VIDHYADEEP INSTITUTE OF COMPUTER & INFORMATION TECHNOLOGY,ANITA-KIM.



**VEER NARAMD SOUTH GUJARAT UNIVERSITY**

**PROJECT REPORT**

**ON**

**“Neural Network”**

**AS A PARTIAL REQUIREMENT FOR THE DEGREE**

**OF**

**BECHOLOR OF COMPUTER APPLICATION( BCA) Year 2023-2024**

**(TYBCA 6th Semester)**

Internal Guide:- SubmittedBy:-

**Mr.Partik Patel Sunny Javiya**



**VIDHYADEEP INSTITUTE OF COMPUTER & INFORMATION TECHNOLOGY**

# CERTIFICATE

This is to certify that project report,submitted for the project entitled “**Neural Network”** has been carried out by **“Sunny Javiya”** at BCA department of vidhyadeep institute of computer & information technology, kim for partial fulfilment of BCA degree to be awarded by veer narmad south Gujarat university. this project work has been carried out under my supervision and is to my satisfaction.

Place :- kim

Date :-

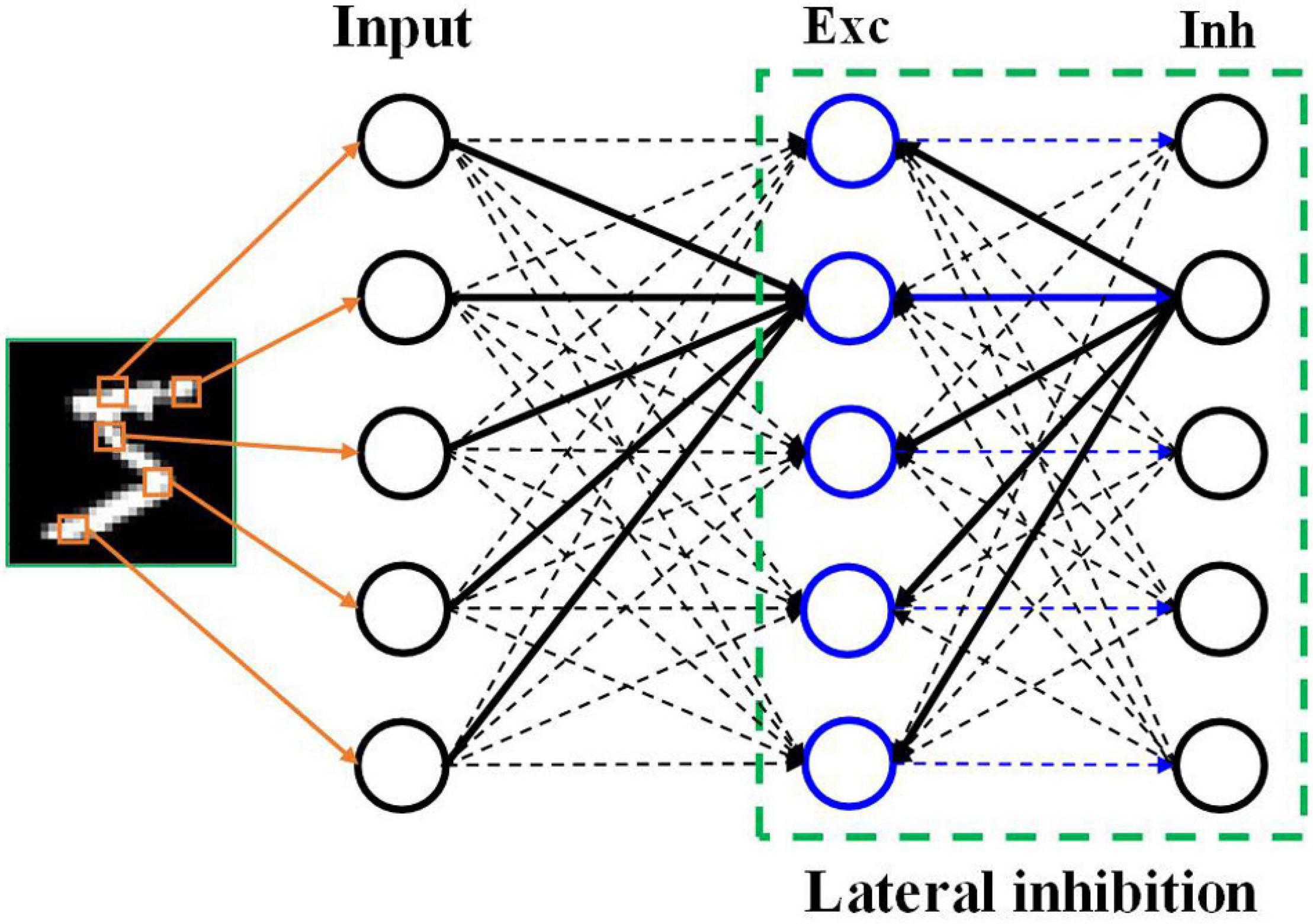
**Internal Guide I/C Principal**

MR. Partik Patel Dr..Himansu Patel

# Neural Network

In machine learning, a neural network (also artificial neural network or neural net, abbreviated ANN or NN) is a model inspired by the neuronal organization found in the biological neural networks in animal brains.An ANN is made of connected units or nodes called artificial neurons, which loosely model the neurons in a brain. These are connected by edges, which model the synapses in a brain. An 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 sum of its inputs, called the activation function. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection.

