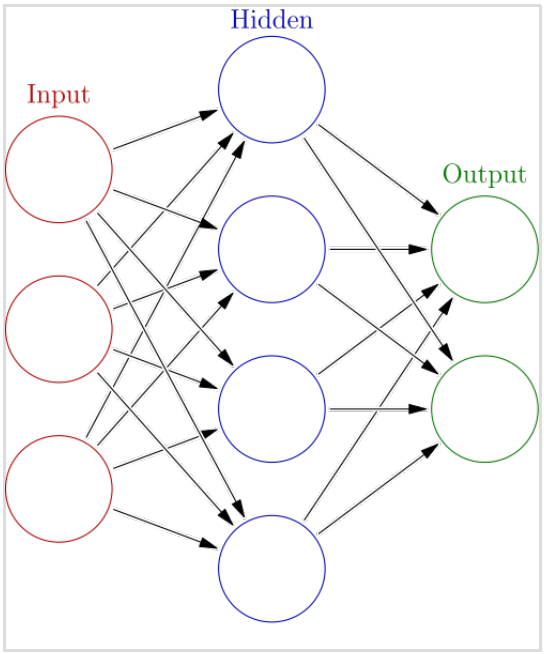
Neural network (machine learning)

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



An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

An ANN consists of connected units or nodes called *artificial neurons*, which loosely model the neurons in the brain. 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 sum 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.

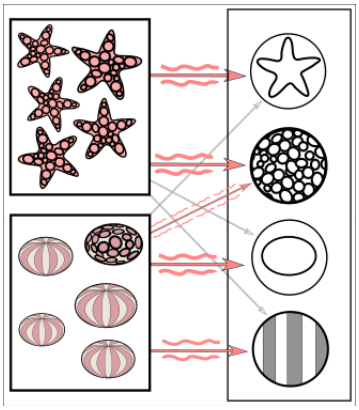
Training

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.

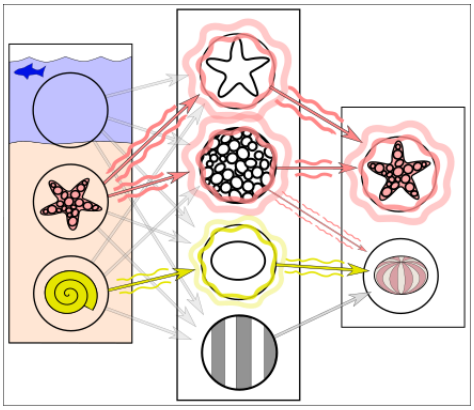
History

Early work

Historically, digital computers evolved from the von Neumann model, and operate via the execution of explicit instructions with 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.



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



Subsequent run of the network on an input image (left):[6] The network correctly detects the starfish. However, the weakly weighted association between ringed texture and sea urchin also confers a weak signal to the latter from one of two intermediate nodes. In addition, a shell that was not included in the training gives a weak signal for the oval shape, also resulting in a weak signal for the sea urchin output. These weak signals may result in a false positive result for sea urchin. In reality, textures and outlines would not be represented by single nodes, but rather by associated weight patterns of multiple nodes.

Warren McCulloch and Walter Pitts[7] (1943) considered a non-learning computational model for neural networks.[8] 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[9] 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.

In 1958, psychologist Frank Rosenblatt described the perceptron, one of the first implemented artificial neural networks,[10][11][12][13] funded by the United States Office of Naval Research.[14] R. D. Joseph (1960)[15] mentions an even earlier perceptron-like device by Farley and Clark[16]: "Farley and Clark of MIT Lincoln Laboratory actually preceded Rosenblatt in the development of a perceptron-like device." However, "they dropped the subject." Farley and Clark[17] (1954) also used computational machines to simulate a Hebbian network. Other neural network computational machines were created by Rochester, Holland, Habit and Duda (1956).[18] 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.[19]

The first perceptrons did not have adaptive hidden units. However, Joseph (1960)[15] also discussed multilayer perceptrons with an adaptive hidden layer. Rosenblatt (1962)[20]: 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 Ukraine (1965). They regarded it as a form of polynomial regression,[21] or a generalization of Rosenblatt's perceptron.[22] A 1971 paper described a deep network with eight layers trained by this method,[23] 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."[16]

The first deep learning multilayer perceptron trained by stochastic gradient descent[24] was published in 1967 by Shun'ichi Amari.[25] In computer experiments conducted by Amari's student Saito, a five layer MLP with two modifiable layers learned internal representations to classify non-linearily separable pattern classes.[16] Subsequent developments in hardware and hyperparameter tunings have made end-to-end stochastic gradient descent the currently dominant training technique.

In 1969, Kunihiko Fukushima introduced the ReLU (rectified linear unit) activation function.[26][27][16] The rectifier has become the most popular activation function for deep learning.[28]

Nevertheless, research stagnated in the United States following the work of Minsky and Papert (1969),[29] 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).

Deep learning architectures for convolutional neural networks (CNNs) with convolutional layers and downsampling layers and weight replication began with the Neocognitron introduced by Kunihiko Fukushima in 1979, though not trained by backpropagation.[30][31][32]

Backpropagation

Backpropagation is an efficient application of the chain rule derived by Gottfried Wilhelm Leibniz in 1673[33] to networks of differentiable nodes. The terminology "back-propagating errors" was actually introduced in 1962 by Rosenblatt,[20] 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.[34] The modern form of backpropagation was developed multiple times in early 1970s. The earliest published instance was Seppo Linnainmaa's master thesis (1970).[35][36] Paul Werbos developed it independently in 1971,[37] but had difficulty publishing it until 1982.[38] In 1986, David E. Rumelhart et al. popularized backpropagation.[39]

Convolutional neural networks

Kunihiko Fukushima's convolutional neural network (CNN) architecture of 1979[30] also introduced max pooling,[40] 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.[41][42] In 1988, Wei Zhang applied a backpropagation-trained CNN to alphabet recognition.[43] In 1989, Yann LeCun et al. created a CNN called LeNet for recognizing handwritten ZIP codes on mail. Training required 3 days.[44] In 1990, Wei Zhang implemented a CNN on optical computing hardware.[45] In 1991, a CNN was applied to medical image object segmentation[46] and breast cancer detection in mammograms.[47] 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.[48]

From 1988 onward,[49][50] 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.[51]

Recurrent networks

One origin of RNN was statistical mechanics. Shun'ichi Amari in 1972 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.[52] This was popularized as the Hopfield network (1982).[53]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.[54] Hebb considered "reverberating circuit" as an explanation for short-term memory.[55] The McCulloch and Pitts paper (1943) considered neural networks that contains cycles, and noted that the current activity of such networks can be affected by activity indefinitely far in the past.[56]

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"[57][58] which introduced the important concepts of self-supervised pre-training (the "P" in ChatGPT) and neural knowledge distillation.[16] 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.[59]

In 1991, Sepp Hochreiter's diploma thesis [60] identified and analyzed the vanishing gradient problem[60][61] 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.[62][63] This was not yet the modern version of LSTM, which required the forget gate, which was introduced in 1999.[64] 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,[65] restricted Boltzmann machine,[66] Helmholtz machine,[67] and the wake-sleep algorithm.[68] 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.[69][70] In 2011, a CNN named *DanNet*[71][72] 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.[32] It then won more contests.[73][74] They also showed how max-pooling CNNs on GPU improved performance significantly.[75]

In October 2012, AlexNet by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton[76] 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[77] and Google's Inceptionv3.[78]

In 2012, Ng and Dean created a network that learned to recognize higher-level concepts, such as cats, only from watching unlabeled images.[79] 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.[80]

Generative adversarial network (GAN) (Ian Goodfellow et al., 2014)[81] 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.[82][83] 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)[84] based on the Progressive GAN by Tero Karras et al.[85] 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.[86] Diffusion models (2015)[87] eclipsed GANs in generative modeling since then, with systems such as DALL·E 2 (2022) and Stable Diffusion (2022).

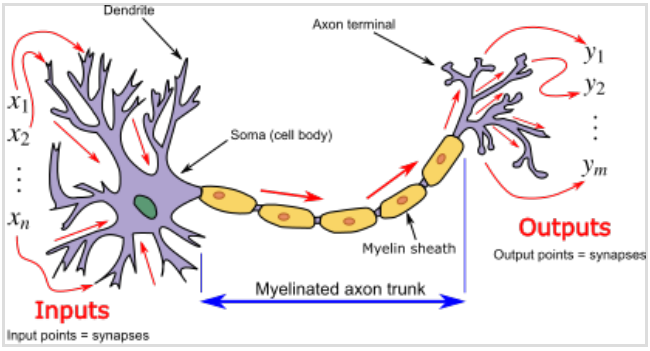
In 2014, the state of the art was training “very deep neural network” with 20 to 30 layers.[88] Stacking too many layers led to a steep reduction in training accuracy,[89] known as the "degradation" problem.[90] In 2015, two techniques were developed to train very deep networks:

the highway network was published in May 2015[91] and the residual neural network (ResNet) in Dec 2015.[92][93] ResNet behaves like an open-gated Highway Net.

During the 2010s period, 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*.[94] It requires computation time that is quadratic in the size of the context window. Jürgen Schmidhuber's fast weight controller (1992)[95] scales linearly and was later shown to be equivalent to the unnormalized linear Transformer.[96][97][16] Transformers have increasingly become the model of choice for natural language processing.[98] 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.[99]



Neuron and myelinated axon, with signal flow from inputs at dendrites to outputs at axon terminals

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,[100] 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.[101] 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.[102] 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.[103]

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.[104] Neurons with only such connections form a directed acyclic graph and are known as *feedforward networks*.[105] Alternatively, networks that allow connections between neurons in the same or previous layers are known as *recurrent networks*.[106]

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.

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.[99][107]

Learning rate

The learning rate defines the size of the corrective steps that the model takes to adjust for errors in each observation.[108] 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.[109] 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*,[110] "no-prop" networks,[111] training without backtracking,[112] "weightless" networks,[113][114] and non-connectionist neural networks.

Learning paradigms

Machine learning is commonly separated into three main learning paradigms, supervised learning,[115] unsupervised learning[116] and reinforcement learning.[117] 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.[118] 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 $\boldsymbol{x}$ 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 $s_1, \ldots, s_n \in S$ and actions $a_1, \ldots, a_m \in A$. 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.[119][120] Dynamic programming coupled with ANNs (giving neurodynamic programming)[121] has been applied to problems such as those involved in vehicle routing,[122] video games, natural resource management[123][124] and medicine[125] 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).[126] 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.[127] 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.[128]

Neuroevolution

Neuroevolution can create neural network topologies and weights using evolutionary computation. It is competitive with sophisticated gradient descent approaches.[129][130] One advantage of neuroevolution is that it may be less prone to get caught in "dead ends".[131]

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.[132] Stochastic neural networks trained using a Bayesian approach are known as **Bayesian neural networks**.[133]

Other

In a Bayesian framework, a distribution over the set of allowed models is chosen to minimize the cost. Evolutionary methods,[134] gene expression programming,[135] simulated annealing,[136] expectation–maximization, non-parametric methods and particle swarm optimization[137] are other learning algorithms. Convergent recursion is a learning algorithm for cerebellar model articulation controller (CMAC) neural networks.[138][139]

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;[140][141] where long short-term memory avoids the vanishing gradient problem[142] and can handle signals that have a mix of low and high frequency components aiding large-vocabulary speech recognition,[143][144] text-to-speech synthesis,[145][146][147] and photo-real talking heads;[148]
- 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[149] or on deceiving the opponent about the authenticity of an input.[81]

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[150] 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.[151] Available systems include AutoML and AutoKeras.[152] 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.[153]

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:

```python
def train(X, y, n_hidden, learning_rate, n_iter):
    m, n_input = X.shape

    # 1. random initialize weights and biases
    w1 = np.random.randn(n_input, n_hidden)
    b1 = np.zeros((1, n_hidden))
    w2 = np.random.randn(n_hidden, 1)
    b2 = np.zeros((1, 1))

    # 2. in each iteration, feed all layers with the latest weights and biases
    for i in range(n_iter + 1):
        z2 = np.dot(X, w1) + b1
        a2 = sigmoid(z2)
        z3 = np.dot(a2, w2) + b2
        a3 = z3
        dz3 = a3 - y
        dw2 = np.dot(a2.T, dz3)
        db2 = np.sum(dz3, axis=0, keepdims=True)
        dz2 = np.dot(dz3, w2.T) * sigmoid_derivative(z2)
        dw1 = np.dot(X.T, dz2)
        db1 = np.sum(dz2, axis=0)

        # 3. update weights and biases with gradients
        w1 -= learning_rate * dw1 / m
        w2 -= learning_rate * dw2 / m
        b1 -= learning_rate * db1 / m
        b2 -= learning_rate * db2 / m

        if i % 1000 == 0:
            print("Epoch", i, "loss: ", np.mean(np.square(dz3)))

    model = {"w1": w1, "b1": b1, "w2": w2, "b2": b2}
    return model
```

Applications

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

- Function approximation,[154] or regression analysis,[155] (including time series prediction, fitness approximation,[156] and modeling)
- Data processing[157] (including filtering, clustering, blind source separation,[158] and compression)
- Nonlinear system identification[80] and control (including vehicle control, trajectory prediction,[159] adaptive control, process control, and natural resource management)
- Pattern recognition (including radar systems, face identification, signal classification,[160] novelty detection, 3D reconstruction,[161] object recognition, and sequential decision making[162])
- Sequence recognition (including gesture, speech, and handwritten and printed text

recognition[163])
- Sensor data analysis[164] (including image analysis)
- Robotics (including directing manipulators and prostheses)
- Data mining (including knowledge discovery in databases)
- Finance[165] (such as ex-ante models for specific financial long-run forecasts and artificial financial markets)
- Quantum chemistry[166]
- General game playing[167]
- Generative AI[168]
- Data visualization
- Machine translation
- Social network filtering[169]
- E-mail spam filtering
- Medical diagnosis

ANNs have been used to diagnose several types of cancers[170][171] and to distinguish highly invasive cancer cell lines from less invasive lines using only cell shape information.[172][173]

ANNs have been used to accelerate reliability analysis of infrastructures subject to natural disasters[174][175] and to predict foundation settlements.[176] It can also be useful to mitigate flood by the use of ANNs for modelling rainfall-runoff.[177] ANNs have also been used for building black-box models in geoscience: hydrology,[178][179] ocean modelling and coastal engineering,[180][181] and geomorphology.[182] 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,[183] for identifying domains belonging to threat actors and for detecting URLs posing a security risk.[184] Research is underway on ANN systems designed for penetration testing, for detecting botnets,[185] credit cards frauds[186] and network intrusions.

