatics.^[82]

Both shallow and deep learning (e.g., recurrent nets) of ANNs for speech recognition have been explored for many years.^{[83][84][85]} These methods never outperformed non-uniform internal-handcrafting Gaussian mixture model/Hidden Markov model (GMM-HMM) technology based on generative models of speech trained discriminatively.^[86] Key difficulties have been analyzed, including gradient diminishing^[70] and weak temporal correlation structure in neural predictive models.^{[87][88]} Additional difficulties were the lack of training data and limited computing power.

Most speech recognition researchers moved away from neural nets to pursue generative modeling. An exception was at SRI International in the late 1990s. Funded by the US government's NSA and DARPA, SRI researched in speech and speaker recognition. The speaker recognition team led by Larry Heck reported significant success with deep neural networks in speech processing in the 1998 NIST Speaker Recognition benchmark.^{[89][90]} It was deployed in the Nuance Verifier, representing the first major industrial application of deep learning.^[91]

The principle of elevating "raw" features over hand-crafted optimization was first explored successfully in the architecture of deep autoencoder on the "raw" spectrogram or linear filter-bank features in the late 1990s,^[90] showing its superiority over the Mel-Cepstral features that contain stages of fixed transformation from spectrograms. The raw features of speech, waveforms, later produced excellent larger-scale results.^[92]

2000s

Neural networks entered a null, and simpler models that use task-specific handcrafted features such as Gabor filters and support vector machines (SVMs) became the preferred choices in the 1990s and 2000s, because of artificial neural networks' computational cost and a lack of understanding of how the brain wires its biological networks.

In 2003, LSTM became competitive with traditional speech recognizers on certain tasks.^[93] In 2006, Alex Graves, Santiago Fernández, Faustino Gomez, and Schmidhuber combined it with connectionist temporal classification (CTC)^[94] in stacks of LSTMs.^[95] In 2009, it became the first RNN to win a pattern recognition contest, in connected handwriting recognition.^{[96][9]}

In 2006, publications by Geoff Hinton, Ruslan Salakhutdinov, Osindero and Teh^{[97][98]} deep belief networks were developed for generative modeling. They are trained by training one restricted Boltzmann machine, then freezing it and training another one on top of the first one, and so on, then optionally fine-tuned using supervised backpropagation.^[99] They could model high-dimensional probability distributions, such as the distribution of MNIST images, but convergence was slow.^{[100][101][102]}

The impact of deep learning in industry began in the early 2000s, when CNNs already processed an estimated 10% to 20% of all the checks written in the US, according to Yann LeCun.^[103] Industrial applications of deep learning to large-scale speech recognition started around 2010.

The 2009 NIPS Workshop on Deep Learning for Speech Recognition was motivated by the limitations of deep generative models of speech, and the possibility that given more capable hardware and large-scale data sets that deep neural nets might become practical. It was believed that pre-training DNNs using generative models of deep belief nets (DBN) would overcome the main difficulties of neural nets. However, it was discovered that replacing pre-training with large amounts of training data for straightforward backpropagation when using DNNs with large, context-dependent output layers produced error rates dramatically lower than then-state-of-the-art Gaussian mixture model (GMM)/Hidden Markov Model (HMM) and also than more-advanced generative model-based systems.^[104] The nature of the recognition errors produced by the two types of systems was characteristically different,^[105] offering technical insights into how to integrate deep learning into the existing highly efficient, run-time speech decoding system deployed by all major speech recognition systems.^{[23][106][107]} Analysis around 2009–2010, contrasting the GMM (and other generative speech models) vs. DNN models, stimulated early industrial investment in deep learning for speech recognition.^[105] That analysis was done with comparable performance (less than 1.5% in error rate) between discriminative DNNs and generative models.^{[104][105][108]} In 2010, researchers extended deep learning from TIMIT to large vocabulary speech recognition, by adopting large output layers of the DNN based on context-dependent HMM states constructed by decision trees.^{[109][110][111][106]}

Deep learning revolution

The deep learning revolution started around CNN- and GPU-based computer vision.

Although CNNs trained by backpropagation had been around for decades and GPU implementations of NNs for years,^[112] including CNNs,^[113] faster implementations of CNNs on GPUs were needed to progress on computer vision. Later, as deep learning becomes widespread, specialized hardware and algorithm optimizations were developed specifically for deep learning.^[114]

A key advance for the deep learning revolution was hardware advances, especially GPU. Some early work dated back to 2004.^{[112][113]} In 2009, Raina, Madhavan, and Andrew Ng reported a 100M deep belief network trained on 30 Nvidia GeForce GTX 280 GPUs, an early demonstration of GPU-based deep learning. They reported up to 70 times faster training.^[115]

In 2011, a CNN named *DanNet*^{[116][117]} by Dan Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber achieved for the first time superhuman performance in a visual pattern recognition contest, outperforming traditional methods by a factor of 3.^[9] It then won more contests.^{[118][119]} They also showed how max-pooling CNNs on GPU improved performance significantly.^[3]

In 2012, Andrew Ng and Jeff Dean created an FNN that learned to recognize higher-level concepts, such as cats, only from watching unlabeled images taken from YouTube videos.^[120]

In October 2012, AlexNet by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton^[4] won the large-scale ImageNet competition by a significant margin over shallow machine learning methods. Further incremental improvements included the VGG-16 network by Karen Simonyan and Andrew Zisserman^[121] and Google's Inceptionv3.^[122]

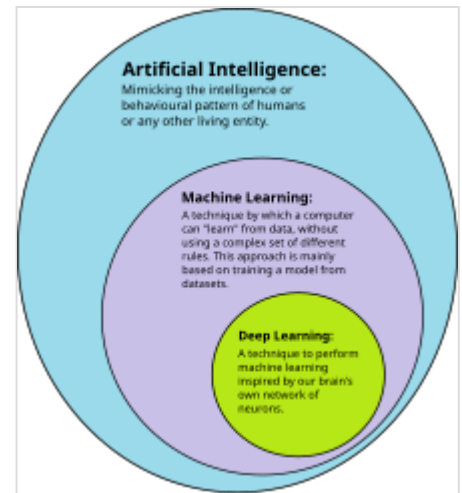
The success in image classification was then extended to the more challenging task of generating descriptions (captions) for images, often as a combination of CNNs and LSTMs.^{[123][124][125]}

In 2014, the state of the art was training “very deep neural network” with 20 to 30 layers.^[126] Stacking too many layers led to a steep reduction in training accuracy,^[127] known as the “degradation” problem.^[128] In 2015, two techniques were developed to train very deep networks: the Highway Network was published in May 2015, and the residual neural network (ResNet)^[129] in Dec 2015. ResNet behaves like an open-gated Highway Net.

Around the same time, deep learning started impacting the field of art. Early examples included Google DeepDream (2015), and neural style transfer (2015),^[130] both of which were based on pretrained image classification neural networks, such as VGG-19.

Generative adversarial network (GAN) by (Ian Goodfellow et al., 2014)^[131] (based on Jürgen Schmidhuber's principle of artificial curiosity^{[74][76]}) became state of the art in generative modeling during 2014-2018 period. Excellent image quality is achieved by Nvidia's StyleGAN (2018)^[132] based on the Progressive GAN by Tero Karras et al.^[133] Here the GAN generator is grown from small to large scale in a pyramidal fashion. Image generation by GAN reached popular success, and provoked discussions concerning deepfakes.^[134] Diffusion models (2015)^[135] eclipsed GANs in generative modeling since then, with systems such as DALL·E 2 (2022) and Stable Diffusion (2022).

