Deep learning is the subset of machine learning methods based on artificial neural networks (ANNs) with representation learning. The adjective "deep" refers to the use of multiple layers in the network. Methods used can be either supervised, semi-supervised or unsupervised.[2]

Deep-learning architectures such as deep neural networks, deep belief networks, recurrent neural networks, convolutional neural networks and transformers 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]

Artificial neural networks were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, artificial neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analog.[6][7] ANNs are generally seen as low quality models for brain function.[8]

Definition

Deep learning is a class of machine learning algorithms that[9]:199–200 uses multiple layers to progressively extract higher-level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

From another angle to view deep learning, deep learning refers to "computer-simulate" or "automate" human learning processes from a source (e.g., an image of dogs) to a learned object (dogs). Therefore, a notion coined as "deeper" learning or "deepest" learning[10] makes sense. The deepest learning refers to the fully automatic learning from a source to a final learned object. A deeper learning thus refers to a mixed learning process: a human learning process from a source to a learned semi-object, followed by a computer learning process from the human learned semi-object to a final learned object.

Overview

Most modern deep learning models are based on multi-layered artificial 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.[11]

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; 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. Importantly, a deep learning process can learn which features to optimally place in which level on its own.

This does not eliminate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.[12][13]

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.[14] 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 2. CAP of depth 2 has been shown to be a universal approximator in the sense that it can emulate any function.[15] Beyond that, more layers do not add to the function approximator ability of the network. Deep models (CAP > 2) 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.[16] Deep learning helps to disentangle these abstractions and pick out which features improve performance.[12]

For supervised learning tasks, deep learning methods enable elimination of feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures that remove redundancy in representation.

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.[12][17]

Machine learning models are now adept at identifying complex patterns in financial market data. Due to the benefits of artificial intelligence, investors are increasingly utilizing deep learning techniques to forecast and analyze trends in stock and foreign exchange markets.[18]

Interpretations

Deep neural networks are generally interpreted in terms of the universal approximation theorem[19][20][21][22][23] or probabilistic inference.[24][9][12][14][25]

The classic universal approximation theorem concerns the capacity of feedforward neural networks with a single hidden layer of finite size to approximate continuous functions.[19][20][21][22] In 1989, the first proof was published by George Cybenko for sigmoid activation functions[19] and was generalised to feed-forward multi-layer architectures in 1991 by Kurt Hornik.[20] Recent work also showed that universal approximation also holds for non-bounded activation functions such as Kunihiko Fukushima's

rectified linear unit.[26][27]

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.[23] 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[25] derives from the field of machine learning. It features inference,[9][11][12][14][17][25] 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.[25] 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.[28]

History

There are two types of artificial neural network (ANN): feedforward neural networks (FNNs) and recurrent neural networks (RNNs). RNNs have cycles in their connectivity structure, FNNs don't. In the 1920s, Wilhelm Lenz and Ernst Ising created and analyzed the Ising model[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 popularised by John Hopfield in 1982.[32] RNNs have become central for speech recognition and language processing.

Charles Tappert writes that Frank Rosenblatt developed and explored all of the basic ingredients of the deep learning systems of today,[33] referring to Rosenblatt's 1962 book[34] which introduced multilayer perceptron (MLP) with 3 layers: an input layer, a hidden layer with randomized weights that did not learn, and an output layer. It also introduced variants, including a version with four-layer perceptrons where the last two layers have learned weights (and thus a proper multilayer perceptron).[34]: section 16 In addition, term deep learning was proposed in 1986 by Rina Dechter[35] although the history of its appearance is apparently more complicated.[36]

The first general, working learning algorithm for supervised, deep, feedforward, multilayer perceptrons was published by Alexey Ivakhnenko and Lapa in 1967.[37] A 1971 paper described a deep network with eight layers trained by the group method of data handling.[38]

The first deep learning multilayer perceptron trained by stochastic gradient descent[39] was published in 1967 by Shun'ichi Amari.[40][31] In computer experiments conducted by Amari's student Saito, a five layer MLP with two modifiable layers learned internal representations to classify non-linearily separable pattern classes.[31] In 1987 Matthew Brand reported that wide

12-layer nonlinear perceptrons could be fully end-to-end trained to reproduce logic functions of nontrivial circuit depth via gradient descent on small batches of random input/output samples, but concluded that training time on contemporary hardware (sub-megaflop computers) made the technique impractical, and proposed using fixed random early layers as an input hash for a single modifiable layer.[41] Instead, subsequent developments in hardware and hyperparameter tunings have made end-to-end stochastic gradient descent the currently dominant training technique.

