

Title: Neural network (machine learning)

URL: [https://en.wikipedia.org/wiki/Neural_network_\(machine_learning\)](https://en.wikipedia.org/wiki/Neural_network_(machine_learning))

PageID: 21523

Categories: Category:Artificial neural networks, Category:Bioinspiration, Category:Classification algorithms, Category:Computational neuroscience, Category:Computational statistics, Category:Market research, Category:Mathematical and quantitative methods (economics), Category:Mathematical psychology

Source: Wikipedia (CC BY-SA 4.0).

Supervised learning

Unsupervised learning

Semi-supervised learning

Self-supervised learning

Reinforcement learning

Meta-learning

Online learning

Batch learning

Curriculum learning

Rule-based learning

Neuro-symbolic AI

Neuromorphic engineering

Quantum machine learning

Classification

Generative modeling

Regression

Clustering

Dimensionality reduction

Density estimation

Anomaly detection

Data cleaning

AutoML

Association rules

Semantic analysis

Structured prediction

Feature engineering

Feature learning

Learning to rank

Grammar induction

Ontology learning
Multimodal learning
Apprenticeship learning
Decision trees
Ensembles Bagging Boosting Random forest
Bagging
Boosting
Random forest
k -NN
Linear regression
Naïve Bayes
Artificial neural networks
Logistic regression
Perceptron
Relevance vector machine (RVM)
Support vector machine (SVM)
BIRCH
CURE
Hierarchical
k -means
Fuzzy
Expectation–maximization (EM)
DBSCAN
OPTICS
Mean shift
Factor analysis
CCA
ICA
LDA
NMF
PCA
PGD
t-SNE
SDL
Graphical models Bayes net Conditional random field Hidden Markov
Bayes net
Conditional random field
Hidden Markov

RANSAC
k -NN
Local outlier factor
Isolation forest
Autoencoder
Deep learning
Feedforward neural network
Recurrent neural network LSTM GRU ESN reservoir computing
LSTM
GRU
ESN
reservoir computing
Boltzmann machine Restricted
Restricted
GAN
Diffusion model
SOM
Convolutional neural network U-Net LeNet AlexNet DeepDream
U-Net
LeNet
AlexNet
DeepDream
Neural field Neural radiance field Physics-informed neural networks
Neural radiance field
Physics-informed neural networks
Transformer Vision
Vision
Mamba
Spiking neural network
Memtransistor
Electrochemical RAM (ECRAM)
Q-learning
Policy gradient
SARSA
Temporal difference (TD)
Multi-agent Self-play
Self-play
Active learning

Crowdsourcing

Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

In machine learning , a neural network (also artificial neural network or neural net , abbreviated ANN or NN) is a computational model inspired by the structure and functions of biological neural networks . [1] [2]

A neural network consists of connected units or nodes called artificial neurons , which loosely model the neurons in the brain. Artificial neuron models that mimic biological neurons more closely have also been recently investigated and shown to significantly improve performance. These are connected by edges , which model the synapses in the brain. Each artificial neuron receives signals

from connected neurons, then processes them and sends a signal to other connected neurons. The "signal" is a real number, and the output of each neuron is computed by some non-linear function of the totality of its inputs, called the activation function. The strength of the signal at each connection is determined by a weight, which adjusts during the learning process.

Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly passing through multiple intermediate layers (hidden layers). A network is typically called a deep neural network if it has at least two hidden layers. [3]

Artificial neural networks are used for various tasks, including predictive modeling, adaptive control, and solving problems in artificial intelligence. They can learn from experience, and can derive conclusions from a complex and seemingly unrelated set of information.

Training

Neural networks are typically trained through empirical risk minimization. This method is based on the idea of optimizing the network's parameters to minimize the difference, or empirical risk, between the predicted output and the actual target values in a given dataset. [4] Gradient-based methods such as backpropagation are usually used to estimate the parameters of the network. [4] During the training phase, ANNs learn from labeled training data by iteratively updating their parameters to minimize a defined loss function. [5] This method allows the network to generalize to unseen data.

History

Early work

Today's deep neural networks are based on early work in statistics over 200 years ago. The simplest kind of feedforward neural network (FNN) is a linear network, which consists of a single layer of output nodes with linear activation functions; the inputs are fed directly to the outputs via a series of weights. The sum of the products of the weights and the inputs is calculated at each node. The mean squared errors between these calculated outputs and the given target values are minimized by creating an adjustment to the weights. This technique has been known for over two centuries as the method of least squares or linear regression. It was used as a means of finding a good rough linear fit to a set of points by Legendre (1805) and Gauss (1795) for the prediction of planetary movement. [7] [8] [9] [10] [11]

Historically, digital computers such as the von Neumann model operate via the execution of explicit instructions with access to memory by a number of processors. Some neural networks, on the other hand, originated from efforts to model information processing in biological systems through the framework of connectionism. Unlike the von Neumann model, connectionist computing does not separate memory and processing.

Warren McCulloch and Walter Pitts [12] (1943) considered a non-learning computational model for neural networks. [13] This model paved the way for research to split into two approaches. One approach focused on biological processes while the other focused on the application of neural networks to artificial intelligence.

In the late 1940s, D. O. Hebb [14] proposed a learning hypothesis based on the mechanism of neural plasticity that became known as Hebbian learning. It was used in many early neural networks, such as Rosenblatt's perceptron and the Hopfield network. Farley and Clark [15] (1954) used computational machines to simulate a Hebbian network. Other neural network computational machines were created by Rochester, Holland, Habit and Duda (1956). [16]

In 1958, psychologist Frank Rosenblatt described the perceptron, one of the first implemented artificial neural networks, [17] [18] [19] [20] funded by the United States Office of Naval Research. [21] R. D. Joseph (1960) [22] mentions an even earlier perceptron-like device by Farley and Clark: [10] "Farley and Clark of MIT Lincoln Laboratory actually preceded Rosenblatt in the development of a perceptron-like device." However, "they dropped the subject."

The perceptron raised public excitement for research in Artificial Neural Networks, causing the US government to drastically increase funding. This contributed to "the Golden Age of AI" fueled by the

optimistic claims made by computer scientists regarding the ability of perceptrons to emulate human intelligence. [23]

The first perceptrons did not have adaptive hidden units. However, Joseph (1960) [22] also discussed multilayer perceptrons with an adaptive hidden layer. Rosenblatt (1962) [24] : section 16 cited and adopted these ideas, also crediting work by H. D. Block and B. W. Knight. Unfortunately, these early efforts did not lead to a working learning algorithm for hidden units, i.e., deep learning .

Deep learning breakthroughs in the 1960s and 1970s

Fundamental research was conducted on ANNs in the 1960s and 1970s. 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 the Soviet Union (1965). They regarded it as a form of polynomial regression, [25] or a generalization of Rosenblatt's perceptron. [26] A 1971 paper described a deep network with eight layers trained by this method, [27] 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." [10]

The first deep learning multilayer perceptron trained by stochastic gradient descent [28] was published in 1967 by Shun'ichi Amari . [29] In computer experiments conducted by Amari's student Saito, a five layer MLP with two modifiable layers learned internal representations to classify non-linearly separable pattern classes. [10] Subsequent developments in hardware and hyperparameter tunings have made end-to-end stochastic gradient descent the currently dominant training technique.

In 1969, Kunihiro Fukushima introduced the ReLU (rectified linear unit) activation function. [10] [30] [31] The rectifier has become the most popular activation function for deep learning. [32]

Nevertheless, research stagnated in the United States following the work of Minsky and Papert (1969), [33] who emphasized that basic perceptrons were incapable of processing the exclusive-or circuit. This insight was irrelevant for the deep networks of Ivakhnenko (1965) and Amari (1967).