ANNs have been proposed as a tool to solve partial differential equations in physics[187][188][189] and simulate the properties of many-body open quantum systems.[190][191][192][193] In brain research ANNs have studied short-term behavior of individual neurons,[194] 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.[195]

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.[196]

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,[197] 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.[198][199]

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[200] which summarizes work by Thomas Cover.[201] The capacity of a network of standard neurons (not convolutional) can be derived by four rules[202] 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,[200] 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.[203]

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.

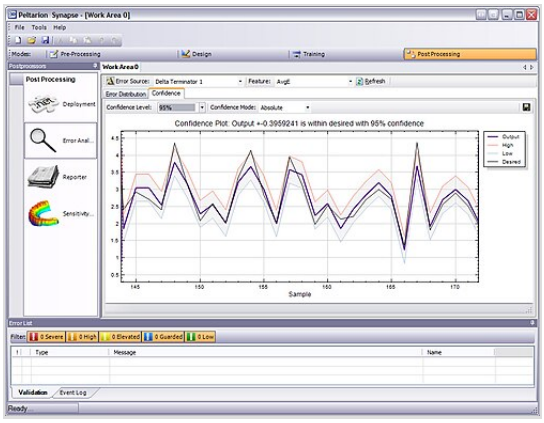
The convergence behavior of certain types of ANN architectures are more understood than others. When the width of network approaches to infinity, the ANN is well described by its first order Taylor expansion throughout training, and so inherits the convergence behavior of affine models.[204][205] Another 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.[206][207][208][209] 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.[210]

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.



Confidence analysis of a neural network

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:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{c} e^{x_j}}$$

Criticism

Training

A common criticism of neural networks, particularly in robotics, is that they require too many training samples for real-world operation.[211] 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.[138] 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).[212]

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".[213] One response to Dewdney is that neural networks have been successfully used to handle many complex and diverse tasks, ranging from autonomously flying aircraft[214] 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.[215]

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.[216]

Biological brains use both shallow and deep circuits as reported by brain anatomy,[217] displaying a wide variety of invariance. Weng[218] 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.[219] 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.

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.[32] The use of accelerators such as FPGAs and GPUs can reduce training times from months to days.[219][220]

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.[221]

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.[222]

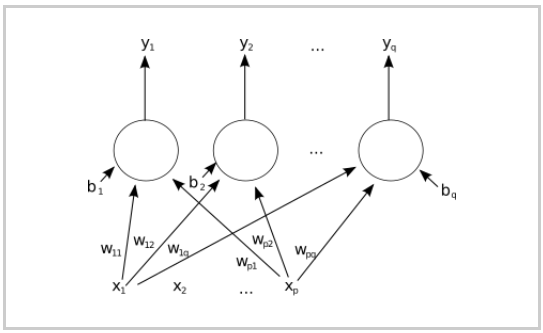
Hybrid approaches

Advocates of hybrid models (combining neural networks and symbolic approaches) say that such a mixture can better capture the mechanisms of the human mind.[223][224]

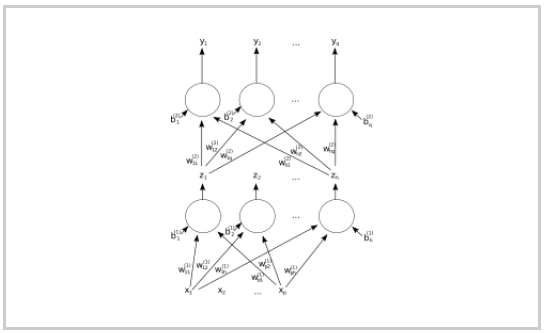
Dataset bias

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.[225][226] 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.[225] 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.[226][227] 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.[227] 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.[228]

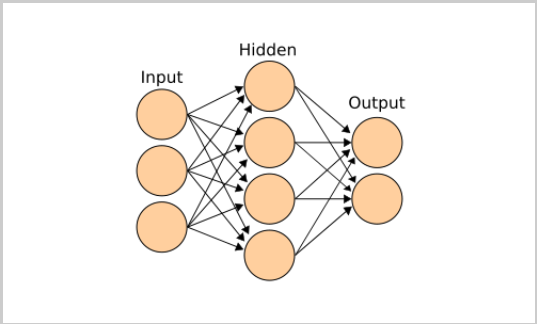
Gallery



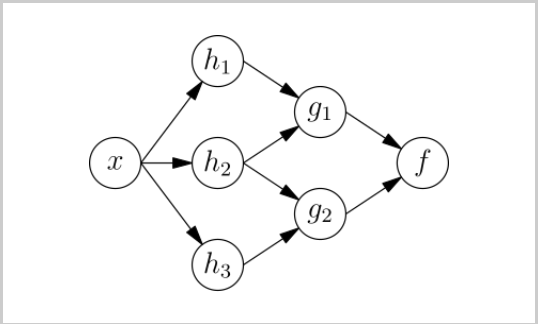
A single-layer feedforward artificial neural network. Arrows originating from x_2 are omitted for clarity. There are p inputs to this network and q outputs. In this system, the value of the qth output, y_q, is calculated as $y_q = K*(\sum_i (x_i * w_{iq}) - b_q)$.



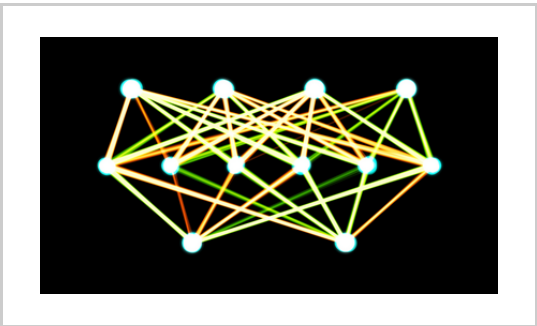
A two-layer feedforward artificial neural network



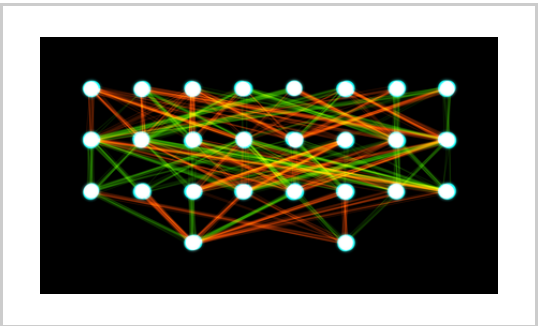
An artificial neural network



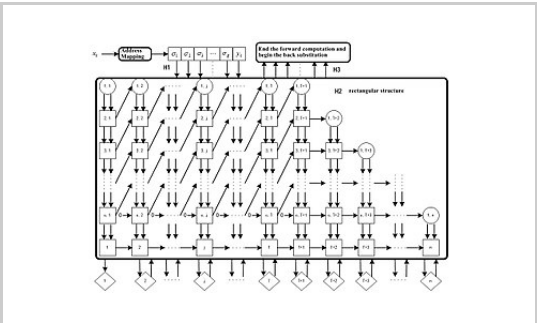
An ANN dependency graph



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.

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.[229] 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.[229]

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.[229][230] 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.[229][230] 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.[229]
- In credit scoring, ANNs offer data-driven, personalized assessments of creditworthiness, improving the accuracy of default predictions and automating the lending process.[230]

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.[230] In drug discovery, ANNs

speed up the identification of potential drug candidates and predict their efficacy and safety, significantly reducing development time and costs.[229] Additionally, their application in personalized medicine and healthcare data analysis allows tailored therapies and efficient patient care management.[230] 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.[231] 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.[232] In the field of music, transformers are used to create original music for commercials and documentaries through companies such as AIVA and Jukedeck.[233] In the marketing industry generative models are used to create personalized advertisements for consumers.[231] 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.[234] 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.[235]

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
- 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

- 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

External links

- A Brief Introduction to Neural Networks (D. Kriesel) (http://www.dkriesel.com/en/science/neural_networks) - 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 (http://www.msm.cam.ac.uk/phase-trans/abstracts/neural.review.html) Archived (https://web.archive.org/web/20150607101310/http://www.msm.cam.ac.uk/phase-trans/abstracts/neural.review.html) 7 June 2015 at the Wayback Machine
- Artificial Neural Networks Tutorial in three languages (Univ. Politécnica de Madrid) (https://web.archive.org/web/20090318133122/http://www.gc.ssr.upm.es/inves/neural/ann1/anntutorial.html)
- Another introduction to ANN (https://web.archive.org/web/20091216110504/http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html)
- Next Generation of Neural Networks (https://www.youtube.com/watch?v=AyzOUbkUf3M) Archived (https://web.archive.org/web/20110124234328/http://www.youtube.com/watch?v=AyzOUbkUf3M) 24 January 2011 at the Wayback Machine - Google Tech Talks
- Performance of Neural Networks (http://www.msm.cam.ac.uk/phase-trans/2009/performance.html)
- Neural Networks and Information (http://www.msm.cam.ac.uk/phase-trans/2009/review_Bhadeshia_SADM.pdf) Archived (https://web.archive.org/web/20090709153828/http://www.msm.cam.ac.uk/phase-trans/2009/review_Bhadeshia_SADM.pdf) 9 July 2009 at the Wayback Machine
- Sanderson G (5 October 2017). "But what *is* a Neural Network?" (https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi). *3Blue1Brown*. Archived (https://ghostarchive.org/varchive/youtube/20211107/aircAruvnKk) from the original on 7 November 2021 – via YouTube.

Notes

References

1. Hardesty L (14 April 2017). "Explained: Neural networks" (https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414). MIT News Office. Archived (https://web.archive.org/web/20240318120205/https://news.mit.edu/2017/explained-neural-networks-deep-learnin

g-0414) from the original on 18 March 2024. Retrieved 2 June 2022.
2. Yang Z, Yang Z (2014). *Comprehensive Biomedical Physics* (https://www.sciencedirect.com/topics/neuroscience/artificial-neural-network). Karolinska Institute, Stockholm, Sweden: Elsevier. p. 1. ISBN 978-0-444-53633-4. Archived (https://web.archive.org/web/20220728183237/https://www.sciencedirect.com/topics/neuroscience/artificial-neural-network) from the original on 28 July 2022. Retrieved 28 July 2022.
3. Bishop CM (17 August 2006). *Pattern Recognition and Machine Learning*. New York: Springer. ISBN 978-0-387-31073-2.
4. Vapnik VN, Vapnik VN (1998). *The nature of statistical learning theory* (Corrected 2nd print. ed.). New York Berlin Heidelberg: Springer. ISBN 978-0-387-94559-0.
5. Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). *Deep Learning* (http://www.deeplearningbook.org/). MIT Press. Archived (https://web.archive.org/web/20160416111010/http://www.deeplearningbook.org/) from the original on 16 April 2016. Retrieved 1 June 2016.
6. Ferrie, C., Kaiser, S. (2019). *Neural Networks for Babies*. Sourcebooks. ISBN 978-1-4926-7120-6.
7. McCulloch W, Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics*. **5** (4): 115–133. doi:10.1007/BF02478259 (https://doi.org/10.1007%2FBF02478259).
8. Kleene S (1956). "Representation of Events in Nerve Nets and Finite Automata" (https://www.degruyter.com/view/books/9781400882618/9781400882618-002/9781400882618-002.xml). *Annals of Mathematics Studies*. No. 34. Princeton University Press. pp. 3–41. Retrieved 17 June 2017.
9. Hebb D (1949). *The Organization of Behavior* (https://books.google.com/books?id=ddB4AgAAQBAJ). New York: Wiley. ISBN 978-1-135-63190-1.
10. Haykin (2008) Neural Networks and Learning Machines, 3rd edition
11. Rosenblatt F (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization in the Brain". *Psychological Review*. **65** (6): 386–408. CiteSeerX 10.1.1.588.3775 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.588.3775). doi:10.1037/h0042519 (https://doi.org/10.1037%2Fh0042519). PMID 13602029 (https://pubmed.ncbi.nlm.nih.gov/13602029). S2CID 12781225 (https://api.semanticscholar.org/CorpusID:12781225).
12. Werbos P (1975). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences* (https://books.google.com/books?id=z81XmgEACAAJ).
13. Rosenblatt F (1957). "The Perceptron—a perceiving and recognizing automaton". *Report 85-460-1*. Cornell Aeronautical Laboratory.
14. Olazaran M (1996). "A Sociological Study of the Official History of the Perceptrons Controversy". *Social Studies of Science*. **26** (3): 611–659. doi:10.1177/030631296026003005 (https://doi.org/10.1177%2F030631296026003005). JSTOR 285702 (https://www.jstor.org/stable/285702). S2CID 16786738 (https://api.semanticscholar.org/CorpusID:16786738).
15. Joseph RD (1960). *Contributions to Perceptron Theory, Cornell Aeronautical Laboratory Report No. VG-11 96--G-7, Buffalo*.
16. Schmidhuber J (2022). "Annotated History of Modern AI and Deep Learning". arXiv:2212.11279 (https://arxiv.org/abs/2212.11279) [cs.NE (https://arxiv.org/archive/cs.NE)].
17. Farley B, W.A. Clark (1954). "Simulation of Self-Organizing Systems by Digital Computer". *IRE Transactions on Information Theory*. **4** (4): 76–84. doi:10.1109/TIT.1954.1057468 (https://doi.org/10.1109%2FTIT.1954.1057468).
18. Rochester N, J.H. Holland, L.H. Habit, W.L. Duda (1956). "Tests on a cell assembly theory of the action of the brain, using a large digital computer". *IRE Transactions on Information Theory*. **2** (3): 80–93. doi:10.1109/TIT.1956.1056810 (https://doi.org/10.1109%2FTIT.1956.1