In 1970, Seppo Linnainmaa published the reverse mode of automatic differentiation of discrete connected networks of nested differentiable functions.[42][43][44] This became known as backpropagation.[14] It is an efficient application of the chain rule derived by Gottfried Wilhelm Leibniz in 1673[45] to networks of differentiable nodes.[31] The terminology "back-propagating errors" was actually introduced in 1962 by Rosenblatt,[34][31] but he did not know how to implement this, although Henry J. Kelley had a continuous precursor of backpropagation[46] already in 1960 in the context of control theory.[31] In 1982, Paul Werbos applied backpropagation to MLPs in the way that has become standard.[47][48][31] In 1985, David E. Rumelhart et al. published an experimental analysis of the technique.[49]

Deep learning architectures for convolutional neural networks (CNNs) with convolutional layers and downsampling layers began with the Neocognitron introduced by Kunihiko Fukushima in 1980.[50] In 1969, he also introduced the ReLU (rectified linear unit) activation function.[26][31] The rectifier has become the most popular activation function for CNNs and deep learning in general.[51] CNNs have become an essential tool for computer vision.

The term Deep Learning was introduced to the machine learning community by Rina Dechter in 1986,[35] and to artificial neural networks by Igor Aizenberg and colleagues in 2000, in the context of Boolean threshold neurons.[52][53]

In 1988, Wei Zhang et al. applied the backpropagation algorithm to a convolutional neural network (a simplified Neocognitron with convolutional interconnections between the image feature layers and the last fully connected layer) for alphabet recognition. They also proposed an implementation of the CNN with an optical computing system.[54][55] In 1989, Yann LeCun et al. applied backpropagation to a CNN with the purpose of recognizing handwritten ZIP codes on mail. While the algorithm worked, training required 3 days.[56] Subsequently, Wei Zhang, et al. modified their model by removing the last fully connected layer and applied it for medical image object segmentation in 1991[57] and breast cancer detection in mammograms in 1994.[58] LeNet-5 (1998), a 7-level CNN by Yann LeCun et al.,[59] that classifies digits, was applied by several banks to recognize hand-written numbers on checks digitized in 32x32 pixel images.

In the 1980s, backpropagation did not work well for deep learning with long credit assignment paths. To overcome this problem, Jürgen Schmidhuber (1992) proposed a hierarchy of RNNs pre-trained one level at a time by self-supervised learning.[60] It 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.[60][31] In 1993, a chunker solved a deep learning task whose depth exceeded 1000.[61]

In 1992, Jürgen Schmidhuber also published an alternative to RNNs[62] which is now called a linear Transformer or a Transformer with linearized self-attention[63][64][31] (save for a normalization operator). It learns internal spotlights of attention:[65] a slow feedforward neural network learns by gradient descent to control the fast weights of another neural network through outer products of self-generated activation patterns FROM and TO (which are now called key and value for self-attention).[63] This fast weight attention mapping is applied to a query pattern.

The modern Transformer was introduced by Ashish Vaswani et al. in their 2017 paper "Attention Is All You Need".[66] It combines this with a softmax operator and a projection matrix.[31] Transformers have increasingly become the model of choice for natural language processing.[67] Many modern large language models such as ChatGPT, GPT-4, and BERT use it. Transformers are also increasingly being used in computer vision.[68]

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.[69][70][71] 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 a generative adversarial network (GAN) by Ian Goodfellow et al.[72] Here the environmental reaction is 1 or 0 depending on whether the first network's output is in a given set. This can be used to create realistic deepfakes.[73] Excellent image quality is achieved by Nvidia's StyleGAN (2018)[74] based on the Progressive GAN by Tero Karras et al.[75] Here the GAN generator is grown from small to large scale in a pyramidal fashion.