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 2 hidden layers.Artificial neural networks are used for predictive modeling, adaptive control, and other applications where they can be trained via a dataset. They are also used to solve problems in artificial intelligence. Networks can learn from experience, and can derive conclusions from a complex and seemingly unrelated set of information.



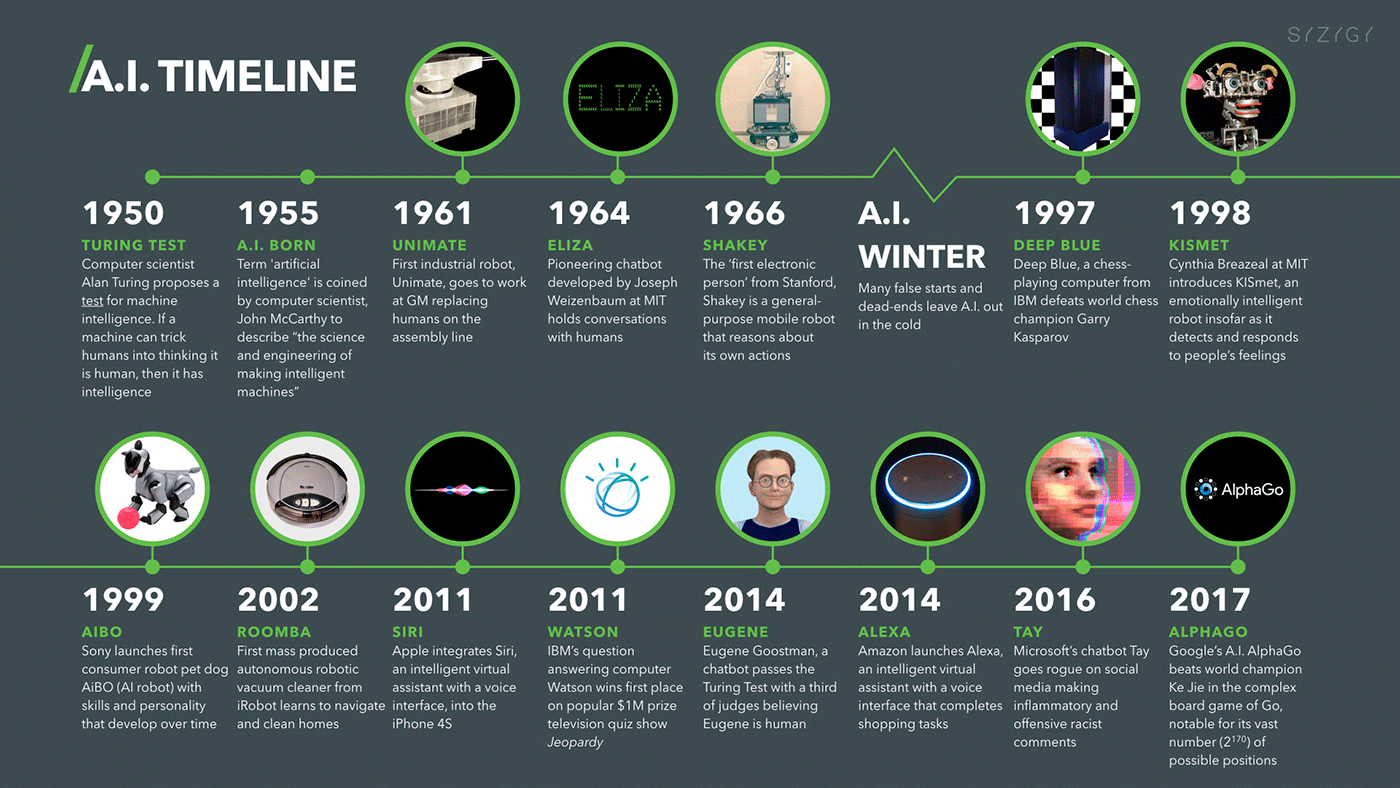
[This Photo](https://www.frontiersin.org/articles/10.3389/fnins.2021.638474/full) by Unknown Author is licensed under [CC BY](https://creativecommons.org/licenses/by/3.0/)

# == 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. Gradient based methods such as backpropagation are usually used to estimate the parameters of the network. During the training phase, ANNs learn from labeled training data by iteratively updating their parameters to minimize a defined loss function. This method allows the network to generalize to unseen data.

# == History ==

Historically, digital computers evolved from the von Neumann model, and operate via the execution of explicit instructions via access to memory by a number of processors. 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.



[This Photo](https://www.futuremagazineonline.com/es/que-es-realmente-la-inteligencia-artificial-y-por-que-puede-dejarte-sin-trabajo/) by Unknown Author is licensed under [CC BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/3.0/)

The simplest kind of feedforward neural network (FNN) is a linear network, which consists of a single layer of output nodes; 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.Warren McCulloch and Walter Pitts (1943) also considered a non-learning computational model for neural networks.In the late 1940s, D. O. Hebb created a learning hypothesis based on the mechanism of neural plasticity that became known as Hebbian learning. Hebbian learning is considered to be a 'typical' unsupervised learning rule and its later variants were early models for long term potentiation. These ideas started being applied to computational models in 1948 with Turing's "unorganized machines". Farley and Wesley A. Clark were the first to simulate a Hebbian network in 1954 at MIT. They used computational machines, then called "calculators". Other neural network computational machines were created by Rochester, Holland, Habit, and Duda in 1956. In 1958, psychologist Frank Rosenblatt invented the perceptron, the first implemented artificial neural network, funded by the United States Office of Naval Research.