In 2015, Google's speech recognition improved by 49% by an LSTM-based model, which they made available through Google Voice Search on smartphone.^{[136][137]}



How deep learning is a subset of machine learning and how machine learning is a subset of artificial intelligence (AI)

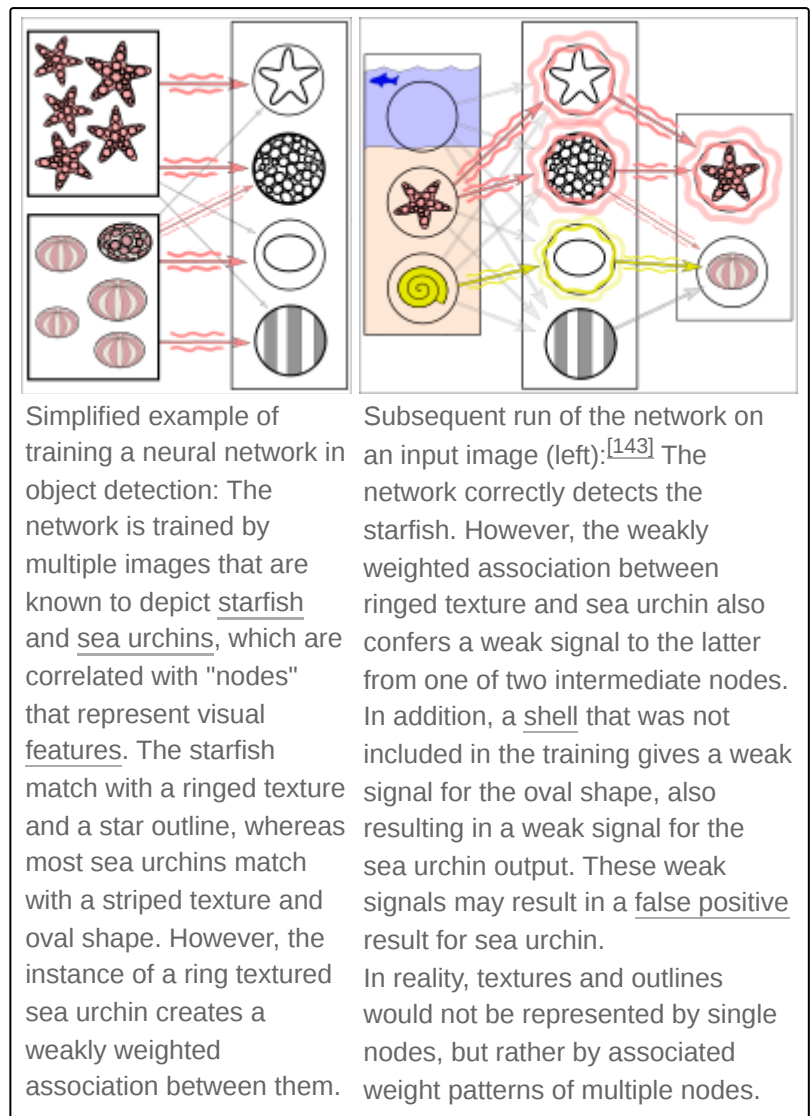
Deep learning is part of state-of-the-art systems in various disciplines, particularly computer vision and automatic speech recognition (ASR). Results on commonly used evaluation sets such as TIMIT (ASR) and MNIST (image classification), as well as a range of large-vocabulary speech recognition tasks have steadily improved.^{[104][138]} Convolutional neural networks were superseded for ASR by LSTM,^{[137][139][140][141]} but are more successful in computer vision.

Yoshua Bengio, Geoffrey Hinton and Yann LeCun were awarded the 2018 Turing Award for "conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing."^[142]

Neural networks

Artificial neural networks (ANNs) or **connectionist systems** are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve their ability) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images. They have found most use in applications difficult to express with a traditional computer algorithm using rule-based programming.

An ANN is based on a collection of connected units called artificial neurons, (analogous to biological neurons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream.



Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times.

The original goal of the neural network approach was to solve problems in the same way that a human brain would. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as backpropagation, or passing information in the reverse direction and adjusting the network to reflect that information.

Neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

As of 2017, neural networks typically have a few thousand to a few million units and millions of connections. Despite this number being several order of magnitude less than the number of neurons on a human brain, these networks can perform many tasks at a level beyond that of humans (e.g., recognizing faces, or playing "Go"^[144]).

Deep neural networks

A deep neural network (DNN) is an artificial neural network with multiple layers between the input and output layers.^{[7][9]} There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions.^[145] These components as a whole function in a way that mimics functions of the human brain, and can be trained like any other ML algorithm.

For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks.

DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives.^[146] The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.^[7] For instance, it was proved that sparse multivariate polynomials are exponentially easier to approximate with DNNs than with shallow networks.^[147]

Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network did not accurately recognize a particular pattern, an algorithm would adjust the weights.^[148] That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data.

Recurrent neural networks, in which data can flow in any direction, are used for applications such as language modeling.^{[149][150][151][152][153]} Long short-term memory is particularly effective for this use.^{[154][155]}

Convolutional neural networks (CNNs) are used in computer vision.^[156] CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).^[157]

Challenges

As with ANNs, many issues can arise with naively trained DNNs. Two common issues are overfitting and computation time.

DNNs are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. Regularization methods such as Ivakhnenko's unit pruning^[41] or weight decay (ℓ_2 -regularization) or sparsity (ℓ_1 -regularization) can be applied during training to combat overfitting.^[158] Alternatively dropout regularization randomly omits units from the hidden layers during training. This helps to exclude rare dependencies.^[159] Finally, data can be augmented via methods such as cropping and rotating such that smaller training sets can be increased in size to reduce the chances of overfitting.^[160]

DNNs must consider many training parameters, such as the size (number of layers and number of units per layer), the learning rate, and initial weights. Sweeping through the parameter space for optimal parameters may not be feasible due to the cost in time and computational resources. Various tricks, such as batching (computing the gradient on several training examples at once rather than individual examples)^[161] speed up computation. Large processing capabilities of many-core architectures (such as GPUs or the Intel Xeon Phi) have produced significant speedups in training, because of the suitability of such processing architectures for the matrix and vector computations.^{[162][163]}

Alternatively, engineers may look for other types of neural networks with more straightforward and convergent training algorithms. CMAC (cerebellar model articulation controller) is one such kind of neural network. It doesn't require learning rates or randomized initial weights. The training process can be guaranteed to converge in one step with a new batch of data, and the computational complexity of the training algorithm is linear with respect to the number of neurons involved.^{[164][165]}

Hardware

Since the 2010s, advances in both machine learning algorithms and computer hardware have led to more efficient methods for training deep neural networks that contain many layers of non-linear hidden units and a very large output layer.^[166] By 2019, graphics processing units (GPUs), often with AI-specific enhancements, had displaced CPUs as the dominant method for training large-scale commercial cloud AI.^[167] OpenAI estimated the hardware computation used in the largest deep learning projects from AlexNet (2012) to AlphaZero (2017) and found a 300,000-fold increase in the amount of computation required, with a doubling-time trendline of 3.4 months.^{[168][169]}

Special electronic circuits called deep learning processors were designed to speed up deep learning algorithms. Deep learning processors include neural processing units (NPUs) in Huawei cellphones^[170] and cloud computing servers such as tensor processing units (TPU) in the Google Cloud Platform.^[171]

Cerebras Systems has also built a dedicated system to handle large deep learning models, the CS-2, based on the largest processor in the industry, the second-generation Wafer Scale Engine (WSE-2).^{[172][173]}

