Jump to content

□ Main menu

Main menu

move to sidebar hide

Navigation

- Main page
- Contents
- Current events
- Random article
- About Wikipedia
- Contact us

Contribute

- Help
- Learn to edit
- Community portal
- Recent changes
- Upload file
- Special pages

Wikipedia The Free Encyclopedia

Search

Search Wikipedia

Search

□ Appearance

- Donate
- Create account
- Log in

□ Personal tools

- Donate
- Create account
- Log in

Pages for logged out editors learn more

- Contributions
- Talk

Contents

move to sidebar hide

- (Top)
- 1 Overview
- 2 Interpretations

- 3 History
  Toggle History subsection
  - 3.1 Before 1980
  - 3.2 1980s-2000s
  - 3.3 2000s
  - 3.4 Deep learning revolution
- 4 Neural networks
  Toggle Neural networks subsection
  - 4.1 Deep neural networks
    - 4.1.1 Challenges
- 5 Hardware
- 6 Applications
  Toggle Applications subsection
  - 6.1 Automatic speech recognition
  - 6.2 Image recognition
  - 6.3 Visual art processing
  - 6.4 Natural language processing
  - 6.5 Drug discovery and toxicology
  - 6.6 Customer relationship management
  - 6.7 Recommendation systems
  - 6.8 Bioinformatics
  - 6.9 Deep Neural Network Estimations
  - 6.10 Medical image analysis
  - 6.11 Mobile advertising
  - 6.12 Image restoration
  - 6.13 Financial fraud detection
  - 6.14 Materials science
  - 6.15 Military
  - 6.16 Partial differential equations
  - 6.17 Deep backward stochastic differential equation method
  - 6.18 Image reconstruction
  - 6.19 Weather prediction
  - 6.20 Epigenetic clock
- 7 Relation to human cognitive and brain development
- 8 Commercial activity
- 9 Criticism and comment
  Toggle Criticism and comment subsection
  - 9.1 Theory
  - 9.2 Errors
  - 9.3 Cyber threat
  - 9.4 Data collection ethics
- 10 See also
- 11 References
- 12 Further reading

☐ Toggle the table of contents

Deep learning

☐ 58 languages

- Afrikaans

- العربية
- Azərbaycanca
- বাংলা
- 閩南語 / Bân-lâm-gú
- Български
- Bosanski
- Català
- Čeština
- Dansk
- Deutsch
- Eesti
- Ελληνικά
- Español
- Esperanto
- Euskara
- فارسی
- Français
- Gaeilge
- Galego
- 한국어
- Հայերեն
- हिन्दी
- Bahasa Indonesia
- Italiano
- עברית
- Magyar
- മലയാളം
- Bahasa Melayu
- Монгол
- Nederlands
- 日本語
- Norsk bokmål
- Occitan
- پښتو
- Polski
- Português
- Qaraqalpaqsha
- Română
- Runa Simi
- Русский
- Shqip
- සිංහල
- Simple English
- Slovenščina
- کوردی
- Српски / srpski
- Srpskohrvatski / српскохрватски
- Suomi
- Svenska
- தமிழ்
- ไทย

- Türkçe
- Українська
- اردو
- Tiếng Việt
- 粵語
- 中文

Edit links

- Article
- Talk

□ English

- Read
- Edit
- View history

□ Tools
Tools
move to sidebar hide
Actions

- Read
- Edit
- View history

General

- What links here
- Related changes
- Upload file
- Permanent link
- Page information
- Cite this page
- Get shortened URL
- Download QR code

Print/export

- Download as PDF
- Printable version

In other projects

- Wikimedia Commons
- Wikidata item

Appearance
move to sidebar hide

From Wikipedia, the free encyclopedia

Representing images on multiple layers of abstraction in deep learning

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

Deep learning is a subset of machine learning that focuses on utilizing neural networks to perform tasks such as classification, regression, and 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

[edit]

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 progressively 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.

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

[edit]

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

[edit]

Before 1980

[edit]

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-linearily 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

[edit]

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

[edit]

Neural networks entered a lull, 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.[*citation needed*]

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

[edit]

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

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).

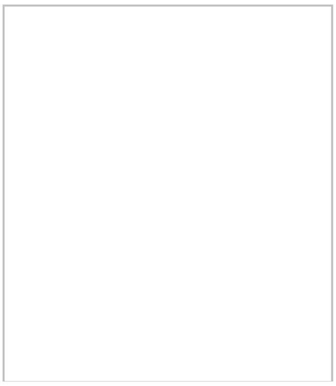
In 2015, Google's speech recognition improved by 49% by an LSTM-based model, which they made available through Google Voice Search on smartphone.[136][137]

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

[edit]



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):[143] 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"[144]).

Deep neural networks

[edit]

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.[*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, [146] 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.[147] 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.[148]

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.[146]

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.[149] 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.[150][151][152][153][154] Long short-term memory is particularly effective for this use.[155][156]

Convolutional neural networks (CNNs) are used in computer vision.[157] CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).[158]

Challenges

[edit]

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.[159] Alternatively dropout regularization randomly omits units from the hidden layers during training. This helps to exclude rare dependencies.[160] Another interesting recent development is research into models of just enough complexity through an estimation of the intrinsic complexity of the task being modelled. This approach has been successfully applied for multivariate time series prediction tasks such as traffic prediction.[161] 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.[162]

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)[163] 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.[164][165]

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.[166][167]

Hardware

[edit]

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.[168] 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 .[169] 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.[170][171]

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[172] and cloud computing servers such as tensor processing units (TPU) in the Google Cloud Platform.[173] 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).[174][175]

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).[176]

In 2021, J. Feldmann et al. proposed an integrated photonic hardware accelerator for parallel convolutional processing.[177] 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.[177] Their system can execute trillions of multiply-accumulate operations per second, indicating the potential of integrated photonics in data-heavy AI applications.[177]

Applications

[edit]

Automatic speech recognition

[edit]

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[156] 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.[178] 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[179]	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[180]	20.0
Convolutional DNN w. Heterogeneous Pooling	18.7
Ensemble DNN/CNN/RNN[181]	18.3
Bidirectional LSTM	17.8
Hierarchical Convolutional Deep Maxout Network[182]	16.5

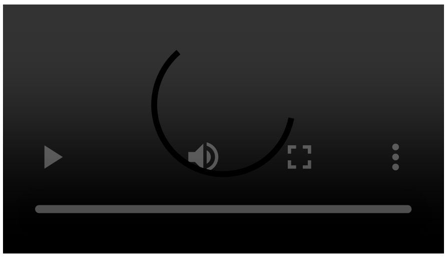
The debut of DNNs for speaker recognition in the late 1990s and speech recognition around 2009-2011 and of LSTM around 2003–2007, accelerated progress in eight major areas:[23][108][106]

- 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][183][184]

Image recognition

[edit]



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

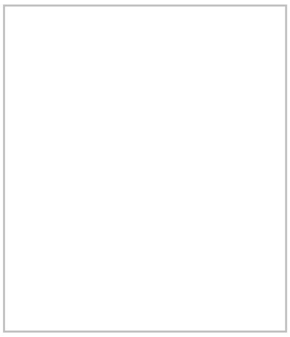
A common evaluation set for image classification is the MNIST database data set. MNIST is composed of handwritten digits and includes 60,000 training examples and 10,000 test examples. As with TIMIT, its small size lets users test multiple configurations. A comprehensive list of results on this set is available.[185]

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.[186][187]

Deep learning-trained vehicles now interpret 360° camera views.[188] 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

[edit]



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

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[189][190]
- Neural Style Transfer – capturing the style of a given artwork and applying it in a visually pleasing manner to an arbitrary photograph or video[189][190]
- generating striking imagery based on random visual input fields.[189][190]

Natural language processing

[edit]

Neural networks have been used for implementing language models since the early 2000s.[150] LSTM helped to improve machine translation and language modeling.[151][152][153]

Other key techniques in this field are negative sampling[191] 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.[192] Recursive auto-encoders built atop word embeddings can assess sentence similarity and detect paraphrasing.[192] Deep neural architectures provide the best results for constituency parsing,[193] sentiment analysis,[194] information retrieval,[195][196] spoken language understanding,[197] machine translation,[151][198] contextual entity linking,[198] writing style recognition,[199] named-entity recognition (token classification),[200] text classification, and others.[201]

Recent developments generalize word embedding to sentence embedding.

Google Translate (GT) uses a large end-to-end long short-term memory (LSTM) network.[202][203][204][205] Google Neural Machine Translation (GNMT) uses an example-based machine translation method in which the system "learns from millions of examples".[203] It translates "whole sentences at a time, rather than pieces". Google Translate supports over one hundred languages.[203] The network encodes the "semantics of the sentence rather than simply memorizing phrase-to-phrase translations".[203][206] GT uses English as an intermediate between most language pairs.[206]

Drug discovery and toxicology

[edit]

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.[207][208] Research has explored use of deep learning to predict the biomolecular targets,[209][210] off-targets, and toxic effects of environmental chemicals in nutrients, household products and drugs.[211][212][213]

AtomNet is a deep learning system for structure-based rational drug design.[214] AtomNet was used to predict novel candidate biomolecules for disease targets such as the Ebola virus[215] and multiple sclerosis.[216][215]

In 2017 graph neural networks were used for the first time to predict various properties of molecules in a large toxicology data set.[217] In 2019, generative neural networks were used to produce molecules that were validated experimentally all the way into mice.[218][219]

Customer relationship management

[edit]

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.[220]

Recommendation systems

[edit]

Recommendation systems have used deep learning to extract meaningful features for a latent factor model for content-based music and journal recommendations.[221][222] Multi-view deep learning has been applied for learning user preferences from multiple domains.[223] The model uses a hybrid collaborative and content-based approach and enhances recommendations in multiple tasks.

Bioinformatics

[edit]

An autoencoder ANN was used in bioinformatics, to predict gene ontology annotations and gene-function relationships.[224]

In medical informatics, deep learning was used to predict sleep quality based on data from wearables[225] and predictions of health complications from electronic health record data.[226]

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.[227][228]

Deep Neural Network Estimations

[edit]

Deep neural networks can be used to estimate the entropy of a stochastic process and called Neural Joint Entropy Estimator (NJEE).[229] 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.[229]

Medical image analysis

[edit]

Deep learning has been shown to produce competitive results in medical application such as cancer cell classification, lesion detection, organ segmentation and image enhancement.[230][231] 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.[232][233]

Mobile advertising

[edit]

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.[234] 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

[edit]

Deep learning has been successfully applied to inverse problems such as denoising, super-resolution, inpainting, and film colorization.[235] These applications include learning methods such as "Shrinkage Fields for Effective Image Restoration"[236] which trains on an image dataset, and Deep Image Prior, which trains on the image that needs restoration.