The invention of the perceptron raised public excitement for research in Artificial Neural Networks, causing the US government to drastically increase funding into deep learning research. This led to "the golden age of AI" fueled by the optimistic claims made by computer scientists regarding the ability of perceptrons to emulate human intelligence. For example, in 1957 Herbert Simon famously said:It is not my aim to surprise or shock you—but the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until—in a visible future—the range of problems they can handle will be coextensive with the range to which the human mind has been applied.However, this wasn't the case as research stagnated in the United States following the work of Minsky and Papert (1969), who discovered that basic perceptrons were incapable of processing the exclusive-or circuit and that computers lacked sufficient power to train useful neural networks. This, along with other factors such as the 1973 Lighthill report by James Lighthill stating that research in Artificial Intelligence has not "produced the major impact that was then promised," shutting funding in research into the field of AI in all but two universities in the UK and in many major institutions across the world. This ushered an era called the AI Winter with reduced research into connectionism due to a decrease in government funding and an increased stress on symbolic artificial intelligence in the United States and other Western countries.During the AI Winter era, however, research outside the United States continued, especially in Eastern Europe. By the time Minsky and Papert's book on Perceptrons came out, methods for training multilayer perceptrons (MLPs) were already known. The first deep learning MLP was published by Alexey Grigorevich Ivakhnenko and Valentin Lapa in 1965, as the Group Method of Data Handling. The first deep learning MLP trained by stochastic gradient descent was published in 1967 by Shun'ichi Amari. In computer experiments conducted by Amari's student Saito, a five layer MLP with two modifiable layers learned useful internal representations to classify non-linearily separable pattern classes.Self-organizing maps (SOMs) were described by Teuvo Kohonen in 1982. SOMs are neurophysiologically inspired neural networks that learn low-dimensional representations of high-dimensional data while preserving the topological structure of the data. They are trained using competitive learning.The convolutional neural network (CNN) architecture with convolutional layers and downsampling layers was introduced by Kunihiko Fukushima in 1980. He called it the neocognitron. In 1969, he also introduced the ReLU (rectified linear unit) activation function. The rectifier has become the most popular activation function for CNNs and deep neural networks in general. CNNs have become an essential tool for computer vision.

A key in later advances in artificial neural network research was the backpropagation algorithm, an efficient application of the Leibniz chain rule (1673) to networks of differentiable nodes. It is also known as

the reverse mode of automatic differentiation or reverse accumulation, due to Seppo Linnainmaa (1970). The term "back-propagating errors" was introduced in 1962 by Frank Rosenblatt, but he did not have an implementation of this procedure, although Henry J. Kelley and Bryson had dynamic programming based continuous precursors of backpropagation already in 1960–61 in the context of control theory.

In 1973, Dreyfus used backpropagation to adapt parameters of controllers in proportion to error gradients.

In 1982, Paul Werbos applied backpropagation to MLPs in the way that has become standard. In 1986 Rumelhart, Hinton and Williams showed that backpropagation learned interesting internal representations of words as feature vectors when trained to predict the next word in a sequence.In the late 1970s to early 1980s, interest briefly emerged in theoretically investigating the Ising model created by Wilhelm Lenz (1920) and Ernst Ising (1925)

in relation to Cayley tree topologies and large neural networks.

The Ising model is essentially a non-learning artificial recurrent neural network (RNN) consisting of neuron-like threshold elements.

In 1972, Shun'ichi Amari described an adaptive version of this architecture,

In 1981, the Ising model was solved exactly by Peter Barth for the general case of closed Cayley trees (with loops) with an arbitrary branching ratio

and found to exhibit unusual phase transition behavior in its local-apex and long-range site-site correlations.John Hopfield popularised this architecture in 1982,

and it is now known as a Hopfield network.

The time delay neural network (TDNN) of Alex Waibel (1987) combined convolutions and weight sharing and backpropagation. In 1988, Wei Zhang et al. applied backpropagation to a CNN (a simplified Neocognitron with convolutional interconnections between the image feature layers and the last fully connected layer) for alphabet recognition. In 1989, Yann LeCun et al. trained a CNN to recognize handwritten ZIP codes on mail.

In 1992, max-pooling for CNNs was introduced by Juan Weng et al. to help with least-shift invariance and tolerance to deformation to aid 3D object recognition.

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.

From 1988 onward, 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.In 1991, Sepp Hochreiter's diploma thesis identified and analyzed the vanishing gradient problem and proposed recurrent residual connections to solve it. His thesis was called "one of the most important documents in the history of machine learning" by his supervisor Juergen Schmidhuber.In 1991, Juergen Schmidhuber 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. 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 1992, Juergen Schmidhuber proposed a hierarchy of RNNs pre-trained one level at a time by self-supervised learning. It uses predictive coding to learn internal representations at multiple self-organizing time scales. This can substantially facilitate downstream deep learning. The RNN hierarchy can be collapsed into a single RNN, by distilling a higher level chunker network into a lower level automatizer network. In the same year he also published an alternative to RNNs which is a precursor of a linear Transformer. It introduces the concept internal spotlights of attention: a slow feedforward neural network learns by gradient descent to control the fast weights of another neural network through outer products of self-generated activation patterns.