Atomically thin semiconductors are considered promising for energy-efficient deep learning hardware where the same basic device structure is used for both logic operations and data storage. In 2020, Marega et al. published experiments with a large-area active channel material for developing logic-in-memory devices and circuits based on floating-gate field-effect transistors (FGFETs).^[174]

In 2021, J. Feldmann et al. proposed an integrated photonic hardware accelerator for parallel convolutional processing.^[175] The authors identify two key advantages of integrated photonics over its electronic counterparts: (1) massively parallel data transfer through wavelength division multiplexing in conjunction with frequency combs, and (2) extremely high data modulation speeds.^[175] Their system can execute trillions of multiply-accumulate operations per second, indicating the potential of integrated photonics in data-heavy AI applications.^[175]

Applications

Automatic speech recognition

Large-scale automatic speech recognition is the first and most convincing successful case of deep learning. LSTM RNNs can learn "Very Deep Learning" tasks^[9] that involve multi-second intervals containing speech events separated by thousands of discrete time steps, where one time step corresponds to about 10 ms. LSTM with forget gates^[155] is competitive with traditional speech recognizers on certain tasks.^[93]

The initial success in speech recognition was based on small-scale recognition tasks based on TIMIT. The data set contains 630 speakers from eight major dialects of American English, where each speaker reads 10 sentences.^[176] Its small size lets many configurations be tried. More importantly, the TIMIT task concerns phone-sequence recognition, which, unlike word-sequence recognition, allows weak phone bigram language models. This lets the strength of the acoustic modeling aspects of speech recognition be more easily analyzed. The error rates listed below, including these early results and measured as percent phone error rates (PER), have been summarized since 1991.

Method	Percent phone error rate (PER) (%)
Randomly Initialized RNN ^[177]	26.1
Bayesian Triphone GMM-HMM	25.6
Hidden Trajectory (Generative) Model	24.8
Monophone Randomly Initialized DNN	23.4
Monophone DBN-DNN	22.4
Triphone GMM-HMM with BMMI Training	21.7
Monophone DBN-DNN on fbank	20.7
Convolutional DNN ^[178]	20.0
Convolutional DNN w. Heterogeneous Pooling	18.7
Ensemble DNN/CNN/RNN ^[179]	18.3
Bidirectional LSTM	17.8
Hierarchical Convolutional Deep Maxout Network ^[180]	16.5

The debut of DNNs for speaker recognition in the late 1990s and speech recognition around 2009-2011 and of LSTM around 2003–2007, accelerated progress in eight major areas:^{[23][108][106]}

- Scale-up/out and accelerated DNN training and decoding
- Sequence discriminative training
- Feature processing by deep models with solid understanding of the underlying mechanisms
- Adaptation of DNNs and related deep models
- Multi-task and transfer learning by DNNs and related deep models
- CNNs and how to design them to best exploit domain knowledge of speech
- RNN and its rich LSTM variants
- Other types of deep models including tensor-based models and integrated deep generative/discriminative models.

All major commercial speech recognition systems (e.g., Microsoft Cortana, Xbox, Skype Translator, Amazon Alexa, Google Now, Apple Siri, Baidu and iFlyTek voice search, and a range of Nuance speech products, etc.) are based on deep learning.^{[23][181][182]}

Image recognition

A common evaluation set for image classification is the MNIST database data set. MNIST is composed of handwritten digits and includes 60,000 training examples and 10,000 test examples. As with TIMIT, its small size lets users test multiple configurations. A comprehensive list of results on this set is available.^[183]

Deep learning-based image recognition has become "superhuman", producing more accurate results than human contestants. This first occurred in 2011 in recognition of traffic signs, and in 2014, with recognition of human faces.^{[184][185]}

Deep learning-trained vehicles now interpret 360° camera views.^[186] Another example is Facial Dysmorphology Novel Analysis (FDNA) used to analyze cases of human malformation connected to a large database of genetic syndromes.

Visual art processing

Closely related to the progress that has been made in image recognition is the increasing application of deep learning techniques to various visual art tasks. DNNs have proven themselves capable, for example, of

- identifying the style period of a given painting^{[187][188]}
- Neural Style Transfer – capturing the style of a given artwork and applying it in a visually pleasing manner to an arbitrary photograph or video^{[187][188]}
- generating striking imagery based on random visual input fields.^{[187][188]}

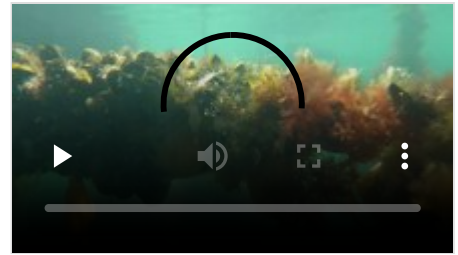
Natural language processing

Neural networks have been used for implementing language models since the early 2000s.^[149] LSTM helped to improve machine translation and language modeling.^{[150][151][152]}

Other key techniques in this field are negative sampling^[189] and word embedding. Word embedding, such as word2vec, can be thought of as a representational layer in a deep learning architecture that transforms an atomic word into a positional representation of the word relative to other words in the dataset; the position is represented as a point in a vector space. Using word embedding as an RNN input layer allows the network to parse sentences and phrases using an effective compositional vector grammar. A compositional vector grammar can be thought of as probabilistic context free grammar (PCFG) implemented by an RNN.^[190] Recursive auto-encoders built atop word embeddings can assess sentence similarity and detect paraphrasing.^[190] Deep neural architectures provide the best results for constituency parsing,^[191] sentiment analysis,^[192] information retrieval,^{[193][194]} spoken language understanding,^[195] machine translation,^{[150][196]} contextual entity linking,^[196] writing style recognition,^[197] named-entity recognition (token classification),^[198] text classification, and others.^[199]

Recent developments generalize word embedding to sentence embedding.

Google Translate (GT) uses a large end-to-end long short-term memory (LSTM) network.^{[200][201][202][203]} Google Neural Machine Translation (GNMT) uses an example-based machine translation method in which the system "learns from millions of examples".^[201] It translates "whole



Richard Green explains how deep learning is used with a remotely operated vehicle in mussel aquaculture



Visual art processing of Jimmy Wales in France, with the style of Munch's "The Scream" applied using neural style transfer

sentences at a time, rather than pieces". Google Translate supports over one hundred languages.^[201] The network encodes the "semantics of the sentence rather than simply memorizing phrase-to-phrase translations".^{[201][204]} GT uses English as an intermediate between most language pairs.^[204]

Drug discovery and toxicology

A large percentage of candidate drugs fail to win regulatory approval. These failures are caused by insufficient efficacy (on-target effect), undesired interactions (off-target effects), or unanticipated toxic effects.^{[205][206]} Research has explored use of deep learning to predict the biomolecular targets,^{[207][208]} off-targets, and toxic effects of environmental chemicals in nutrients, household products and drugs.^{[209][210][211]}

AtomNet is a deep learning system for structure-based rational drug design.^[212] AtomNet was used to predict novel candidate biomolecules for disease targets such as the Ebola virus^[213] and multiple sclerosis.^{[214][213]}

In 2017 graph neural networks were used for the first time to predict various properties of molecules in a large toxicology data set.^[215] In 2019, generative neural networks were used to produce molecules that were validated experimentally all the way into mice.^{[216][217]}

Customer relationship management

Deep reinforcement learning has been used to approximate the value of possible direct marketing actions, defined in terms of RFM variables. The estimated value function was shown to have a natural interpretation as customer lifetime value.^[218]

Recommendation systems

Recommendation systems have used deep learning to extract meaningful features for a latent factor model for content-based music and journal recommendations.^{[219][220]} Multi-view deep learning has been applied for learning user preferences from multiple domains.^[221] The model uses a hybrid collaborative and content-based approach and enhances recommendations in multiple tasks.