Financial fraud detection

[edit]

Deep learning is being successfully applied to financial fraud detection, tax evasion detection,[237] and anti-money laundering.[238]

Materials science

[edit]

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.[239][240][241]

Military

[edit]

The United States Department of Defense applied deep learning to train robots in new tasks through observation.[242]

Partial differential equations

[edit]

Physics informed neural networks have been used to solve partial differential equations in both forward and inverse problems in a data driven manner.[243] 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 rely on.[244][245]

Deep backward stochastic differential equation method

[edit]

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.[246]

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

[edit]

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 [247] and ultrasound imaging.[248]

Weather prediction

[edit]

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.[249][250]

Epigenetic clock

[edit]

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.[251] 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

[edit]

Deep learning is closely related to a class of theories of brain development (specifically, neocortical development) proposed by cognitive neuroscientists in the early 1990s.[252][253][254][255] 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".[256]

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.[257][258] 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.[259][260] In this respect, generative neural network models have been related to neurobiological evidence about sampling-based processing in the cerebral cortex.[261]

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[262] and neural populations.[263] Similarly, the representations developed by deep learning models are similar to those measured in the primate visual system[264] both at the single-unit[265] and at the population[266] levels.

Commercial activity

[edit]

Facebook's AI lab performs tasks such as automatically tagging uploaded pictures with the names of the people in them.[267]

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.[268][269][270] 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.[271]

As of 2008,[272] 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.[242] 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.[242] 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".[273]

Criticism and comment

[edit]

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

Theory

[edit]

A main criticism concerns the lack of theory surrounding some methods.[274] 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.[*citation needed*] (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.[275]

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[276] 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*[277] website.

Errors

[edit]

Some deep learning architectures display problematic behaviors,[278] such as confidently classifying unrecognizable images as belonging to a familiar category of ordinary images (2014)[279] and misclassifying minuscule perturbations of correctly classified images (2013).[280] 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.[278] These issues may possibly be addressed by deep learning architectures that internally form states homologous to image-grammar[281] decompositions of observed entities and events.[278] 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[282] and artificial intelligence (AI).[283]

Cyber threat

[edit]

As deep learning moves from the lab into the world, research and experience show that artificial neural networks are vulnerable to hacks and deception.[284] 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".[285]

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.[286] 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. [287]

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.[286]

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.[286]

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)".[286]

In "data poisoning", false data is continually smuggled into a machine learning system's training set to prevent it from achieving mastery.[286]

Data collection ethics

[edit]

The deep learning systems that are trained using supervised learning often rely on data that is created and/or annotated by humans.[288] 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.[289] 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.[289]

See also

[edit]

- 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

[edit]

1. ^ Schulz, Hannes; Behnke, Sven (1 November 2012). "Deep Learning". *KI - Künstliche Intelligenz*. **26** (4): 357–363. doi:10.1007/s13218-012-0198-z. ISSN 1610-1987. S2CID 220523562.

2. ^ *a* *b* LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey (2015). "Deep Learning" (PDF). *Nature*. **521** (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. S2CID 3074096.
3. ^ *a* *b* Ciresan, D.; Meier, U.; Schmidhuber, J. (2012). "Multi-column deep neural networks for image classification". *2012 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3642–3649. arXiv:1202.2745. doi:10.1109/cvpr.2012.6248110. ISBN 978-1-4673-1228-8. S2CID 2161592.
4. ^ *a* *b* Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey (2012). "ImageNet Classification with Deep Convolutional Neural Networks" (PDF). *NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada*. Archived (PDF) from the original on 2017-01-10. Retrieved 2017-05-24.
5. ^ "Google's AlphaGo AI wins three-match series against the world's best Go player". *TechCrunch*. 25 May 2017. Archived from the original on 17 June 2018. Retrieved 17 June 2018.
6. ^ "Study urges caution when comparing neural networks to the brain". *MIT News | Massachusetts Institute of Technology*. 2022-11-02. Retrieved 2023-12-06.
7. ^ *a* *b* *c* *d* Bengio, Yoshua (2009). "Learning Deep Architectures for AI" (PDF). *Foundations and Trends in Machine Learning*. **2** (1): 1–127. CiteSeerX 10.1.1.701.9550. doi:10.1561/2200000006. S2CID 207178999. Archived from the original (PDF) on 4 March 2016. Retrieved 3 September 2015.
8. ^ *a* *b* *c* *d* *e* Bengio, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **35** (8): 1798–1828. arXiv:1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338. S2CID 393948.
9. ^ *a* *b* *c* *d* *e* *f* *g* *h* Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. **61**: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.
10. ^ Shigeki, Sugiyama (12 April 2019). *Human Behavior and Another Kind in Consciousness: Emerging Research and Opportunities: Emerging Research and Opportunities*. IGI Global. ISBN 978-1-5225-8218-2.
11. ^ Bengio, Yoshua; Lamblin, Pascal; Popovici, Dan; Larochelle, Hugo (2007). *Greedy layer-wise training of deep networks* (PDF). Advances in neural information processing systems. pp. 153–160. Archived (PDF) from the original on 2019-10-20. Retrieved 2019-10-06.
12. ^ *a* *b* Hinton, G.E. (2009). "Deep belief networks". *Scholarpedia*. **4** (5): 5947. Bibcode:2009SchpJ...4.5947H. doi:10.4249/scholarpedia.5947.
13. ^ Rina Dechter (1986). Learning while searching in constraint-satisfaction problems. University of California, Computer Science Department, Cognitive Systems Laboratory.Online Archived 2016-04-19 at the Wayback Machine
14. ^ Aizenberg, I.N.; Aizenberg, N.N.; Vandewalle, J. (2000). *Multi-Valued and Universal Binary Neurons*. Science & Business Media. doi:10.1007/978-1-4757-3115-6. ISBN 978-0-7923-7824-2. Retrieved 27 December 2023.
15. ^ Co-evolving recurrent neurons learn deep memory POMDPs. Proc. GECCO, Washington, D. C., pp. 1795–1802, ACM Press, New York, NY, USA, 2005.
16. ^ Fradkov, Alexander L. (2020-01-01). "Early History of Machine Learning". *IFAC-PapersOnLine*. 21st IFAC World Congress. **53** (2): 1385–1390. doi:10.1016/j.ifacol.2020.12.1888. ISSN 2405-8963. S2CID 235081987.
17. ^ *a* *b* *c* Cybenko (1989). "Approximations by superpositions of sigmoidal functions" (PDF). *Mathematics of Control, Signals, and Systems*. **2** (4): 303–314. Bibcode:1989MCSS....2..303C. doi:10.1007/bf02551274. S2CID 3958369. Archived from the original (PDF) on 10 October 2015.
18. ^ *a* *b* *c* Hornik, Kurt (1991). "Approximation Capabilities of Multilayer Feedforward Networks". *Neural Networks*. **4** (2): 251–257. doi:10.1016/0893-6080(91)90009-t. S2CID 7343126.

19. ^ a b Haykin, Simon S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall. ISBN 978-0-13-273350-2.
20. ^ a b Hassoun, Mohamad H. (1995). *Fundamentals of Artificial Neural Networks*. MIT Press. p. 48. ISBN 978-0-262-08239-6.
21. ^ a b Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The Expressive Power of Neural Networks: A View from the Width Archived 2019-02-13 at the Wayback Machine. Neural Information Processing Systems, 6231-6239.
22. ^ Orhan, A. E.; Ma, W. J. (2017). "Efficient probabilistic inference in generic neural networks trained with non-probabilistic feedback". *Nature Communications*. **8** (1): 138. Bibcode:2017NatCo...8..138O. doi:10.1038/s41467-017-00181-8 . PMC 5527101 . PMID 28743932.
23. ^ a b c d e Deng, L.; Yu, D. (2014). "Deep Learning: Methods and Applications" (PDF). *Foundations and Trends in Signal Processing*. **7** (3–4): 1–199. doi:10.1561/2000000039. Archived (PDF) from the original on 2016-03-14. Retrieved 2014-10-18.
24. ^ a b c d Murphy, Kevin P. (24 August 2012). *Machine Learning: A Probabilistic Perspective*. MIT Press. ISBN 978-0-262-01802-9.
25. ^ a b Fukushima, K. (1969). "Visual feature extraction by a multilayered network of analog threshold elements". *IEEE Transactions on Systems Science and Cybernetics*. **5** (4): 322–333. doi:10.1109/TSSC.1969.300225.
26. ^ Sonoda, Sho; Murata, Noboru (2017). "Neural network with unbounded activation functions is universal approximator". *Applied and Computational Harmonic Analysis*. **43** (2): 233–268. arXiv:1505.03654 . doi:10.1016/j.acha.2015.12.005. S2CID 12149203.
27. ^ Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning* (PDF). Springer. ISBN 978-0-387-31073-2. Archived (PDF) from the original on 2017-01-11. Retrieved 2017-08-06.
28. ^ a b "bibliotheca Augustana". *www.hs-augsburg.de*.
29. ^ Brush, Stephen G. (1967). "History of the Lenz-Ising Model". *Reviews of Modern Physics*. **39** (4): 883–893. Bibcode:1967RvMP...39..883B. doi:10.1103/RevModPhys.39.883.
30. ^ a b Amari, Shun-Ichi (1972). "Learning patterns and pattern sequences by self-organizing nets of threshold elements". *IEEE Transactions*. **C** (21): 1197–1206.
31. ^ a b c d e f g Schmidhuber, Jürgen (2022). "Annotated History of Modern AI and Deep Learning". arXiv:2212.11279 [cs.NE].
32. ^ Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities". *Proceedings of the National Academy of Sciences*. **79** (8): 2554–2558. Bibcode:1982PNAS...79.2554H. doi:10.1073/pnas.79.8.2554 . PMC 346238 . PMID 6953413.
33. ^ Nakano, Kaoru (1971). "Learning Process in a Model of Associative Memory". *Pattern Recognition and Machine Learning*. pp. 172–186. doi:10.1007/978-1-4615-7566-5_15. ISBN 978-1-4615-7568-9.
34. ^ Nakano, Kaoru (1972). "Associatron-A Model of Associative Memory". *IEEE Transactions on Systems, Man, and Cybernetics*. SMC-2 (3): 380–388. doi:10.1109/TSMC.1972.4309133.
35. ^ Turing, Alan (1948). "Intelligent Machinery". *Unpublished (Later Published in Ince DC, Editor, Collected Works of AM Turing—Mechanical Intelligence, Elsevier Science Publishers, 1992)*.
36. ^ Rosenblatt, F. (1958). "The perceptron: A probabilistic model for information storage and organization in the brain". *Psychological Review*. **65** (6): 386–408. doi:10.1037/h0042519. ISSN 1939-1471. PMID 13602029.
37. ^ a b Rosenblatt, Frank (1962). *Principles of Neurodynamics*. Spartan, New York.
38. ^ Joseph, R. D. (1960). *Contributions to Perceptron Theory, Cornell Aeronautical Laboratory Report No. VG-11 96--G-7, Buffalo*.