In 1976 transfer learning was introduced in neural networks learning. [34] [35]

Deep learning architectures for convolutional neural networks (CNNs) with convolutional layers and downsampling layers and weight replication began with the Neocognitron introduced by Kunihiro Fukushima in 1979, though not trained by backpropagation. [36] [37] [38]

Backpropagation

Backpropagation is an efficient application of the chain rule derived by Gottfried Wilhelm Leibniz in 1673 [39] to networks of differentiable nodes. The terminology "back-propagating errors" was actually introduced in 1962 by Rosenblatt, [24] 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 . [40] In 1970, Seppo Linnainmaa published the modern form of backpropagation in his Master's thesis (1970). [41] [42] [10] G.M. Ostrovski et al. republished it in 1971. [43] [44] Paul Werbos applied backpropagation to neural networks in 1982 [45] [46] (his 1974 PhD thesis, reprinted in a 1994 book, [47] did not yet describe the algorithm [44]). In 1986, David E. Rumelhart et al. popularised backpropagation but did not cite the original work. [48]

Convolutional neural networks

Kunihiro Fukushima's convolutional neural network (CNN) architecture of 1979 [36] also introduced max pooling , [49] a popular downsampling procedure for CNNs. CNNs have become an essential tool for computer vision .

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. [50] [51] In 1988, Wei Zhang applied a backpropagation-trained CNN to alphabet recognition. [52] In 1989, Yann LeCun et al. created a CNN called LeNet for recognizing handwritten ZIP codes on mail. Training required 3 days. [53] In 1990, Wei Zhang implemented a CNN on optical computing

hardware. [54] In 1991, a CNN was applied to medical image object segmentation [55] and breast cancer detection in mammograms. [56] 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. [57]

From 1988 onward, [58] [59] the use of neural networks transformed the field of protein structure prediction , in particular when the first cascading networks were trained on profiles (matrices) produced by multiple sequence alignments . [60]

Recurrent neural networks

One origin of RNN was statistical mechanics . In 1972, Shun'ichi Amari proposed to modify the weights of an Ising model by Hebbian learning rule as a model of associative memory , adding in the component of learning. [61] This was popularized as the Hopfield network by John Hopfield (1982). [62] Another origin of RNN was neuroscience. The word "recurrent" is used to describe loop-like structures in anatomy. In 1901, Cajal observed "recurrent semicircles" in the cerebellar cortex . [63] Hebb considered "reverberating circuit" as an explanation for short-term memory. [64] The McCulloch and Pitts paper (1943) considered neural networks that contain cycles, and noted that the current activity of such networks can be affected by activity indefinitely far in the past. [12]

In 1982 a recurrent neural network with an array architecture (rather than a multilayer perceptron architecture), namely a Crossbar Adaptive Array, [65] [66] used direct recurrent connections from the output to the supervisor (teaching) inputs. In addition of computing actions (decisions), it computed internal state evaluations (emotions) of the consequence situations. Eliminating the external supervisor, it introduced the self-learning method in neural networks.

In cognitive psychology, the journal American Psychologist in early 1980's carried out a debate on the relation between cognition and emotion. Zajonc in 1980 stated that emotion is computed first and is independent from cognition, while Lazarus in 1982 stated that cognition is computed first and is inseparable from emotion. [67] [68] In 1982 the Crossbar Adaptive Array gave a neural network model of cognition-emotion relation. [65] [69] It was an example of a debate where an AI system, a recurrent neural network, contributed to an issue in the same time addressed by cognitive psychology.

Two early influential works were the Jordan network (1986) and the Elman network (1990), which applied RNN to study cognitive psychology .

In the 1980s, backpropagation did not work well for deep RNNs. To overcome this problem, in 1991, Jürgen Schmidhuber proposed the "neural sequence chunker" or "neural history compressor" [70] [71] which introduced the important concepts of self-supervised pre-training (the "P" in ChatGPT) and neural knowledge distillation . [10] In 1993, a neural history compressor system solved a "Very Deep Learning" task that required more than 1000 subsequent layers in an RNN unfolded in time. [72]

In 1991, Sepp Hochreiter 's diploma thesis [73] identified and analyzed the vanishing gradient problem [73] [74] and proposed recurrent residual connections to solve it. He and Schmidhuber introduced long short-term memory (LSTM), which set accuracy records in multiple applications domains. [75] [76] This was not yet the modern version of LSTM, which required the forget gate, which was introduced in 1999. [77] It became the default choice for RNN architecture.

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 , [78] restricted Boltzmann machine , [79] Helmholtz machine , [80] and the wake-sleep algorithm . [81] These were designed for unsupervised learning of deep generative models.

Deep learning

Between 2009 and 2012, ANNs began winning prizes in image recognition contests, approaching human level performance on various tasks, initially in pattern recognition and handwriting recognition . [82] [83] In 2011, a CNN named DanNet [84] [85] 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. [38] It then won more contests. [86] [87] They also showed how max-pooling CNNs on GPU improved performance significantly. [88]

In October 2012, AlexNet by Alex Krizhevsky , Ilya Sutskever , and Geoffrey Hinton [89] 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 [90] and Google's Inceptionv3 . [91]

In 2012, Ng and Dean created a network that learned to recognize higher-level concepts, such as cats, only from watching unlabeled images. [92] Unsupervised pre-training and increased computing power from GPUs and distributed computing allowed the use of larger networks, particularly in image and visual recognition problems, which became known as "deep learning". [5]

Radial basis function and wavelet networks were introduced in 2013. These can be shown to offer best approximation properties and have been applied in nonlinear system identification and classification applications. [93]

Generative adversarial network (GAN) (Ian Goodfellow et al., 2014) [94] became state of the art in generative modeling during 2014–2018 period. The GAN principle was originally published in 1991 by Jürgen Schmidhuber who called it "artificial curiosity": two neural networks contest with each other in the form of a zero-sum game , where one network's gain is the other network's loss. [95] [96] 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. Excellent image quality is achieved by Nvidia 's StyleGAN (2018) [97] based on the Progressive GAN by Tero Karras et al. [98] 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 . [99] Diffusion models (2015) [100] eclipsed GANs in generative modeling since then, with systems such as DALL·E 2 (2022) and Stable Diffusion (2022).

In 2014, the state of the art was training "very deep neural network" with 20 to 30 layers. [101] Stacking too many layers led to a steep reduction in training accuracy, [102] known as the "degradation" problem. [103] In 2015, two techniques were developed to train very deep networks: the highway network was published in May 2015, [104] and the residual neural network (ResNet) in December 2015. [105] [106] ResNet behaves like an open-gated Highway Net.

During the 2010s, the seq2seq model was developed, and attention mechanisms were added. It led to the modern Transformer architecture in 2017 in Attention Is All You Need . [107] It requires computation time that is quadratic in the size of the context window. Jürgen Schmidhuber's fast weight controller (1992) [108] scales linearly and was later shown to be equivalent to the unnormalized linear Transformer. [109] [110] [10] Transformers have increasingly become the model of choice for natural language processing . [111] Many modern large language models such as ChatGPT , GPT-4 , and BERT use this architecture.

Models

ANNs began as an attempt to exploit the architecture of the human brain to perform tasks that conventional algorithms had little success with. They soon reoriented towards improving empirical results, abandoning attempts to remain true to their biological precursors. ANNs have the ability to learn and model non-linearities and complex relationships. This is achieved by neurons being connected in various patterns, allowing the output of some neurons to become the input of others. The network forms a directed , weighted graph . [112]

An artificial neural network consists of simulated neurons. Each neuron is connected to other nodes via links like a biological axon-synapse-dendrite connection. All the nodes connected by links take in some data and use it to perform specific operations and tasks on the data. Each link has a weight, determining the strength of one node's influence on another, [113] allowing weights to choose the signal between neurons.