Bioinformatics

An autoencoder ANN was used in bioinformatics, to predict gene ontology annotations and gene-function relationships.^[222]

In medical informatics, deep learning was used to predict sleep quality based on data from wearables^[223] and predictions of health complications from electronic health record data.^[224]

Deep neural networks have shown unparalleled performance in predicting protein structure, according to the sequence of the amino acids that make it up. In 2020, AlphaFold, a deep-learning based system, achieved a level of accuracy significantly higher than all previous computational methods.^{[225][226]}

Deep Neural Network Estimations

Deep neural networks can be used to estimate the entropy of a stochastic process and called Neural Joint Entropy Estimator (NJEE).^[227] Such an estimation provides insights on the effects of input random variables on an independent random variable. Practically, the DNN is trained as a classifier that maps an input vector or matrix X to an output probability distribution over the possible classes of random variable Y, given input X. For example, in image classification tasks, the NJEE maps a vector of pixels' color values to probabilities over possible image classes. In practice, the probability distribution of Y is obtained by a Softmax layer with number of nodes that is equal to the alphabet size of Y. NJEE uses continuously differentiable activation functions, such that the conditions for the universal approximation theorem holds. It is shown that this method provides a strongly consistent estimator and outperforms other methods in case of large alphabet sizes.^[227]

Medical image analysis

Deep learning has been shown to produce competitive results in medical application such as cancer cell classification, lesion detection, organ segmentation and image enhancement.^{[228][229]} Modern deep learning tools demonstrate the high accuracy of detecting various diseases and the helpfulness of their use by specialists to improve the diagnosis efficiency.^{[230][231]}

Mobile advertising

Finding the appropriate mobile audience for mobile advertising is always challenging, since many data points must be considered and analyzed before a target segment can be created and used in ad serving by any ad server.^[232] Deep learning has been used to interpret large, many-dimensioned advertising datasets. Many data points are collected during the request/serve/click internet advertising cycle. This information can form the basis of machine learning to improve ad selection.

Image restoration

Deep learning has been successfully applied to inverse problems such as denoising, super-resolution, inpainting, and film colorization.^[233] These applications include learning methods such as "Shrinkage Fields for Effective Image Restoration"^[234] which trains on an image dataset, and Deep Image Prior, which trains on the image that needs restoration.

Financial fraud detection

Deep learning is being successfully applied to financial fraud detection, tax evasion detection,^[235] and anti-money laundering.^[236]

Materials science

In November 2023, researchers at Google DeepMind and Lawrence Berkeley National Laboratory announced that they had developed an AI system known as GNoME. This system has contributed to materials science by discovering over 2 million new materials within a relatively short timeframe. GNoME employs deep learning techniques to efficiently explore potential material structures, achieving a significant increase in the identification of stable inorganic crystal structures. The system's predictions were validated through autonomous robotic experiments, demonstrating a noteworthy success rate of 71%. The data of newly discovered materials is publicly available through the Materials Project database,

offering researchers the opportunity to identify materials with desired properties for various applications. This development has implications for the future of scientific discovery and the integration of AI in material science research, potentially expediting material innovation and reducing costs in product development. The use of AI and deep learning suggests the possibility of minimizing or eliminating manual lab experiments and allowing scientists to focus more on the design and analysis of unique compounds.^{[237][238][239]}

Military

The United States Department of Defense applied deep learning to train robots in new tasks through observation.^[240]

Partial differential equations

Physics informed neural networks have been used to solve partial differential equations in both forward and inverse problems in a data driven manner.^[241] One example is the reconstructing fluid flow governed by the Navier-Stokes equations. Using physics informed neural networks does not require the often expensive mesh generation that conventional CFD methods relies on.^{[242][243]}

Deep backward stochastic differential equation method

Deep backward stochastic differential equation method is a numerical method that combines deep learning with Backward stochastic differential equation (BSDE). This method is particularly useful for solving high-dimensional problems in financial mathematics. By leveraging the powerful function approximation capabilities of deep neural networks, deep BSDE addresses the computational challenges faced by traditional numerical methods in high-dimensional settings. Specifically, traditional methods like finite difference methods or Monte Carlo simulations often struggle with the curse of dimensionality, where computational cost increases exponentially with the number of dimensions. Deep BSDE methods, however, employ deep neural networks to approximate solutions of high-dimensional partial differential equations (PDEs), effectively reducing the computational burden.^[244]

In addition, the integration of Physics-informed neural networks (PINNs) into the deep BSDE framework enhances its capability by embedding the underlying physical laws directly into the neural network architecture. This ensures that the solutions not only fit the data but also adhere to the governing stochastic differential equations. PINNs leverage the power of deep learning while respecting the constraints imposed by the physical models, resulting in more accurate and reliable solutions for financial mathematics problems.

Image reconstruction

Image reconstruction is the reconstruction of the underlying images from the image-related measurements. Several works showed the better and superior performance of the deep learning methods compared to analytical methods for various applications, e.g., spectral imaging ^[245] and ultrasound imaging.^[246]

Weather prediction

Traditional weather prediction systems solve a very complex system of partial differential equations. GraphCast is a deep learning based model, trained on a long history of weather data to predict how weather patterns change over time. It is able to predict weather conditions for up to 10 days globally, at a very detailed level, and in under a minute, with precision similar to state of the art systems.^{[247][248]}

Epigenetic clock

An epigenetic clock is a biochemical test that can be used to measure age. Galkin et al. used deep neural networks to train an epigenetic aging clock of unprecedented accuracy using >6,000 blood samples.^[249] The clock uses information from 1000 CpG sites and predicts people with certain conditions older than healthy controls: IBD, frontotemporal dementia, ovarian cancer, obesity. The aging clock was planned to be released for public use in 2021 by an Insilico Medicine spinoff company Deep Longevity.

Relation to human cognitive and brain development

Deep learning is closely related to a class of theories of brain development (specifically, neocortical development) proposed by cognitive neuroscientists in the early 1990s.^{[250][251][252][253]} These developmental theories were instantiated in computational models, making them predecessors of deep learning systems. These developmental models share the property that various proposed learning dynamics in the brain (e.g., a wave of nerve growth factor) support the self-organization somewhat analogous to the neural networks utilized in deep learning models. Like the neocortex, neural networks employ a hierarchy of layered filters in which each layer considers information from a prior layer (or the operating environment), and then passes its output (and possibly the original input), to other layers. This process yields a self-organizing stack of transducers, well-tuned to their operating environment. A 1995 description stated, "...the infant's brain seems to organize itself under the influence of waves of so-called trophic-factors ... different regions of the brain become connected sequentially, with one layer of tissue maturing before another and so on until the whole brain is mature".^[254]

A variety of approaches have been used to investigate the plausibility of deep learning models from a neurobiological perspective. On the one hand, several variants of the backpropagation algorithm have been proposed in order to increase its processing realism.^{[255][256]} Other researchers have argued that unsupervised forms of deep learning, such as those based on hierarchical generative models and deep belief networks, may be closer to biological reality.^{[257][258]} In this respect, generative neural network models have been related to neurobiological evidence about sampling-based processing in the cerebral cortex.^[259]

Although a systematic comparison between the human brain organization and the neuronal encoding in deep networks has not yet been established, several analogies have been reported. For example, the computations performed by deep learning units could be similar to those of actual neurons^[260] and neural populations.^[261] Similarly, the representations developed by deep learning models are similar to those measured in the primate visual system^[262] both at the single-unit^[263] and at the population^[264] levels.