39. ^ Ivakhnenko, A. G.; Lapa, V. G. (1967). *Cybernetics and Forecasting Techniques*. American Elsevier Publishing Co. ISBN 978-0-444-00020-0.
40. ^ Ivakhnenko, A.G. (March 1970). "Heuristic self-organization in problems of engineering cybernetics". *Automatica*. **6** (2): 207–219. doi:10.1016/0005-1098(70)90092-0.
41. ^ *a* *b* Ivakhnenko, Alexey (1971). "Polynomial theory of complex systems" (PDF). *IEEE Transactions on Systems, Man, and Cybernetics*. SMC-1 (4): 364–378. doi:10.1109/TSMC.1971.4308320. Archived (PDF) from the original on 2017-08-29. Retrieved 2019-11-05.
42. ^ Robbins, H.; Monro, S. (1951). "A Stochastic Approximation Method". *The Annals of Mathematical Statistics*. **22** (3): 400. doi:10.1214/aoms/1177729586 .
43. ^ Amari, Shun'ichi (1967). "A theory of adaptive pattern classifier". *IEEE Transactions*. **EC** (16): 279–307.
44. ^ Ramachandran, Prajit; Barret, Zoph; Quoc, V. Le (October 16, 2017). "Searching for Activation Functions". arXiv:1710.05941 [cs.NE].
45. ^ Fukushima, K. (1979). "Neural network model for a mechanism of pattern recognition unaffected by shift in position—Neocognitron". *Trans. IECE (In Japanese)*. J62-A (10): 658–665. doi:10.1007/bf00344251. PMID 7370364. S2CID 206775608.
46. ^ Fukushima, K. (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". *Biol. Cybern.* **36** (4): 193–202. doi:10.1007/bf00344251. PMID 7370364. S2CID 206775608.
47. ^ Leibniz, Gottfried Wilhelm Freiherr von (1920). *The Early Mathematical Manuscripts of Leibniz: Translated from the Latin Texts Published by Carl Immanuel Gerhardt with Critical and Historical Notes (Leibniz published the chain rule in a 1676 memoir)*. Open court publishing Company. ISBN 9780598818461.
48. ^ Kelley, Henry J. (1960). "Gradient theory of optimal flight paths". *ARS Journal*. **30** (10): 947–954. doi:10.2514/8.5282.
49. ^ Linnainmaa, Seppo (1970). *The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors* (Masters) (in Finnish). University of Helsinki. p. 6–7.
50. ^ Linnainmaa, Seppo (1976). "Taylor expansion of the accumulated rounding error". *BIT Numerical Mathematics*. **16** (2): 146–160. doi:10.1007/bf01931367. S2CID 122357351.
51. ^ Ostrovski, G.M., Volin,Y.M., and Boris, W.W. (1971). On the computation of derivatives. Wiss. Z. Tech. Hochschule for Chemistry, 13:382–384.
52. ^ *a* *b* Schmidhuber, Juergen (25 Oct 2014). "Who Invented Backpropagation?". IDSIA, Switzerland. Archived from the original on 30 July 2024. Retrieved 14 Sep 2024.
53. ^ Werbos, Paul (1982). "Applications of advances in nonlinear sensitivity analysis" (PDF). *System modeling and optimization*. Springer. pp. 762–770. Archived (PDF) from the original on 14 April 2016. Retrieved 2 July 2017.
54. ^ Werbos, Paul J. (1994). *The Roots of Backpropagation : From Ordered Derivatives to Neural Networks and Political Forecasting*. New York: John Wiley & Sons. ISBN 0-471-59897-6.
55. ^ Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (October 1986). "Learning representations by back-propagating errors". *Nature*. **323** (6088): 533–536. Bibcode:1986Natur.323..533R. doi:10.1038/323533a0. ISSN 1476-4687.
56. ^ Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation Archived 2022-10-13 at the Wayback Machine". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundation. MIT Press, 1986.
57. ^ Waibel, Alex (December 1987). *Phoneme Recognition Using Time-Delay Neural Networks* (PDF). Meeting of the Institute of Electrical, Information and Communication Engineers (IEICE). Tokyo, Japan.

58. ^ Alexander Waibel et al., *Phoneme Recognition Using Time-Delay Neural Networks* IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 37, No. 3, pp. 328. – 339 March 1989.
59. ^ Zhang, Wei (1988). "Shift-invariant pattern recognition neural network and its optical architecture". *Proceedings of Annual Conference of the Japan Society of Applied Physics*.
60. ^ LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition", *Neural Computation*, 1, pp. 541–551, 1989.
61. ^ Zhang, Wei (1990). "Parallel distributed processing model with local space-invariant interconnections and its optical architecture". *Applied Optics*. **29** (32): 4790–7. Bibcode:1990ApOpt..29.4790Z. doi:10.1364/AO.29.004790. PMID 20577468.
62. ^ Zhang, Wei (1991). "Image processing of human corneal endothelium based on a learning network". *Applied Optics*. **30** (29): 4211–7. Bibcode:1991ApOpt..30.4211Z. doi:10.1364/AO.30.004211. PMID 20706526.
63. ^ Zhang, Wei (1994). "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network". *Medical Physics*. **21** (4): 517–24. Bibcode:1994MedPh..21..517Z. doi:10.1118/1.597177. PMID 8058017.
64. ^ LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998). "Gradient-based learning applied to document recognition" (PDF). *Proceedings of the IEEE*. **86** (11): 2278–2324. CiteSeerX 10.1.1.32.9552 . doi:10.1109/5.726791. S2CID 14542261. Retrieved October 7, 2016.
65. ^ Jordan, Michael I. (1986). "Attractor dynamics and parallelism in a connectionist sequential machine". *Proceedings of the Annual Meeting of the Cognitive Science Society*. **8**.
66. ^ Elman, Jeffrey L. (March 1990). "Finding Structure in Time". *Cognitive Science*. **14** (2): 179–211. doi:10.1207/s15516709cog1402_1. ISSN 0364-0213.
67. ^ *a* *b* *c* Schmidhuber, Jürgen (April 1991). "Neural Sequence Chunkers" (PDF). *TR FKI-148, TU Munich*.
68. ^ *a* *b* Schmidhuber, Jürgen (1992). "Learning complex, extended sequences using the principle of history compression (based on TR FKI-148, 1991)" (PDF). *Neural Computation*. **4** (2): 234–242. doi:10.1162/neco.1992.4.2.234. S2CID 18271205.
69. ^ Schmidhuber, Jürgen (1993). *Habilitation thesis: System modeling and optimization* (PDF). Archived from the original (PDF) on May 16, 2022. Page 150 ff demonstrates credit assignment across the equivalent of 1,200 layers in an unfolded RNN.
70. ^ *a* *b* *c* S. Hochreiter., "Untersuchungen zu dynamischen neuronalen Netzen". Archived 2015-03-06 at the Wayback Machine. *Diploma thesis. Institut f. Informatik, Technische Univ. Munich. Advisor: J. Schmidhuber*, 1991.
71. ^ Hochreiter, S.; et al. (15 January 2001). "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". In Kolen, John F.; Kremer, Stefan C. (eds.). *A Field Guide to Dynamical Recurrent Networks*. John Wiley & Sons. ISBN 978-0-7803-5369-5.
72. ^ Sepp Hochreiter; Jürgen Schmidhuber (21 August 1995), *Long Short Term Memory*, Wikidata Q98967430
73. ^ Gers, Felix; Schmidhuber, Jürgen; Cummins, Fred (1999). "Learning to forget: Continual prediction with LSTM". *9th International Conference on Artificial Neural Networks: ICANN '99*. Vol. 1999. pp. 850–855. doi:10.1049/cp:19991218. ISBN 0-85296-721-7.
74. ^ *a* *b* Schmidhuber, Jürgen (1991). "A possibility for implementing curiosity and boredom in model-building neural controllers". *Proc. SAB'1991*. MIT Press/Bradford Books. pp. 222–227.
75. ^ Schmidhuber, Jürgen (2010). "Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990-2010)". *IEEE Transactions on Autonomous Mental Development*. **2** (3): 230–247. doi:10.1109/TAMD.2010.2056368. S2CID 234198.
76. ^ *a* *b* Schmidhuber, Jürgen (2020). "Generative Adversarial Networks are Special Cases of Artificial Curiosity (1990) and also Closely Related to Predictability Minimization (1991)". *Neural Networks*. **127**: 58–66. arXiv:1906.04493 . doi:10.1016/j.neunet.2020.04.008. PMID 32334341. S2CID 216056336.

77. ^ Ackley, David H.; Hinton, Geoffrey E.; Sejnowski, Terrence J. (1985-01-01). "A learning algorithm for boltzmann machines". *Cognitive Science*. **9** (1): 147–169. doi:10.1016/S0364-0213(85)80012-4. ISSN 0364-0213.
78. ^ Smolensky, Paul (1986). "Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory" (PDF). In Rumelhart, David E.; McLelland, James L. (eds.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. MIT Press. pp. 194–281. ISBN 0-262-68053-X.
79. ^ Peter, Dayan; Hinton, Geoffrey E.; Neal, Radford M.; Zemel, Richard S. (1995). "The Helmholtz machine". *Neural Computation*. **7** (5): 889–904. doi:10.1162/neco.1995.7.5.889. hdl:21.11116/0000-0002-D6D3-E. PMID 7584891. S2CID 1890561. Closed access icon
80. ^ Hinton, Geoffrey E.; Dayan, Peter; Frey, Brendan J.; Neal, Radford (1995-05-26). "The wake-sleep algorithm for unsupervised neural networks". *Science*. **268** (5214): 1158–1161. Bibcode:1995Sci...268.1158H. doi:10.1126/science.7761831. PMID 7761831. S2CID 871473.
81. ^ Sejnowski, Terrence J. (2018). *The Deep Learning Revolution*. Cambridge, Massachusetts: The MIT Press. ISBN 978-0-262-03803-4.
82. ^ Qian, Ning; Sejnowski, Terrence J. (1988-08-20). "Predicting the secondary structure of globular proteins using neural network models". *Journal of Molecular Biology*. **202** (4): 865–884. doi:10.1016/0022-2836(88)90564-5. ISSN 0022-2836. PMID 3172241.
83. ^ Morgan, Nelson; Bourlard, Hervé; Renals, Steve; Cohen, Michael; Franco, Horacio (1 August 1993). "Hybrid neural network/hidden markov model systems for continuous speech recognition". *International Journal of Pattern Recognition and Artificial Intelligence*. **07** (4): 899–916. doi:10.1142/s0218001493000455. ISSN 0218-0014.
84. ^ Robinson, T. (1992). "A real-time recurrent error propagation network word recognition system". *ICASSP*. Icassp'92: 617–620. ISBN 9780780305328. Archived from the original on 2021-05-09. Retrieved 2017-06-12.
85. ^ Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K. J. (March 1989). "Phoneme recognition using time-delay neural networks" (PDF). *IEEE Transactions on Acoustics, Speech, and Signal Processing*. **37** (3): 328–339. doi:10.1109/29.21701. hdl:10338.dmlcz/135496. ISSN 0096-3518. S2CID 9563026. Archived (PDF) from the original on 2021-04-27. Retrieved 2019-09-24.
86. ^ Baker, J.; Deng, Li; Glass, Jim; Khudanpur, S.; Lee, C.-H.; Morgan, N.; O'Shaughnessy, D. (2009). "Research Developments and Directions in Speech Recognition and Understanding, Part 1". *IEEE Signal Processing Magazine*. **26** (3): 75–80. Bibcode:2009ISPM...26...75B. doi:10.1109/msp.2009.932166. hdl:1721.1/51891. S2CID 357467.
87. ^ Bengio, Y. (1991). "Artificial Neural Networks and their Application to Speech/Sequence Recognition". McGill University Ph.D. thesis. Archived from the original on 2021-05-09. Retrieved 2017-06-12.
88. ^ Deng, L.; Hassanein, K.; Elmasry, M. (1994). "Analysis of correlation structure for a neural predictive model with applications to speech recognition". *Neural Networks*. **7** (2): 331–339. doi:10.1016/0893-6080(94)90027-2.
89. ^ Doddington, G.; Przybocki, M.; Martin, A.; Reynolds, D. (2000). "The NIST speaker recognition evaluation ± Overview, methodology, systems, results, perspective". *Speech Communication*. **31** (2): 225–254. doi:10.1016/S0167-6393(99)00080-1.
90. ^ *a* *b* Heck, L.; Konig, Y.; Sonmez, M.; Weintraub, M. (2000). "Robustness to Telephone Handset Distortion in Speaker Recognition by Discriminative Feature Design". *Speech Communication*. **31** (2): 181–192. doi:10.1016/s0167-6393(99)00077-1.
91. ^ L.P Heck and R. Teunen. "Secure and Convenient Transactions with Nuance Verifier". Nuance Users Conference, April 1998.
92. ^ "Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR (PDF Download Available)". *ResearchGate*. Archived from the original on 9 May 2021. Retrieved 14 June 2017.