056810).
19. Russel, Stuart, Norvig, Peter (2010). *Artificial Intelligence A Modern Approach* (https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf) (PDF) (3rd ed.). United States of America: Pearson Education. pp. 16–28. ISBN 978-0-13-604259-4.
20. Rosenblatt F (1962). *Principles of Neurodynamics*. Spartan, New York.
21. Ivakhnenko AG, Lapa VG (1967). *Cybernetics and Forecasting Techniques* (https://books.google.com/books?id=rGFgAAAAMAAJ). American Elsevier Publishing Co. ISBN 978-0-444-00020-0.
22. Ivakhnenko A (March 1970). "Heuristic self-organization in problems of engineering cybernetics" (https://linkinghub.elsevier.com/retrieve/pii/0005109870900920). *Automatica*. **6** (2): 207–219. doi:10.1016/0005-1098(70)90092-0 (https://doi.org/10.1016%2F0005-1098%2870%2990092-0).
23. Ivakhnenko A (1971). "Polynomial theory of complex systems" (http://gmdh.net/articles/history/polynomial.pdf) (PDF). *IEEE Transactions on Systems, Man, and Cybernetics*. SMC-1 (4): 364–378. doi:10.1109/TSMC.1971.4308320 (https://doi.org/10.1109%2FTSMC.1971.4308320). Archived (https://web.archive.org/web/20170829230621/http://www.gmdh.net/articles/history/polynomial.pdf) (PDF) from the original on 29 August 2017. Retrieved 5 November 2019.
24. Robbins H, Monro S (1951). "A Stochastic Approximation Method" (https://doi.org/10.1214%2Faoms%2F1177729586). *The Annals of Mathematical Statistics*. **22** (3): 400. doi:10.1214/aoms/1177729586 (https://doi.org/10.1214%2Faoms%2F1177729586).
25. Amari S (1967). "A theory of adaptive pattern classifier". *IEEE Transactions*. **EC** (16): 279–307.
26. Fukushima K (1969). "Visual feature extraction by a multilayered network of analog threshold elements". *IEEE Transactions on Systems Science and Cybernetics*. **5** (4): 322–333. doi:10.1109/TSSC.1969.300225 (https://doi.org/10.1109%2FTSSC.1969.300225).
27. Sonoda S, Murata N (2017). "Neural network with unbounded activation functions is universal approximator". *Applied and Computational Harmonic Analysis*. **43** (2): 233–268. arXiv:1505.03654 (https://arxiv.org/abs/1505.03654). doi:10.1016/j.acha.2015.12.005 (https://doi.org/10.1016%2Fj.acha.2015.12.005). S2CID 12149203 (https://api.semanticscholar.org/CorpusID:12149203).
28. Ramachandran P, Barret Z, Quoc VL (16 October 2017). "Searching for Activation Functions". arXiv:1710.05941 (https://arxiv.org/abs/1710.05941) [cs.NE (https://arxiv.org/archive/cs.NE)].
29. Minsky M, Papert S (1969). *Perceptrons: An Introduction to Computational Geometry* (https://books.google.com/books?id=Ow1OAQAAIAAJ). MIT Press. ISBN 978-0-262-63022-1.
30. Fukushima K (1979). "Neural network model for a mechanism of pattern recognition unaffected by shift in position—Neocognitron". *Trans. IECE (In Japanese)*. J62-A (10): 658–665. doi:10.1007/bf00344251 (https://doi.org/10.1007%2Fbf00344251). PMID 7370364 (https://pubmed.ncbi.nlm.nih.gov/7370364). S2CID 206775608 (https://api.semanticscholar.org/CorpusID:206775608).
31. Fukushima K (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". *Biol. Cybern*. **36** (4): 193–202. doi:10.1007/bf00344251 (https://doi.org/10.1007%2Fbf00344251). PMID 7370364 (https://pubmed.ncbi.nlm.nih.gov/7370364). S2CID 206775608 (https://api.semanticscholar.org/CorpusID:206775608).
32. Schmidhuber J (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. **61**: 85–117. arXiv:1404.7828 (https://arxiv.org/abs/1404.7828). doi:10.1016/j.neunet.2014.09.003 (https://doi.org/10.1016%2Fj.neunet.2014.09.003). PMID 25462637 (https://pubmed.ncbi.nlm.nih.gov/25462637). S2CID 11715509 (https://api.semanticscholar.org/CorpusID:11715509).

33. Leibniz GW (1920). *The Early Mathematical Manuscripts of Leibniz: Translated from the Latin Texts Published by Carl Immanuel Gerhardt with Critical and Historical Notes (Leibniz published the chain rule in a 1676 memoir)* (https://books.google.com/books?id=bOIGAAAAYAAJ&q=leibniz+altered+manuscripts&pg=PA90). Open court publishing Company. ISBN 9780598818461.
34. Kelley HJ (1960). "Gradient theory of optimal flight paths". *ARS Journal*. **30** (10): 947–954. doi:10.2514/8.5282 (https://doi.org/10.2514%2F8.5282).
35. Linnainmaa S (1970). *The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors* (Masters) (in Finnish). University of Helsinki. p. 6–7.
36. Linnainmaa S (1976). "Taylor expansion of the accumulated rounding error". *BIT Numerical Mathematics*. **16** (2): 146–160. doi:10.1007/bf01931367 (https://doi.org/10.1007%2Fbf01931367). S2CID 122357351 (https://api.semanticscholar.org/CorpusID:122357351).
37. Anderson JA, Rosenfeld E, eds. (2000). *Talking Nets: An Oral History of Neural Networks* (https://direct.mit.edu/books/book/4886/Talking-NetsAn-Oral-History-of-Neural-Networks). The MIT Press. doi:10.7551/mitpress/6626.003.0016 (https://doi.org/10.7551%2Fmitpress%2F6626.003.0016). ISBN 978-0-262-26715-1.
38. Werbos P (1982). "Applications of advances in nonlinear sensitivity analysis" (http://werbos.com/Neural/SensitivityIFIPSeptember1981.pdf) (PDF). *System modeling and optimization*. Springer. pp. 762–770. Archived (https://web.archive.org/web/20160414055503/http://werbos.com/Neural/SensitivityIFIPSeptember1981.pdf) (PDF) from the original on 14 April 2016. Retrieved 2 July 2017.
39. Rumelhart DE, Hinton GE, Williams RJ (October 1986). "Learning representations by back-propagating errors" (https://www.nature.com/articles/323533a0). *Nature*. **323** (6088): 533–536. Bibcode:1986Natur.323..533R (https://ui.adsabs.harvard.edu/abs/1986Natur.323..533R). doi:10.1038/323533a0 (https://doi.org/10.1038%2F323533a0). ISSN 1476-4687 (https://search.worldcat.org/issn/1476-4687).
40. Fukushima K, Miyake S (1 January 1982). "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position" (https://www.sciencedirect.com/science/article/abs/pii/0031320382900243). *Pattern Recognition*. **15** (6): 455–469. doi:10.1016/0031-3203(82)90024-3 (https://doi.org/10.1016%2F0031-3203%2882%2990024-3). ISSN 0031-3203 (https://search.worldcat.org/issn/0031-3203).
41. Waibel A (December 1987). *Phoneme Recognition Using Time-Delay Neural Networks* (https://isl.anthropomatik.kit.edu/pdf/Waibel1987a.pdf) (PDF). Meeting of the Institute of Electrical, Information and Communication Engineers (IEICE). Tokyo, Japan.
42. Alexander Waibel et al., *Phoneme Recognition Using Time-Delay Neural Networks (http://www.inf.ufrgs.br/~engel/data/media/file/cmp121/waibel89_TDNN.pdf)* IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 37, No. 3, pp. 328. – 339 March 1989.
43. Zhang W (1988). "Shift-invariant pattern recognition neural network and its optical architecture" (https://drive.google.com/file/d/1nN_5odSG_QVae54EsQN_qSz-0ZsX6wA0/view?usp=sharing). *Proceedings of Annual Conference of the Japan Society of Applied Physics*.
44. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition", *Neural Computation*, 1, pp. 541–551, 1989.
45. Zhang W (1990). "Parallel distributed processing model with local space-invariant interconnections and its optical architecture" (https://drive.google.com/file/d/0B65v6Wo67Tk5ODRzZmhSR29VeDg/view?usp=sharing). *Applied Optics*. **29** (32): 4790–7. Bibcode:1990ApOpt..29.4790Z (https://ui.adsabs.harvard.edu/abs/1990ApOpt..29.4790Z). doi:10.1364/AO.29.004790 (https://doi.org/10.1364%2FAO.29.004790). PMID 20577468 (https://pubmed.ncbi.nlm.nih.gov/20577468).
46. Zhang W (1991). "Image processing of human corneal endothelium based on a learning

network" (https://drive.google.com/file/d/0B65v6Wo67Tk5cm5DTlNGd0NPUmM/view?usp=sharing). *Applied Optics*. **30** (29): 4211–7. Bibcode:1991ApOpt..30.4211Z (https://ui.adsabs.harvard.edu/abs/1991ApOpt..30.4211Z). doi:10.1364/AO.30.004211 (https://doi.org/10.1364%2FAO.30.004211). PMID 20706526 (https://pubmed.ncbi.nlm.nih.gov/20706526).

47. Zhang W (1994). "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network" (https://drive.google.com/file/d/0B65v6Wo67Tk5Ml9qeW5nQ3poVTQ/view?usp=sharing). *Medical Physics*. **21** (4): 517–24. Bibcode:1994MedPh..21..517Z (https://ui.adsabs.harvard.edu/abs/1994MedPh..21..517Z). doi:10.1118/1.597177 (https://doi.org/10.1118%2F1.597177). PMID 8058017 (https://pubmed.ncbi.nlm.nih.gov/8058017).
48. LeCun Y, Léon Bottou, Yoshua Bengio, Patrick Haffner (1998). "Gradient-based learning applied to document recognition" (http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf) (PDF). *Proceedings of the IEEE*. **86** (11): 2278–2324. CiteSeerX 10.1.1.32.9552 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.9552). doi:10.1109/5.726791 (https://doi.org/10.1109%2F5.726791). S2CID 14542261 (https://api.semanticscholar.org/CorpusID:14542261). Retrieved 7 October 2016.
49. Qian, Ning, and Terrence J. Sejnowski. "Predicting the secondary structure of globular proteins using neural network models." *Journal of molecular biology* 202, no. 4 (1988): 865-884.
50. Bohr, Henrik, Jakob Bohr, Søren Brunak, Rodney MJ Cotterill, Benny Lautrup, Leif Nørskov, Ole H. Olsen, and Steffen B. Petersen. "Protein secondary structure and homology by neural networks The α-helices in rhodopsin." *FEBS letters* 241, (1988): 223-228
51. Rost, Burkhard, and Chris Sander. "Prediction of protein secondary structure at better than 70% accuracy." *Journal of molecular biology* 232, no. 2 (1993): 584-599.
52. Amari SI (November 1972). "Learning Patterns and Pattern Sequences by Self-Organizing Nets of Threshold Elements" (https://ieeexplore.ieee.org/document/1672070). *IEEE Transactions on Computers*. **C-21** (11): 1197–1206. doi:10.1109/T-C.1972.223477 (https://doi.org/10.1109%2FT-C.1972.223477). ISSN 0018-9340 (https://search.worldcat.org/issn/0018-9340).
53. Hopfield JJ (1982). "Neural networks and physical systems with emergent collective computational abilities" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC346238). *Proceedings of the National Academy of Sciences*. **79** (8): 2554–2558. Bibcode:1982PNAS...79.2554H (https://ui.adsabs.harvard.edu/abs/1982PNAS...79.2554H). doi:10.1073/pnas.79.8.2554 (https://doi.org/10.1073%2Fpnas.79.8.2554). PMC 346238 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC346238). PMID 6953413 (https://pubmed.ncbi.nlm.nih.gov/6953413).
54. Espinosa-Sanchez JM, Gomez-Marin A, de Castro F (5 July 2023). "The Importance of Cajal's and Lorente de Nó's Neuroscience to the Birth of Cybernetics" (http://journals.sagepub.com/doi/10.1177/10738584231179932). *The Neuroscientist*. doi:10.1177/10738584231179932 (https://doi.org/10.1177%2F10738584231179932). hdl:10261/348372 (https://hdl.handle.net/10261%2F348372). ISSN 1073-8584 (https://search.worldcat.org/issn/1073-8584). PMID 37403768 (https://pubmed.ncbi.nlm.nih.gov/37403768).
55. "reverberating circuit" (https://www.oxfordreference.com/display/10.1093/oi/authority.20110803100417461). *Oxford Reference*. Retrieved 27 July 2024.
56. McCulloch WS, Pitts W (December 1943). "A logical calculus of the ideas immanent in nervous activity" (http://link.springer.com/10.1007/BF02478259). *The Bulletin of Mathematical Biophysics*. **5** (4): 115–133. doi:10.1007/BF02478259 (https://doi.org/10.1007%2FBF02478259). ISSN 0007-4985 (https://search.worldcat.org/issn/0007-4985).
57. Schmidhuber J (April 1991). "Neural Sequence Chunkers" (https://people.idsia.ch/~juergen/FKI-148-91ocr.pdf) (PDF). *TR FKI-148, TU Munich*.
58. Schmidhuber J (1992). "Learning complex, extended sequences using the principle of

history compression (based on TR FKI-148, 1991)" (https://sferics.idsia.ch/pub/juergen/chunker.pdf) (PDF). *Neural Computation*. **4** (2): 234–242. doi:10.1162/neco.1992.4.2.234 (https://doi.org/10.1162%2Fneco.1992.4.2.234). S2CID 18271205 (https://api.semanticscholar.org/CorpusID:18271205).