Sepp Hochreiter's diploma thesis (1991)[76] was called "one of the most important documents in the history of machine learning" by his supervisor Schmidhuber.[31] It not only tested the neural history compressor,[60] but also identified and analyzed the vanishing gradient problem.[76][77] Hochreiter proposed recurrent residual connections to solve this problem. This led to the deep learning method called long short-term memory (LSTM), published in 1997.[78] LSTM recurrent neural networks can learn "very deep learning" tasks[14] with long credit assignment paths that require memories of events that happened thousands of discrete time steps before. The "vanilla LSTM" with forget gate was introduced in 1999 by Felix Gers, Schmidhuber and Fred Cummins.[79] LSTM has become the most cited neural network of the 20th century.[31] In 2015, Rupesh Kumar Srivastava, Klaus Greff, and Schmidhuber used LSTM principles to create the Highway network, a feedforward neural network with hundreds of layers, much deeper than previous networks.[80][81] 7 months later, Kaiming He, Xiangyu Zhang; Shaoqing Ren, and Jian Sun won the ImageNet 2015 competition with an open-gated or gateless Highway network variant called Residual neural network.[82] This has become the

most cited neural network of the 21st century.[31]

In 1994, André de Carvalho, together with Mike Fairhurst and David Bisset, published experimental results of a multi-layer boolean neural network, also known as a weightless neural network, composed of a 3-layers self-organising feature extraction neural network module (SOFT) followed by a multi-layer classification neural network module (GSN), which were independently trained. Each layer in the feature extraction module extracted features with growing complexity regarding the previous layer.[83]

In 1995, Brendan Frey demonstrated that it was possible to train (over two days) a network containing six fully connected layers and several hundred hidden units using the wake-sleep algorithm, co-developed with Peter Dayan and Hinton.[84]

Since 1997, Sven Behnke extended the feed-forward hierarchical convolutional approach in the Neural Abstraction Pyramid[85] by lateral and backward connections in order to flexibly incorporate context into decisions and iteratively resolve local ambiguities.

Simpler models that use task-specific handcrafted features such as Gabor filters and support vector machines (SVMs) were a popular choice 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.

Both shallow and deep learning (e.g., recurrent nets) of ANNs for speech recognition have been explored for many years.[86][87][88] 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.[89] Key difficulties have been analyzed, including gradient diminishing[76] and weak temporal correlation structure in neural predictive models.[90][91] 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 studied deep neural networks (DNNs) 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 National Institute of Standards and Technology Speaker Recognition evaluation.[92] The SRI deep neural network was then deployed in the Nuance Verifier, representing the first major industrial application of deep learning.[93] 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,[93] 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.[94]

Speech recognition was taken over by LSTM. In 2003, LSTM started to become competitive with traditional speech recognizers on certain tasks.[95] In 2006, Alex Graves, Santiago

Fernández, Faustino Gomez, and Schmidhuber combined it with connectionist temporal classification (CTC)[96] in stacks of LSTM RNNs.[97] In 2015, Google's speech recognition reportedly experienced a dramatic performance jump of 49% through CTC-trained LSTM, which they made available through Google Voice Search.[98]

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.[99] Industrial applications of deep learning to large-scale speech recognition started around 2010.

In 2006, publications by Geoff Hinton, Ruslan Salakhutdinov, Osindero and Teh[100][101][102] showed how a many-layered feedforward neural network could be effectively pre-trained one layer at a time, treating each layer in turn as an unsupervised restricted Boltzmann machine, then fine-tuning it using supervised backpropagation.[103] The papers referred to learning for deep belief nets.

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.[9][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 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][112] Convolutional neural networks were superseded for ASR by CTC[96] for LSTM.[78][98][113][114][115] but are more successful in computer vision.

Advances in hardware have driven renewed interest in deep learning. In 2009, Nvidia was

involved in what was called the "big bang" of deep learning, "as deep-learning neural networks were trained with Nvidia graphics processing units (GPUs)".[116] That year, Andrew Ng determined that GPUs could increase the speed of deep-learning systems by about 100 times.[117] In particular, GPUs are well-suited for the matrix/vector computations involved in machine learning.[118][119][120] GPUs speed up training algorithms by orders of magnitude, reducing running times from weeks to days.[121][122] Further, specialized hardware and algorithm optimizations can be used for efficient processing of deep learning models.[123]