Commercial activity

Facebook's AI lab performs tasks such as automatically tagging uploaded pictures with the names of the people in them.^[265]

Google's DeepMind Technologies developed a system capable of learning how to play Atari video games using only pixels as data input. In 2015 they demonstrated their AlphaGo system, which learned the game of Go well enough to beat a professional Go player.^{[266][267][268]} Google Translate uses a neural network to translate between more than 100 languages.

In 2017, Covariant.ai was launched, which focuses on integrating deep learning into factories.^[269]

As of 2008,^[270] researchers at The University of Texas at Austin (UT) developed a machine learning framework called Training an Agent Manually via Evaluative Reinforcement, or TAMER, which proposed new methods for robots or computer programs to learn how to perform tasks by interacting with a human instructor.^[240] First developed as TAMER, a new algorithm called Deep TAMER was later introduced in 2018 during a collaboration between U.S. Army Research Laboratory (ARL) and UT researchers. Deep TAMER used deep learning to provide a robot with the ability to learn new tasks through observation.^[240] Using Deep TAMER, a robot learned a task with a human trainer, watching video streams or observing a human perform a task in-person. The robot later practiced the task with the help of some coaching from the trainer, who provided feedback such as "good job" and "bad job".^[271]

Criticism and comment

Deep learning has attracted both criticism and comment, in some cases from outside the field of computer science.

Theory

A main criticism concerns the lack of theory surrounding some methods.^[272] Learning in the most common deep architectures is implemented using well-understood gradient descent. However, the theory surrounding other algorithms, such as contrastive divergence is less clear. (e.g., Does it converge? If so, how fast? What is it approximating?) Deep learning methods are often looked at as a black box, with most confirmations done empirically, rather than theoretically.^[273]

Others point out that deep learning should be looked at as a step towards realizing strong AI, not as an all-encompassing solution. Despite the power of deep learning methods, they still lack much of the functionality needed to realize this goal entirely. Research psychologist Gary Marcus noted:

Realistically, deep learning is only part of the larger challenge of building intelligent machines. Such techniques lack ways of representing causal relationships (...) have no obvious ways of performing logical inferences, and they are also still a long way from integrating abstract knowledge, such as information about what objects are, what they are for, and how they are typically used. The most powerful A.I. systems, like Watson (...) use techniques like deep learning as just one element in a very complicated ensemble of techniques, ranging from the statistical technique of Bayesian inference to deductive reasoning.^[274]

In further reference to the idea that artistic sensitivity might be inherent in relatively low levels of the cognitive hierarchy, a published series of graphic representations of the internal states of deep (20-30 layers) neural networks attempting to discern within essentially random data the images on which they were trained^[275] demonstrate a visual appeal: the original research notice received well over 1,000 comments, and was the subject of what was for a time the most frequently accessed article on *The Guardian's*^[276] website.

Errors

Some deep learning architectures display problematic behaviors,^[277] such as confidently classifying unrecognizable images as belonging to a familiar category of ordinary images (2014)^[278] and misclassifying minuscule perturbations of correctly classified images (2013).^[279] Goertzel hypothesized that these behaviors are due to limitations in their internal representations and that these limitations would inhibit integration into heterogeneous multi-component artificial general intelligence (AGI) architectures.^[277] These issues may possibly be addressed by deep learning architectures that internally form states homologous to image-grammar^[280] decompositions of observed entities and events.^[277] Learning a grammar (visual or linguistic) from training data would be equivalent to restricting the system to commonsense reasoning that operates on concepts in terms of grammatical production rules and is a basic goal of both human language acquisition^[281] and artificial intelligence (AI).^[282]

Cyber threat

As deep learning moves from the lab into the world, research and experience show that artificial neural networks are vulnerable to hacks and deception.^[283] By identifying patterns that these systems use to function, attackers can modify inputs to ANNs in such a way that the ANN finds a match that human observers would not recognize. For example, an attacker can make subtle changes to an image such that the ANN finds a match even though the image looks to a human nothing like the search target. Such manipulation is termed an "adversarial attack".^[284]

In 2016 researchers used one ANN to doctor images in trial and error fashion, identify another's focal points, and thereby generate images that deceived it. The modified images looked no different to human eyes. Another group showed that printouts of doctored images then photographed successfully tricked an image classification system.^[285] One defense is reverse image search, in which a possible fake image is submitted to a site such as TinEye that can then find other instances of it. A refinement is to search using only parts of the image, to identify images from which that piece may have been taken.^[286]

Another group showed that certain psychedelic spectacles could fool a facial recognition system into thinking ordinary people were celebrities, potentially allowing one person to impersonate another. In 2017 researchers added stickers to stop signs and caused an ANN to misclassify them.^[285]

ANNs can however be further trained to detect attempts at deception, potentially leading attackers and defenders into an arms race similar to the kind that already defines the malware defense industry. ANNs have been trained to defeat ANN-based anti-malware software by repeatedly attacking a defense with malware that was continually altered by a genetic algorithm until it tricked the anti-malware while retaining its ability to damage the target.^[285]

In 2016, another group demonstrated that certain sounds could make the Google Now voice command system open a particular web address, and hypothesized that this could "serve as a stepping stone for further attacks (e.g., opening a web page hosting drive-by malware)".^[285]

In "data poisoning", false data is continually smuggled into a machine learning system's training set to prevent it from achieving mastery.^[285]

Data collection ethics

The deep learning systems that are trained using supervised learning often rely on data that is created and/or annotated by humans.^[287] It has been argued that not only low-paid clickwork (such as on Amazon Mechanical Turk) is regularly deployed for this purpose, but also implicit forms of human microwork that are often not recognized as such.^[288] The philosopher Rainer Mühlhoff distinguishes five types of "machinic capture" of human microwork to generate training data: (1) gamification (the embedding of annotation or computation tasks in the flow of a game), (2) "trapping and tracking" (e.g. CAPTCHAs for image recognition or click-tracking on Google search results pages), (3) exploitation of social motivations (e.g. tagging faces on Facebook to obtain labeled facial images), (4) information mining (e.g. by leveraging quantified-self devices such as activity trackers) and (5) clickwork.^[288]

See also

- Applications of artificial intelligence
- Comparison of deep learning software
- Compressed sensing
- Differentiable programming
- Echo state network
- List of artificial intelligence projects
- Liquid state machine
- List of datasets for machine-learning research
- Reservoir computing
- Scale space and deep learning
- Sparse coding
- Stochastic parrot
- Topological deep learning

References

1. Schulz, Hannes; Behnke, Sven (1 November 2012). "Deep Learning" (<https://www.semanticscholar.org/paper/51a80649d16a38d41dbd20472deb3bc9b61b59a0>). *KI - Künstliche Intelligenz*. **26** (4): 357–363. doi:10.1007/s13218-012-0198-z (<https://doi.org/10.1007%2Fs13218-012-0198-z>). ISSN 1610-1987 (<https://search.worldcat.org/issn/1610-1987>). S2CID 220523562 (<https://api.semanticscholar.org/CorpusID:220523562>).