59. Schmidhuber J (1993). *Habilitation thesis: System modeling and optimization* (https://sferics.idsia.ch/pub/juergen/habilitation.pdf) (PDF). Page 150 ff demonstrates credit assignment across the equivalent of 1,200 layers in an unfolded RNN.
60. S. Hochreiter., "Untersuchungen zu dynamischen neuronalen Netzen (http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf) Archived (https://web.archive.org/web/20150306075401/http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf) 2015-03-06 at the Wayback Machine," *Diploma thesis. Institut f. Informatik, Technische Univ. Munich. Advisor: J. Schmidhuber*, 1991.
61. Hochreiter S, et al. (15 January 2001). "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies" (https://books.google.com/books?id=NWOcMVA64aAC). In Kolen JF, Kremer SC (eds.). *A Field Guide to Dynamical Recurrent Networks*. John Wiley & Sons. ISBN 978-0-7803-5369-5. Archived (https://web.archive.org/web/20240519081124/https://books.google.com/books?id=NWOcMVA64aAC) from the original on 19 May 2024. Retrieved 26 June 2017.
62. Sepp Hochreiter, Jürgen Schmidhuber (21 August 1995), *Long Short Term Memory* (ftp://ftp.idsia.ch/pub/juergen/fki-207-95.ps.gz), Wikidata Q98967430
63. Hochreiter S, Schmidhuber J (1 November 1997). "Long Short-Term Memory". *Neural Computation*. **9** (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735 (https://doi.org/10.1162%2Fneco.1997.9.8.1735). PMID 9377276 (https://pubmed.ncbi.nlm.nih.gov/9377276). S2CID 1915014 (https://api.semanticscholar.org/CorpusID:1915014).
64. Gers F, Schmidhuber J, Cummins F (1999). "Learning to forget: Continual prediction with LSTM". *9th International Conference on Artificial Neural Networks: ICANN '99*. Vol. 1999. pp. 850–855. doi:10.1049/cp:19991218 (https://doi.org/10.1049%2Fcp%3A19991218). ISBN 0-85296-721-7.
65. Ackley DH, Hinton GE, Sejnowski TJ (1 January 1985). "A learning algorithm for boltzmann machines" (https://www.sciencedirect.com/science/article/pii/S0364021385800124). *Cognitive Science*. **9** (1): 147–169. doi:10.1016/S0364-0213(85)80012-4 (https://doi.org/10.1016%2FS0364-0213%2885%2980012-4) (inactive 7 August 2024). ISSN 0364-0213 (https://search.worldcat.org/issn/0364-0213).
66. Smolensky P (1986). "Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory" (https://stanford.edu/~jlmcc/papers/PDP/Volume%201/Chap6_PDP86.pdf) (PDF). In Rumelhart DE, McLelland JL (eds.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. MIT Press. pp. 194–281 (https://archive.org/details/paralleldistribu00rume/page/194). ISBN 0-262-68053-X.
67. Peter D, Hinton GE, Neal RM, Zemel RS (1995). "The Helmholtz machine". *Neural Computation*. **7** (5): 889–904. doi:10.1162/neco.1995.7.5.889 (https://doi.org/10.1162%2Fneco.1995.7.5.889). hdl:21.11116/0000-0002-D6D3-E (https://hdl.handle.net/21.11116%2F0000-0002-D6D3-E). PMID 7584891 (https://pubmed.ncbi.nlm.nih.gov/7584891). S2CID 1890561 (https://api.semanticscholar.org/CorpusID:1890561). 🔒
68. Hinton GE, Dayan P, Frey BJ, Neal R (26 May 1995). "The wake-sleep algorithm for unsupervised neural networks". *Science*. **268** (5214): 1158–1161. Bibcode:1995Sci...268.1158H (https://ui.adsabs.harvard.edu/abs/1995Sci...268.1158H). doi:10.1126/science.7761831 (https://doi.org/10.1126%2Fscience.7761831). PMID 7761831 (https://pubmed.ncbi.nlm.nih.gov/7761831). S2CID 871473 (https://api.semanticscholar.org/CorpusID:871473).
69. 2012 Kurzweil AI Interview (http://www.kurzweilai.net/how-bio-inspired-deep-learning-keeps-winning-competitions) Archived (https://web.archive.org/web/20180831075249/http://www.k

urzweilai.net/how-bio-inspired-deep-learning-keeps-winning-competitions) 31 August 2018 at the Wayback Machine with Juergen Schmidhuber on the eight competitions won by his Deep Learning team 2009–2012

70. "How bio-inspired deep learning keeps winning competitions | KurzweilAI" (https://web.archive.org/web/20180831075249/http://www.kurzweilai.net/how-bio-inspired-deep-learning-keeps-winning-competitions). *www.kurzweilai.net*. Archived from the original (http://www.kurzweilai.net/how-bio-inspired-deep-learning-keeps-winning-competitions) on 31 August 2018. Retrieved 16 June 2017.
71. Cireşan DC, Meier U, Gambardella LM, Schmidhuber J (21 September 2010). "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". *Neural Computation*. **22** (12): 3207–3220. arXiv:1003.0358 (https://arxiv.org/abs/1003.0358). doi:10.1162/neco_a_00052 (https://doi.org/10.1162%2Fneco_a_00052). ISSN 0899-7667 (https://search.worldcat.org/issn/0899-7667). PMID 20858131 (https://pubmed.ncbi.nlm.nih.gov/20858131). S2CID 1918673 (https://api.semanticscholar.org/CorpusID:1918673).
72. Ciresan DC, Meier U, Masci J, Gambardella L, Schmidhuber J (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (http://ijcai.org/papers11/Papers/IJCAI11-210.pdf) (PDF). *International Joint Conference on Artificial Intelligence*. doi:10.5591/978-1-57735-516-8/ijcai11-210 (https://doi.org/10.5591%2F978-1-57735-516-8%2Fijcai11-210). Archived (https://web.archive.org/web/20140929094040/http://ijcai.org/papers11/Papers/IJCAI11-210.pdf) (PDF) from the original on 29 September 2014. Retrieved 13 June 2017.
73. Ciresan D, Giusti A, Gambardella LM, Schmidhuber J (2012). Pereira F, Burges CJ, Bottou L, Weinberger KQ (eds.). *Advances in Neural Information Processing Systems 25* (http://papers.nips.cc/paper/4741-deep-neural-networks-segment-neuronal-membranes-in-electron-microscopy-images.pdf) (PDF). Curran Associates, Inc. pp. 2843–2851. Archived (https://web.archive.org/web/20170809081713/http://papers.nips.cc/paper/4741-deep-neural-networks-segment-neuronal-membranes-in-electron-microscopy-images.pdf) (PDF) from the original on 9 August 2017. Retrieved 13 June 2017.
74. Ciresan D, Giusti A, Gambardella L, Schmidhuber J (2013). "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks". *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. Lecture Notes in Computer Science. Vol. 7908. pp. 411–418. doi:10.1007/978-3-642-40763-5_51 (https://doi.org/10.1007%2F978-3-642-40763-5_51). ISBN 978-3-642-38708-1. PMID 24579167 (https://pubmed.ncbi.nlm.nih.gov/24579167).
75. Ciresan D, Meier U, Schmidhuber J (2012). "Multi-column deep neural networks for image classification". *2012 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3642–3649. arXiv:1202.2745 (https://arxiv.org/abs/1202.2745). doi:10.1109/cvpr.2012.6248110 (https://doi.org/10.1109%2Fcvpr.2012.6248110). ISBN 978-1-4673-1228-8. S2CID 2161592 (https://api.semanticscholar.org/CorpusID:2161592).
76. Krizhevsky A, Sutskever I, Hinton G (2012). "ImageNet Classification with Deep Convolutional Neural Networks" (https://www.cs.toronto.edu/~kriz/imagenet_classification_with_deep_convolutional.pdf) (PDF). *NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada*. Archived (https://web.archive.org/web/20170110123024/http://www.cs.toronto.edu/~kriz/imagenet_classification_with_deep_convolutional.pdf) (PDF) from the original on 10 January 2017. Retrieved 24 May 2017.
77. Simonyan K, Andrew Z (2014). "Very Deep Convolution Networks for Large Scale Image Recognition". arXiv:1409.1556 (https://arxiv.org/abs/1409.1556) [cs.CV (https://arxiv.org/archive/cs.CV)].
78. Szegedy C (2015). "Going deeper with convolutions" (https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43022.pdf) (PDF). *Cvpr2015*.
79. Ng A, Dean J (2012). "Building High-level Features Using Large Scale Unsupervised Learning". arXiv:1112.6209 (https://arxiv.org/abs/1112.6209) [cs.LG (https://arxiv.org/archive/

cs.LG)].
80. Billings SA (2013). *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Wiley. ISBN 978-1-119-94359-4.
81. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. (2014). *Generative Adversarial Networks* (https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf) (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680. Archived (https://web.archive.org/web/20191122034612/http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf) (PDF) from the original on 22 November 2019. Retrieved 20 August 2019.
82. Schmidhuber J (1991). "A possibility for implementing curiosity and boredom in model-building neural controllers". *Proc. SAB'1991*. MIT Press/Bradford Books. pp. 222–227.
83. Schmidhuber J (2020). "Generative Adversarial Networks are Special Cases of Artificial Curiosity (1990) and also Closely Related to Predictability Minimization (1991)". *Neural Networks*. **127**: 58–66. arXiv:1906.04493 (https://arxiv.org/abs/1906.04493). doi:10.1016/j.neunet.2020.04.008 (https://doi.org/10.1016%2Fj.neunet.2020.04.008). PMID 32334341 (https://pubmed.ncbi.nlm.nih.gov/32334341). S2CID 216056336 (https://api.semanticscholar.org/CorpusID:216056336).
84. "GAN 2.0: NVIDIA's Hyperrealistic Face Generator" (https://syncedreview.com/2018/12/14/gan-2-0-nvidias-hyperrealistic-face-generator/). *SyncedReview.com*. 14 December 2018. Retrieved 3 October 2019.
85. Karras T, Aila T, Laine S, Lehtinen J (26 February 2018). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". arXiv:1710.10196 (https://arxiv.org/abs/1710.10196) [cs.NE (https://arxiv.org/archive/cs.NE)].
86. "Prepare, Don't Panic: Synthetic Media and Deepfakes" (https://lab.witness.org/projects/synthetic-media-and-deep-fakes/). witness.org. Archived (https://web.archive.org/web/20201202231744/https://lab.witness.org/projects/synthetic-media-and-deep-fakes/) from the original on 2 December 2020. Retrieved 25 November 2020.
87. Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S (1 June 2015). "Deep Unsupervised Learning using Nonequilibrium Thermodynamics" (http://proceedings.mlr.press/v37/sohl-dickstein15.pdf) (PDF). *Proceedings of the 32nd International Conference on Machine Learning*. **37**. PMLR: 2256–2265.
88. Simonyan K, Zisserman A (10 April 2015), *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv:1409.1556 (https://arxiv.org/abs/1409.1556)
89. He K, Zhang X, Ren S, Sun J (2016). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". arXiv:1502.01852 (https://arxiv.org/abs/1502.01852) [cs.CV (https://arxiv.org/archive/cs.CV)].
90. He K, Zhang X, Ren S, Sun J (10 December 2015). *Deep Residual Learning for Image Recognition*. arXiv:1512.03385 (https://arxiv.org/abs/1512.03385).
91. Srivastava RK, Greff K, Schmidhuber J (2 May 2015). "Highway Networks". arXiv:1505.00387 (https://arxiv.org/abs/1505.00387) [cs.LG (https://arxiv.org/archive/cs.LG)].
92. He K, Zhang X, Ren S, Sun J (2016). *Deep Residual Learning for Image Recognition* (https://ieeexplore.ieee.org/document/7780459). *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE. pp. 770–778. arXiv:1512.03385 (https://arxiv.org/abs/1512.03385). doi:10.1109/CVPR.2016.90 (https://doi.org/10.1109%2FCVPR.2016.90). ISBN 978-1-4673-8851-1.
93. Linn A (10 December 2015). "Microsoft researchers win ImageNet computer vision challenge" (https://blogs.microsoft.com/ai/microsoft-researchers-win-imagenet-computer-vision-challenge/). *The AI Blog*. Retrieved 29 June 2024.
94. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. (12 June 2017). "Attention Is All You Need". arXiv:1706.03762 (https://arxiv.org/abs/1706.03762) [cs.CL (https://arxiv.org/archive/cs.CL)].

95. Schmidhuber J (1992). "Learning to control fast-weight memories: an alternative to recurrent nets" (https://archive.org/download/wikipedia-scholarly-sources-corpus/10.1162.zip/10.1162%252Fneco.1992.4.1.131.pdf) (PDF). *Neural Computation*. **4** (1): 131–139. doi:10.1162/neco.1992.4.1.131 (https://doi.org/10.1162%2Fneco.1992.4.1.131). S2CID 16683347 (https://api.semanticscholar.org/CorpusID:16683347).
96. Katharopoulos A, Vyas A, Pappas N, Fleuret F (2020). "Transformers are RNNs: Fast autoregressive Transformers with linear attention" (https://paperswithcode.com/paper/a-decomposable-attention-model-for-natural). *ICML 2020*. PMLR. pp. 5156–5165.
97. Schlag I, Irie K, Schmidhuber J (2021). "Linear Transformers Are Secretly Fast Weight Programmers". *ICML 2021*. Springer. pp. 9355–9366.
98. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. (2020). "Transformers: State-of-the-Art Natural Language Processing". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6 (https://doi.org/10.18653%2Fv1%2F2020.emnlp-demos.6). S2CID 208117506 (https://api.semanticscholar.org/CorpusID:208117506).
99. Zell A (2003). "chapter 5.2". *Simulation neuronaler Netze* [*Simulation of Neural Networks*] (in German) (1st ed.). Addison-Wesley. ISBN 978-3-89319-554-1. OCLC 249017987 (https://search.worldcat.org/oclc/249017987).
100. *Artificial intelligence* (3rd ed.). Addison-Wesley Pub. Co. 1992. ISBN 0-201-53377-4.
101. Abbod MF (2007). "Application of Artificial Intelligence to the Management of Urological Cancer". *The Journal of Urology*. **178** (4): 1150–1156. doi:10.1016/j.juro.2007.05.122 (https://doi.org/10.1016%2Fj.juro.2007.05.122). PMID 17698099 (https://pubmed.ncbi.nlm.nih.gov/17698099).
102. Dawson CW (1998). "An artificial neural network approach to rainfall-runoff modelling" (https://doi.org/10.1080%2F02626669809492102). *Hydrological Sciences Journal*. **43** (1): 47–66. Bibcode:1998HydSJ..43...47D (https://ui.adsabs.harvard.edu/abs/1998HydSJ..43...47D). doi:10.1080/02626669809492102 (https://doi.org/10.1080%2F02626669809492102).
103. "The Machine Learning Dictionary" (https://web.archive.org/web/20180826151959/http://www.cse.unsw.edu.au/~billw/mldict.html#activnfn). *www.cse.unsw.edu.au*. Archived from the original (http://www.cse.unsw.edu.au/~billw/mldict.html#activnfn) on 26 August 2018. Retrieved 4 November 2009.
104. Ciresan D, Ueli Meier, Jonathan Masci, Luca M. Gambardella, Jurgen Schmidhuber (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (https://people.idsia.ch/~juergen/ijcai2011.pdf) (PDF). *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two*. **2**: 1237–1242. Archived (https://web.archive.org/web/20220405190128/https://people.idsia.ch/~juergen/ijcai2011.pdf) (PDF) from the original on 5 April 2022. Retrieved 7 July 2022.
105. Zell A (1994). *Simulation Neuronaler Netze* [*Simulation of Neural Networks*] (in German) (1st ed.). Addison-Wesley. p. 73. ISBN 3-89319-554-8.
106. Miljanovic M (February–March 2012). "Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction" (http://www.ijcse.com/docs/INDJCSE12-03-01-028.pdf) (PDF). *Indian Journal of Computer and Engineering*. **3** (1). Archived (https://web.archive.org/web/20240519081156/http://www.ijcse.com/docs/INDJCSE12-03-01-028.pdf) (PDF) from the original on 19 May 2024. Retrieved 21 August 2019.
107. Kelleher JD, Mac Namee B, D'Arcy A (2020). "7-8". *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies* (2nd ed.). Cambridge, MA: The MIT Press. ISBN 978-0-262-36110-1. OCLC 1162184998 (https://search.worldcat.org/oclc/1162184998).
108. Wei J (26 April 2019). "Forget the Learning Rate, Decay Loss". arXiv:1905.00094 (https://arxiv.org/abs/1905.00094) [cs.LG (https://arxiv.org/archive/cs.LG)].
109. Li Y, Fu Y, Li H, Zhang SW (1 June 2009). "The Improved Training Algorithm of Back