Artificial neurons

ANNs are composed of artificial neurons which are conceptually derived from biological neurons . Each artificial neuron has inputs and produces a single output which can be sent to multiple other neurons. [114] The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons. The outputs of the final output neurons of the neural net accomplish the task, such as recognizing an object in an image. [citation needed]

To find the output of the neuron we take the weighted sum of all the inputs, weighted by the weights of the connections from the inputs to the neuron. We add a bias term to this sum. [115] This weighted sum is sometimes called the activation . This weighted sum is then passed through a (usually nonlinear) activation function to produce the output. The initial inputs are external data, such as images and documents. The ultimate outputs accomplish the task, such as recognizing an object in an image. [116]

Organization

The neurons are typically organized into multiple layers, especially in deep learning. Neurons of one layer connect only to neurons of the immediately preceding and immediately following layers. The layer that receives external data is the input layer . The layer that produces the ultimate result is the output layer . In between them are zero or more hidden layers . Single layer and unlayered networks are also used. Between two layers, multiple connection patterns are possible. They can be 'fully connected', with every neuron in one layer connecting to every neuron in the next layer. They can be pooling , where a group of neurons in one layer connects to a single neuron in the next layer, thereby reducing the number of neurons in that layer. [117] Neurons with only such connections form a directed acyclic graph and are known as feedforward networks . [118] Alternatively, networks that allow connections between neurons in the same or previous layers are known as recurrent networks . [119]

Hyperparameter

A hyperparameter is a constant parameter whose value is set before the learning process begins. Examples of hyperparameters include learning rate , the number of hidden layers and batch size. [citation needed] The values of some hyperparameters can be dependent on those of other hyperparameters. For example, the size of some layers can depend on the overall number of layers. [citation needed]

Learning

Learning is the adaptation of the network to better handle a task by considering sample observations. Learning involves adjusting the weights (and optional thresholds) of the network to improve the accuracy of the result. This is done by minimizing the observed errors. Learning is complete when examining additional observations does not usefully reduce the error rate. Even after learning, the error rate typically does not reach 0. If after learning, the error rate is too high, the network typically must be redesigned. Practically this is done by defining a cost function that is evaluated periodically during learning. As long as its output continues to decline, learning continues. The cost is frequently defined as a statistic whose value can only be approximated. The outputs are actually numbers, so when the error is low, the difference between the output (almost certainly a cat) and the correct answer (cat) is small. Learning attempts to reduce the total of the differences across the observations. Most learning models can be viewed as a straightforward application of optimization theory and statistical estimation . [112] [120]

Learning rate

The learning rate defines the size of the corrective steps that the model takes to adjust for errors in each observation. [121] A high learning rate shortens the training time, but with lower ultimate accuracy, while a lower learning rate takes longer, but with the potential for greater accuracy. Optimizations such as Quickprop are primarily aimed at speeding up error minimization, while other improvements mainly try to increase reliability. In order to avoid oscillation inside the network such as alternating connection weights, and to improve the rate of convergence, refinements use an adaptive learning rate that increases or decreases as appropriate. [122] The concept of momentum allows the balance between the gradient and the previous change to be weighted such that the weight adjustment depends to some degree on the previous change. A momentum close to

0 emphasizes the gradient, while a value close to 1 emphasizes the last change. [citation needed]

Cost function

While it is possible to define a cost function ad hoc , frequently the choice is determined by the function's desirable properties (such as convexity) because it arises from the model (e.g. in a probabilistic model, the model's posterior probability can be used as an inverse cost). [citation needed]

Backpropagation

Backpropagation is a method used to adjust the connection weights to compensate for each error found during learning. The error amount is effectively divided among the connections. Technically, backpropagation calculates the gradient (the derivative) of the cost function associated with a given state with respect to the weights. The weight updates can be done via stochastic gradient descent or other methods, such as extreme learning machines , [123] "no-prop" networks, [124] training without backtracking, [125] "weightless" networks, [126] [127] and non-connectionist neural networks . [citation needed]

Learning paradigms

Machine learning is commonly separated into three main learning paradigms, supervised learning , [128] unsupervised learning [129] and reinforcement learning . [130] Each corresponds to a particular learning task.

Supervised learning

Supervised learning uses a set of paired inputs and desired outputs. The learning task is to produce the desired output for each input. In this case, the cost function is related to eliminating incorrect deductions. [131] A commonly used cost is the mean-squared error , which tries to minimize the average squared error between the network's output and the desired output. Tasks suited for supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation). Supervised learning is also applicable to sequential data (e.g., for handwriting, speech and gesture recognition). This can be thought of as learning with a "teacher", in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.

Unsupervised learning

In unsupervised learning , input data is given along with the cost function, some function of the data x and the network's output. The cost function is dependent on the task (the model domain) and any a priori assumptions (the implicit properties of the model, its parameters and the observed variables). As a trivial example, consider the model $f(x) = a$ where a is a constant and the cost $C = E[(x - f(x))^2]$. Minimizing this cost produces a value of a that is equal to the mean of the data. The cost function can be much more complicated. Its form depends on the application: for example, in compression it could be related to the mutual information between x and $f(x)$, whereas in statistical modeling, it could be related to the posterior probability of the model given the data (note that in both of those examples, those quantities would be maximized rather than minimized). Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering , the estimation of statistical distributions , compression and filtering .

Reinforcement learning

In applications such as playing video games, an actor takes a string of actions, receiving a generally unpredictable response from the environment after each one. The goal is to win the game, i.e., generate the most positive (lowest cost) responses. In reinforcement learning , the aim is to weight the network (devise a policy) to perform actions that minimize long-term (expected cumulative) cost. At each point in time the agent performs an action and the environment generates an observation and an instantaneous cost, according to some (usually unknown) rules. The rules and the long-term cost usually only can be estimated. At any juncture, the agent decides whether to

explore new actions to uncover their costs or to exploit prior learning to proceed more quickly.

Formally, the environment is modeled as a Markov decision process (MDP) with states $s_1, \dots, s_n \in S$ and actions $a_1, \dots, a_m \in A$. Because the state transitions are not known, probability distributions are used instead: the instantaneous cost distribution $P(c_t | s_t)$, the observation distribution $P(x_t | s_t)$ and the transition distribution $P(s_{t+1} | s_t, a_t)$, while a policy is defined as the conditional distribution over actions given the observations. Taken together, the two define a Markov chain (MC). The aim is to discover the lowest-cost MC.

ANNs serve as the learning component in such applications. [132] [133] Dynamic programming coupled with ANNs (giving neurodynamic programming) [134] has been applied to problems such as those involved in vehicle routing , [135] video games, natural resource management [136] [137] and medicine [138] because of ANNs ability to mitigate losses of accuracy even when reducing the discretization grid density for numerically approximating the solution of control problems. Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.