93. ^ a b Graves, Alex; Eck, Douglas; Beringer, Nicole; Schmidhuber, Jürgen (2003). "Biologically Plausible Speech Recognition with LSTM Neural Nets" (PDF). *1st Intl. Workshop on Biologically Inspired Approaches to Advanced Information Technology, Bio-ADIT 2004, Lausanne, Switzerland*. pp. 175–184. Archived (PDF) from the original on 2021-05-09. Retrieved 2016-04-09.
94. ^ Graves, Alex; Fernández, Santiago; Gomez, Faustino; Schmidhuber, Jürgen (2006). "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks". *Proceedings of the International Conference on Machine Learning, ICML 2006*: 369–376. CiteSeerX 10.1.1.75.6306 .
95. ^ Santiago Fernandez, Alex Graves, and Jürgen Schmidhuber (2007). An application of recurrent neural networks to discriminative keyword spotting Archived 2018-11-18 at the Wayback Machine. Proceedings of ICANN (2), pp. 220–229.
96. ^ Graves, Alex; and Schmidhuber, Jürgen; *Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks*, in Bengio, Yoshua; Schuurmans, Dale; Lafferty, John; Williams, Chris K. I.; and Culotta, Aron (eds.), *Advances in Neural Information Processing Systems 22 (NIPS'22), December 7th–10th, 2009, Vancouver, BC*, Neural Information Processing Systems (NIPS) Foundation, 2009, pp. 545–552
97. ^ Hinton, Geoffrey E. (1 October 2007). "Learning multiple layers of representation". *Trends in Cognitive Sciences*. **11** (10): 428–434. doi:10.1016/j.tics.2007.09.004. ISSN 1364-6613. PMID 17921042. S2CID 15066318. Archived from the original on 11 October 2013. Retrieved 12 June 2017.
98. ^ Hinton, G. E.; Osindero, S.; Teh, Y. W. (2006). "A Fast Learning Algorithm for Deep Belief Nets" (PDF). *Neural Computation*. **18** (7): 1527–1554. doi:10.1162/neco.2006.18.7.1527. PMID 16764513. S2CID 2309950. Archived (PDF) from the original on 2015-12-23. Retrieved 2011-07-20.
99. ^ G. E. Hinton., "Learning multiple layers of representation". Archived 2018-05-22 at the Wayback Machine. *Trends in Cognitive Sciences*, 11, pp. 428–434, 2007.
100. ^ Hinton, Geoffrey E. (October 2007). "Learning multiple layers of representation". *Trends in Cognitive Sciences*. **11** (10): 428–434. doi:10.1016/j.tics.2007.09.004. PMID 17921042.
101. ^ Hinton, Geoffrey E.; Osindero, Simon; Teh, Yee-Whye (July 2006). "A Fast Learning Algorithm for Deep Belief Nets". *Neural Computation*. **18** (7): 1527–1554. doi:10.1162/neco.2006.18.7.1527. ISSN 0899-7667. PMID 16764513.
102. ^ Hinton, Geoffrey E. (2009-05-31). "Deep belief networks". *Scholarpedia*. **4** (5): 5947. Bibcode:2009SchpJ...4.5947H. doi:10.4249/scholarpedia.5947 . ISSN 1941-6016.
103. ^ Yann LeCun (2016). Slides on Deep Learning Online Archived 2016-04-23 at the Wayback Machine
104. ^ a b c Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; Kingsbury, B. (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". *IEEE Signal Processing Magazine*. **29** (6): 82–97. Bibcode:2012ISPM...29...82H. doi:10.1109/msp.2012.2205597. S2CID 206485943.
105. ^ a b c Deng, L.; Hinton, G.; Kingsbury, B. (May 2013). "New types of deep neural network learning for speech recognition and related applications: An overview (ICASSP)" (PDF). Microsoft. Archived (PDF) from the original on 2017-09-26. Retrieved 27 December 2023.
106. ^ a b c Yu, D.; Deng, L. (2014). *Automatic Speech Recognition: A Deep Learning Approach (Publisher: Springer)*. Springer. ISBN 978-1-4471-5779-3.
107. ^ "Deng receives prestigious IEEE Technical Achievement Award - Microsoft Research". *Microsoft Research*. 3 December 2015. Archived from the original on 16 March 2018. Retrieved 16 March 2018.
108. ^ a b Li, Deng (September 2014). "Keynote talk: 'Achievements and Challenges of Deep Learning - From Speech Analysis and Recognition To Language and Multimodal Processing' ". *Interspeech*. Archived from the original on 2017-09-26. Retrieved 2017-06-12.

109. ^ Yu, D.; Deng, L. (2010). "Roles of Pre-Training and Fine-Tuning in Context-Dependent DBN-HMMs for Real-World Speech Recognition". *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. Archived from the original on 2017-10-12. Retrieved 2017-06-14.
110. ^ Seide, F.; Li, G.; Yu, D. (2011). "Conversational speech transcription using context-dependent deep neural networks". *Interspeech 2011*. pp. 437–440. doi:10.21437/Interspeech.2011-169. S2CID 398770. Archived from the original on 2017-10-12. Retrieved 2017-06-14.
111. ^ Deng, Li; Li, Jinyu; Huang, Jui-Ting; Yao, Kaisheng; Yu, Dong; Seide, Frank; Seltzer, Mike; Zweig, Geoff; He, Xiaodong (1 May 2013). "Recent Advances in Deep Learning for Speech Research at Microsoft". *Microsoft Research*. Archived from the original on 12 October 2017. Retrieved 14 June 2017.
112. ^ *a* *b* Oh, K.-S.; Jung, K. (2004). "GPU implementation of neural networks". *Pattern Recognition*. **37** (6): 1311–1314. Bibcode:2004PatRe..37.1311O. doi:10.1016/j.patcog.2004.01.013.
113. ^ *a* *b* Chellapilla, Kumar; Puri, Sidd; Simard, Patrice (2006), *High performance convolutional neural networks for document processing*, archived from the original on 2020-05-18, retrieved 2021-02-14
114. ^ Sze, Vivienne; Chen, Yu-Hsin; Yang, Tien-Ju; Emer, Joel (2017). "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". arXiv:1703.09039 [cs.CV].
115. ^ Raina, Rajat; Madhavan, Anand; Ng, Andrew Y. (2009-06-14). "Large-scale deep unsupervised learning using graphics processors". *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. New York, NY, USA: Association for Computing Machinery. pp. 873–880. doi:10.1145/1553374.1553486. ISBN 978-1-60558-516-1.
116. ^ Cireşan, Dan Claudiu; Meier, Ueli; Gambardella, Luca Maria; Schmidhuber, Jürgen (21 September 2010). "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". *Neural Computation*. **22** (12): 3207–3220. arXiv:1003.0358 . doi:10.1162/neco_a_00052. ISSN 0899-7667. PMID 20858131. S2CID 1918673.
117. ^ Ciresan, D. C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (PDF). *International Joint Conference on Artificial Intelligence*. doi:10.5591/978-1-57735-516-8/ijcai11-210. Archived (PDF) from the original on 2014-09-29. Retrieved 2017-06-13.
118. ^ Ciresan, Dan; Giusti, Alessandro; Gambardella, Luca M.; Schmidhuber, Jürgen (2012). Pereira, F.; Burges, C. J. C.; Bottou, L.; Weinberger, K. Q. (eds.). *Advances in Neural Information Processing Systems 25* (PDF). Curran Associates, Inc. pp. 2843–2851. Archived (PDF) from the original on 2017-08-09. Retrieved 2017-06-13.
119. ^ Ciresan, D.; Giusti, A.; Gambardella, L.M.; Schmidhuber, J. (2013). "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks". *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. Lecture Notes in Computer Science. Vol. 7908. pp. 411–418. doi:10.1007/978-3-642-40763-5_51. ISBN 978-3-642-38708-1. PMID 24579167.
120. ^ Ng, Andrew; Dean, Jeff (2012). "Building High-level Features Using Large Scale Unsupervised Learning". arXiv:1112.6209 [cs.LG].
121. ^ Simonyan, Karen; Andrew, Zisserman (2014). "Very Deep Convolution Networks for Large Scale Image Recognition". arXiv:1409.1556 [cs.CV].
122. ^ Szegedy, Christian (2015). "Going deeper with convolutions" (PDF). *Cvpr2015*. arXiv:1409.4842 .
123. ^ Vinyals, Oriol; Toshev, Alexander; Bengio, Samy; Erhan, Dumitru (2014). "Show and Tell: A Neural Image Caption Generator". arXiv:1411.4555 [cs.CV]..
124. ^ Fang, Hao; Gupta, Saurabh; Iandola, Forrest; Srivastava, Rupesh; Deng, Li; Dollár, Piotr; Gao, Jianfeng; He, Xiaodong; Mitchell, Margaret; Platt, John C; Lawrence Zitnick, C; Zweig, Geoffrey (2014). "From Captions to Visual Concepts and Back". arXiv:1411.4952 [cs.CV]..
125. ^ Kiros, Ryan; Salakhutdinov, Ruslan; Zemel, Richard S (2014). "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models". arXiv:1411.2539 [cs.LG]..
126. ^ Simonyan, Karen; Zisserman, Andrew (2015-04-10), *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv:1409.1556