Deep learning revolution

How deep learning is a subset of machine learning and how machine learning is a subset of artificial intelligence (AI)
In the late 2000s, deep learning started to outperform other methods in machine learning competitions. In 2009, a long short-term memory trained by connectionist temporal classification (Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, 2006)[96] was the first RNN to win pattern recognition contests, winning three competitions in connected handwriting recognition.[124][14] Google later used CTC-trained LSTM for speech recognition on the smartphone.[125][98]

Significant impacts in image or object recognition were felt from 2011 to 2012. Although CNNs trained by backpropagation had been around for decades,[54][56] and GPU implementations of NNs for years,[118] including CNNs,[120][14] faster implementations of CNNs on GPUs were needed to progress on computer vision. In 2011, the DanNet[126][3] 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.[14] Also in 2011, DanNet won the ICDAR Chinese handwriting contest, and in May 2012, it won the ISBI image segmentation contest.[127] Until 2011, CNNs did not play a major role at computer vision conferences, but in June 2012, a paper by Ciresan et al. at the leading conference CVPR[3] showed how max-pooling CNNs on GPU can dramatically improve many vision benchmark records. In September 2012, DanNet also won the ICPR contest on analysis of large medical images for cancer detection, and in the following year also the MICCAI Grand Challenge on the same topic.[128] In October 2012, the similar 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. The VGG-16 network by Karen Simonyan and Andrew Zisserman[129] further reduced the error rate and won the ImageNet 2014 competition, following a similar trend in large-scale speech recognition.

Image classification was then extended to the more challenging task of generating descriptions (captions) for images, often as a combination of CNNs and LSTMs.[130][131][132]

In 2012, a team led by George E. Dahl won the "Merck Molecular Activity Challenge" using multi-task deep neural networks to predict the biomolecular target of one drug.[133][134] In 2014, Sepp Hochreiter's group used deep learning to detect off-target and toxic effects of

environmental chemicals in nutrients, household products and drugs and won the "Tox21 Data Challenge" of NIH, FDA and NCATS.[135][136][137]

In 2016, Roger Parloff mentioned a "deep learning revolution" that has transformed the AI industry.[138]

In March 2019, Yoshua Bengio, Geoffrey Hinton and Yann LeCun were awarded the Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing.

Neural networks
Main article: Artificial neural network

Simplified example of training a neural network in object detection: The network is trained by multiple images that are known to depict starfish and sea urchins, which are correlated with "nodes" that represent visual features. The starfish match with a ringed texture and a star outline, whereas most sea urchins match with a striped texture and oval shape. However, the instance of a ring textured sea urchin creates a weakly weighted association between them.

Subsequent run of the network on an input image (left):[139] The network correctly detects the starfish. However, the weakly weighted association between ringed texture and sea urchin also confers a weak signal to the latter from one of two intermediate nodes. In addition, a shell that was not included in the training gives a weak signal for the oval shape, also resulting in a weak signal for the sea urchin output. These weak signals may result in a false positive result for sea urchin.
In reality, textures and outlines would not be represented by single nodes, but rather by associated weight patterns of multiple nodes.
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"[140]).

Deep neural networks

A deep neural network (DNN) is an artificial neural network with multiple layers between the input and output layers.[11][14] There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions.[141] 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.[citation needed]

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,[citation needed] 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.[142] The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.[11] For instance, it was proved that sparse multivariate polynomials are exponentially easier to approximate with DNNs than with shallow networks.[143]

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.[144] 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.[145][146][147][148][149] Long short-term memory is particularly effective for this use.[78][150]

Convolutional neural networks (CNNs) are used in computer vision.[151] CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).[152]

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[38] or weight decay (
 can be applied during training to combat overfitting.[153] Alternatively dropout regularization randomly omits units from the hidden layers during training. This helps to exclude rare dependencies.[154] 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.[155]

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)[156] 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.[157][158]

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.[159][160]

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.[161] By 2019, graphic processing units (GPUs), often with AI-specific enhancements, had displaced CPUs as the dominant method of training large-scale commercial cloud AI.[162] 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.[163][164]

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[165] and cloud computing servers such as tensor processing units (TPU) in the Google Cloud Platform.[166] 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).[167][168]

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).[169]

In 2021, J. Feldmann et al. proposed an integrated photonic hardware accelerator for parallel convolutional processing.[170] 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.[170] Their system can execute trillions of multiply-accumulate operations per second, indicating the potential of integrated photonics in data-heavy AI applications.[170]