Self-learning

Self-learning in neural networks was introduced in 1982 along with a neural network capable of self-learning named crossbar adaptive array (CAA). [139] It is a system with only one input, situation s , and only one output, action (or behavior) a . It has neither external advice input nor external reinforcement input from the environment. The CAA computes, in a crossbar fashion, both decisions about actions and emotions (feelings) about encountered situations. The system is driven by the interaction between cognition and emotion. [140] Given the memory matrix, $W = ||w(a,s)||$, the crossbar self-learning algorithm in each iteration performs the following computation:

The backpropagated value (secondary reinforcement) is the emotion toward the consequence situation. The CAA exists in two environments, one is behavioral environment where it behaves, and the other is genetic environment, where from it receives initial emotions (only once) about to be encountered situations in the behavioral environment. Having received the genome vector (species vector) from the genetic environment, the CAA will learn a goal-seeking behavior, in the behavioral environment that contains both desirable and undesirable situations. [141]

Neuroevolution

Neuroevolution can create neural network topologies and weights using evolutionary computation . It is competitive with sophisticated gradient descent approaches. [142] [143] One advantage of neuroevolution is that it may be less prone to get caught in "dead ends". [144]

Stochastic neural network

Stochastic neural networks originating from Sherrington–Kirkpatrick models are a type of artificial neural network built by introducing random variations into the network, either by giving the network's artificial neurons stochastic transfer functions [citation needed] , or by giving them stochastic weights. This makes them useful tools for optimization problems, since the random fluctuations help the network escape from local minima . [145] Stochastic neural networks trained using a Bayesian approach are known as Bayesian neural networks . [146]

Topological deep learning

Topological deep learning , first introduced in 2017, [147] is an emerging approach in machine learning that integrates topology with deep neural networks to address highly intricate and high-order data. Initially rooted in algebraic topology , TDL has since evolved into a versatile framework incorporating tools from other mathematical disciplines, such as differential topology and geometric topology . As a successful example of mathematical deep learning, TDL continues to inspire advancements in mathematical artificial intelligence , fostering a mutually beneficial relationship between AI and mathematics .

Other

In a Bayesian framework, a distribution over the set of allowed models is chosen to minimize the cost. Evolutionary methods , [148] gene expression programming , [149] simulated annealing , [150] expectation–maximization , non-parametric methods and particle swarm optimization [151] are other learning algorithms. Convergent recursion is a learning algorithm for cerebellar model articulation controller (CMAC) neural networks. [152] [153]

Modes

Two modes of learning are available: stochastic and batch. In stochastic learning, each input creates a weight adjustment. In batch learning, weights are adjusted based on a batch of inputs, accumulating errors over the batch. Stochastic learning introduces "noise" into the process, using the local gradient calculated from one data point; this reduces the chance of the network getting stuck in local minima. However, batch learning typically yields a faster, more stable descent to a local minimum, since each update is performed in the direction of the batch's average error. A common compromise is to use "mini-batches", small batches with samples in each batch selected stochastically from the entire data set.

Types

ANNs have evolved into a broad family of techniques that have advanced the state of the art across multiple domains. The simplest types have one or more static components, including number of units, number of layers, unit weights and topology . Dynamic types allow one or more of these to evolve via learning. The latter is much more complicated but can shorten learning periods and produce better results. Some types allow/require learning to be "supervised" by the operator, while others operate independently. Some types operate purely in hardware, while others are purely software and run on general purpose computers.

Some of the main breakthroughs include:

Convolutional neural networks that have proven particularly successful in processing visual and other two-dimensional data; [154] [155] where long short-term memory avoids the vanishing gradient problem [156] and can handle signals that have a mix of low and high frequency components aiding large-vocabulary speech recognition, [157] [158] text-to-speech synthesis , [159] [160] [161] and photo-real talking heads; [162]

Competitive networks such as generative adversarial networks in which multiple networks (of varying structure) compete with each other, on tasks such as winning a game [163] or on deceiving the opponent about the authenticity of an input. [94]

Network design

Using artificial neural networks requires an understanding of their characteristics.

Choice of model: This depends on the data representation and the application. Model parameters include the number, type, and connectedness of network layers, as well as the size of each and the connection type (full, pooling, etc.). Overly complex models learn slowly.

Learning algorithm : Numerous trade-offs exist between learning algorithms. Almost any algorithm will work well with the correct hyperparameters [164] for training on a particular data set. However, selecting and tuning an algorithm for training on unseen data requires significant experimentation.

Robustness : If the model, cost function and learning algorithm are selected appropriately, the resulting ANN can become robust.

Neural architecture search (NAS) uses machine learning to automate ANN design. Various approaches to NAS have designed networks that compare well with hand-designed systems. The basic search algorithm is to propose a candidate model, evaluate it against a dataset, and use the results as feedback to teach the NAS network. [165] Available systems include AutoML and AutoKeras. [166] scikit-learn library provides functions to help with building a deep network from scratch. We can then implement a deep network with TensorFlow or Keras .

Hyperparameters must also be defined as part of the design (they are not learned), governing matters such as how many neurons are in each layer, learning rate, step, stride, depth, receptive

field and padding (for CNNs), etc. [167] The Python code snippet provides an overview of the training function, which uses the training dataset, number of hidden layer units, learning rate, and number of iterations as parameters: [citation needed]

Monitoring and concept drift detection of ANNs

When neural networks are deployed in real-world applications, the statistical properties of the input data may change over time, a phenomenon known as concept drift or non-stationarity . Drift can reduce predictive accuracy and lead to unreliable or biased decisions if it is not detected and corrected. In practice, this means that the model's accuracy in deployment may differ substantially from the levels observed during training or cross-validation.

Several strategies have been developed to monitor neural networks for drift and degradation:

Error-based monitoring : comparing current predictions against ground-truth labels when they become available. This approach directly quantifies predictive performance but may be impractical when labels are delayed or costly to obtain.

Data distribution monitoring : detecting changes in the input data distribution using statistical tests, divergence measures, or density-ratio estimation.

Representation monitoring : tracking the distribution of internal embeddings or hidden-layer features. Shifts in the latent representation can indicate nonstationarity even when labels are unavailable. Statistical methods such as statistical process control charts have been adapted for this purpose. [168]

Applications

Because of their ability to model and reproduce nonlinear processes, artificial neural networks have found applications in many disciplines. These include:

Function approximation , [169] or regression analysis , [170] (including time series prediction , fitness approximation , [171] and modeling)

Data processing [172] (including filtering, clustering, blind source separation , [173] and compression)

Nonlinear system identification [93] and control (including vehicle control, trajectory prediction, [174] adaptive control , process control , and natural resource management)

Pattern recognition (including radar systems, face identification , signal classification, [175] novelty detection , 3D reconstruction , [176] object recognition, and sequential decision making [177])

Sequence recognition (including gesture , speech , and handwritten and printed text recognition [178])

Sensor data analysis [179] (including image analysis)

Robotics (including directing manipulators and prostheses)

Data mining (including knowledge discovery in databases)

Finance [180] (such as ex-ante models for specific financial long-run forecasts and artificial financial markets)

Quantum chemistry [181]

General game playing [182]

Generative AI [183]

Data visualization

Machine translation

Social network filtering [184]

E-mail spam filtering

Medical diagnosis [185]

ANNs have been used to diagnose several types of cancers [186] [187] and to distinguish highly invasive cancer cell lines from less invasive lines using only cell shape information. [188] [189]

ANNs have been used to accelerate reliability analysis of infrastructures subject to natural disasters [190] [191] and to predict foundation settlements. [192] It can also be useful to mitigate flood by the use of ANNs for modelling rainfall-runoff. [193] ANNs have also been used for building black-box models in geoscience : hydrology , [194] [195] ocean modelling and coastal engineering , [196] [197] and geomorphology . [198] ANNs have been employed in cybersecurity , with the objective to discriminate between legitimate activities and malicious ones. For example, machine learning has been used for classifying Android malware, [199] for identifying domains belonging to threat actors and for detecting URLs posing a security risk. [200] Research is underway on ANN systems designed for penetration testing, for detecting botnets, [201] credit cards frauds [202] and network intrusions.