127. ^ He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". arXiv:1502.01852 [cs.CV].
128. ^ He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (10 Dec 2015). *Deep Residual Learning for Image Recognition*. arXiv:1512.03385 .
129. ^ He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). *Deep Residual Learning for Image Recognition*. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE. pp. 770–778. arXiv:1512.03385 . doi:10.1109/CVPR.2016.90. ISBN 978-1-4673-8851-1.
130. ^ Gatys, Leon A.; Ecker, Alexander S.; Bethge, Matthias (26 August 2015). "A Neural Algorithm of Artistic Style". arXiv:1508.06576 [cs.CV].
131. ^ Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). *Generative Adversarial Networks* (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680. Archived (PDF) from the original on 22 November 2019. Retrieved 20 August 2019.
132. ^ "GAN 2.0: NVIDIA's Hyperrealistic Face Generator". *SyncedReview.com*. December 14, 2018. Retrieved October 3, 2019.
133. ^ Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. (26 February 2018). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". arXiv:1710.10196 [cs.NE].
134. ^ "Prepare, Don't Panic: Synthetic Media and Deepfakes". witness.org. Archived from the original on 2 December 2020. Retrieved 25 November 2020.
135. ^ Sohl-Dickstein, Jascha; Weiss, Eric; Maheswaranathan, Niru; Ganguli, Surya (2015-06-01). "Deep Unsupervised Learning using Nonequilibrium Thermodynamics" (PDF). *Proceedings of the 32nd International Conference on Machine Learning*. **37**. PMLR: 2256–2265. arXiv:1503.03585 .
136. ^ Google Research Blog. The neural networks behind Google Voice transcription. August 11, 2015. By Françoise Beaufays http://googleresearch.blogspot.co.at/2015/08/the-neural-networks-behind-google-voice.html
137. ^ *a b* Sak, Haşim; Senior, Andrew; Rao, Kanishka; Beaufays, Françoise; Schalkwyk, Johan (September 2015). "Google voice search: faster and more accurate". Archived from the original on 2016-03-09. Retrieved 2016-04-09.
138. ^ Singh, Premjeet; Saha, Goutam; Sahidullah, Md (2021). "Non-linear frequency warping using constant-Q transformation for speech emotion recognition". *2021 International Conference on Computer Communication and Informatics (ICCCI)*. pp. 1–4. arXiv:2102.04029 . doi:10.1109/ICCCI50826.2021.9402569. ISBN 978-1-7281-5875-4. S2CID 231846518.
139. ^ Sak, Hasim; Senior, Andrew; Beaufays, Francoise (2014). "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling" (PDF). Archived from the original (PDF) on 24 April 2018.
140. ^ Li, Xiangang; Wu, Xihong (2014). "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition". arXiv:1410.4281 [cs.CL].
141. ^ Zen, Heiga; Sak, Hasim (2015). "Unidirectional Long Short-Term Memory Recurrent Neural Network with Recurrent Output Layer for Low-Latency Speech Synthesis" (PDF). *Google.com*. ICASSP. pp. 4470–4474. Archived (PDF) from the original on 2021-05-09. Retrieved 2017-06-13.
142. ^ "2018 ACM A.M. Turing Award Laureates". *awards.acm.org*. Retrieved 2024-08-07.
143. ^ Ferrie, C., & Kaiser, S. (2019). *Neural Networks for Babies*. Sourcebooks. ISBN 978-1492671206.

144. ^ Silver, David; Huang, Aja; Maddison, Chris J.; Guez, Arthur; Sifre, Laurent; Driessche, George van den; Schrittwieser, Julian; Antonoglou, Ioannis; Panneershelvam, Veda (January 2016). "Mastering the game of Go with deep neural networks and tree search". *Nature*. **529** (7587): 484–489. Bibcode:2016Natur.529..484S. doi:10.1038/nature16961. ISSN 1476-4687. PMID 26819042. S2CID 515925.
145. ^ *A Guide to Deep Learning and Neural Networks*, archived from the original on 2020-11-02, retrieved 2020-11-16
146. ^ *a* *b* Kumar, Nishant; Raubal, Martin (2021). "Applications of deep learning in congestion detection, prediction and alleviation: A survey". *Transportation Research Part C: Emerging Technologies*. **133**: 103432. arXiv:2102.09759. Bibcode:2021TRPC..13303432K. doi:10.1016/j.trc.2021.103432. hdl:10230/42143. S2CID 240420107.
147. ^ Szegedy, Christian; Toshev, Alexander; Erhan, Dumitru (2013). "Deep neural networks for object detection". *Advances in Neural Information Processing Systems*: 2553–2561. Archived from the original on 2017-06-29. Retrieved 2017-06-13.
148. ^ Rolnick, David; Tegmark, Max (2018). "The power of deeper networks for expressing natural functions". *International Conference on Learning Representations*. ICLR 2018. Archived from the original on 2021-01-07. Retrieved 2021-01-05.
149. ^ Hof, Robert D. "Is Artificial Intelligence Finally Coming into Its Own?". *MIT Technology Review*. Archived from the original on 31 March 2019. Retrieved 10 July 2018.
150. ^ *a* *b* Gers, Felix A.; Schmidhuber, Jürgen (2001). "LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages". *IEEE Transactions on Neural Networks*. **12** (6): 1333–1340. doi:10.1109/72.963769. PMID 18249962. S2CID 10192330. Archived from the original on 2020-01-26. Retrieved 2020-02-25.
151. ^ *a* *b* *c* Sutskever, L.; Vinyals, O.; Le, Q. (2014). "Sequence to Sequence Learning with Neural Networks" (PDF). *Proc. NIPS*. arXiv:1409.3215. Bibcode:2014arXiv1409.3215S. Archived (PDF) from the original on 2021-05-09. Retrieved 2017-06-13.
152. ^ *a* *b* Jozefowicz, Rafal; Vinyals, Oriol; Schuster, Mike; Shazeer, Noam; Wu, Yonghui (2016). "Exploring the Limits of Language Modeling". arXiv:1602.02410 [cs.CL].
153. ^ *a* *b* Gillick, Dan; Brunk, Cliff; Vinyals, Oriol; Subramanya, Amarnag (2015). "Multilingual Language Processing from Bytes". arXiv:1512.00103 [cs.CL].
154. ^ Mikolov, T.; et al. (2010). "Recurrent neural network based language model" (PDF). *Interspeech*: 1045–1048. doi:10.21437/Interspeech.2010-343. S2CID 17048224. Archived (PDF) from the original on 2017-05-16. Retrieved 2017-06-13.
155. ^ Hochreiter, Sepp; Schmidhuber, Jürgen (1 November 1997). "Long Short-Term Memory". *Neural Computation*. **9** (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. ISSN 0899-7667. PMID 9377276. S2CID 1915014.
156. ^ *a* *b* "Learning Precise Timing with LSTM Recurrent Networks (PDF Download Available)". *ResearchGate*. Archived from the original on 9 May 2021. Retrieved 13 June 2017.
157. ^ LeCun, Y.; et al. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*. **86** (11): 2278–2324. doi:10.1109/5.726791. S2CID 14542261.
158. ^ Sainath, Tara N.; Mohamed, Abdel-Rahman; Kingsbury, Brian; Ramabhadran, Bhuvana (2013). "Deep convolutional neural networks for LVCSR". *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. pp. 8614–8618. doi:10.1109/icassp.2013.6639347. ISBN 978-1-4799-0356-6. S2CID 13816461.
159. ^ Bengio, Yoshua; Boulanger-Lewandowski, Nicolas; Pascanu, Razvan (2013). "Advances in optimizing recurrent networks". *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. pp. 8624–8628. arXiv:1212.0901. CiteSeerX 10.1.1.752.9151. doi:10.1109/icassp.2013.6639349. ISBN 978-1-4799-0356-6. S2CID 12485056.
160. ^ Dahl, G.; et al. (2013). "Improving DNNs for LVCSR using rectified linear units and dropout" (PDF). *ICASSP*. Archived (PDF) from the original on 2017-08-12. Retrieved 2017-06-13.

161. ^ Kumar, Nishant; Martin, Henry; Raubal, Martin (2024). "Enhancing Deep Learning-Based City-Wide Traffic Prediction Pipelines Through Complexity Analysis". *Data Science for Transportation*. **6** (3): Article 24. doi:10.1007/s42421-024-00109-x . hdl:20.500.11850/695425 .
162. ^ "Data Augmentation - deeplearning.ai | Coursera". *Coursera*. Archived from the original on 1 December 2017. Retrieved 30 November 2017.
163. ^ Hinton, G. E. (2010). "A Practical Guide to Training Restricted Boltzmann Machines". *Tech. Rep. UTML TR 2010-003*. Archived from the original on 2021-05-09. Retrieved 2017-06-13.
164. ^ You, Yang; Buluç, Aydın; Demmel, James (November 2017). "Scaling deep learning on GPU and knights landing clusters". *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis on - SC '17*. SC '17, ACM. pp. 1–12. doi:10.1145/3126908.3126912. ISBN 9781450351140. S2CID 8869270. Archived from the original on 29 July 2020. Retrieved 5 March 2018.
165. ^ Viebke, André; Memeti, Suejb; Pllana, Sabri; Abraham, Ajith (2019). "CHAOS: a parallelization scheme for training convolutional neural networks on Intel Xeon Phi". *The Journal of Supercomputing*. **75**: 197–227. arXiv:1702.07908 . Bibcode:2017arXiv170207908V. doi:10.1007/s11227-017-1994-x. S2CID 14135321.
166. ^ Ting Qin, et al. "A learning algorithm of CMAC based on RLS". Neural Processing Letters 19.1 (2004): 49-61.
167. ^ Ting Qin, et al. "Continuous CMAC-QRLS and its systolic array". Archived 2018-11-18 at the Wayback Machine. Neural Processing Letters 22.1 (2005): 1-16.
168. ^ Research, AI (23 October 2015). "Deep Neural Networks for Acoustic Modeling in Speech Recognition". *airesearch.com*. Archived from the original on 1 February 2016. Retrieved 23 October 2015.
169. ^ "GPUs Continue to Dominate the AI Accelerator Market for Now". *InformationWeek*. December 2019. Archived from the original on 10 June 2020. Retrieved 11 June 2020.
170. ^ Ray, Tiernan (2019). "AI is changing the entire nature of computation". *ZDNet*. Archived from the original on 25 May 2020. Retrieved 11 June 2020.
171. ^ "AI and Compute". *OpenAI*. 16 May 2018. Archived from the original on 17 June 2020. Retrieved 11 June 2020.
172. ^ "HUAWEI Reveals the Future of Mobile AI at IFA 2017 | HUAWEI Latest News | HUAWEI Global". *consumer.huawei.com*.
173. ^ P, JouppiNorman; YoungCliff; PatilNishant; PattersonDavid; AgrawalGaurav; BajwaRaminder; BatesSarah; BhatiaSuresh; BodenNan; BorchersAl; BoyleRick (2017-06-24). "In-Datacenter Performance Analysis of a Tensor Processing Unit". *ACM SIGARCH Computer Architecture News*. **45** (2): 1–12. arXiv:1704.04760 . doi:10.1145/3140659.3080246 .
174. ^ Woodie, Alex (2021-11-01). "Cerebras Hits the Accelerator for Deep Learning Workloads". *Datanami*. Retrieved 2022-08-03.
175. ^ "Cerebras launches new AI supercomputing processor with 2.6 trillion transistors". *VentureBeat*. 2021-04-20. Retrieved 2022-08-03.
176. ^ Marega, Guilherme Migliato; Zhao, Yanfei; Avsar, Ahmet; Wang, Zhenyu; Tripati, Mukesh; Radenovic, Aleksandra; Kis, Anras (2020). "Logic-in-memory based on an atomically thin semiconductor". *Nature*. **587** (2): 72–77. Bibcode:2020Natur.587...72M. doi:10.1038/s41586-020-2861-0. PMC 7116757 . PMID 33149289.
177. ^ *a b c* Feldmann, J.; Youngblood, N.; Karpov, M.; et al. (2021). "Parallel convolutional processing using an integrated photonic tensor". *Nature*. **589** (2): 52–58. arXiv:2002.00281 . doi:10.1038/s41586-020-03070-1. PMID 33408373. S2CID 211010976.
178. ^ Garofolo, J.S.; Lamel, L.F.; Fisher, W.M.; Fiscus, J.G.; Pallett, D.S.; Dahlgren, N.L.; Zue, V. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Linguistic Data Consortium. doi:10.35111/17gk-bn40. ISBN 1-58563-019-5. Retrieved 27 December 2023.
179. ^ Robinson, Tony (30 September 1991). "Several Improvements to a Recurrent Error Propagation Network Phone Recognition System". *Cambridge University Engineering Department Technical Report*. CUED/F-INFENG/TR82. doi:10.13140/RG.2.2.15418.90567.