Propagation Neural Network with Self-adaptive Learning Rate". *2009 International Conference on Computational Intelligence and Natural Computing*. Vol. 1. pp. 73–76. doi:10.1109/CINC.2009.111 (https://doi.org/10.1109%2FCINC.2009.111). ISBN 978-0-7695-3645-3. S2CID 10557754 (https://api.semanticscholar.org/CorpusID:10557754).

110. Huang GB, Zhu QY, Siew CK (2006). "Extreme learning machine: theory and applications". *Neurocomputing*. **70** (1): 489–501. CiteSeerX 10.1.1.217.3692 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.217.3692). doi:10.1016/j.neucom.2005.12.126 (https://doi.org/10.1016%2Fj.neucom.2005.12.126). S2CID 116858 (https://api.semanticscholar.org/CorpusID:116858).
111. Widrow B, et al. (2013). "The no-prop algorithm: A new learning algorithm for multilayer neural networks". *Neural Networks*. **37**: 182–188. doi:10.1016/j.neunet.2012.09.020 (https://doi.org/10.1016%2Fj.neunet.2012.09.020). PMID 23140797 (https://pubmed.ncbi.nlm.nih.gov/23140797).
112. Ollivier Y, Charpiat G (2015). "Training recurrent networks without backtracking". arXiv:1507.07680 (https://arxiv.org/abs/1507.07680) [cs.NE (https://arxiv.org/archive/cs.NE)].
113. Hinton GE (2010). "A Practical Guide to Training Restricted Boltzmann Machines" (https://www.researchgate.net/publication/221166159). *Tech. Rep. UTML TR 2010-003*. Archived (https://web.archive.org/web/20210509123211/https://www.researchgate.net/publication/221166159_A_brief_introduction_to_Weightless_Neural_Systems) from the original on 9 May 2021. Retrieved 27 June 2017.
114. ESANN. 2009.
115. Bernard E (2021). *Introduction to machine learning* (https://www.wolfram.com/language/introduction-machine-learning/machine-learning-paradigms/#p-9). Champaign: Wolfram Media. p. 9. ISBN 978-1-57955-048-6. Archived (https://web.archive.org/web/20240519081126/https://www.wolfram.com/language/introduction-machine-learning/machine-learning-paradigms/#p-9) from the original on 19 May 2024. Retrieved 22 March 2023.
116. Bernard E (2021). *Introduction to machine learning* (https://www.wolfram.com/language/introduction-machine-learning/machine-learning-paradigms/#p-9). Champaign: Wolfram Media. p. 12. ISBN 978-1-57955-048-6. Archived (https://web.archive.org/web/20240519081126/https://www.wolfram.com/language/introduction-machine-learning/machine-learning-paradigms/#p-9) from the original on 19 May 2024. Retrieved 22 March 2023.
117. Bernard E (2021). *Introduction to Machine Learning* (https://www.wolfram.com/language/introduction-machine-learning/). Wolfram Media Inc. p. 9. ISBN 978-1-57955-048-6. Archived (https://web.archive.org/web/20240519081126/https://www.wolfram.com/language/introduction-machine-learning/) from the original on 19 May 2024. Retrieved 28 July 2022.
118. Ojha VK, Abraham A, Snášel V (1 April 2017). "Metaheuristic design of feedforward neural networks: A review of two decades of research". *Engineering Applications of Artificial Intelligence*. **60**: 97–116. arXiv:1705.05584 (https://arxiv.org/abs/1705.05584). Bibcode:2017arXiv170505584O (https://ui.adsabs.harvard.edu/abs/2017arXiv170505584O). doi:10.1016/j.engappai.2017.01.013 (https://doi.org/10.1016%2Fj.engappai.2017.01.013). S2CID 27910748 (https://api.semanticscholar.org/CorpusID:27910748).
119. Dominic, S., Das, R., Whitley, D., Anderson, C. (July 1991). "Genetic reinforcement learning for neural networks" (https://archive.org/details/ijcnn91seattlein01ieee). *IJCNN-91-Seattle International Joint Conference on Neural Networks*. IJCNN-91-Seattle International Joint Conference on Neural Networks. Seattle, Washington, US: IEEE. pp. 71–76. doi:10.1109/IJCNN.1991.155315 (https://doi.org/10.1109%2FIJCNN.1991.155315). ISBN 0-7803-0164-1.
120. Hoskins J, Himmelblau, D.M. (1992). "Process control via artificial neural networks and reinforcement learning". *Computers & Chemical Engineering*. **16** (4): 241–251. doi:10.1016/0098-1354(92)80045-B (https://doi.org/10.1016%2F0098-1354%2892%2980045-B).

121. Bertsekas D, Tsitsiklis J (1996). *Neuro-dynamic programming* (https://papers.nips.cc/paper/4741-deep-neural-networks-segment-neuronal-membranes-in-electron-microscopy-images). Athena Scientific. p. 512. ISBN 978-1-886529-10-6. Archived (https://web.archive.org/web/20170629172039/http://papers.nips.cc/paper/4741-deep-neural-networks-segment-neuronal-membranes-in-electron-microscopy-images) from the original on 29 June 2017. Retrieved 17 June 2017.
122. Secomandi N (2000). "Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands". *Computers & Operations Research*. **27** (11–12): 1201–1225. CiteSeerX 10.1.1.392.4034 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.392.4034). doi:10.1016/S0305-0548(99)00146-X (https://doi.org/10.1016%2FS0305-0548%2899%2900146-X).
123. de Rigo, D., Rizzoli, A. E., Soncini-Sessa, R., Weber, E., Zenesi, P. (2001). "Neuro-dynamic programming for the efficient management of reservoir networks" (http://www.mssanz.org.au/MODSIM01/MODSIM01.htm). *Proceedings of MODSIM 2001, International Congress on Modelling and Simulation*. MODSIM 2001, International Congress on Modelling and Simulation. Canberra, Australia: Modelling and Simulation Society of Australia and New Zealand. doi:10.5281/zenodo.7481 (https://doi.org/10.5281%2Fzenodo.7481). ISBN 0-86740-525-2. Archived (https://web.archive.org/web/20130807223658/http://mssanz.org.au/MODSIM01/MODSIM01.htm) from the original on 7 August 2013. Retrieved 29 July 2013.
124. Damas, M., Salmeron, M., Diaz, A., Ortega, J., Prieto, A., Olivares, G. (2000). "Genetic algorithms and neuro-dynamic programming: application to water supply networks". *Proceedings of 2000 Congress on Evolutionary Computation*. 2000 Congress on Evolutionary Computation. Vol. 1. La Jolla, California, US: IEEE. pp. 7–14. doi:10.1109/CEC.2000.870269 (https://doi.org/10.1109%2FCEC.2000.870269). ISBN 0-7803-6375-2.
125. Deng G, Ferris, M.C. (2008). "Neuro-dynamic programming for fractionated radiotherapy planning". *Optimization in Medicine*. Springer Optimization and Its Applications. Vol. 12. pp. 47–70. CiteSeerX 10.1.1.137.8288 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.137.8288). doi:10.1007/978-0-387-73299-2_3 (https://doi.org/10.1007%2F978-0-387-73299-2_3). ISBN 978-0-387-73298-5.
126. Bozinovski, S. (1982). "A self-learning system using secondary reinforcement". In R. Trappl (ed.) Cybernetics and Systems Research: Proceedings of the Sixth European Meeting on Cybernetics and Systems Research. North Holland. pp. 397–402. ISBN 978-0-444-86488-8.
127. Bozinovski, S. (2014) "Modeling mechanisms of cognition-emotion interaction in artificial neural networks, since 1981 (https://core.ac.uk/download/pdf/81973924.pdf) Archived (https://web.archive.org/web/20190323204838/https://core.ac.uk/download/pdf/81973924.pdf) 23 March 2019 at the Wayback Machine." Procedia Computer Science p. 255-263
128. Bozinovski S, Bozinovska L (2001). "Self-learning agents: A connectionist theory of emotion based on crossbar value judgment". *Cybernetics and Systems*. **32** (6): 637–667. doi:10.1080/01969720118145 (https://doi.org/10.1080%2F01969720118145). S2CID 8944741 (https://api.semanticscholar.org/CorpusID:8944741).
129. Salimans T, Ho J, Chen X, Sidor S, Sutskever I (7 September 2017). "Evolution Strategies as a Scalable Alternative to Reinforcement Learning". arXiv:1703.03864 (https://arxiv.org/abs/1703.03864) [stat.ML (https://arxiv.org/archive/stat.ML)].
130. Such FP, Madhavan V, Conti E, Lehman J, Stanley KO, Clune J (20 April 2018). "Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning". arXiv:1712.06567 (https://arxiv.org/abs/1712.06567) [cs.NE (https://arxiv.org/archive/cs.NE)].
131. "Artificial intelligence can 'evolve' to solve problems" (https://www.science.org/content/article/artificial-intelligence-can-evolve-solve-problems). *Science | AAAS*. 10 January 2018. Archived (https://web.archive.org/web/20211209231714/https://www.science.org/content/article/artificial-intelligence-can-evolve-solve-problems) from the original on 9 December 2021.

Retrieved 7 February 2018.

132. Turchetti C (2004), *Stochastic Models of Neural Networks*, Frontiers in artificial intelligence and applications: Knowledge-based intelligent engineering systems, vol. 102, IOS Press, ISBN 978-1-58603-388-0
133. Jospin LV, Laga H, Boussaid F, Buntine W, Bennamoun M (2022). "Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users". *IEEE Computational Intelligence Magazine*. Vol. 17, no. 2. pp. 29–48. arXiv:2007.06823 (https://arxiv.org/abs/2007.06823). doi:10.1109/mci.2022.3155327 (https://doi.org/10.1109%2Fmci.2022.3155327). ISSN 1556-603X (https://search.worldcat.org/issn/1556-603X). S2CID 220514248 (https://api.semanticscholar.org/CorpusID:220514248).
134. de Rigo, D., Castelletti, A., Rizzoli, A. E., Soncini-Sessa, R., Weber, E. (January 2005). "A selective improvement technique for fastening Neuro-Dynamic Programming in Water Resources Network Management" (http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2005/Papers/Paper4269.html). In Pavel Zítek (ed.). *Proceedings of the 16th IFAC World Congress – IFAC-PapersOnLine*. 16th IFAC World Congress (http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2005/Index.html). Vol. 16. Prague, Czech Republic: IFAC. pp. 7–12. doi:10.3182/20050703-6-CZ-1902.02172 (https://doi.org/10.3182%2F20050703-6-CZ-1902.02172). hdl:11311/255236 (https://hdl.handle.net/11311%2F255236). ISBN 978-3-902661-75-3. Archived (https://web.archive.org/web/20120426012450/http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2005/Papers/Paper4269.html) from the original on 26 April 2012. Retrieved 30 December 2011.
135. Ferreira C (2006). "Designing Neural Networks Using Gene Expression Programming". In A. Abraham, B. de Baets, M. Köppen, B. Nickolay (eds.). *Applied Soft Computing Technologies: The Challenge of Complexity* (http://www.gene-expression-programming.com/webpapers/Ferreira-ASCT2006.pdf) (PDF). Springer-Verlag. pp. 517–536. Archived (https://web.archive.org/web/20131219022806/http://www.gene-expression-programming.com/webpapers/Ferreira-ASCT2006.pdf) (PDF) from the original on 19 December 2013. Retrieved 8 October 2012.
136. Da, Y., Xiurun, G. (July 2005). "An improved PSO-based ANN with simulated annealing technique" (https://web.archive.org/web/20120425233611/http://www.dice.ucl.ac.be/esann/proceedings/electronicproceedings.htm). In T. Villmann (ed.). *New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks*. Vol. 63. Elsevier. pp. 527–533. doi:10.1016/j.neucom.2004.07.002 (https://doi.org/10.1016%2Fj.neucom.2004.07.002). Archived from the original (http://www.dice.ucl.ac.be/esann/proceedings/electronicproceedings.htm) on 25 April 2012. Retrieved 30 December 2011.
137. Wu, J., Chen, E. (May 2009). "A Novel Nonparametric Regression Ensemble for Rainfall Forecasting Using Particle Swarm Optimization Technique Coupled with Artificial Neural Network" (https://web.archive.org/web/20141231221755/http://www2.mae.cuhk.edu.hk/~isnn2009/). In Wang, H., Shen, Y., Huang, T., Zeng, Z. (eds.). *6th International Symposium on Neural Networks, ISNN 2009*. Lecture Notes in Computer Science. Vol. 5553. Springer. pp. 49–58. doi:10.1007/978-3-642-01513-7_6 (https://doi.org/10.1007%2F978-3-642-01513-7_6). ISBN 978-3-642-01215-0. Archived from the original (http://www2.mae.cuhk.edu.hk/~isnn2009/) on 31 December 2014. Retrieved 1 January 2012.
138. Ting Qin, Zonghai Chen, Haitao Zhang, Sifu Li, Wei Xiang, Ming Li (2004). "A learning algorithm of CMAC based on RLS" (http://www-control.eng.cam.ac.uk/Homepage/papers/cued_control_998.pdf) (PDF). *Neural Processing Letters*. **19** (1): 49–61. doi:10.1023/B:NEPL.0000016847.18175.60 (https://doi.org/10.1023%2FB%3ANEPL.0000016847.18175.60). S2CID 6233899 (https://api.semanticscholar.org/CorpusID:6233899). Archived (https://web.archive.org/web/20210414103815/http://www-control.eng.cam.ac.uk/Homepage/papers/cued_control_998.pdf) (PDF) from the original on 14 April 2021. Retrieved 30 January 2019.
139. Ting Qin, Haitao Zhang, Zonghai Chen, Wei Xiang (2005). "Continuous CMAC-QRLS and