The development of metal–oxide–semiconductor (MOS) very-large-scale integration (VLSI), in the form of complementary MOS (CMOS) technology, enabled increasing MOS transistor counts in digital electronics. This provided more processing power for the development of practical artificial neural networks in the 1980s.Neural networks' early successes included predicting the stock market and in 1995 a (mostly) self-driving car.1997, Sepp Hochreite and Juergen Schmidhuber introduced the deep learning method called long short-term memory (LSTM), published in Neural Computation. LSTM recurrent neural networks can learn "very deep learning" tasks with long credit assignment paths that require memories of events that happened thousands of discrete time steps before. The "vanilla LSTM" with forget gate was introduced in 1999 by Felix Gers, Schmidhuber and Fred Cummins.Geoffrey Hinton et al. (2006) proposed learning a high-level representation using successive layers of binary or real-valued latent variables with a restricted Boltzmann machine to model each layer. In 2012, Ng and Dean created a network that learned to recognize higher-level concepts, such as cats, only from watching unlabeled images. 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".Variants of the back-propagation algorithm, as well as unsupervised methods by Geoff Hinton and colleagues at the University of Toronto, can be used to train deep, highly nonlinear neural architectures, similar to the 1980 Neocognitron by Kunihiko Fukushima, and the "standard architecture of vision", inspired by the simple and complex cells identified by David H. Hubel and Torsten Wiesel in the primary visual cortex.

Computational devices have been created in CMOS for both biophysical simulation and neuromorphic computing. More recent efforts show promise for creating nanodevices for very large scale principal components analyses and convolution. If successful, these efforts could usher in a new era of neural computing that is a step beyond digital computing, because it depends on learning rather than programming and because it is fundamentally analog rather than digital even though the first instantiations may in fact be with CMOS digital devices.

Ciresan and colleagues (2010) showed that despite the vanishing gradient problem, GPUs make backpropagation feasible for many-layered feedforward neural networks. 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. For example, the bi-directional and multi-dimensional long short-term memory (LSTM) of Graves et al. won three competitions in connected handwriting recognition in 2009 without any prior knowledge about the three languages to be learned.Ciresan and colleagues built the first pattern recognizers to achieve human-competitive/superhuman performance on benchmarks such as traffic sign recognition (IJCNN 2012).

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.In 2014, the adversarial network principle was used in a generative adversarial network (GAN) by Ian Goodfellow et al. Here the adversarial network (discriminator) outputs a value between 1 and 0 depending on the likelihood of the first network's (generator) output is in a given set. This can be used to create realistic deepfakes.

Excellent image quality is achieved by Nvidia's StyleGAN (2018) based on the Progressive GAN by Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Here the GAN generator is grown from small to large scale in a pyramidal fashion.

In 2015, Rupesh Kumar Srivastava, Klaus Greff, and Schmidhuber used the LSTM principle to create the Highway network, a feedforward neural network with hundreds of layers, much deeper than previous networks. 7 months later, Kaiming He, Xiangyu Zhang; Shaoqing Ren, and Jian Sun won the ImageNet 2015 competition with an open-gated or gateless Highway network variant called Residual neural network.In 2017, Ashish Vaswani et al. introduced the modern Transformer architecture in their paper "Attention Is All You Need."

It combines this with a softmax operator and a projection matrix.

Transformers have increasingly become the model of choice for natural language processing. Many modern large language models such as ChatGPT, GPT-4, and BERT use it. Transformers are also increasingly being used in computer vision.Ramenzanpour et al. showed in 2020 that analytical and computational techniques derived from statistical physics of disordered systems can be extended to large-scale problems, including machine learning, e.g., to analyze the weight space of deep neural networks.

**== 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.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, allowing weights to choose the signal between neurons.

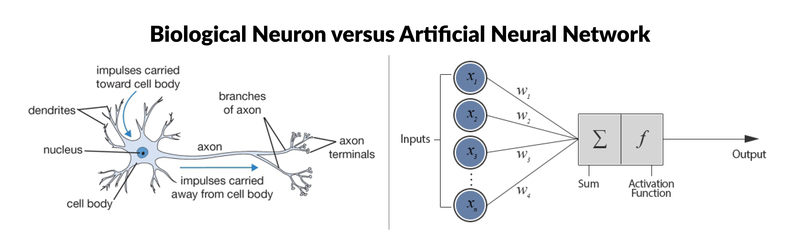


[This Photo](https://technofaq.org/posts/2018/01/the-role-of-big-data-in-strengthening-machine-learning-projects/) by Unknown Author is licensed under [CC BY-SA-NC](https://creativecommons.org/licenses/by-nc-sa/3.0/)

# === 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. 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.

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. 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.