Distributed computing

Distributed computing is a field of computer science that studies **distributed systems**, defined as computer systems whose inter-communicating components are located on different networked computers.^{[1][2]}

The components of a distributed system communicate and coordinate their actions by passing messages to one another in order to achieve a common goal. Three significant challenges of distributed systems are: maintaining concurrency of components, overcoming the lack of a global clock, and managing the independent failure of components.^[1] When a component of one system fails, the entire system does not fail.^[3] Examples of distributed systems vary from SOA-based systems to microservices to massively multiplayer online games to peer-to-peer applications. Distributed systems cost significantly more than monolithic architectures, primarily due to increased needs for additional hardware, servers, gateways, firewalls, new subnets, proxies, and so on.^[4] Also, distributed systems are prone to fallacies of distributed computing. On the other hand, a well designed distributed system is more scalable, more durable, more changeable and more fine-tuned than a monolithic application deployed on a single machine.^[5] According to Marc Brooker: "a system is scalable in the range where marginal cost of additional workload is nearly constant." Serverless technologies fit this definition but you need to consider total cost of ownership not just the infra cost.^[6]

A computer program that runs within a distributed system is called a **distributed program**,^[7] and *distributed programming* is the process of writing such programs.^[8] There are many different types of implementations for the message passing mechanism, including pure HTTP, RPC-like connectors and message queues.^[9]

Distributed computing also refers to the use of distributed systems to solve computational problems. In *distributed computing*, a problem is divided into many tasks, each of which is solved by one or more computers,^[10] which communicate with each other via message passing.^[11]

Introduction

The word *distributed* in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area.^[12] The terms are nowadays used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing.^[11]

While there is no single definition of a distributed system,^[13] the following defining properties are commonly used as:

- There are several autonomous computational entities (*computers* or nodes), each of which has its own local memory.^[14]
- The entities communicate with each other by message passing.^[15]

A distributed system may have a common goal, such as solving a large computational problem;^[16] the user then perceives the collection of autonomous processors as a unit. Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users.^[17]

Other typical properties of distributed systems include the following:

- The system has to tolerate failures in individual computers.^[18]
- The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program.^[19]
- Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input.^[20]

Patterns

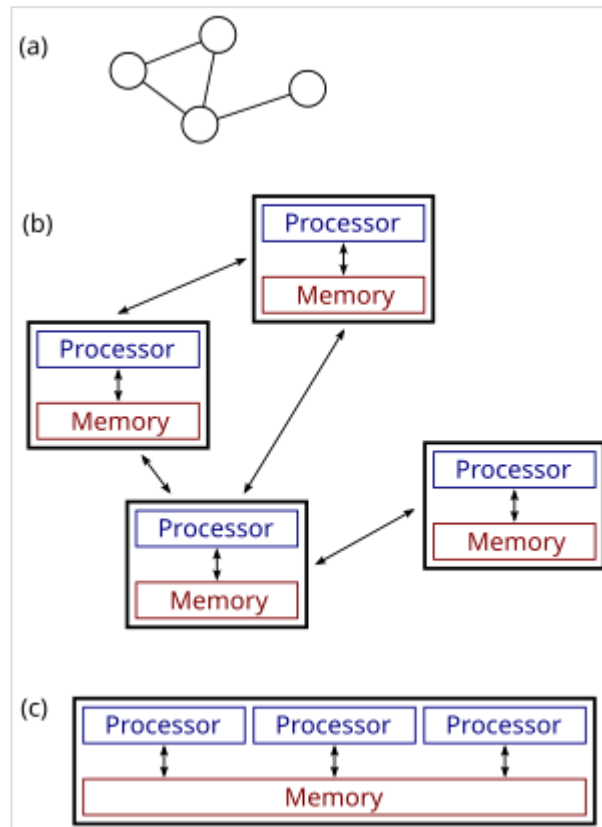
Here are common architectural patterns used for distributed computing:^[21]

- Saga interaction pattern
- Microservices
- Event driven architecture

Parallel and distributed computing

Distributed systems are groups of networked computers which share a common goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have much overlap, and no clear distinction exists between them.^[22] The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel.^[23] Parallel computing may be seen as a particularly tightly coupled form of distributed computing,^[24] and distributed computing may be seen as a loosely coupled form of parallel computing.^[13] Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

- In parallel computing, all processors may have access to a shared memory to exchange information between processors.^[25]
- In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.^[26]



(a), (b): a distributed system.

(c): a parallel system.

The figure on the right illustrates the difference between distributed and parallel systems. Figure (a) is a schematic view of a typical distributed system; the system is represented as a network topology in which each node is a computer and each line connecting the nodes is a communication link. Figure (b) shows the same distributed system in more detail: each computer has its own local memory, and information can be exchanged only by passing messages from one node to another by using the available communication links. Figure (c) shows a parallel system in which each processor has a direct access to a shared memory.

The situation is further complicated by the traditional uses of the terms parallel and distributed *algorithm* that do not quite match the above definitions of parallel and distributed *systems* (see [below](#) for more detailed discussion). Nevertheless, as a rule of thumb, high-performance parallel computation in a shared-memory multiprocessor uses parallel algorithms while the coordination of a large-scale distributed system uses distributed algorithms.^[27]

History

The use of concurrent processes which communicate through message-passing has its roots in [operating system](#) architectures studied in the 1960s.^[28] The first widespread distributed systems were [local-area networks](#) such as [Ethernet](#), which was invented in the 1970s.^[29]

[ARPANET](#), one of the predecessors of the [Internet](#), was introduced in the late 1960s, and [ARPANET e-mail](#) was invented in the early 1970s. E-mail became the most successful application of [ARPANET](#),^[30] and it is probably the earliest example of a large-scale [distributed application](#). In addition to [ARPANET](#) (and its successor, the global Internet), other early worldwide computer networks included [Usenet](#) and [FidoNet](#) from the 1980s, both of which were used to support distributed discussion systems.^[31]

The study of distributed computing became its own branch of computer science in the late 1970s and early 1980s. The first conference in the field, [Symposium on Principles of Distributed Computing](#) (PODC), dates back to 1982, and its counterpart [International Symposium on Distributed Computing](#) (DISC) was first held in Ottawa in 1985 as the International Workshop on Distributed Algorithms on Graphs.^[32]

Architectures

Various hardware and software architectures are used for distributed computing. At a lower level, it is necessary to interconnect multiple CPUs with some sort of network, regardless of whether that network is printed onto a circuit board or made up of loosely coupled devices and cables. At a higher level, it is necessary to interconnect [processes](#) running on those CPUs with some sort of [communication system](#).^[33]

Whether these CPUs share resources or not determines a first distinction between three types of architecture:

- [Shared memory](#)
- [Shared disk](#)
- [Shared nothing](#).

Distributed programming typically falls into one of several basic architectures: [client-server](#), [three-tier](#), [n-tier](#), or [peer-to-peer](#); or categories: [loose coupling](#), or [tight coupling](#).^[34]

- Client-server: architectures where smart clients contact the server for data then format and display it to the users. Input at the client is committed back to the server when it represents a permanent change.
- Three-tier: architectures that move the client intelligence to a middle tier so that stateless clients can be used. This simplifies application deployment. Most web applications are three-tier.
- n-tier: architectures that refer typically to web applications which further forward their requests to other enterprise services. This type of application is the one most responsible for the success of application servers.
- Peer-to-peer: architectures where there are no special machines that provide a service or manage the network resources.^{[35]:227} Instead all responsibilities are uniformly divided among all machines, known as peers. Peers can serve both as clients and as servers.^[36] Examples of this architecture include BitTorrent and the bitcoin network.