180. ^ Abdel-Hamid, O.; et al. (2014). "Convolutional Neural Networks for Speech Recognition". *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. **22** (10): 1533–1545. doi:10.1109/taslp.2014.2339736. S2CID 206602362. Archived from the original on 2020-09-22. Retrieved 2018-04-20.
181. ^ Deng, L.; Platt, J. (2014). "Ensemble Deep Learning for Speech Recognition". *Proc. Interspeech*: 1915–1919. doi:10.21437/Interspeech.2014-433. S2CID 15641618.
182. ^ Tóth, Laszló (2015). "Phone Recognition with Hierarchical Convolutional Deep Maxout Networks" (PDF). *EURASIP Journal on Audio, Speech, and Music Processing*. **2015**. doi:10.1186/s13636-015-0068-3. S2CID 217950236. Archived (PDF) from the original on 2020-09-24. Retrieved 2019-04-01.
183. ^ McMillan, Robert (17 December 2014). "How Skype Used AI to Build Its Amazing New Language Translator | WIRED". *Wired*. Archived from the original on 8 June 2017. Retrieved 14 June 2017.
184. ^ Hannun, Awni; Case, Carl; Casper, Jared; Catanzaro, Bryan; Diamos, Greg; Elsen, Erich; Prenger, Ryan; Satheesh, Sanjeev; Sengupta, Shubho; Coates, Adam; Ng, Andrew Y (2014). "Deep Speech: Scaling up end-to-end speech recognition". arXiv:1412.5567 [cs.CL].
185. ^ "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges". *yann.lecun.com*. Archived from the original on 2014-01-13. Retrieved 2014-01-28.
186. ^ Cireşan, Dan; Meier, Ueli; Masci, Jonathan; Schmidhuber, Jürgen (August 2012). "Multi-column deep neural network for traffic sign classification". *Neural Networks*. Selected Papers from IJCNN 2011. **32**: 333–338. CiteSeerX 10.1.1.226.8219. doi:10.1016/j.neunet.2012.02.023. PMID 22386783.
187. ^ Chaochao Lu; Xiaoou Tang (2014). "Surpassing Human Level Face Recognition". arXiv:1404.3840 [cs.CV].
188. ^ Nvidia Demos a Car Computer Trained with "Deep Learning" (6 January 2015), David Talbot, *MIT Technology Review*
189. ^ *a* *b* *c* G. W. Smith; Frederic Fol Leymarie (10 April 2017). "The Machine as Artist: An Introduction". *Arts*. **6** (4): 5. doi:10.3390/arts6020005.
190. ^ *a* *b* *c* Blaise Agüera y Arcas (29 September 2017). "Art in the Age of Machine Intelligence". *Arts*. **6** (4): 18. doi:10.3390/arts6040018.
191. ^ Goldberg, Yoav; Levy, Omar (2014). "word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method". arXiv:1402.3722 [cs.CL].
192. ^ *a* *b* Socher, Richard; Manning, Christopher. "Deep Learning for NLP" (PDF). Archived (PDF) from the original on 6 July 2014. Retrieved 26 October 2014.
193. ^ Socher, Richard; Bauer, John; Manning, Christopher; Ng, Andrew (2013). "Parsing With Compositional Vector Grammars" (PDF). *Proceedings of the ACL 2013 Conference*. Archived (PDF) from the original on 2014-11-27. Retrieved 2014-09-03.
194. ^ Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.; Potts, C. (October 2013). "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank" (PDF). *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. pp. 1631–1642. doi:10.18653/v1/D13-1170. Archived (PDF) from the original on 28 December 2016. Retrieved 21 December 2023.
195. ^ Shen, Yelong; He, Xiaodong; Gao, Jianfeng; Deng, Li; Mesnil, Gregoire (1 November 2014). "A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval". *Microsoft Research*. Archived from the original on 27 October 2017. Retrieved 14 June 2017.
196. ^ Huang, Po-Sen; He, Xiaodong; Gao, Jianfeng; Deng, Li; Acero, Alex; Heck, Larry (1 October 2013). "Learning Deep Structured Semantic Models for Web Search using Clickthrough Data". *Microsoft Research*. Archived from the original on 27 October 2017. Retrieved 14 June 2017.
197. ^ Mesnil, G.; Dauphin, Y.; Yao, K.; Bengio, Y.; Deng, L.; Hakkani-Tur, D.; He, X.; Heck, L.; Tur, G.; Yu, D.; Zweig, G. (2015). "Using recurrent neural networks for slot filling in spoken language understanding". *IEEE Transactions on Audio, Speech, and Language Processing*. **23** (3): 530–539. doi:10.1109/taslp.2014.2383614. S2CID 1317136.

198. ^ a b Gao, Jianfeng; He, Xiaodong; Yih, Scott Wen-tau; Deng, Li (1 June 2014). "Learning Continuous Phrase Representations for Translation Modeling". *Microsoft Research*. Archived from the original on 27 October 2017. Retrieved 14 June 2017.
199. ^ Brocardo, Marcelo Luiz; Traore, Issa; Woungang, Isaac; Obaidat, Mohammad S. (2017). "Authorship verification using deep belief network systems". *International Journal of Communication Systems*. **30** (12): e3259. doi:10.1002/dac.3259. S2CID 40745740.
200. ^ Kariampuzha, William; Alyea, Gioconda; Qu, Sue; Sanjak, Jaleal; Mathé, Ewy; Sid, Eric; Chatelaine, Haley; Yadaw, Arjun; Xu, Yanji; Zhu, Qian (2023). "Precision information extraction for rare disease epidemiology at scale". *Journal of Translational Medicine*. **21** (1): 157. doi:10.1186/s12967-023-04011-y. PMC 9972634. PMID 36855134.
201. ^ "Deep Learning for Natural Language Processing: Theory and Practice (CIKM2014 Tutorial) - Microsoft Research". *Microsoft Research*. Archived from the original on 13 March 2017. Retrieved 14 June 2017.
202. ^ Turovsky, Barak (15 November 2016). "Found in translation: More accurate, fluent sentences in Google Translate". *The Keyword Google Blog*. Archived from the original on 7 April 2017. Retrieved 23 March 2017.
203. ^ a b c d Schuster, Mike; Johnson, Melvin; Thorat, Nikhil (22 November 2016). "Zero-Shot Translation with Google's Multilingual Neural Machine Translation System". *Google Research Blog*. Archived from the original on 10 July 2017. Retrieved 23 March 2017.
204. ^ Wu, Yonghui; Schuster, Mike; Chen, Zhifeng; Le, Quoc V; Norouzi, Mohammad; Macherey, Wolfgang; Krikun, Maxim; Cao, Yuan; Gao, Qin; Macherey, Klaus; Klingner, Jeff; Shah, Apurva; Johnson, Melvin; Liu, Xiaobing; Kaiser, Łukasz; Gouws, Stephan; Kato, Yoshikiyo; Kudo, Taku; Kazawa, Hideto; Stevens, Keith; Kurian, George; Patil, Nishant; Wang, Wei; Young, Cliff; Smith, Jason; Riesa, Jason; Rudnick, Alex; Vinyals, Oriol; Corrado, Greg; et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". arXiv:1609.08144 [cs.CL].
205. ^ Metz, Cade (27 September 2016). "An Infusion of AI Makes Google Translate More Powerful Than Ever". *Wired*. Archived from the original on 8 November 2020. Retrieved 12 October 2017.
206. ^ a b Boitet, Christian; Blanchon, Hervé; Seligman, Mark; Bellynck, Valérie (2010). "MT on and for the Web" (PDF). Archived from the original (PDF) on 29 March 2017. Retrieved 1 December 2016.
207. ^ Arrowsmith, J; Miller, P (2013). "Trial watch: Phase II and phase III attrition rates 2011-2012". *Nature Reviews Drug Discovery*. **12** (8): 569. doi:10.1038/nrd4090. PMID 23903212. S2CID 20246434.
208. ^ Verbist, B; Klambauer, G; Vervoort, L; Talloen, W; The Qstar, Consortium; Shkedy, Z; Thas, O; Bender, A; Göhlmann, H. W.; Hochreiter, S (2015). "Using transcriptomics to guide lead optimization in drug discovery projects: Lessons learned from the QSTAR project". *Drug Discovery Today*. **20** (5): 505–513. doi:10.1016/j.drudis.2014.12.014. hdl:1942/18723. PMID 25582842.
209. ^ "Merck Molecular Activity Challenge". *kaggle.com*. Archived from the original on 2020-07-16. Retrieved 2020-07-16.
210. ^ "Multi-task Neural Networks for QSAR Predictions | Data Science Association". *www.datascienceassn.org*. Archived from the original on 30 April 2017. Retrieved 14 June 2017.
211. ^ "Toxicology in the 21st century Data Challenge"
212. ^ "NCATS Announces Tox21 Data Challenge Winners". Archived from the original on 2015-09-08. Retrieved 2015-03-05.
213. ^ "NCATS Announces Tox21 Data Challenge Winners". Archived from the original on 28 February 2015. Retrieved 5 March 2015.
214. ^ Wallach, Izhar; Dzamba, Michael; Heifets, Abraham (9 October 2015). "AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery". arXiv:1510.02855 [cs.LG].
215. ^ a b "Toronto startup has a faster way to discover effective medicines". *The Globe and Mail*. Archived from the original on 20 October 2015. Retrieved 9 November 2015.