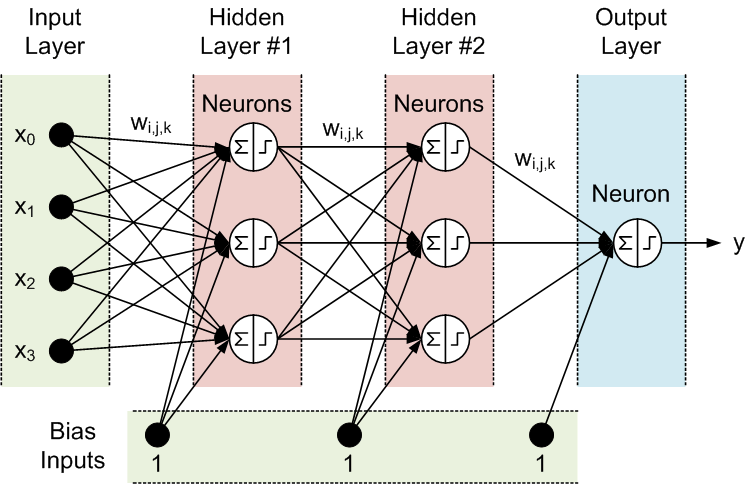


[This Photo](https://devopedia.org/artificial-neural-network) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

# === 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. Neurons with only such connections form a directed acyclic graph and are known as feedforward networks. Alternatively, networks that allow connections between neurons in the same or previous layers are known as recurrent networks.

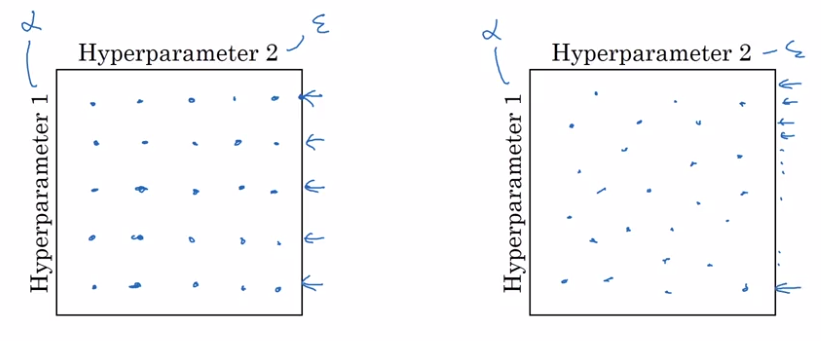
Neurons organization



[This Photo](http://stackoverflow.com/questions/28288489/neural-networks-does-the-input-layer-consist-of-neurons) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

# === Hyperparameter ===

A hyperparameter is a constant parameter whose value is set before the learning process begins. The values of parameters are derived via learning. Examples of hyperparameters include learning rate, the number of hidden layers and batch size. 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.



[This Photo](https://x-wei.github.io/Ng_DLMooc_c2wk3.html) by Unknown Author is licensed under [CC BY-SA-NC](https://creativecommons.org/licenses/by-nc-sa/3.0/)

# === 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.

**==== Learning rate ====**

The learning rate defines the size of the corrective steps that the model takes to adjust for errors in each observation. 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. 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.

# ==== 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) or because it arises from the model (e.g. in a probabilistic model the model's posterior probability can be used as an inverse cost).

# ==== 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, backprop 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, "no-prop" networks, training without backtracking, "weightless" networks, and non-connectionist neural networks.



[This Photo](http://stackoverflow.com/questions/39090431/backpropagation-neural-networks) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

**=== Learning paradigms ===**

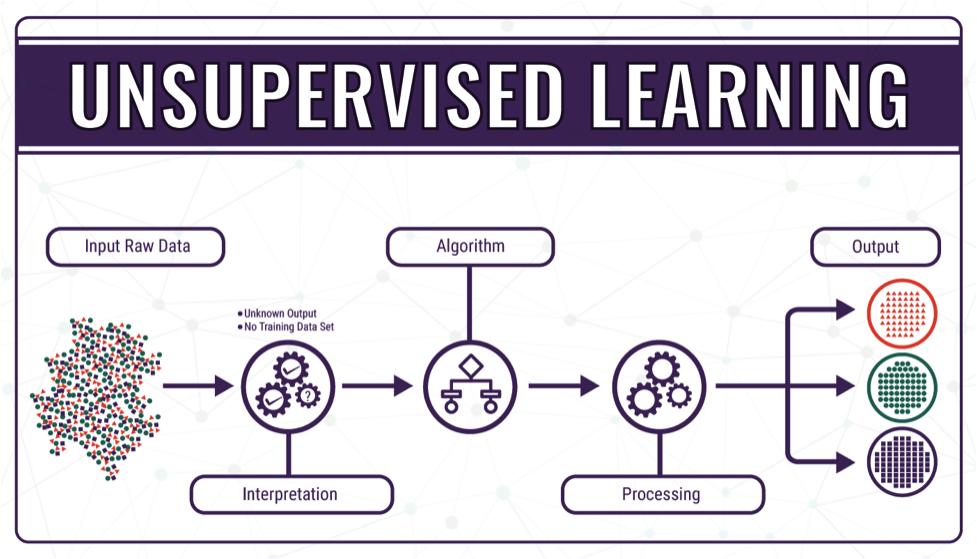
Machine learning is commonly separated into three main learning paradigms, supervised learning, unsupervised learning and reinforcement learning. Each corresponds to a particular learning task.