ANNs have been proposed as a tool to solve partial differential equations in physics [203] [204] [205] and simulate the properties of many-body open quantum systems . [206] [207] [208] [209] In brain research ANNs have studied short-term behavior of individual neurons , [210] the dynamics of neural circuitry arise from interactions between individual neurons and how behavior can arise from abstract neural modules that represent complete subsystems. Studies considered long-and short-term plasticity of neural systems and their relation to learning and memory from the individual neuron to the system level.

It is possible to create a profile of a user's interests from pictures, using artificial neural networks trained for object recognition. [211]

Beyond their traditional applications, artificial neural networks are increasingly being utilized in interdisciplinary research, such as materials science. For instance, graph neural networks (GNNs) have demonstrated their capability in scaling deep learning for the discovery of new stable materials by efficiently predicting the total energy of crystals. This application underscores the adaptability and potential of ANNs in tackling complex problems beyond the realms of predictive modeling and artificial intelligence, opening new pathways for scientific discovery and innovation. [212]

Theoretical properties

Computational power

The multilayer perceptron is a universal function approximator, as proven by the universal approximation theorem . However, the proof is not constructive regarding the number of neurons required, the network topology, the weights and the learning parameters.

A specific recurrent architecture with rational -valued weights (as opposed to full precision real number-valued weights) has the power of a universal Turing machine , [213] using a finite number of neurons and standard linear connections. Further, the use of irrational values for weights results in a machine with super-Turing power. [214] [215] [failed verification]

Capacity

A model's "capacity" property corresponds to its ability to model any given function. It is related to the amount of information that can be stored in the network and to the notion of complexity.

Two notions of capacity are known by the community. The information capacity and the VC Dimension. The information capacity of a perceptron is intensively discussed in Sir David MacKay 's book [216] which summarizes work by Thomas Cover . [217] The capacity of a network of standard neurons (not convolutional) can be derived by four rules [218] that derive from understanding a neuron as an electrical element. The information capacity captures the functions modelable by the network given any data as input. The second notion, is the VC dimension . VC Dimension uses the principles of measure theory and finds the maximum capacity under the best possible circumstances. This is, given input data in a specific form. As noted in, [216] the VC Dimension for arbitrary inputs is half the information capacity of a perceptron. The VC Dimension for arbitrary points is sometimes referred to as Memory Capacity. [219]

Convergence

Models may not consistently converge on a single solution, firstly because local minima may exist, depending on the cost function and the model. Secondly, the optimization method used might not guarantee to converge when it begins far from any local minimum. Thirdly, for sufficiently large data or parameters, some methods become impractical.

Another issue worthy to mention is that training may cross some saddle point which may lead the convergence to the wrong direction.

The convergence behavior of certain types of ANN architectures are more understood than others. When the width of network approaches to infinity, the ANN is well described by its first order Taylor expansion throughout training, and so inherits the convergence behavior of affine models . [220] [221] Another example is when parameters are small, it is observed that ANNs often fit target functions from low to high frequencies. This behavior is referred to as the spectral bias, or frequency principle, of neural networks. [222] [223] [224] [225] This phenomenon is the opposite to the behavior of some well studied iterative numerical schemes such as Jacobi method . Deeper neural networks have been observed to be more biased towards low frequency functions. [226]

Generalization and statistics

Applications whose goal is to create a system that generalizes well to unseen examples, face the possibility of over-training . This arises in convoluted or over-specified systems when the network capacity significantly exceeds the needed free parameters.

Two approaches address over-training. The first is to use cross-validation and similar techniques to check for the presence of over-training and to select hyperparameters to minimize the generalization error. The second is to use some form of regularization . This concept emerges in a probabilistic (Bayesian) framework, where regularization can be performed by selecting a larger prior probability over simpler models; but also in statistical learning theory, where the goal is to minimize over two quantities: the 'empirical risk' and the 'structural risk', which roughly corresponds to the error over the training set and the predicted error in unseen data due to overfitting.

Supervised neural networks that use a mean squared error (MSE) cost function can use formal statistical methods to determine the confidence of the trained model. The MSE on a validation set can be used as an estimate for variance. This value can then be used to calculate the confidence interval of network output, assuming a normal distribution . A confidence analysis made this way is statistically valid as long as the output probability distribution stays the same and the network is not modified.

By assigning a softmax activation function , a generalization of the logistic function , on the output layer of the neural network (or a softmax component in a component-based network) for categorical target variables, the outputs can be interpreted as posterior probabilities. This is useful in classification as it gives a certainty measure on classifications.

The softmax activation function is:

Criticism

Training

A common criticism of neural networks, particularly in robotics, is that they require too many training samples for real-world operation. [227] Any learning machine needs sufficient representative examples in order to capture the underlying structure that allows it to generalize to new cases. Potential solutions include randomly shuffling training examples, by using a numerical optimization algorithm that does not take too large steps when changing the network connections following an example, grouping examples in so-called mini-batches and/or introducing a recursive least squares algorithm for CMAC . [152] Dean Pomerleau uses a neural network to train a robotic vehicle to drive on multiple types of roads (single lane, multi-lane, dirt, etc.), and a large amount of his research is devoted to extrapolating multiple training scenarios from a single training experience, and preserving past training diversity so that the system does not become overtrained (if, for

example, it is presented with a series of right turns—it should not learn to always turn right). [228]

Theory

A central claim [citation needed] of ANNs is that they embody new and powerful general principles for processing information. These principles are ill-defined . It is often claimed [by whom?] that they are emergent from the network itself. This allows simple statistical association (the basic function of artificial neural networks) to be described as learning or recognition. In 1997, Alexander Dewdney , a former Scientific American columnist, commented that as a result, artificial neural networks have a

something-for-nothing quality, one that imparts a peculiar aura of laziness and a distinct lack of curiosity about just how good these computing systems are. No human hand (or mind) intervenes; solutions are found as if by magic; and no one, it seems, has learned anything. [229]

One response to Dewdney is that neural networks have been successfully used to handle many complex and diverse tasks, ranging from autonomously flying aircraft [230] to detecting credit card fraud to mastering the game of Go .

Technology writer Roger Bridgman commented:

Neural networks, for instance, are in the dock not only because they have been hyped to high heaven, (what hasn't?) but also because you could create a successful net without understanding how it worked: the bunch of numbers that captures its behaviour would in all probability be "an opaque, unreadable table...valueless as a scientific resource".

In spite of his emphatic declaration that science is not technology, Dewdney seems here to pillory neural nets as bad science when most of those devising them are just trying to be good engineers. An unreadable table that a useful machine could read would still be well worth having. [231]

Although it is true that analyzing what has been learned by an artificial neural network is difficult, it is much easier to do so than to analyze what has been learned by a biological neural network. Moreover, recent emphasis on the explainability of AI has contributed towards the development of methods, notably those based on attention mechanisms, for visualizing and explaining learned neural networks. Furthermore, researchers involved in exploring learning algorithms for neural networks are gradually uncovering generic principles that allow a learning machine to be successful. For example, Bengio and LeCun (2007) wrote an article regarding local vs non-local learning, as well as shallow vs deep architecture. [232]

Biological brains use both shallow and deep circuits as reported by brain anatomy, [233] displaying a wide variety of invariance. Weng [234] argued that the brain self-wires largely according to signal statistics and therefore, a serial cascade cannot catch all major statistical dependencies.