216. ^ "Startup Harnesses Supercomputers to Seek Cures". *KQED Future of You*. 27 May 2015. Archived from the original on 24 December 2015. Retrieved 9 November 2015.
217. ^ Gilmer, Justin; Schoenholz, Samuel S.; Riley, Patrick F.; Vinyals, Oriol; Dahl, George E. (2017-06-12). "Neural Message Passing for Quantum Chemistry". arXiv:1704.01212 [cs.LG].
218. ^ Zhavoronkov, Alex (2019). "Deep learning enables rapid identification of potent DDR1 kinase inhibitors". *Nature Biotechnology*. **37** (9): 1038–1040. doi:10.1038/s41587-019-0224-x. PMID 31477924. S2CID 201716327.
219. ^ Gregory, Barber. "A Molecule Designed By AI Exhibits 'Druglike' Qualities". *Wired*. Archived from the original on 2020-04-30. Retrieved 2019-09-05.
220. ^ Tkachenko, Yegor (8 April 2015). "Autonomous CRM Control via CLV Approximation with Deep Reinforcement Learning in Discrete and Continuous Action Space". arXiv:1504.01840 [cs.LG].
221. ^ van den Oord, Aaron; Dieleman, Sander; Schrauwen, Benjamin (2013). Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; Weinberger, K. Q. (eds.). *Advances in Neural Information Processing Systems 26* (PDF). Curran Associates, Inc. pp. 2643–2651. Archived (PDF) from the original on 2017-05-16. Retrieved 2017-06-14.
222. ^ Feng, X.Y.; Zhang, H.; Ren, Y.J.; Shang, P.H.; Zhu, Y.; Liang, Y.C.; Guan, R.C.; Xu, D. (2019). "The Deep Learning–Based Recommender System "Pubmender" for Choosing a Biomedical Publication Venue: Development and Validation Study". *Journal of Medical Internet Research*. **21** (5): e12957. doi:10.2196/12957 . PMC 6555124 . PMID 31127715.
223. ^ Elkahky, Ali Mamdouh; Song, Yang; He, Xiaodong (1 May 2015). "A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems". *Microsoft Research*. Archived from the original on 25 January 2018. Retrieved 14 June 2017.
224. ^ Chicco, Davide; Sadowski, Peter; Baldi, Pierre (1 January 2014). "Deep autoencoder neural networks for gene ontology annotation predictions". *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM. pp. 533–540. doi:10.1145/2649387.2649442. hdl:11311/964622. ISBN 9781450328944. S2CID 207217210. Archived from the original on 9 May 2021. Retrieved 23 November 2015.
225. ^ Sathyanarayana, Aarti (1 January 2016). "Sleep Quality Prediction From Wearable Data Using Deep Learning". *JMIR mHealth and uHealth*. **4** (4): e125. doi:10.2196/mhealth.6562 . PMC 5116102 . PMID 27815231. S2CID 3821594.
226. ^ Choi, Edward; Schuetz, Andy; Stewart, Walter F.; Sun, Jimeng (13 August 2016). "Using recurrent neural network models for early detection of heart failure onset". *Journal of the American Medical Informatics Association*. **24** (2): 361–370. doi:10.1093/jamia/ocw112. ISSN 1067-5027. PMC 5391725 . PMID 27521897.
227. ^ "DeepMind's protein-folding AI has solved a 50-year-old grand challenge of biology". *MIT Technology Review*. Retrieved 2024-05-10.
228. ^ Shead, Sam (2020-11-30). "DeepMind solves 50-year-old 'grand challenge' with protein folding A.I." *CNBC*. Retrieved 2024-05-10.
229. ^ *a* *b* Shalev, Y.; Painsky, A.; Ben-Gal, I. (2022). "Neural Joint Entropy Estimation" (PDF). *IEEE Transactions on Neural Networks and Learning Systems*. **PP** (4): 5488–5500. arXiv:2012.11197 . doi:10.1109/TNNLS.2022.3204919. PMID 36155469. S2CID 229339809.
230. ^ Litjens, Geert; Kooi, Thijs; Bejnordi, Babak Ehteshami; Setio, Arnaud Arindra Adiyoso; Ciompi, Francesco; Ghafoorian, Mohsen; van der Laak, Jeroen A.W.M.; van Ginneken, Bram; Sánchez, Clara I. (December 2017). "A survey on deep learning in medical image analysis". *Medical Image Analysis*. **42**: 60–88. arXiv:1702.05747 . Bibcode:2017arXiv170205747L. doi:10.1016/j.media.2017.07.005. PMID 28778026. S2CID 2088679.
231. ^ Forslid, Gustav; Wieslander, Hakan; Bengtsson, Ewert; Wahlby, Carolina; Hirsch, Jan-Michael; Stark, Christina Runow; Sadanandan, Sajith Kecheril (2017). "Deep Convolutional Neural Networks for Detecting Cellular Changes Due to Malignancy". *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. pp. 82–89. doi:10.1109/ICCVW.2017.18. ISBN 9781538610343. S2CID 4728736. Archived from the original on 2021-05-09. Retrieved 2019-11-12.

232. ^ Dong, Xin; Zhou, Yizhao; Wang, Lantian; Peng, Jingfeng; Lou, Yanbo; Fan, Yiqun (2020). "Liver Cancer Detection Using Hybridized Fully Convolutional Neural Network Based on Deep Learning Framework". *IEEE Access*. **8**: 129889–129898. Bibcode:2020IEEEA...8l9889D. doi:10.1109/ACCESS.2020.3006362 . ISSN 2169-3536. S2CID 220733699.
233. ^ Lyakhov, Pavel Alekseevich; Lyakhova, Ulyana Alekseevna; Nagornov, Nikolay Nikolaevich (2022-04-03). "System for the Recognizing of Pigmented Skin Lesions with Fusion and Analysis of Heterogeneous Data Based on a Multimodal Neural Network". *Cancers*. **14** (7): 1819. doi:10.3390/cancers14071819 . ISSN 2072-6694. PMC 8997449 . PMID 35406591.
234. ^ De, Shaunak; Maity, Abhishek; Goel, Vritti; Shitole, Sanjay; Bhattacharya, Avik (2017). "Predicting the popularity of instagram posts for a lifestyle magazine using deep learning". *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*. pp. 174–177. doi:10.1109/CSCITA.2017.8066548. ISBN 978-1-5090-4381-1. S2CID 35350962.
235. ^ "Colorizing and Restoring Old Images with Deep Learning". *FloydHub Blog*. 13 November 2018. Archived from the original on 11 October 2019. Retrieved 11 October 2019.
236. ^ Schmidt, Uwe; Roth, Stefan. *Shrinkage Fields for Effective Image Restoration* (PDF). Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. Archived (PDF) from the original on 2018-01-02. Retrieved 2018-01-01.
237. ^ Kleanthous, Christos; Chatzis, Sotirios (2020). "Gated Mixture Variational Autoencoders for Value Added Tax audit case selection". *Knowledge-Based Systems*. **188**: 105048. doi:10.1016/j.knosys.2019.105048. S2CID 204092079.
238. ^ Czech, Tomasz (28 June 2018). "Deep learning: the next frontier for money laundering detection". *Global Banking and Finance Review*. Archived from the original on 2018-11-16. Retrieved 2018-07-15.
239. ^ Nuñez, Michael (2023-11-29). "Google DeepMind's materials AI has already discovered 2.2 million new crystals". *VentureBeat*. Retrieved 2023-12-19.
240. ^ Merchant, Amil; Batzner, Simon; Schoenholz, Samuel S.; Aykol, Muratahan; Cheon, Gowoon; Cubuk, Ekin Dogus (December 2023). "Scaling deep learning for materials discovery". *Nature*. **624** (7990): 80–85. Bibcode:2023Natur.624...80M. doi:10.1038/s41586-023-06735-9 . ISSN 1476-4687. PMC 10700131 . PMID 38030720.
241. ^ Peplow, Mark (2023-11-29). "Google AI and robots join forces to build new materials". *Nature*. doi:10.1038/d41586-023-03745-5. PMID 38030771. S2CID 265503872.
242. ^ *a b c* "Army researchers develop new algorithms to train robots". *EurekAlert!*. Archived from the original on 28 August 2018. Retrieved 29 August 2018.
243. ^ Raissi, M.; Perdikaris, P.; Karniadakis, G. E. (2019-02-01). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics*. **378**: 686–707. Bibcode:2019JCoPh.378..686R. doi:10.1016/j.jcp.2018.10.045 . ISSN 0021-9991. OSTI 1595805. S2CID 57379996.
244. ^ Mao, Zhiping; Jagtap, Ameya D.; Karniadakis, George Em (2020-03-01). "Physics-informed neural networks for high-speed flows". *Computer Methods in Applied Mechanics and Engineering*. **360**: 112789. Bibcode:2020CMAME.360k2789M. doi:10.1016/j.cma.2019.112789 . ISSN 0045-7825. S2CID 212755458.
245. ^ Raissi, Maziar; Yazdani, Alireza; Karniadakis, George Em (2020-02-28). "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations". *Science*. **367** (6481): 1026–1030. Bibcode:2020Sci...367.1026R. doi:10.1126/science.aaw4741. PMC 7219083 . PMID 32001523.
246. ^ Han, J.; Jentzen, A.; E, W. (2018). "Solving high-dimensional partial differential equations using deep learning". *Proceedings of the National Academy of Sciences*. **115** (34): 8505–8510. arXiv:1707.02568 . Bibcode:2018PNAS..115.8505H. doi:10.1073/pnas.1718942115 . PMC 6112690 . PMID 30082389.