Another basic aspect of distributed computing architecture is the method of communicating and coordinating work among concurrent processes. Through various message passing protocols, processes may communicate directly with one another, typically in a main/sub relationship. Alternatively, a "database-centric" architecture can enable distributed computing to be done without any form of direct inter-process communication, by utilizing a shared database.^[37] Database-centric architecture in particular provides relational processing analytics in a schematic architecture allowing for live environment relay. This enables distributed computing functions both within and beyond the parameters of a networked database.^[38]

Applications

Reasons for using distributed systems and distributed computing may include:

- The very nature of an application may *require* the use of a communication network that connects several computers: for example, data produced in one physical location and required in another location.
- There are many cases in which the use of a single computer would be possible in principle, but the use of a distributed system is *beneficial* for practical reasons. For example:
 - It can allow for much larger storage and memory, faster compute, and higher bandwidth than a single machine.
 - It can provide more reliability than a non-distributed system, as there is no single point of failure. Moreover, a distributed system may be easier to expand and manage than a monolithic uniprocessor system.^[39]
 - It may be more cost-efficient to obtain the desired level of performance by using a cluster of several low-end computers, in comparison with a single high-end computer.

Examples

Examples of distributed systems and applications of distributed computing include the following:^[40]

- telecommunications networks:
 - telephone networks and cellular networks,
 - computer networks such as the Internet,
 - wireless sensor networks,

- routing algorithms;
- network applications:
 - World Wide Web and peer-to-peer networks,
 - massively multiplayer online games and virtual reality communities,
 - distributed databases and distributed database management systems,
 - network file systems,
 - distributed cache such as burst buffers,
 - distributed information processing systems such as banking systems and airline reservation systems;
- real-time process control:
 - aircraft control systems,
 - industrial control systems;
- parallel computation:
 - scientific computing, including cluster computing, grid computing, cloud computing,^[41] and various volunteer computing projects,
 - distributed rendering in computer graphics.
- peer-to-peer

Reactive distributed systems

According to Reactive Manifesto, reactive distributed systems are responsive, resilient, elastic and message-driven. Subsequently, Reactive systems are more flexible, loosely-coupled and scalable. To make your systems reactive, you are advised to implement Reactive Principles. Reactive Principles are a set of principles and patterns which help to make your cloud native application as well as edge native applications more reactive. ^[42]

Theoretical foundations

Models

Many tasks that we would like to automate by using a computer are of question–answer type: we would like to ask a question and the computer should produce an answer. In theoretical computer science, such tasks are called computational problems. Formally, a computational problem consists of *instances* together with a *solution* for each instance. Instances are questions that we can ask, and solutions are desired answers to these questions.

Theoretical computer science seeks to understand which computational problems can be solved by using a computer (computability theory) and how efficiently (computational complexity theory). Traditionally, it is said that a problem can be solved by using a computer if we can design an algorithm that produces a correct solution for any given instance. Such an algorithm can be implemented as a computer program that runs on a general-purpose computer: the program reads a problem instance from input, performs

some computation, and produces the solution as output. Formalisms such as random-access machines or universal Turing machines can be used as abstract models of a sequential general-purpose computer executing such an algorithm.^{[43][44]}

The field of concurrent and distributed computing studies similar questions in the case of either multiple computers, or a computer that executes a network of interacting processes: which computational problems can be solved in such a network and how efficiently? However, it is not at all obvious what is meant by "solving a problem" in the case of a concurrent or distributed system: for example, what is the task of the algorithm designer, and what is the concurrent or distributed equivalent of a sequential general-purpose computer?

The discussion below focuses on the case of multiple computers, although many of the issues are the same for concurrent processes running on a single computer.

Three viewpoints are commonly used:

Parallel algorithms in shared-memory model

- All processors have access to a shared memory. The algorithm designer chooses the program executed by each processor.
- One theoretical model is the parallel random-access machines (PRAM) that are used.^[45] However, the classical PRAM model assumes synchronous access to the shared memory.
- Shared-memory programs can be extended to distributed systems if the underlying operating system encapsulates the communication between nodes and virtually unifies the memory across all individual systems.
- A model that is closer to the behavior of real-world multiprocessor machines and takes into account the use of machine instructions, such as Compare-and-swap (CAS), is that of *asynchronous shared memory*. There is a wide body of work on this model, a summary of which can be found in the literature.^{[46][47]}

Parallel algorithms in message-passing model

- The algorithm designer chooses the structure of the network, as well as the program executed by each computer.
- Models such as Boolean circuits and sorting networks are used.^[48] A Boolean circuit can be seen as a computer network: each gate is a computer that runs an extremely simple computer program. Similarly, a sorting network can be seen as a computer network: each comparator is a computer.

Distributed algorithms in message-passing model

- The algorithm designer only chooses the computer program. All computers run the same program. The system must work correctly regardless of the structure of the network.
- A commonly used model is a graph with one finite-state machine per node.

In the case of distributed algorithms, computational problems are typically related to graphs. Often the graph that describes the structure of the computer network is the problem instance. This is illustrated in the following example.^[49]

An example

Consider the computational problem of finding a coloring of a given graph G . Different fields might take the following approaches:

Centralized algorithms^[49]

- The graph G is encoded as a string, and the string is given as input to a computer. The computer program finds a coloring of the graph, encodes the coloring as a string, and outputs the result.

Parallel algorithms

- Again, the graph G is encoded as a string. However, multiple computers can access the same string in parallel. Each computer might focus on one part of the graph and produce a coloring for that part.
- The main focus is on high-performance computation that exploits the processing power of multiple computers in parallel.

Distributed algorithms

- The graph G is the structure of the computer network. There is one computer for each node of G and one communication link for each edge of G . Initially, each computer only knows about its immediate neighbors in the graph G ; the computers must exchange messages with each other to discover more about the structure of G . Each computer must produce its own color as output.
- The main focus is on coordinating the operation of an arbitrary distributed system.^[49]

While the field of parallel algorithms has a different focus than the field of distributed algorithms, there is much interaction between the two fields. For example, the Cole–Vishkin algorithm for graph coloring^[50] was originally presented as a parallel algorithm, but the same technique can also be used directly as a distributed algorithm.

Moreover, a parallel algorithm can be implemented either in a parallel system (using shared memory) or in a distributed system (using message passing).^[51] The traditional boundary between parallel and distributed algorithms (choose a suitable network vs. run in any given network) does not lie in the same place as the boundary between parallel and distributed systems (shared memory vs. message passing).

Complexity measures

In parallel algorithms, yet another resource in addition to time and space is the number of computers. Indeed, often there is a trade-off between the running time and the number of computers: the problem can be solved faster if there are more computers running in parallel (see speedup). If a decision problem can be solved in polylogarithmic time by using a polynomial number of processors, then the problem is said to be in the class NC.^[52] The class NC can be defined equally well by using the PRAM formalism or Boolean circuits—PRAM machines can simulate Boolean circuits efficiently and vice versa.^[53]

In the analysis of distributed algorithms, more attention is usually paid on communication operations than computational steps. Perhaps the simplest model of distributed computing is a synchronous system where all nodes operate in a lockstep fashion. This model is commonly known as the LOCAL model. During each *communication round*, all nodes in parallel (1) receive the latest messages from their neighbours,

(2) perform arbitrary local computation, and (3) send new messages to their neighbors. In such systems, a central complexity measure is the number of synchronous communication rounds required to complete the task.^[54]

This complexity measure is closely related to the diameter of the network. Let D be the diameter of the network. On the one hand, any computable problem can be solved trivially in a synchronous distributed system in approximately $2D$ communication rounds: simply gather all information in one location (D rounds), solve the problem, and inform each node about the solution (D rounds).