its systolic array" (http://www-control.eng.cam.ac.uk/Homepage/papers/cued_control_997.pdf) (PDF). *Neural Processing Letters*. **22** (1): 1–16. doi:10.1007/s11063-004-2694-0 (https://doi.org/10.1007%2Fs11063-004-2694-0). S2CID 16095286 (https://api.semanticscholar.org/CorpusID:16095286). Archived (https://web.archive.org/web/20181118122850/http://www-control.eng.cam.ac.uk/Homepage/papers/cued_control_997.pdf) (PDF) from the original on 18 November 2018. Retrieved 30 January 2019.

140. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Computation*. **1** (4): 541–551. doi:10.1162/neco.1989.1.4.541 (https://doi.org/10.1162%2Fneco.1989.1.4.541). S2CID 41312633 (https://api.semanticscholar.org/CorpusID:41312633).
141. Yann LeCun (2016). Slides on Deep Learning Online (https://indico.cern.ch/event/510372/) Archived (https://web.archive.org/web/20160423021403/https://indico.cern.ch/event/510372/) 23 April 2016 at the Wayback Machine
142. Hochreiter S, Schmidhuber J (1 November 1997). "Long Short-Term Memory". *Neural Computation*. **9** (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735 (https://doi.org/10.1162%2Fneco.1997.9.8.1735). ISSN 0899-7667 (https://search.worldcat.org/issn/0899-7667). PMID 9377276 (https://pubmed.ncbi.nlm.nih.gov/9377276). S2CID 1915014 (https://api.semanticscholar.org/CorpusID:1915014).
143. Sak H, Senior A, Beaufays F (2014). "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling" (https://web.archive.org/web/20180424203806/https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf) (PDF). Archived from the original (https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf) (PDF) on 24 April 2018.
144. Li X, Wu X (15 October 2014). "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition". arXiv:1410.4281 (https://arxiv.org/abs/1410.4281) [cs.CL (https://arxiv.org/archive/cs.CL)].
145. Fan Y, Qian Y, Xie F, Soong FK (2014). "TTS synthesis with bidirectional LSTM based Recurrent Neural Networks" (https://www.researchgate.net/publication/287741874). *Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech*: 1964–1968. Retrieved 13 June 2017.
146. Schmidhuber J (2015). "Deep Learning" (https://doi.org/10.4249%2Fscholarpedia.32832). *Scholarpedia*. **10** (11): 85–117. Bibcode:2015SchpJ..1032832S (https://ui.adsabs.harvard.edu/abs/2015SchpJ..1032832S). doi:10.4249/scholarpedia.32832 (https://doi.org/10.4249%2Fscholarpedia.32832).
147. Zen H, Sak H (2015). "Unidirectional Long Short-Term Memory Recurrent Neural Network with Recurrent Output Layer for Low-Latency Speech Synthesis" (https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43266.pdf) (PDF). *Google.com*. ICASSP. pp. 4470–4474. Archived (https://web.archive.org/web/20210509123113/https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43266.pdf) (PDF) from the original on 9 May 2021. Retrieved 27 June 2017.
148. Fan B, Wang L, Soong FK, Xie L (2015). "Photo-Real Talking Head with Deep Bidirectional LSTM" (https://www.microsoft.com/en-us/research/wp-content/uploads/2015/04/icassp2015_fanbo_1009.pdf) (PDF). *Proceedings of ICASSP*. Archived (https://web.archive.org/web/20171101052317/https://www.microsoft.com/en-us/research/wp-content/uploads/2015/04/icassp2015_fanbo_1009.pdf) (PDF) from the original on 1 November 2017. Retrieved 27 June 2017.
149. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. (5 December 2017). "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". arXiv:1712.01815 (https://arxiv.org/abs/1712.01815) [cs.AI (https://arxiv.org/archive/cs.AI)].
150. Probst P, Boulesteix AL, Bischl B (26 February 2018). "Tunability: Importance of Hyperparameters of Machine Learning Algorithms". *J. Mach. Learn. Res.* **20**: 53:1–53:32.

S2CID 88515435 (https://api.semanticscholar.org/CorpusID:88515435).

151. Zoph B, Le QV (4 November 2016). "Neural Architecture Search with Reinforcement Learning". arXiv:1611.01578 (https://arxiv.org/abs/1611.01578) [cs.LG (https://arxiv.org/archive/cs.LG)].
152. Haifeng Jin, Qingquan Song, Xia Hu (2019). "Auto-keras: An efficient neural architecture search system" (https://autokeras.com/). *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. arXiv:1806.10282 (https://arxiv.org/abs/1806.10282). Archived (https://web.archive.org/web/20190821163310/https://autokeras.com/) from the original on 21 August 2019. Retrieved 21 August 2019 – via autokeras.com.
153. Claesen M, De Moor B (2015). "Hyperparameter Search in Machine Learning". arXiv:1502.02127 (https://arxiv.org/abs/1502.02127) [cs.LG (https://arxiv.org/archive/cs.LG)]. Bibcode:2015arXiv150202127C (https://ui.adsabs.harvard.edu/abs/2015arXiv150202127C/abstract)
154. Esch R (1990). "Functional Approximation". *Handbook of Applied Mathematics* (Springer US ed.). Boston, MA: Springer US. pp. 928–987. doi:10.1007/978-1-4684-1423-3_17 (https://doi.org/10.1007%2F978-1-4684-1423-3_17). ISBN 978-1-4684-1423-3.
155. Sarstedt M, Moo E (2019). "Regression Analysis" (https://link.springer.com/chapter/10.1007/978-3-662-56707-4_7#Sec1). *A Concise Guide to Market Research*. Springer Texts in Business and Economics. Springer Berlin Heidelberg. pp. 209–256. doi:10.1007/978-3-662-56707-4_7 (https://doi.org/10.1007%2F978-3-662-56707-4_7). ISBN 978-3-662-56706-7. S2CID 240396965 (https://api.semanticscholar.org/CorpusID:240396965). Archived (https://web.archive.org/web/20230320212723/https://link.springer.com/chapter/10.1007/978-3-662-56707-4_7#Sec1) from the original on 20 March 2023. Retrieved 20 March 2023.
156. Tian J, Tan Y, Sun C, Zeng J, Jin Y (December 2016). "A self-adaptive similarity-based fitness approximation for evolutionary optimization" (https://ieeexplore.ieee.org/document/7850209). *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. pp. 1–8. doi:10.1109/SSCI.2016.7850209 (https://doi.org/10.1109%2FSSCI.2016.7850209). ISBN 978-1-5090-4240-1. S2CID 14948018 (https://api.semanticscholar.org/CorpusID:14948018). Archived (https://web.archive.org/web/20240519082200/https://ieeexplore.ieee.org/document/7850209) from the original on 19 May 2024. Retrieved 22 March 2023.
157. Alaloul WS, Qureshi AH (2019). "Data Processing Using Artificial Neural Networks" (https://www.intechopen.com/chapters/71673). *Dynamic Data Assimilation - Beating the Uncertainties*. doi:10.5772/intechopen.91935 (https://doi.org/10.5772%2Fintechopen.91935). ISBN 978-1-83968-083-0. S2CID 219735060 (https://api.semanticscholar.org/CorpusID:219735060). Archived (https://web.archive.org/web/20230320212722/https://www.intechopen.com/chapters/71673) from the original on 20 March 2023. Retrieved 20 March 2023.
158. Pal M, Roy R, Basu J, Bepari MS (2013). "Blind source separation: A review and analysis" (https://ieeexplore.ieee.org/document/6709849). *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*. IEEE. pp. 1–5. doi:10.1109/ICSDA.2013.6709849 (https://doi.org/10.1109%2FICSDA.2013.6709849). ISBN 978-1-4799-2378-6. S2CID 37566823 (https://api.semanticscholar.org/CorpusID:37566823). Archived (https://web.archive.org/web/20230320212720/https://ieeexplore.ieee.org/document/6709849) from the original on 20 March 2023. Retrieved 20 March 2023.
159. Zissis D (October 2015). "A cloud based architecture capable of perceiving and predicting multiple vessel behaviour" (https://zenodo.org/record/848743). *Applied Soft Computing*. **35**: 652–661. doi:10.1016/j.asoc.2015.07.002 (https://doi.org/10.1016%2Fj.asoc.2015.07.002). Archived (https://web.archive.org/web/20200726091505/https://zenodo.org/record/848743) from the original on 26 July 2020. Retrieved 18 July 2019.
160. Sengupta N, Sahidullah, Md, Saha, Goutam (August 2016). "Lung sound classification using cepstral-based statistical features". *Computers in Biology and Medicine*. **75** (1): 118–129.

doi:10.1016/j.compbiomed.2016.05.013 (https://doi.org/10.1016%2Fj.compbiomed.2016.05.013). PMID 27286184 (https://pubmed.ncbi.nlm.nih.gov/27286184).

161. Choy, Christopher B., et al. "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction (https://arxiv.org/abs/1604.00449) Archived (https://web.archive.org/web/20200726091721/https://arxiv.org/abs/1604.00449) 26 July 2020 at the Wayback Machine." European conference on computer vision. Springer, Cham, 2016.
162. Turek, Fred D. (March 2007). "Introduction to Neural Net Machine Vision" (http://www.vision-systems.com/articles/print/volume-12/issue-3/features/introduction-to-neural-net-machine-vision.html). *Vision Systems Design*. **12** (3). Archived (https://web.archive.org/web/20130516124148/http://www.vision-systems.com/articles/print/volume-12/issue-3/features/introduction-to-neural-net-machine-vision.html) from the original on 16 May 2013. Retrieved 5 March 2013.
163. Maitra DS, Bhattacharya U, Parui SK (August 2015). "CNN based common approach to handwritten character recognition of multiple scripts" (https://ieeexplore.ieee.org/document/7333916). *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. pp. 1021–1025. doi:10.1109/ICDAR.2015.7333916 (https://doi.org/10.1109%2FICDAR.2015.7333916). ISBN 978-1-4799-1805-8. S2CID 25739012 (https://api.semanticscholar.org/CorpusID:25739012). Archived (https://web.archive.org/web/20231016190918/https://ieeexplore.ieee.org/document/7333916) from the original on 16 October 2023. Retrieved 18 March 2021.
164. Gessler J (August 2021). "Sensor for food analysis applying impedance spectroscopy and artificial neural networks" (https://riunet.upv.es/handle/10251/174498). *RiuNet UPV* (1): 8–12. Archived (https://web.archive.org/web/20211021115443/https://riunet.upv.es/handle/10251/174498) from the original on 21 October 2021. Retrieved 21 October 2021.
165. French J (2016). "The time traveller's CAPM". *Investment Analysts Journal*. **46** (2): 81–96. doi:10.1080/10293523.2016.1255469 (https://doi.org/10.1080%2F10293523.2016.1255469). S2CID 157962452 (https://api.semanticscholar.org/CorpusID:157962452).
166. Roman M. Balabin, Ekaterina I. Lomakina (2009). "Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies". *J. Chem. Phys.* **131** (7): 074104. Bibcode:2009JChPh.131g4104B (https://ui.adsabs.harvard.edu/abs/2009JChPh.131g4104B). doi:10.1063/1.3206326 (https://doi.org/10.1063%2F1.3206326). PMID 19708729 (https://pubmed.ncbi.nlm.nih.gov/19708729).
167. Silver D, et al. (2016). "Mastering the game of Go with deep neural networks and tree search" (http://web.iitd.ac.in/~sumeet/Silver16.pdf) (PDF). *Nature*. **529** (7587): 484–489. Bibcode:2016Natur.529..484S (https://ui.adsabs.harvard.edu/abs/2016Natur.529..484S). doi:10.1038/nature16961 (https://doi.org/10.1038%2Fnature16961). PMID 26819042 (https://pubmed.ncbi.nlm.nih.gov/26819042). S2CID 515925 (https://api.semanticscholar.org/CorpusID:515925). Archived (https://web.archive.org/web/20181123112812/http://web.iitd.ac.in/~sumeet/Silver16.pdf) (PDF) from the original on 23 November 2018. Retrieved 31 January 2019.
168. Pasick A (27 March 2023). "Artificial Intelligence Glossary: Neural Networks and Other Terms Explained" (https://www.nytimes.com/article/ai-artificial-intelligence-glossary.html). *The New York Times*. ISSN 0362-4331 (https://search.worldcat.org/issn/0362-4331). Archived (https://web.archive.org/web/20230901183440/https://www.nytimes.com/article/ai-artificial-intelligence-glossary.html) from the original on 1 September 2023. Retrieved 22 April 2023.
169. Schechner S (15 June 2017). "Facebook Boosts A.I. to Block Terrorist Propaganda" (https://www.wsj.com/articles/facebook-boosts-a-i-to-block-terrorist-propaganda-1497546000). *The Wall Street Journal*. ISSN 0099-9660 (https://search.worldcat.org/issn/0099-9660). Archived (https://web.archive.org/web/20240519082135/https://www.wsj.com/articles/facebook-boosts-a-i-to-block-terrorist-propaganda-1497546000) from the original on 19 May 2024. Retrieved 16 June 2017.
170. Ganesan N (2010). "Application of Neural Networks in Diagnosing Cancer Disease Using