Hardware

Large and effective neural networks require considerable computing resources. [235] While the brain has hardware tailored to the task of processing signals through a graph of neurons, simulating even a simplified neuron on von Neumann architecture may consume vast amounts of memory and storage. Furthermore, the designer often needs to transmit signals through many of these connections and their associated neurons – which require enormous CPU power and time. [citation needed]

Some argue that the resurgence of neural networks in the twenty-first century is largely attributable to advances in hardware: from 1991 to 2015, computing power, especially as delivered by GPGPUs (on GPUs), has increased around a million-fold, making the standard backpropagation algorithm feasible for training networks that are several layers deeper than before. [38] The use of accelerators such as FPGAs and GPUs can reduce training times from months to days. [235] [236]

Neuromorphic engineering or a physical neural network addresses the hardware difficulty directly, by constructing non-von-Neumann chips to directly implement neural networks in circuitry. Another type of chip optimized for neural network processing is called a Tensor Processing Unit , or TPU. [

237]

Practical counterexamples

Analyzing what has been learned by an ANN is much easier than analyzing what has been learned by a biological neural network. Furthermore, researchers involved in exploring learning algorithms for neural networks are gradually uncovering general principles that allow a learning machine to be successful. For example, local vs. non-local learning and shallow vs. deep architecture. [238]

Hybrid approaches

Advocates of hybrid models (combining neural networks and symbolic approaches) say that such a mixture can better capture the mechanisms of the human mind. [239] [240]

Dataset bias

Neural networks are dependent on the quality of the data they are trained on, thus low quality data with imbalanced representativeness can lead to the model learning and perpetuating societal biases. [241] [242] These inherited biases become especially critical when the ANNs are integrated into real-world scenarios where the training data may be imbalanced due to the scarcity of data for a specific race, gender or other attribute. [241] This imbalance can result in the model having inadequate representation and understanding of underrepresented groups, leading to discriminatory outcomes that exacerbate societal inequalities, especially in applications like facial recognition , hiring processes, and law enforcement . [242] [243] For example, in 2018, Amazon had to scrap a recruiting tool because the model favored men over women for jobs in software engineering due to the higher number of male workers in the field. [243] The program would penalize any resume with the word "woman" or the name of any women's college. However, the use of synthetic data can help reduce dataset bias and increase representation in datasets. [244]

Gallery

A single-layer feedforward artificial neural network. Arrows originating from x_2 are omitted for clarity. There are p inputs to this network and q outputs. In this system, the value of the q th output, y_q , is calculated as $y_q = K * (\sum_i (x_i * w_{iq}) - b_q)$.

A two-layer feedforward artificial neural network

An artificial neural network

An ANN dependency graph

A single-layer feedforward artificial neural network with 4 inputs, 6 hidden nodes and 2 outputs. Given position state and direction, it outputs wheel based control values.

A two-layer feedforward artificial neural network with 8 inputs, 2x8 hidden nodes and 2 outputs. Given position state, direction and other environment values, it outputs thruster based control values.

Parallel pipeline structure of CMAC neural network. This learning algorithm can converge in one step.

Recent advancements and future directions

Artificial neural networks (ANNs) have undergone significant advancements, particularly in their ability to model complex systems, handle large data sets, and adapt to various types of applications. Their evolution over the past few decades has been marked by a broad range of applications in fields such as image processing, speech recognition, natural language processing, finance, and medicine. [citation needed]

Image processing

In the realm of image processing, ANNs are employed in tasks such as image classification, object recognition, and image segmentation. For instance, deep convolutional neural networks (CNNs) have been important in handwritten digit recognition, achieving state-of-the-art performance. [245] This demonstrates the ability of ANNs to effectively process and interpret complex visual

information, leading to advancements in fields ranging from automated surveillance to medical imaging. [245]

Speech recognition

By modeling speech signals, ANNs are used for tasks like speaker identification and speech-to-text conversion. Deep neural network architectures have introduced significant improvements in large vocabulary continuous speech recognition, outperforming traditional techniques. [245] [246] These advancements have enabled the development of more accurate and efficient voice-activated systems, enhancing user interfaces in technology products. [citation needed]

Natural language processing

In natural language processing, ANNs are used for tasks such as text classification, sentiment analysis, and machine translation. They have enabled the development of models that can accurately translate between languages, understand the context and sentiment in textual data, and categorize text based on content. [245] [246] This has implications for automated customer service, content moderation, and language understanding technologies. [247]

Control systems

In the domain of control systems, ANNs are used to model dynamic systems for tasks such as system identification, control design, and optimization. For instance, deep feedforward neural networks are important in system identification and control applications. [248]

Finance

ANNs are used for stock market prediction and credit scoring :

In investing, ANNs can process vast amounts of financial data, recognize complex patterns, and forecast stock market trends, aiding investors and risk managers in making informed decisions. [245]

In credit scoring, ANNs offer data-driven, personalized assessments of creditworthiness, improving the accuracy of default predictions and automating the lending process. [246]

ANNs require high-quality data and careful tuning, and their "black-box" nature can pose challenges in interpretation. Nevertheless, ongoing advancements suggest that ANNs continue to play a role in finance, offering valuable insights and enhancing risk management strategies . [citation needed]

Medicine

ANNs are able to process and analyze vast medical datasets. They enhance diagnostic accuracy, especially by interpreting complex medical imaging for early disease detection, and by predicting patient outcomes for personalized treatment planning. [246] In drug discovery , ANNs speed up the identification of potential drug candidates and predict their efficacy and safety, significantly reducing development time and costs. [245] Additionally, their application in personalized medicine and healthcare data analysis allows tailored therapies and efficient patient care management. [246] Ongoing research is aimed at addressing remaining challenges such as data privacy and model interpretability, as well as expanding the scope of ANN applications in medicine. [citation needed]

Content creation

ANNs such as generative adversarial networks (GAN) and transformers are used for content creation across numerous industries. [249] This is because deep learning models are able to learn the style of an artist or musician from huge datasets and generate completely new artworks and music compositions. For instance, DALL-E is a deep neural network trained on 650 million pairs of images and texts across the internet that can create artworks based on text entered by the user. [250] In the field of music, transformers are used to create original music for commercials and documentaries through companies such as AIVA and Jukedeck . [251] In the marketing industry, generative models are used to create personalized advertisements for consumers. [249] Additionally, major film companies are partnering with technology companies to analyze the financial success of a film, such as the partnership between Warner Bros and technology company

Cinelytic established in 2020. [252] Furthermore, neural networks have found uses in video game creation, where Non Player Characters (NPCs) can make decisions based on all the characters currently in the game. [253]

See also

ADALINE

Autoencoder

Bio-inspired computing

Blue Brain Project

Catastrophic interference

Cognitive architecture

Connectionist expert system

Connectomics

Deep image prior

Digital morphogenesis

Efficiently updatable neural network

Evolutionary algorithm

Family of curves

Genetic algorithm

Hyperdimensional computing

In situ adaptive tabulation

Large width limits of neural networks

List of machine learning concepts

Memristor

Neural gas

Neural network software

Optical neural network

Parallel distributed processing

Philosophy of artificial intelligence

Predictive analytics

Quantum neural network

Support vector machine

Spiking neural network

Stochastic parrot

Tensor product network

Topological deep learning

References

Bibliography

Bhadeshia H. K. D. H. (1999). "Neural Networks in Materials Science" (PDF) . ISIJ International . 39 (10): 966– 979. doi : 10.2355/isijinternational.39.966 .

Bishop CM (1995). Neural networks for pattern recognition . Clarendon Press. ISBN 978-0-19-853849-3 . OCLC 33101074 .

Borgelt C (2003). Neuro-Fuzzy-Systeme: von den Grundlagen künstlicher Neuronaler Netze zur Kopplung mit Fuzzy-Systemen . Vieweg. ISBN 978-3-528-25265-6 . OCLC 76538146 .