# ==== 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. 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{\displaystyle \textstyle 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{\displaystyle \textstyle f(x)=a} where a{\displaystyle \textstyle a} is a constant and the cost C=E[(x−f(x))2]{\displaystyle \textstyle C=E[(x-f(x))^{2}]}. Minimizing this cost produces a value of a{\displaystyle \textstyle 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{\displaystyle \textstyle x} and f(x){\displaystyle \textstyle 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.



[This Photo](https://ocw.cs.pub.ro/courses/ep/labs/10) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

# ==== 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 s1,...,sn∈S{\displaystyle \textstyle {s\_{1},...,s\_{n}}\in S} and actions a1,...,am∈A{\displaystyle \textstyle {a\_{1},...,a\_{m}}\in A}. Because the state transitions are not known, probability distributions are used instead: the instantaneous cost distribution P(ct|st){\displaystyle \textstyle P(c\_{t}|s\_{t})}, the observation distribution P(xt|st){\displaystyle \textstyle P(x\_{t}|s\_{t})} and the transition distribution P(st+1|st,at){\displaystyle \textstyle 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. Dynamic programming coupled with ANNs (giving neurodynamic programming) has been applied to problems such as those involved in vehicle routing, video games, natural resource management and medicine 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). 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. Given the memory matrix, W =||w(a,s)||, the crossbar self-learning algorithm in each iteration performs the following computation:

In situation s perform action a;

Receive consequence situation s';

Compute emotion of being in consequence situation v(s');

Update crossbar memory w'(a,s) = w(a,s) + v(s').

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 initially and only once receives initial emotions 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.



[This Photo](https://ecampusontario.pressbooks.pub/learningtolearnonlineatfanshawe/chapter/identify-characteristics-of-a-self-directed-learner/) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

# ==== Neuroevolution ====

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

# === 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, 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. Stochastic neural networks trained using a Bayesian approach are known as Bayesian neural networks.

# === Other ===

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

# ==== 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; where long short-term memory avoids the vanishing gradient problem and can handle signals that have a mix of low and high frequency components aiding large-vocabulary speech recognition, text-to-speech synthesis, and photo-real talking heads;

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 or on deceiving the opponent about the authenticity of an input.

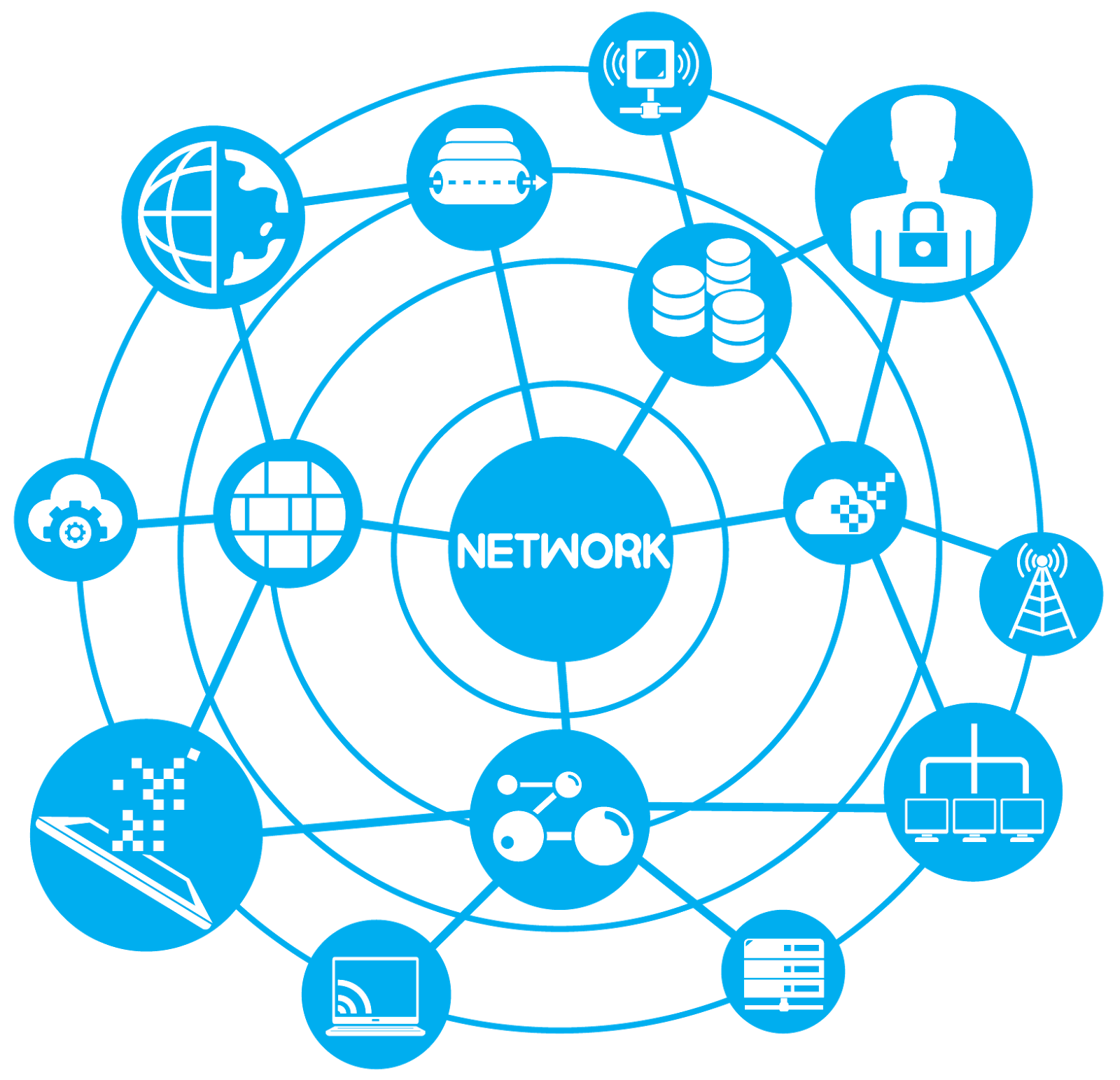
**== Network design ==**

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 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. Available systems include AutoML and AutoKeras. 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. 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:

# == Applications ==

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

Function approximation, or regression analysis, (including time series prediction, fitness approximation, and modeling)

Data processing (including filtering, clustering, blind source separation, and compression)

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

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

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

Sensor data analysis (including image analysis)

Robotics (including directing manipulators and prostheses)

Data mining (including knowledge discovery in databases)

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

Quantum chemistry

General game playing

Generative AI

Data visualization

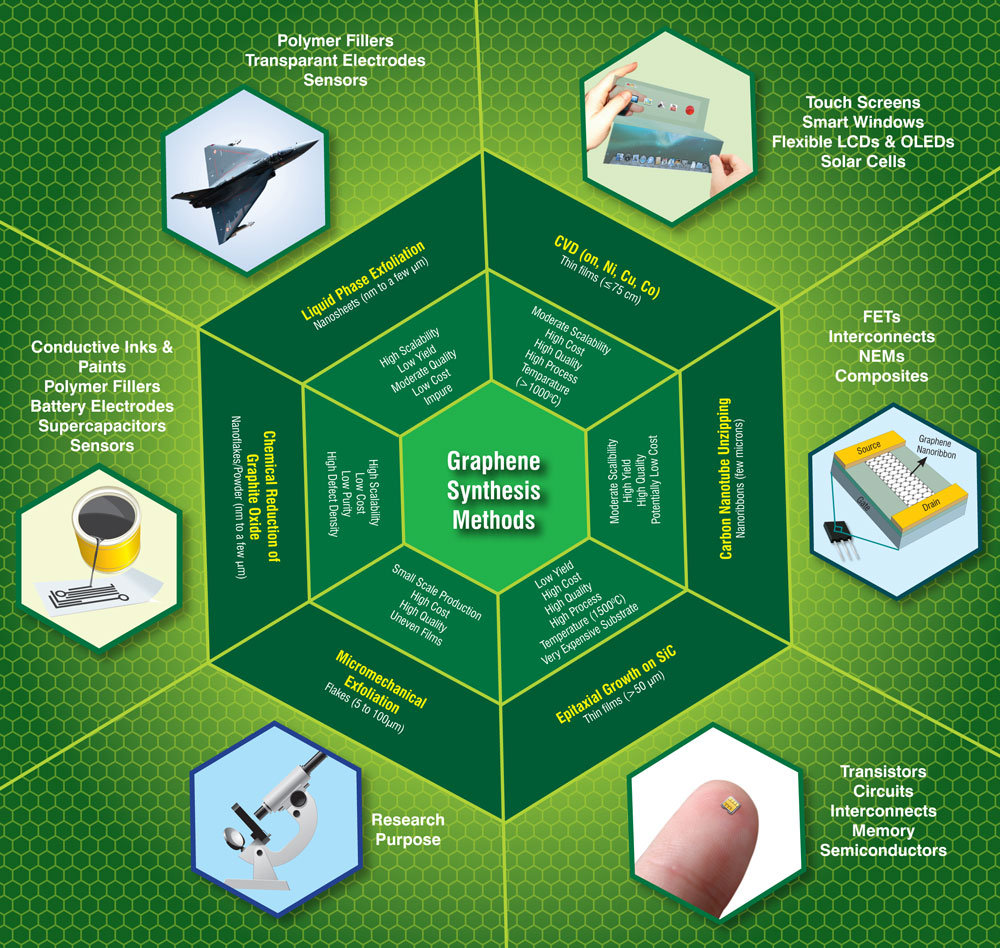
Machine translation

Social network filtering

E-mail spam filtering

Medical diagnosisANNs have been used to diagnose several types of cancers and to distinguish highly invasive cancer cell lines from less invasive lines using only cell shape information.ANNs have been used to accelerate reliability analysis of infrastructures subject to natural disasters and to predict foundation settlements. It can also be useful to mitigate flood by the use of ANNs for modelling rainfall-runoff. ANNs have also been used for building black-box models in geoscience: hydrology, ocean modelling and coastal engineering, and geomorphology. 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, for identifying domains belonging to threat actors and for detecting URLs posing a security risk. Research is underway on ANN systems designed for penetration testing, for detecting botnets, credit cards frauds and network intrusions.

ANNs have been proposed as a tool to solve partial differential equations in physics and simulate the properties of many-body open quantum systems. In brain research ANNs have studied short-term behavior of individual neurons, 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.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.s

**=== 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, 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

**=== 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 which summarizes work by Thomas Cover. The capacity of a network of standard neurons (not convolutional) can be derived by four rules 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, 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.