170. Ganesan N (2010). "Application of Neural Networks in Diagnosing Cancer Disease Using Demographic Data" (https://doi.org/10.5120%2F476-783). *International Journal of Computer Applications*. **1** (26): 81–97. Bibcode:2010IJCA....1z..81G (https://ui.adsabs.harvard.edu/abs/2010IJCA....1z..81G). doi:10.5120/476-783 (https://doi.org/10.5120%2F476-783).
171. Bottaci L (1997). "Artificial Neural Networks Applied to Outcome Prediction for Colorectal Cancer Patients in Separate Institutions" (https://web.archive.org/web/20181123170444/http://www.lcc.uma.es/~jja/recidiva/042.pdf) (PDF). *Lancet*. **350** (9076). The Lancet: 469–72. doi:10.1016/S0140-6736(96)11196-X (https://doi.org/10.1016%2FS0140-6736%2896%2911196-X). PMID 9274582 (https://pubmed.ncbi.nlm.nih.gov/9274582). S2CID 18182063 (https://api.semanticscholar.org/CorpusID:18182063). Archived from the original (http://www.lcc.uma.es/~jja/recidiva/042.pdf) (PDF) on 23 November 2018. Retrieved 2 May 2012.
172. Alizadeh E, Lyons SM, Castle JM, Prasad A (2016). "Measuring systematic changes in invasive cancer cell shape using Zernike moments" (http://pubs.rsc.org/en/Content/ArticleLanding/2016/IB/C6IB00100A). *Integrative Biology*. **8** (11): 1183–1193. doi:10.1039/C6IB00100A (https://doi.org/10.1039%2FC6IB00100A). PMID 27735002 (https://pubmed.ncbi.nlm.nih.gov/27735002). Archived (https://web.archive.org/web/20240519082133/https://pubs.rsc.org/en/Content/ArticleLanding/2016/IB/C6IB00100A) from the original on 19 May 2024. Retrieved 28 March 2017.
173. Lyons S (2016). "Changes in cell shape are correlated with metastatic potential in murine" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4810736). *Biology Open*. **5** (3): 289–299. doi:10.1242/bio.013409 (https://doi.org/10.1242%2Fbio.013409). PMC 4810736 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4810736). PMID 26873952 (https://pubmed.ncbi.nlm.nih.gov/26873952).
174. Nabian MA, Meidani H (28 August 2017). "Deep Learning for Accelerated Reliability Analysis of Infrastructure Networks". *Computer-Aided Civil and Infrastructure Engineering*. **33** (6): 443–458. arXiv:1708.08551 (https://arxiv.org/abs/1708.08551). Bibcode:2017arXiv170808551N (https://ui.adsabs.harvard.edu/abs/2017arXiv170808551N). doi:10.1111/mice.12359 (https://doi.org/10.1111%2Fmice.12359). S2CID 36661983 (https://api.semanticscholar.org/CorpusID:36661983).
175. Nabian MA, Meidani H (2018). "Accelerating Stochastic Assessment of Post-Earthquake Transportation Network Connectivity via Machine-Learning-Based Surrogates" (https://trid.trb.org/view/1496617). *Transportation Research Board 97th Annual Meeting*. Archived (https://web.archive.org/web/20180309120108/https://trid.trb.org/view/1496617) from the original on 9 March 2018. Retrieved 14 March 2018.
176. Díaz E, Brotons V, Tomás R (September 2018). "Use of artificial neural networks to predict 3-D elastic settlement of foundations on soils with inclined bedrock" (https://doi.org/10.1016%2Fj.sandf.2018.08.001). *Soils and Foundations*. **58** (6): 1414–1422. Bibcode:2018SoFou..58.1414D (https://ui.adsabs.harvard.edu/abs/2018SoFou..58.1414D). doi:10.1016/j.sandf.2018.08.001 (https://doi.org/10.1016%2Fj.sandf.2018.08.001). hdl:10045/81208 (https://hdl.handle.net/10045%2F81208). ISSN 0038-0806 (https://search.worldcat.org/issn/0038-0806).
177. Tayebiyan A, Mohammad TA, Ghazali AH, Mashohor S. "Artificial Neural Network for Modelling Rainfall-Runoff" (http://www.pertanika.upm.edu.my/pjtas/browse/regular-issue?article=JST-0566-2015). *Pertanika Journal of Science & Technology*. **24** (2): 319–330. Archived (https://web.archive.org/web/20230517014047/http://www.pertanika.upm.edu.my/pjtas/browse/regular-issue?article=JST-0566-2015) from the original on 17 May 2023. Retrieved 17 May 2023.
178. Govindaraju RS (1 April 2000). "Artificial Neural Networks in Hydrology. I: Preliminary Concepts". *Journal of Hydrologic Engineering*. **5** (2): 115–123. doi:10.1061/(ASCE)1084-0699(2000)5:2(115) (https://doi.org/10.1061%2F%28ASCE%291084-0699%282000%295%3A2%28115%29).
179. Govindaraju RS (1 April 2000). "Artificial Neural Networks in Hydrology. II: Hydrologic Applications". *Journal of Hydrologic Engineering*. **5** (2): 124–137. doi:10.1061/(ASCE)1084-

Applications". *Journal of Hydrologic Engineering*. **5** (2): 124–131. doi:10.1061/(ASCE)1084-0699(2000)5:2(124) (https://doi.org/10.1061%2F%28ASCE%291084-0699%282000%295%3A2%28124%29).

180. Peres DJ, Iuppa C, Cavallaro L, Cancelliere A, Foti E (1 October 2015). "Significant wave height record extension by neural networks and reanalysis wind data". *Ocean Modelling*. **94**: 128–140. Bibcode:2015OcMod..94..128P (https://ui.adsabs.harvard.edu/abs/2015OcMod..94..128P). doi:10.1016/j.ocemod.2015.08.002 (https://doi.org/10.1016%2Fj.ocemod.2015.08.002).
181. Dwarakish GS, Rakshith S, Natesan U (2013). "Review on Applications of Neural Network in Coastal Engineering" (http://www.ciitresearch.org/dl/index.php/aiml/article/view/AIML072013007). *Artificial Intelligent Systems and Machine Learning*. **5** (7): 324–331. Archived (https://web.archive.org/web/20170815185634/http://www.ciitresearch.org/dl/index.php/aiml/article/view/AIML072013007) from the original on 15 August 2017. Retrieved 5 July 2017.
182. Ermini L, Catani F, Casagli N (1 March 2005). "Artificial Neural Networks applied to landslide susceptibility assessment". *Geomorphology*. Geomorphological hazard and human impact in mountain environments. **66** (1): 327–343. Bibcode:2005Geomo..66..327E (https://ui.adsabs.harvard.edu/abs/2005Geomo..66..327E). doi:10.1016/j.geomorph.2004.09.025 (https://doi.org/10.1016%2Fj.geomorph.2004.09.025).
183. Nix R, Zhang J (May 2017). "Classification of Android apps and malware using deep neural networks". *2017 International Joint Conference on Neural Networks (IJCNN)*. pp. 1871–1878. doi:10.1109/IJCNN.2017.7966078 (https://doi.org/10.1109%2FIJCNN.2017.7966078). ISBN 978-1-5090-6182-2. S2CID 8838479 (https://api.semanticscholar.org/CorpusID:8838479).
184. "Detecting Malicious URLs" (https://web.archive.org/web/20190714201955/http://www.sysnet.ucsd.edu/projects/url/). *The systems and networking group at UCSD*. Archived from the original (http://www.sysnet.ucsd.edu/projects/url/) on 14 July 2019. Retrieved 15 February 2019.
185. Homayoun S, Ahmadzadeh M, Hashemi S, Dehghantanha A, Khayami R (2018), Dehghantanha A, Conti M, Dargahi T (eds.), "BoTShark: A Deep Learning Approach for Botnet Traffic Detection", *Cyber Threat Intelligence*, Advances in Information Security, vol. 70, Springer International Publishing, pp. 137–153, doi:10.1007/978-3-319-73951-9_7 (https://doi.org/10.1007%2F978-3-319-73951-9_7), ISBN 978-3-319-73951-9
186. Ghosh, Reilly (January 1994). "Credit card fraud detection with a neural-network". *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences HICSS-94*. Vol. 3. pp. 621–630. doi:10.1109/HICSS.1994.323314 (https://doi.org/10.1109%2FHICSS.1994.323314). ISBN 978-0-8186-5090-1. S2CID 13260377 (https://api.semanticscholar.org/CorpusID:13260377).
187. Ananthaswamy A (19 April 2021). "Latest Neural Nets Solve World's Hardest Equations Faster Than Ever Before" (https://www.quantamagazine.org/new-neural-networks-solve-hardest-equations-faster-than-ever-20210419/). *Quanta Magazine*. Archived (https://web.archive.org/web/20240519082138/https://www.quantamagazine.org/new-neural-networks-solve-hardest-equations-faster-than-ever-20210419/) from the original on 19 May 2024. Retrieved 12 May 2021.
188. "AI has cracked a key mathematical puzzle for understanding our world" (https://www.technologyreview.com/2020/10/30/1011435/ai-fourier-neural-network-cracks-navier-stokes-and-partial-differential-equations/). *MIT Technology Review*. Archived (https://web.archive.org/web/20240519082138/https://www.technologyreview.com/2020/10/30/1011435/ai-fourier-neural-network-cracks-navier-stokes-and-partial-differential-equations/) from the original on 19 May 2024. Retrieved 19 November 2020.
189. "Caltech Open-Sources AI for Solving Partial Differential Equations" (https://www.infoq.com/news/2020/12/caltech-ai-pde/). *InfoQ*. Archived (https://web.archive.org/web/20210125233952/https://www.infoq.com/news/2020/12/caltech-ai-pde/) from the original on 25 January 2021. Retrieved 20 January 2021.

190. Nagy A (28 June 2019). "Variational Quantum Monte Carlo Method with a Neural-Network Ansatz for Open Quantum Systems". *Physical Review Letters*. **122** (25): 250501. arXiv:1902.09483 (https://arxiv.org/abs/1902.09483). Bibcode:2019PhRvL.122y0501N (https://ui.adsabs.harvard.edu/abs/2019PhRvL.122y0501N). doi:10.1103/PhysRevLett.122.250501 (https://doi.org/10.1103%2FPhysRevLett.122.250501). PMID 31347886 (https://pubmed.ncbi.nlm.nih.gov/31347886). S2CID 119074378 (https://api.semanticscholar.org/CorpusID:119074378).
191. Yoshioka N, Hamazaki R (28 June 2019). "Constructing neural stationary states for open quantum many-body systems". *Physical Review B*. **99** (21): 214306. arXiv:1902.07006 (https://arxiv.org/abs/1902.07006). Bibcode:2019PhRvB..99u4306Y (https://ui.adsabs.harvard.edu/abs/2019PhRvB..99u4306Y). doi:10.1103/PhysRevB.99.214306 (https://doi.org/10.1103%2FPhysRevB.99.214306). S2CID 119470636 (https://api.semanticscholar.org/CorpusID:119470636).
192. Hartmann MJ, Carleo G (28 June 2019). "Neural-Network Approach to Dissipative Quantum Many-Body Dynamics". *Physical Review Letters*. **122** (25): 250502. arXiv:1902.05131 (https://arxiv.org/abs/1902.05131). Bibcode:2019PhRvL.122y0502H (https://ui.adsabs.harvard.edu/abs/2019PhRvL.122y0502H). doi:10.1103/PhysRevLett.122.250502 (https://doi.org/10.1103%2FPhysRevLett.122.250502). PMID 31347862 (https://pubmed.ncbi.nlm.nih.gov/31347862). S2CID 119357494 (https://api.semanticscholar.org/CorpusID:119357494).
193. Vicentini F, Biella A, Regnault N, Ciuti C (28 June 2019). "Variational Neural-Network Ansatz for Steady States in Open Quantum Systems". *Physical Review Letters*. **122** (25): 250503. arXiv:1902.10104 (https://arxiv.org/abs/1902.10104). Bibcode:2019PhRvL.122y0503V (https://ui.adsabs.harvard.edu/abs/2019PhRvL.122y0503V). doi:10.1103/PhysRevLett.122.250503 (https://doi.org/10.1103%2FPhysRevLett.122.250503). PMID 31347877 (https://pubmed.ncbi.nlm.nih.gov/31347877). S2CID 119504484 (https://api.semanticscholar.org/CorpusID:119504484).
194. Forrest MD (April 2015). "Simulation of alcohol action upon a detailed Purkinje neuron model and a simpler surrogate model that runs >400 times faster" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4417229). *BMC Neuroscience*. **16** (27): 27. doi:10.1186/s12868-015-0162-6 (https://doi.org/10.1186%2Fs12868-015-0162-6). PMC 4417229 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4417229). PMID 25928094 (https://pubmed.ncbi.nlm.nih.gov/25928094).
195. Wieczorek S, Filipiak D, Filipowska A (2018). "Semantic Image-Based Profiling of Users' Interests with Neural Networks" (https://www.researchgate.net/publication/328964756). *Studies on the Semantic Web*. **36** (Emerging Topics in Semantic Technologies). doi:10.3233/978-1-61499-894-5-179 (https://doi.org/10.3233%2F978-1-61499-894-5-179). Archived (https://web.archive.org/web/20240519082144/https://www.researchgate.net/publication/328964756_Semantic_Image-Based_Profiling_of_Users%27_Interests_with_Neural_Networks) from the original on 19 May 2024. Retrieved 20 January 2024.
196. Merchant A, Batzner S, Schoenholz SS, Aykol M, Cheon G, Cubuk ED (December 2023). "Scaling deep learning for materials discovery" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10700131). *Nature*. **624** (7990): 80–85. Bibcode:2023Natur.624...80M (https://ui.adsabs.harvard.edu/abs/2023Natur.624...80M). doi:10.1038/s41586-023-06735-9 (https://doi.org/10.1038%2Fs41586-023-06735-9). ISSN 1476-4687 (https://search.worldcat.org/issn/1476-4687). PMC 10700131 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10700131). PMID 38030720 (https://pubmed.ncbi.nlm.nih.gov/38030720).
197. Siegelmann H, Sontag E (1991). "Turing computability with neural nets" (http://www.math.rutgers.edu/~sontag/FTPDIR/aml-turing.pdf) (PDF). *Appl. Math. Lett*. **4** (6): 77–80. doi:10.1016/0893-9659(91)90080-F (https://doi.org/10.1016%2F0893-9659%2891%2990080-F). Archived (https://web.archive.org/web/20240519082138/http://www.math.rutgers.edu/~sontag/FTPDIR/aml-turing.pdf) (PDF) from the original on 19 May 2024. Retrieved 10 January 2017.