Cybenko G (2006). "Approximation by Superpositions of a Sigmoidal function" . In van Schuppen JH (ed.). Mathematics of Control, Signals, and Systems . Springer International. pp. 303– 314. PDF

Dewdney AK (1997). Yes, we have no neutrons: an eye-opening tour through the twists and turns of bad science . New York: Wiley. ISBN 978-0-471-10806-1 . OCLC 35558945 .

Duda RO, Hart PE, Stork DG (2001). Pattern classification (2 ed.). Wiley. ISBN 978-0-471-05669-0 . OCLC 41347061 .

Egmont-Petersen M, de Ridder D, Handels H (2002). "Image processing with neural networks – a review". Pattern Recognition . 35 (10): 2279– 2301. CiteSeerX 10.1.1.21.5444 . doi : 10.1016/S0031-3203(01)00178-9 .

Fahlman S, Lebiere C (1991). "The Cascade-Correlation Learning Architecture" (PDF) . Archived from the original (PDF) on 3 May 2013 . Retrieved 28 August 2006 . created for National Science Foundation , Contract Number EET-8716324, and Defense Advanced Research Projects Agency (DOD), ARPA Order No. 4976 under Contract F33615-87-C-1499.

created for National Science Foundation , Contract Number EET-8716324, and Defense Advanced Research Projects Agency (DOD), ARPA Order No. 4976 under Contract F33615-87-C-1499.

Gurney K (1997). An introduction to neural networks . UCL Press. ISBN 978-1-85728-673-1 . OCLC 37875698 .

Haykin SS (1999). Neural networks: a comprehensive foundation . Prentice Hall. ISBN 978-0-13-273350-2 . OCLC 38908586 .

Hertz J, Palmer RG, Krogh AS (1991). Introduction to the theory of neural computation . Addison-Wesley. ISBN 978-0-201-51560-2 . OCLC 21522159 .

Information theory, inference, and learning algorithms . Cambridge University Press. 25 September 2003. Bibcode : 2003itil.book.....M . ISBN 978-0-521-64298-9 . OCLC 52377690 .

Kruse R, Borgelt C, Klawonn F, Moewes C, Steinbrecher M, Held P (2013). Computational intelligence: a methodological introduction . Springer. ISBN 978-1-4471-5012-1 . OCLC 837524179 .

Lawrence J (1994). Introduction to neural networks: design, theory and applications . California Scientific Software. ISBN 978-1-883157-00-5 . OCLC 32179420 .

Masters T (1994). Signal and image processing with neural networks: a C++ sourcebook . J. Wiley. ISBN 978-0-471-04963-0 . OCLC 29877717 .

Maurer H (2021). Cognitive science: integrative synchronization mechanisms in cognitive neuroarchitectures of the modern connectionism . CRC Press. doi : 10.1201/9781351043526 . ISBN 978-1-351-04352-6 . S2CID 242963768 .

Ripley BD (2007). Pattern Recognition and Neural Networks . Cambridge University Press. ISBN 978-0-521-71770-0 .

Siegelmann H, Sontag ED (1994). "Analog computation via neural networks" . Theoretical Computer Science . 131 (2): 331– 360. doi : 10.1016/0304-3975(94)90178-3 . S2CID 2456483 .

Smith M (1993). Neural networks for statistical modeling . Van Nostrand Reinhold. ISBN 978-0-442-01310-3 . OCLC 27145760 .

Wasserman PD (1993). Advanced methods in neural computing . Van Nostrand Reinhold. ISBN 978-0-442-00461-3 . OCLC 27429729 .

Wilson H (2018). Artificial intelligence . Grey House Publishing. ISBN 978-1-68217-867-6 .

External links

A Brief Introduction to Neural Networks (D. Kriesel) – Illustrated, bilingual manuscript about artificial neural networks; Topics so far: Perceptrons, Backpropagation, Radial Basis Functions, Recurrent Neural Networks, Self Organizing Maps, Hopfield Networks.

Review of Neural Networks in Materials Science Archived 7 June 2015 at the Wayback Machine

Artificial Neural Networks Tutorial in three languages (Univ. Politécnica de Madrid)

Another introduction to ANN

Next Generation of Neural Networks Archived 24 January 2011 at the Wayback Machine – Google Tech Talks

Performance of Neural Networks

Neural Networks and Information Archived 9 July 2009 at the Wayback Machine

Sanderson G (5 October 2017). "But what is a Neural Network?" . 3Blue1Brown . Archived from the original on 7 November 2021 – via YouTube .

v

t

e

History timeline

timeline

Companies

Projects

Parameter Hyperparameter

Hyperparameter

Loss functions

Regression Bias–variance tradeoff Double descent Overfitting

Bias–variance tradeoff

Double descent

Overfitting

Clustering

Gradient descent SGD Quasi-Newton method Conjugate gradient method

SGD

Quasi-Newton method

Conjugate gradient method

Backpropagation

Attention

Convolution

Normalization Batchnorm

Batchnorm

Activation Softmax Sigmoid Rectifier

Softmax

Sigmoid
Rectifier
Gating
Weight initialization
Regularization
Datasets Augmentation
Augmentation
Prompt engineering
Reinforcement learning Q-learning SARSA Imitation Policy gradient
Q-learning
SARSA
Imitation
Policy gradient
Diffusion
Latent diffusion model
Autoregression
Adversary
RAG
Uncanny valley
RLHF
Self-supervised learning
Reflection
Recursive self-improvement
Hallucination
Word embedding
Vibe coding
Machine learning In-context learning
In-context learning
Artificial neural network Deep learning
Deep learning
Language model Large language model NMT
Large language model
NMT
Reasoning language model
Model Context Protocol
Intelligent agent
Artificial human companion
Humanity's Last Exam

Artificial general intelligence (AGI)

AlexNet

WaveNet

Human image synthesis

HWR

OCR

Computer vision

Speech synthesis 15.ai ElevenLabs

15.ai

ElevenLabs

Speech recognition Whisper

Whisper

Facial recognition

AlphaFold

Text-to-image models Aurora DALL-E Firefly Flux Ideogram Imagen Midjourney Recraft Stable Diffusion

Aurora

DALL-E

Firefly

Flux

Ideogram

Imagen

Midjourney

Recraft

Stable Diffusion

Text-to-video models Dream Machine Runway Gen Hailuo AI Kling Sora Veo

Dream Machine

Runway Gen

Hailuo AI

Kling

Sora

Veo

Music generation Riffusion Suno AI Udio

Riffusion

Suno AI

Udio

Word2vec

Seq2seq

GloVe

BERT

T5

Llama

Chinchilla AI

PaLM

GPT 1 2 3 J ChatGPT 4 4o o1 o3 4.5 4.1 o4-mini 5

1

2

3

J

ChatGPT

4

4o

o1

o3

4.5

4.1

o4-mini

5

Claude

Gemini Gemini (language model) Gemma

Gemini (language model)

Gemma

Grok

LaMDA

BLOOM

DBRX

Project Debater

IBM Watson

IBM Watsonx

Granite

PanGu- Σ

DeepSeek

Qwen

AlphaGo

AlphaZero

OpenAI Five

Self-driving car
MuZero
Action selection AutoGPT
AutoGPT
Robot control
Alan Turing
Warren Sturgis McCulloch
Walter Pitts
John von Neumann
Claude Shannon
Shun'ichi Amari
Kunihiko Fukushima
Takeo Kanade
Marvin Minsky
John McCarthy
Nathaniel Rochester
Allen Newell
Cliff Shaw
Herbert A. Simon
Oliver Selfridge
Frank Rosenblatt
Bernard Widrow
Joseph Weizenbaum
Seymour Papert
Seppo Linnainmaa
Paul Werbos
Geoffrey Hinton
John Hopfield
Jürgen Schmidhuber
Yann LeCun
Yoshua Bengio
Lotfi A. Zadeh
Stephen Grossberg
Alex Graves
James Goodnight
Andrew Ng
Fei-Fei Li
Alex Krizhevsky