# === 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. Another example is when parameters are small, it is observed that ANNs often fits target functions from low to high frequencies. This behavior is referred to as the spectral bias, or frequency principle, of neural networks. 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.



s

# === 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:

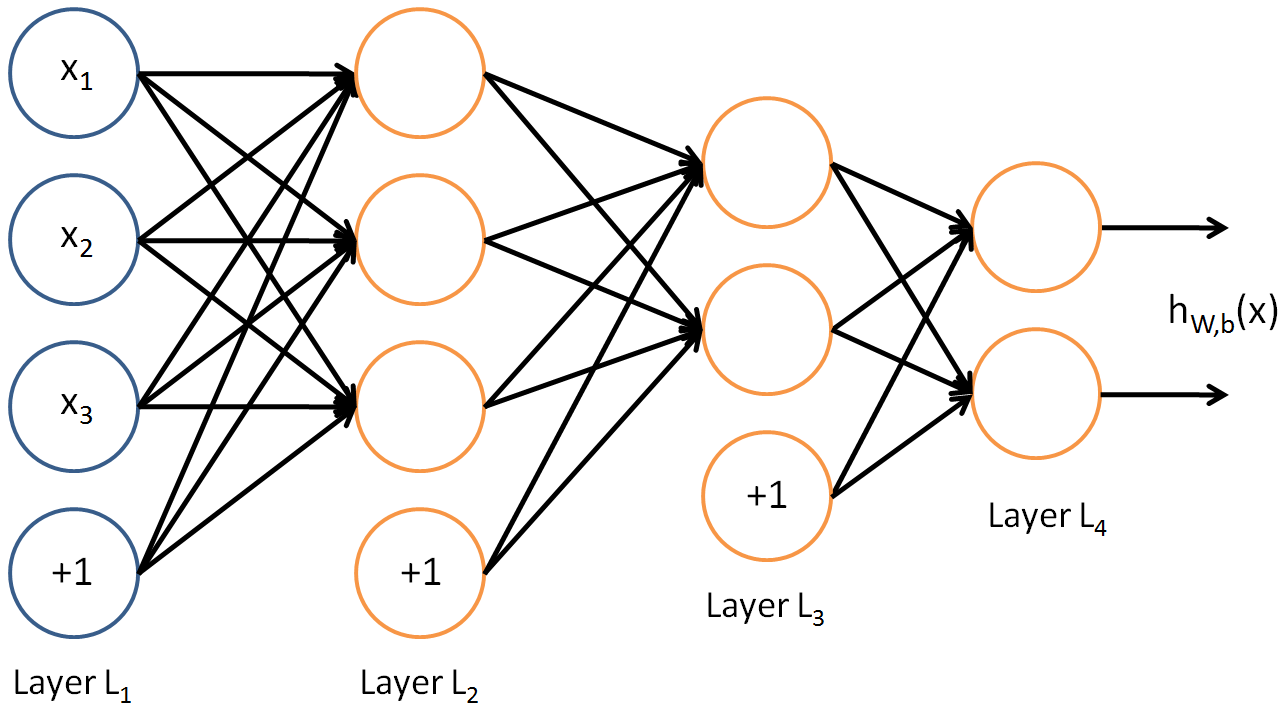
yi=exi∑j=1cexj{\displaystyle y\_{i}={\frac {e^{x\_{i}}}{\sum \_{j=1}^{c}e^{x\_{j}}}}}

# === Training ===

A common criticism of neural networks, particularly in robotics, is that they require too many training samples for real-world operation.

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.

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).



[This Photo](https://stackoverflow.com/questions/42733971/convolutional-layer-to-fully-connected-layer-in-cnn) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

# === Theory ===

A central claim of ANNs is that they embody new and powerful general principles for processing information. These principles are ill-defined. It is often claimed 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". One response to Dewdney is that neural networks have been successfully used to handle many complex and diverse tasks, ranging from autonomously flying aircraft 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.

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.Biological brains use both shallow and deep circuits as reported by brain anatomy, displaying a wide variety of invariance. Weng 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. 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.

Schmidhuber noted 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. The use of accelerators such as FPGAs and GPUs can reduce training times from months to days.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.

# === 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.

# === 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.

# === 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. 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. This imbalance can result in the model having inadequate representation and understanding of underrepresented groups, leading to discriminatory outcomes that exasperate societal inequalities, especially in applications like facial recognition, hiring processes, and law enforcement. 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. 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.

# == 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.

# === 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. 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.

# === 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. These advancements have enabled the development of more accurate and efficient voice-activated systems, enhancing user interfaces in technology products.

# === 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. This has implications for automated customer service, content moderation, and language understanding technologies.

# === 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.

# === 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.

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

# === 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. In drug discovery, ANNs speed up the identification of potential drug candidates and predict their efficacy and safety, significantly reducing development time and costs. Additionally, their application in personalized medicine and healthcare data analysis allows tailored therapies and efficient patient care management. 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.

# === Content creation ===

ANNs such as generative adversarial networks (GAN) and transformers are used for content creation across numerous industries. 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. In the field of music, transformers are used to create original music for commercials and documentaries through companies such as AIVA and Jukedeck. In the marketing industry generative models are used to create personalized advertisements for consumers. 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. 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.

# == 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

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

Another introduction to ANN

Next Generation of Neural Networks - Google Tech Talks

Performance of Neural Networks

Neural Networks and Information

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