247. ^ Oktem, Figen S.; Kar, Oğuzhan Fatih; Bezek, Can Deniz; Kamalabadi, Farzad (2021). "High-Resolution Multi-Spectral Imaging With Diffractive Lenses and Learned Reconstruction". *IEEE Transactions on Computational Imaging*. **7**: 489–504. arXiv:2008.11625. doi:10.1109/TCI.2021.3075349. ISSN 2333-9403. S2CID 235340737.
248. ^ Bernhardt, Melanie; Vishnevskiy, Valery; Rau, Richard; Goksel, Orcun (December 2020). "Training Variational Networks With Multidomain Simulations: Speed-of-Sound Image Reconstruction". *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*. **67** (12): 2584–2594. arXiv:2006.14395. doi:10.1109/TUFFC.2020.3010186. ISSN 1525-8955. PMID 32746211. S2CID 220055785.
249. ^ Lam, Remi; Sanchez-Gonzalez, Alvaro; Willson, Matthew; Wirnsberger, Peter; Fortunato, Meire; Alet, Ferran; Ravuri, Suman; Ewalds, Timo; Eaton-Rosen, Zach; Hu, Weihua; Merose, Alexander; Hoyer, Stephan; Holland, George; Vinyals, Oriol; Stott, Jacklynn (2023-12-22). "Learning skillful medium-range global weather forecasting". *Science*. **382** (6677): 1416–1421. arXiv:2212.12794. Bibcode:2023Sci...382.1416L. doi:10.1126/science.adi2336. ISSN 0036-8075. PMID 37962497.
250. ^ Sivakumar, Ramakrishnan (2023-11-27). "GraphCast: A breakthrough in Weather Forecasting". *Medium*. Retrieved 2024-05-19.
251. ^ Galkin, F.; Mamoshina, P.; Kochetov, K.; Sidorenko, D.; Zhavoronkov, A. (2020). "DeepMAge: A Methylation Aging Clock Developed with Deep Learning". *Aging and Disease*. doi:10.14336/AD.
252. ^ Utgoff, P. E.; Stracuzzi, D. J. (2002). "Many-layered learning". *Neural Computation*. **14** (10): 2497–2529. doi:10.1162/08997660260293319. PMID 12396572. S2CID 1119517.
253. ^ Elman, Jeffrey L. (1998). *Rethinking Innateness: A Connectionist Perspective on Development*. MIT Press. ISBN 978-0-262-55030-7.
254. ^ Shrager, J.; Johnson, MH (1996). "Dynamic plasticity influences the emergence of function in a simple cortical array". *Neural Networks*. **9** (7): 1119–1129. doi:10.1016/0893-6080(96)00033-0. PMID 12662587.
255. ^ Quartz, SR; Sejnowski, TJ (1997). "The neural basis of cognitive development: A constructivist manifesto". *Behavioral and Brain Sciences*. **20** (4): 537–556. CiteSeerX 10.1.1.41.7854. doi:10.1017/s0140525x97001581. PMID 10097006. S2CID 5818342.
256. ^ S. Blakeslee, "In brain's early growth, timetable may be critical", *The New York Times, Science Section*, pp. B5–B6, 1995.
257. ^ Mazzoni, P.; Andersen, R. A.; Jordan, M. I. (15 May 1991). "A more biologically plausible learning rule for neural networks". *Proceedings of the National Academy of Sciences*. **88** (10): 4433–4437. Bibcode:1991PNAS...88.4433M. doi:10.1073/pnas.88.10.4433. ISSN 0027-8424. PMC 51674. PMID 1903542.
258. ^ O'Reilly, Randall C. (1 July 1996). "Biologically Plausible Error-Driven Learning Using Local Activation Differences: The Generalized Recirculation Algorithm". *Neural Computation*. **8** (5): 895–938. doi:10.1162/neco.1996.8.5.895. ISSN 0899-7667. S2CID 2376781.
259. ^ Testolin, Alberto; Zorzi, Marco (2016). "Probabilistic Models and Generative Neural Networks: Towards an Unified Framework for Modeling Normal and Impaired Neurocognitive Functions". *Frontiers in Computational Neuroscience*. **10**: 73. doi:10.3389/fncom.2016.00073. ISSN 1662-5188. PMC 4943066. PMID 27468262. S2CID 9868901.
260. ^ Testolin, Alberto; Stoianov, Ivilin; Zorzi, Marco (September 2017). "Letter perception emerges from unsupervised deep learning and recycling of natural image features". *Nature Human Behaviour*. **1** (9): 657–664. doi:10.1038/s41562-017-0186-2. ISSN 2397-3374. PMID 31024135. S2CID 24504018.
261. ^ Buesing, Lars; Bill, Johannes; Nessler, Bernhard; Maass, Wolfgang (3 November 2011). "Neural Dynamics as Sampling: A Model for Stochastic Computation in Recurrent Networks of Spiking Neurons". *PLOS Computational Biology*. **7** (11): e1002211. Bibcode:2011PLSCB...7E2211B. doi:10.1371/journal.pcbi.1002211. ISSN 1553-7358. PMC 3207943. PMID 22096452. S2CID 7504633.

262. ^ Cash, S.; Yuste, R. (February 1999). "Linear summation of excitatory inputs by CA1 pyramidal neurons". *Neuron*. **22** (2): 383–394. doi:10.1016/s0896-6273(00)81098-3 . ISSN 0896-6273. PMID 10069343. S2CID 14663106.
263. ^ Olshausen, B; Field, D (1 August 2004). "Sparse coding of sensory inputs". *Current Opinion in Neurobiology*. **14** (4): 481–487. doi:10.1016/j.conb.2004.07.007. ISSN 0959-4388. PMID 15321069. S2CID 16560320.
264. ^ Yamins, Daniel L K; DiCarlo, James J (March 2016). "Using goal-driven deep learning models to understand sensory cortex". *Nature Neuroscience*. **19** (3): 356–365. doi:10.1038/nn.4244. ISSN 1546-1726. PMID 26906502. S2CID 16970545.
265. ^ Zorzi, Marco; Testolin, Alberto (19 February 2018). "An emergentist perspective on the origin of number sense". *Phil. Trans. R. Soc. B*. **373** (1740): 20170043. doi:10.1098/rstb.2017.0043. ISSN 0962-8436. PMC 5784047 . PMID 29292348. S2CID 39281431.
266. ^ Güçlü, Umut; van Gerven, Marcel A. J. (8 July 2015). "Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream". *Journal of Neuroscience*. **35** (27): 10005–10014. arXiv:1411.6422 . doi:10.1523/jneurosci.5023-14.2015. PMC 6605414 . PMID 26157000.
267. ^ Metz, C. (12 December 2013). "Facebook's 'Deep Learning' Guru Reveals the Future of AI". *Wired*. Archived from the original on 28 March 2014. Retrieved 26 August 2017.
268. ^ Gibney, Elizabeth (2016). "Google AI algorithm masters ancient game of Go". *Nature*. **529** (7587): 445–446. Bibcode:2016Natur.529..445G. doi:10.1038/529445a . PMID 26819021. S2CID 4460235.
269. ^ Silver, David; Huang, Aja; Maddison, Chris J.; Guez, Arthur; Sifre, Laurent; Driessche, George van den; Schrittwieser, Julian; Antonoglou, Ioannis; Panneershelvam, Veda; Lanctot, Marc; Dieleman, Sander; Grewe, Dominik; Nham, John; Kalchbrenner, Nal; Sutskever, Ilya; Lillicrap, Timothy; Leach, Madeleine; Kavukcuoglu, Koray; Graepel, Thore; Hassabis, Demis (28 January 2016). "Mastering the game of Go with deep neural networks and tree search". *Nature*. **529** (7587): 484–489. Bibcode:2016Natur.529..484S. doi:10.1038/nature16961. ISSN 0028-0836. PMID 26819042. S2CID 515925.Closed access icon
270. ^ "A Google DeepMind Algorithm Uses Deep Learning and More to Master the Game of Go | MIT Technology Review". *MIT Technology Review*. Archived from the original on 1 February 2016. Retrieved 30 January 2016.
271. ^ Metz, Cade (6 November 2017). "A.I. Researchers Leave Elon Musk Lab to Begin Robotics Start-Up". *The New York Times*. Archived from the original on 7 July 2019. Retrieved 5 July 2019.
272. ^ Bradley Knox, W.; Stone, Peter (2008). "TAMER: Training an Agent Manually via Evaluative Reinforcement". *2008 7th IEEE International Conference on Development and Learning*. pp. 292–297. doi:10.1109/devlrn.2008.4640845. ISBN 978-1-4244-2661-4. S2CID 5613334.
273. ^ "Talk to the Algorithms: AI Becomes a Faster Learner". *governmentciomedia.com*. 16 May 2018. Archived from the original on 28 August 2018. Retrieved 29 August 2018.
274. ^ Marcus, Gary (14 January 2018). "In defense of skepticism about deep learning". *Gary Marcus*. Archived from the original on 12 October 2018. Retrieved 11 October 2018.
275. ^ Knight, Will (14 March 2017). "DARPA is funding projects that will try to open up AI's black boxes". *MIT Technology Review*. Archived from the original on 4 November 2019. Retrieved 2 November 2017.
276. ^ Alexander Mordvintsev; Christopher Olah; Mike Tyka (17 June 2015). "Inceptionism: Going Deeper into Neural Networks". Google Research Blog. Archived from the original on 3 July 2015. Retrieved 20 June 2015.
277. ^ Alex Hern (18 June 2015). "Yes, androids do dream of electric sheep". *The Guardian*. Archived from the original on 19 June 2015. Retrieved 20 June 2015.
278. ^ *a* *b* *c* Goertzel, Ben (2015). "Are there Deep Reasons Underlying the Pathologies of Today's Deep Learning Algorithms?" (PDF). Archived (PDF) from the original on 2015-05-13. Retrieved 2015-05-10.

279. ^ Nguyen, Anh; Yosinski, Jason; Clune, Jeff (2014). "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images". arXiv:1412.1897 [cs.CV].
280. ^ Szegedy, Christian; Zaremba, Wojciech; Sutskever, Ilya; Bruna, Joan; Erhan, Dumitru; Goodfellow, Ian; Fergus, Rob (2013). "Intriguing properties of neural networks". arXiv:1312.6199 [cs.CV].
281. ^ Zhu, S.C.; Mumford, D. (2006). "A stochastic grammar of images". *Found. Trends Comput. Graph. Vis.* **2** (4): 259–362. CiteSeerX 10.1.1.681.2190 . doi:10.1561/0600000018.
282. ^ Miller, G. A., and N. Chomsky. "Pattern conception". Paper for Conference on pattern detection, University of Michigan. 1957.
283. ^ Eisner, Jason. "Deep Learning of Recursive Structure: Grammar Induction". Archived from the original on 2017-12-30. Retrieved 2015-05-10.
284. ^ "Hackers Have Already Started to Weaponize Artificial Intelligence". *Gizmodo*. 11 September 2017. Archived from the original on 11 October 2019. Retrieved 11 October 2019.
285. ^ "How hackers can force AI to make dumb mistakes". *The Daily Dot*. 18 June 2018. Archived from the original on 11 October 2019. Retrieved 11 October 2019.
286. ^ *a b c d e* "AI Is Easy to Fool—Why That Needs to Change". *Singularity Hub*. 10 October 2017. Archived from the original on 11 October 2017. Retrieved 11 October 2017.
287. ^ Gibney, Elizabeth (2017). "The scientist who spots fake videos". *Nature*. doi:10.1038/nature.2017.22784. Archived from the original on 2017-10-10. Retrieved 2017-10-11.
288. ^ Tubaro, Paola (2020). "Whose intelligence is artificial intelligence?". *Global Dialogue*: 38–39.
289. ^ *a b* Mühlhoff, Rainer (6 November 2019). "Human-aided artificial intelligence: Or, how to run large computations in human brains? Toward a media sociology of machine learning". *New Media & Society*. **22** (10): 1868–1884. doi:10.1177/1461444819885334 . ISSN 1461-4448. S2CID 209363848.

Further reading

[edit]

- Bishop, Christopher M.; Bishop, Hugh (2024). *Deep learning: foundations and concepts*. Springer. ISBN 978-3-031-45467-7.
- Prince, Simon J. D. (2023). *Understanding deep learning*. The MIT Press. ISBN 9780262048644.
- Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016). *Deep Learning*. MIT Press. ISBN 978-0-26203561-3. Archived from the original on 2016-04-16. Retrieved 2021-05-09, introductory textbook.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=1282368290"
Category:

- Deep learning

Hidden categories:

- Webarchive template wayback links
- CS1: long volume value
- CS1 Finnish-language sources (fi)
- CS1 maint: multiple names: authors list
- Articles with short description
- Short description matches Wikidata
- All articles with unsourced statements
- Articles with unsourced statements from August 2024
- Pages using multiple image with auto scaled images

- Articles with unsourced statements from November 2020
- Articles with unsourced statements from July 2016
- CS1 maint: postscript
- Articles prone to spam from June 2015

- This page was last edited on 26 March 2025, at 00:10 (UTC).
-

- Privacy policy
- About Wikipedia
- Disclaimers
- Contact Wikipedia
- Code of Conduct
- Developers
- Statistics
- Cookie statement
- Mobile view

- Wikimedia Foundation
- Powered by MediaWiki

Search

Search Wikipedia

Search

☐ Toggle the table of contents

Deep learning

58 languages Add topic