On the other hand, if the running time of the algorithm is much smaller than D communication rounds, then the nodes in the network must produce their output without having the possibility to obtain information about distant parts of the network. In other words, the nodes must make globally consistent decisions based on information that is available in their *local D -neighbourhood*. Many distributed algorithms are known with the running time much smaller than D rounds, and understanding which problems can be solved by such algorithms is one of the central research questions of the field.^[55] Typically an algorithm which solves a problem in polylogarithmic time in the network size is considered efficient in this model.

Another commonly used measure is the total number of bits transmitted in the network (cf. communication complexity).^[56] The features of this concept are typically captured with the CONGEST(B) model, which is similarly defined as the LOCAL model, but where single messages can only contain B bits.

Other problems

Traditional computational problems take the perspective that the user asks a question, a computer (or a distributed system) processes the question, then produces an answer and stops. However, there are also problems where the system is required not to stop, including the dining philosophers problem and other similar mutual exclusion problems. In these problems, the distributed system is supposed to continuously coordinate the use of shared resources so that no conflicts or deadlocks occur.

There are also fundamental challenges that are unique to distributed computing, for example those related to *fault-tolerance*. Examples of related problems include consensus problems,^[57] Byzantine fault tolerance,^[58] and self-stabilisation.^[59]

Much research is also focused on understanding the *asynchronous* nature of distributed systems:

- Synchronizers can be used to run synchronous algorithms in asynchronous systems.^[60]
- Logical clocks provide a causal happened-before ordering of events.^[61]
- Clock synchronization algorithms provide globally consistent physical time stamps.^[62]

Note that in distributed systems, latency should be measured through "99th percentile" because "median" and "average" can be misleading.^[63]

Election

Coordinator election (or *leader election*) is the process of designating a single process as the organizer of some task distributed among several computers (nodes). Before the task is begun, all network nodes are either unaware which node will serve as the "coordinator" (or leader) of the task, or unable to

communicate with the current coordinator. After a coordinator election algorithm has been run, however, each node throughout the network recognizes a particular, unique node as the task coordinator.^[64]

The network nodes communicate among themselves in order to decide which of them will get into the "coordinator" state. For that, they need some method in order to break the symmetry among them. For example, if each node has unique and comparable identities, then the nodes can compare their identities, and decide that the node with the highest identity is the coordinator.^[64]

The definition of this problem is often attributed to LeLann, who formalized it as a method to create a new token in a token ring network in which the token has been lost.^[65]

Coordinator election algorithms are designed to be economical in terms of total bytes transmitted, and time. The algorithm suggested by Gallager, Humblet, and Spira^[66] for general undirected graphs has had a strong impact on the design of distributed algorithms in general, and won the Dijkstra Prize for an influential paper in distributed computing.

Many other algorithms were suggested for different kinds of network graphs, such as undirected rings, unidirectional rings, complete graphs, grids, directed Euler graphs, and others. A general method that decouples the issue of the graph family from the design of the coordinator election algorithm was suggested by Korach, Kutten, and Moran.^[67]

In order to perform coordination, distributed systems employ the concept of coordinators. The coordinator election problem is to choose a process from among a group of processes on different processors in a distributed system to act as the central coordinator. Several central coordinator election algorithms exist.^[68]

Properties of distributed systems

So far the focus has been on *designing* a distributed system that solves a given problem. A complementary research problem is *studying* the properties of a given distributed system.^{[69][70]}

The halting problem is an analogous example from the field of centralised computation: we are given a computer program and the task is to decide whether it halts or runs forever. The halting problem is undecidable in the general case, and naturally understanding the behaviour of a computer network is at least as hard as understanding the behaviour of one computer.^[71]

However, there are many interesting special cases that are decidable. In particular, it is possible to reason about the behaviour of a network of finite-state machines. One example is telling whether a given network of interacting (asynchronous and non-deterministic) finite-state machines can reach a deadlock. This problem is PSPACE-complete,^[72] i.e., it is decidable, but not likely that there is an efficient (centralised, parallel or distributed) algorithm that solves the problem in the case of large networks.

See also

- | | |
|----------------------|----------------------------------|
| ▪ <u>Actor model</u> | ▪ <u>Code mobility</u> |
| ▪ <u>AppScale</u> | ▪ <u>Dataflow programming</u> |
| ▪ <u>BOINC</u> | ▪ <u>Decentralized computing</u> |

- Distributed algorithm
- Distributed algorithmic mechanism design
- Distributed cache
- Distributed GIS
- Distributed networking
- Distributed operating system
- Eventual consistency
- Edsger W. Dijkstra Prize in Distributed Computing
- Federation (information technology)
- Flat neighborhood network
- Fog computing
- Folding@home
- Grid computing
- Inferno
- Internet GIS
- Jungle computing
- Layered queueing network
- Library Oriented Architecture (LOA)
- List of distributed computing conferences
- List of volunteer computing projects
- Model checking
- OpenHarmony
- HarmonyOS
- Parallel distributed processing
- Parallel programming model
- Plan 9 from Bell Labs
- Shared nothing architecture
- Web GIS

Notes

1. Tanenbaum, Andrew S.; Steen, Maarten van (2002). *Distributed systems: principles and paradigms* (<https://www.distributed-systems.net/index.php/books/ds3/>). Upper Saddle River, NJ: Pearson Prentice Hall. ISBN 0-13-088893-1. Archived (<https://web.archive.org/web/20200812174339/https://www.distributed-systems.net/index.php/books/ds3/>) from the original on 2020-08-12. Retrieved 2020-08-28.
2. "Distributed Programs". *Texts in Computer Science*. London: Springer London. 2010. pp. 373–406. doi:10.1007/978-1-84882-745-5_11 (https://doi.org/10.1007%2F978-1-84882-745-5_11). ISBN 978-1-84882-744-8. ISSN 1868-0941 (<https://search.worldcat.org/issn/1868-0941>). "Systems consist of a number of physically distributed components that work independently using their private storage, but also communicate from time to time by explicit message passing. Such systems are called distributed systems."
3. Dusseau & Dusseau 2016, p. 1–2.
4. Ford, Neal (March 3, 2020). *Fundamentals of Software Architecture: An Engineering Approach* (1st ed.). O'Reilly Media. pp. 146–147. ISBN 978-1492043454.
5. *Monolith to Microservices Evolutionary Patterns to Transform Your Monolith*. O'Reilly Media. ISBN 9781492047810.
6. *Building Serverless Applications on Knative*. O'Reilly Media. ISBN 9781098142049.
7. "Distributed Programs". *Texts in Computer Science*. London: Springer London. 2010. pp. 373–406. doi:10.1007/978-1-84882-745-5_11 (https://doi.org/10.1007%2F978-1-84882-745-5_11). ISBN 978-1-84882-744-8. ISSN 1868-0941 (<https://search.worldcat.org/issn/1868-0941>). "Distributed programs are abstract descriptions of distributed systems. A distributed program consists of a collection of processes that work concurrently and communicate by explicit message passing. Each process can access a set of variables which are disjoint from the variables that can be changed by any other process."
8. Andrews (2000). Dolev (2000). Ghosh (2007), p. 10.
9. Magnoni, L. (2015). "Modern Messaging for Distributed Sytems (sic)" (<https://doi.org/10.1088%2F1742-6596%2F608%2F1%2F012038>). *Journal of Physics: Conference Series*. **608** (1): 012038. doi:10.1088/1742-6596/608/1/012038 (<https://doi.org/10.1088%2F1742-6596%2F608%2F1%2F012038>). ISSN 1742-6596 (<https://search.worldcat.org/issn/1742-6596>).
10. Godfrey (2002).
11. Andrews (2000), p. 291–292. Dolev (2000), p. 5.