Ilya Sutskever
Oriol Vinyals
Quoc V. Le
Ian Goodfellow
Demis Hassabis
David Silver
Andrej Karpathy
Ashish Vaswani
Noam Shazeer
Aidan Gomez
John Schulman
Mustafa Suleyman
Jan Leike
Daniel Kokotajlo
François Chollet
Neural Turing machine
Differentiable neural computer
Transformer Vision transformer (ViT)
Vision transformer (ViT)
Recurrent neural network (RNN)
Long short-term memory (LSTM)
Gated recurrent unit (GRU)
Echo state network
Multilayer perceptron (MLP)
Convolutional neural network (CNN)
Residual neural network (RNN)
Highway network
Mamba
Autoencoder
Variational autoencoder (VAE)
Generative adversarial network (GAN)
Graph neural network (GNN)
Category
v
t
e
Emergence
Self-organization

Social dynamics
Collective intelligence
Collective action
Collective consciousness
Self-organized criticality
Herd mentality
Phase transition
Agent-based modelling
Synchronization
Ant colony optimization
Particle swarm optimization
Swarm behaviour
Artificial neural network
Evolutionary computation
Genetic algorithms
Genetic programming
Artificial life
Machine learning
Evolutionary developmental biology
Artificial intelligence
Evolutionary robotics
Evolvability
Prisoner's dilemma
Rational choice theory
Bounded rationality
Evolutionary game theory
Social network analysis
Small-world networks
Centrality
Motifs
Graph theory
Scaling
Robustness
Systems biology
Dynamic networks
Adaptive networks
Time series analysis
Ordinary differential equations

Phase space
Attractor
Population dynamics
Chaos
Multistability
Bifurcation
Coupled map lattices
Reaction-diffusion systems
Partial differential equations
Dissipative structures
Percolation
Cellular automata
Spatial ecology
Self-replication
Geomorphology
Homeostasis
Operationalization
Feedback
Self-reference
Goal-oriented
System dynamics
Sensemaking
Entropy
Cybernetics
Autopoiesis
Information theory
Computation theory
v
t
e
Adaptive control
Control theory
Digital control
Energy-shaping control
Fuzzy control
Hybrid control
Intelligent control
Model predictive control

Multivariable control
Neural control
Nonlinear control
Optimal control
Real-time control
Robust control
Stochastic control
Bode plot
Block diagram
Closed-loop transfer function
Controllability
Fourier transform
Frequency response
Laplace transform
Negative feedback
Observability
Performance
Positive feedback
Root locus method
Servomechanism
Signal-flow graph
State space representation
Stability theory
Steady state analysis & design
System dynamics
Transfer function
Discrete-time signal
Digital signal processing
Quantization
Real time software
Sampled data
System identification
Z-transform
Artificial neural network
Coefficient diagram method
Control reconfiguration
Distributed parameter systems
Fractional-order control

Fuzzy logic
H-infinity loop-shaping
Hankel singular value
Kalman filter
Krener's theorem
Least squares
Lyapunov stability
Minor loop feedback
Perceptual control theory
State observer
Vector control
Embedded
Closed-loop
Lead-lag compensator
Numerical
PID
Programmable logic
Automation and Remote Control
Distributed control system
Electric motors
Industrial control systems
Mechatronics
Motion control
Process control
Robotics
Supervisory control and data acquisition (SCADA)
v
t
e
Outline
History
Behavioral epigenetics
Behavioral genetics
Brain mapping
Brain-reading
Cellular neuroscience
Computational neuroscience
Connectomics

Imaging genetics
Integrative neuroscience
Molecular neuroscience
Neural decoding
Neural engineering
Neuroanatomy
Neurobiology
Neurochemistry
Neuroendocrinology
Neurogenetics
Neuroinformatics
Neurometrics
Neuromorphology
Neurophysics
Neurophysiology
Systems neuroscience
Behavioral neurology
Clinical neurophysiology
Epileptology
Neurocardiology
Neuroepidemiology
Neurogastroenterology
Neuroimmunology
Neurointensive care
Neurology
Neuro-oncology
Neuro-ophthalmology
Neuropathology
Neuropharmacology
Neuroprosthetics
Neuropsychiatry
Neuroradiology
Neurorehabilitation
Neurosurgery
Neurotology
Neurovirology
Nutritional neuroscience
Psychiatry

Affective neuroscience
Behavioral neuroscience
Chronobiology
Molecular cellular cognition
Motor control
Neurolinguistics
Neuropsychology
Sensory neuroscience
Social cognitive neuroscience
Consumer neuroscience
Cultural neuroscience
Educational neuroscience
Evolutionary neuroscience
Global neurosurgery
Neuroanthropology
Neural engineering
Neurobiotics
Neurocinema
Neurocriminology
Neuroeconomics
Neuroepistemology
Neuroesthetics
Neuroethics
Neuroethology
Neurohistory
Neurolaw
Neuromarketing
Neuromorphic engineering
Neuroscience of music
Neurophenomenology
Neurophilosophy
Neuropolitics
Neurorobotics
Neurotheology
Paleoneurobiology
Social neuroscience
Brain–computer interface
Development of the nervous system

Neural network (artificial)
Neural network (biological)
Detection theory
Intraoperative neurophysiological monitoring
Neurochip
Neurodegenerative disease
Neurodevelopmental disorder
Neurodiversity
Neurogenesis
Neuroimaging
Neuroimmune system
Neuromanagement
Neuromodulation
Neuroplasticity
Neurotechnology
Neurotoxin
Self-awareness
Category
Commons
v
t
e
History of self-driving cars
Impact of self-driving cars
Intelligent transportation system
Context-aware pervasive systems
Mobile computing
Smart, connected products
Ubiquitous computing
Ambient intelligence
Internet of things
List of self-driving system suppliers
Lane departure warning system
Automatic parking
Automated emergency braking system
Collision avoidance system
Cruise control Adaptive cruise control
Adaptive cruise control

Advanced driver-assistance system
Driver drowsiness detection
Intelligent speed adaptation
Blind spot monitor
Automated Lane Keeping Systems
Vehicular ad hoc network (V2V)
Connected car
Automotive navigation system
VaMP (1994)
Spirit of Berlin (2007)
General Motors EN-V (2010)
MadeInGermany (2011)
Waymo , formerly Google Car (2012)
Tesla Model S with Autopilot (2015)
LUTZ Pathfinder (2015)
Avride (2017)
Honda Legend (2021)
Automated guideway transit
CAVForth
ParkShuttle
Navia shuttle
NuTonomy taxi
Freightliner Inspiration
Driverless tractor
Self-driving truck
Mobility as a service
Legislation
IEEE 802.11p
Safe speed automotive common law
Automated lane keeping system (unece regulation 157)
Regulation (EU) 2019/2144
Radar
Laser
LIDAR
Artificial neural network
Computer stereo vision
Image recognition
Dedicated short-range communications

Real-time Control System

rFpro

Eye tracking

Radio-frequency identification

Automotive navigation system

American Center for Mobility

DAVI

European Land-Robot Trial

Navlab

DARPA Grand Challenge

VisLab Intercontinental Autonomous Challenge

Eureka Prometheus Project

IEEE Intelligent Transportation Systems Society

Harold Goddijn

Alberto Broggi

Anthony Levandowski

GND

United States

Japan

Czech Republic

Latvia

Israel

Yale LUX