198. Bains S (3 November 1998). "Analog computer trumps Turing model" (https://www.eetimes.com/analog-computer-trumps-turing-model/). *EE Times*. Archived (https://web.archive.org/web/20230511152308/https://www.eetimes.com/analog-computer-trumps-turing-model/) from the original on 11 May 2023. Retrieved 11 May 2023.

199. Balcázar J (July 1997). "Computational Power of Neural Networks: A Kolmogorov Complexity Characterization". *IEEE Transactions on Information Theory*. **43** (4): 1175–1183. CiteSeerX 10.1.1.411.7782 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.411.7782). doi:10.1109/18.605580 (https://doi.org/10.1109%2F18.605580).

200. MacKay DJ (2003). *Information Theory, Inference, and Learning Algorithms* (http://www.inference.phy.cam.ac.uk/itprnn/book.pdf) (PDF). Cambridge University Press. ISBN 978-0-521-64298-9. Archived (https://web.archive.org/web/20161019163258/http://www.inference.phy.cam.ac.uk/itprnn/book.pdf) (PDF) from the original on 19 October 2016. Retrieved 11 June 2016.

201. Cover T (1965). "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition" (http://www-isl.stanford.edu/people/cover/papers/paper2.pdf) (PDF). *IEEE Transactions on Electronic Computers*. EC-14 (3). IEEE: 326–334. doi:10.1109/PGEC.1965.264137 (https://doi.org/10.1109%2FPGEC.1965.264137). Archived (https://web.archive.org/web/20160305031348/http://www-isl.stanford.edu/people/cover/papers/paper2.pdf) (PDF) from the original on 5 March 2016. Retrieved 10 March 2020.

202. Gerald F (2019). "Reproducibility and Experimental Design for Machine Learning on Audio and Multimedia Data". *Proceedings of the 27th ACM International Conference on Multimedia*. ACM. pp. 2709–2710. doi:10.1145/3343031.3350545 (https://doi.org/10.1145%2F3343031.3350545). ISBN 978-1-4503-6889-6. S2CID 204837170 (https://api.semanticscholar.org/CorpusID:204837170).

203. "Stop tinkering, start measuring! Predictable experimental design of Neural Network experiments" (https://web.archive.org/web/20220418025904/http://tfmeter.icsi.berkeley.edu/). *The Tensorflow Meter*. Archived from the original (http://tfmeter.icsi.berkeley.edu/) on 18 April 2022. Retrieved 10 March 2020.

204. Lee J, Xiao L, Schoenholz SS, Bahri Y, Novak R, Sohl-Dickstein J, et al. (2020). "Wide neural networks of any depth evolve as linear models under gradient descent". *Journal of Statistical Mechanics: Theory and Experiment*. **2020** (12): 124002. arXiv:1902.06720 (https://arxiv.org/abs/1902.06720). Bibcode:2020JSMTE2020l4002L (https://ui.adsabs.harvard.edu/abs/2020JSMTE2020l4002L). doi:10.1088/1742-5468/abc62b (https://doi.org/10.1088%2F1742-5468%2Fabc62b). S2CID 62841516 (https://api.semanticscholar.org/CorpusID:62841516).

205. Arthur Jacot, Franck Gabriel, Clement Hongler (2018). *Neural Tangent Kernel: Convergence and Generalization in Neural Networks* (https://proceedings.neurips.cc/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf) (PDF). 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, Canada. Archived (https://web.archive.org/web/20220622033100/https://proceedings.neurips.cc/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf) (PDF) from the original on 22 June 2022. Retrieved 4 June 2022.

206. Xu ZJ, Zhang Y, Xiao Y (2019). "Training Behavior of Deep Neural Network in Frequency Domain". In Gedeon T, Wong K, Lee M (eds.). *Neural Information Processing*. Lecture Notes in Computer Science. Vol. 11953. Springer, Cham. pp. 264–274. arXiv:1807.01251 (https://arxiv.org/abs/1807.01251). doi:10.1007/978-3-030-36708-4_22 (https://doi.org/10.1007%2F978-3-030-36708-4_22). ISBN 978-3-030-36707-7. S2CID 49562099 (https://api.semanticscholar.org/CorpusID:49562099).

207. Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, et al. (2019). "On the Spectral Bias of Neural Networks" (http://proceedings.mlr.press/v97/rahaman19a/rahaman19a.pdf) (PDF). *Proceedings of the 36th International Conference on Machine Learning*. **97**: 5301–5310. arXiv:1806.08734 (https://arxiv.org/abs/1806.08734). Archived (https://web.archive.org/web/20221022155951/http://proceedings.mlr.press/v97/ra

Archived (https://web.archive.org/web/20221022153951/http://proceedings.mlr.press/v97/rahaman19a/rahaman19a.pdf) (PDF) from the original on 22 October 2022. Retrieved 4 June 2022.

208. Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, Zheng Ma (2020). "Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks". *Communications in Computational Physics*. **28** (5): 1746–1767. arXiv:1901.06523 (https://arxiv.org/abs/1901.06523). Bibcode:2020CCoPh..28.1746X (https://ui.adsabs.harvard.edu/abs/2020CCoPh..28.1746X). doi:10.4208/cicp.OA-2020-0085 (https://doi.org/10.4208%2Fcicp.OA-2020-0085). S2CID 58981616 (https://api.semanticscholar.org/CorpusID:58981616).
209. Tao Luo, Zheng Ma, Zhi-Qin John Xu, Yaoyu Zhang (2019). "Theory of the Frequency Principle for General Deep Neural Networks". arXiv:1906.09235 (https://arxiv.org/abs/1906.09235) [cs.LG (https://arxiv.org/archive/cs.LG)].
210. Xu ZJ, Zhou H (18 May 2021). "Deep Frequency Principle Towards Understanding Why Deeper Learning is Faster" (https://ojs.aaai.org/index.php/AAAI/article/view/17261). *Proceedings of the AAAI Conference on Artificial Intelligence*. **35** (12): 10541–10550. arXiv:2007.14313 (https://arxiv.org/abs/2007.14313). doi:10.1609/aaai.v35i12.17261 (https://doi.org/10.1609%2Faaai.v35i12.17261). ISSN 2374-3468 (https://search.worldcat.org/issn/2374-3468). S2CID 220831156 (https://api.semanticscholar.org/CorpusID:220831156). Archived (https://web.archive.org/web/20211005142300/https://ojs.aaai.org/index.php/AAAI/article/view/17261) from the original on 5 October 2021. Retrieved 5 October 2021.
211. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (1 May 2019). "Continual lifelong learning with neural networks: A review" (https://doi.org/10.1016%2Fj.neunet.2019.01.012). *Neural Networks*. **113**: 54–71. arXiv:1802.07569 (https://arxiv.org/abs/1802.07569). doi:10.1016/j.neunet.2019.01.012 (https://doi.org/10.1016%2Fj.neunet.2019.01.012). ISSN 0893-6080 (https://search.worldcat.org/issn/0893-6080). PMID 30780045 (https://pubmed.ncbi.nlm.nih.gov/30780045).
212. Dean Pomerleau, "Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving"
213. Dewdney AK (1 April 1997). *Yes, we have no neutrons: an eye-opening tour through the twists and turns of bad science* (https://books.google.com/books?id=KcHaAAAAMAAJ&pg=PA82). Wiley. p. 82. ISBN 978-0-471-10806-1.
214. NASA – Dryden Flight Research Center – News Room: News Releases: NASA NEURAL NETWORK PROJECT PASSES MILESTONE (http://www.nasa.gov/centers/dryden/news/NewsReleases/2003/03-49.html) Archived (https://web.archive.org/web/20100402065100/http://www.nasa.gov/centers/dryden/news/NewsReleases/2003/03-49.html) 2 April 2010 at the Wayback Machine. Nasa.gov. Retrieved on 20 November 2013.
215. "Roger Bridgman's defence of neural networks" (https://web.archive.org/web/20120319163352/http://members.fortunecity.com/templarseries/popper.html). Archived from the original (http://members.fortunecity.com/templarseries/popper.html) on 19 March 2012. Retrieved 12 July 2010.
216. "Scaling Learning Algorithms towards {AI} - LISA - Publications - Aigaion 2.0" (http://www.iro.umontreal.ca/~lisa/publications2/index.php/publications/show/4). *www.iro.umontreal.ca*.
217. D. J. Felleman and D. C. Van Essen, "Distributed hierarchical processing in the primate cerebral cortex (https://archive.today/20150120022056/http://cercor.oxfordjournals.org/content/1/1/1.1.full.pdf+html)," *Cerebral Cortex*, 1, pp. 1–47, 1991.
218. J. Weng, "Natural and Artificial Intelligence: Introduction to Computational Brain-Mind (https://www.amazon.com/Natural-Artificial-Intelligence-Introduction-Computational/dp/0985875720) Archived (https://web.archive.org/web/20240519082645/https://www.amazon.com/Natural-Artificial-Intelligence-Introduction-Computational/dp/0985875720) 19 May 2024 at the Wayback Machine," BMI Press, ISBN 978-0-9858757-2-5, 2012.
219. Edwards C (25 June 2015). "Growing pains for deep learning". *Communications of the ACM*. **58** (7): 14–16. doi:10.1145/2771283 (https://doi.org/10.1145%2F2771283).

S2CID 11026540 (https://api.semanticscholar.org/CorpusID:11026540).

220. "The Bitter Lesson" (http://www.incompleteideas.net/IncIdeas/BitterLesson.html). *www.incompleteideas.net*. Retrieved 7 August 2024.
221. Cade Metz (18 May 2016). "Google Built Its Very Own Chips to Power Its AI Bots" (https://www.wired.com/2016/05/google-tpu-custom-chips/). *Wired*. Archived (https://web.archive.org/web/20180113150305/https://www.wired.com/2016/05/google-tpu-custom-chips/) from the original on 13 January 2018. Retrieved 5 March 2017.
222. "Scaling Learning Algorithms towards AI" (http://yann.lecun.com/exdb/publis/pdf/bengio-lecun-07.pdf) (PDF). Archived (https://web.archive.org/web/20220812081157/http://yann.lecun.com/exdb/publis/pdf/bengio-lecun-07.pdf) (PDF) from the original on 12 August 2022. Retrieved 6 July 2022.
223. Tahmasebi, Hezarkhani (2012). "A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4268588). *Computers & Geosciences*. **42**: 18–27. Bibcode:2012CG.....42...18T (https://ui.adsabs.harvard.edu/abs/2012CG.....42...18T). doi:10.1016/j.cageo.2012.02.004 (https://doi.org/10.1016%2Fj.cageo.2012.02.004). PMC 4268588 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4268588). PMID 25540468 (https://pubmed.ncbi.nlm.nih.gov/25540468).
224. Sun and Bookman, 1990
225. Norori N, Hu Q, Aellen FM, Faraci FD, Tzovara A (October 2021). "Addressing bias in big data and AI for health care: A call for open science" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8515002). *Patterns*. **2** (10): 100347. doi:10.1016/j.patter.2021.100347 (https://doi.org/10.1016%2Fj.patter.2021.100347). PMC 8515002 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8515002). PMID 34693373 (https://pubmed.ncbi.nlm.nih.gov/34693373).
226. Carina W (27 October 2022). "Failing at Face Value: The Effect of Biased Facial Recognition Technology on Racial Discrimination in Criminal Justice" (https://doi.org/10.26689%2Fssr.v4i10.4402). *Scientific and Social Research*. **4** (10): 29–40. doi:10.26689/ssr.v4i10.4402 (https://doi.org/10.26689%2Fssr.v4i10.4402). ISSN 2661-4332 (https://search.worldcat.org/issn/2661-4332).
227. Chang X (13 September 2023). "Gender Bias in Hiring: An Analysis of the Impact of Amazon's Recruiting Algorithm" (https://aemps.ewapublishing.org/article.html?pk=e5b93601b03d453c855d54d3153875ba). *Advances in Economics, Management and Political Sciences*. **23** (1): 134–140. doi:10.54254/2754-1169/23/20230367 (https://doi.org/10.54254%2F2754-1169%2F23%2F20230367). ISSN 2754-1169 (https://search.worldcat.org/issn/2754-1169). Archived (https://web.archive.org/web/20231209135207/https://aemps.ewapublishing.org/article.html?pk=e5b93601b03d453c855d54d3153875ba) from the original on 9 December 2023. Retrieved 9 December 2023.
228. Kortylewski A, Egger B, Schneider A, Gerig T, Morel-Forster A, Vetter T (June 2019). "Analyzing and Reducing the Damage of Dataset Bias to Face Recognition with Synthetic Data". *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (https://edoc.unibas.ch/75257/1/20200128164027_5e3055eb775f1.pdf) (PDF). IEEE. pp. 2261–2268. doi:10.1109/cvprw.2019.00279 (https://doi.org/10.1109%2Fcvprw.2019.00279). ISBN 978-1-7281-2506-0. S2CID 198183828 (https://api.semanticscholar.org/CorpusID:198183828). Archived (https://web.archive.org/web/20240519082642/https://edoc.unibas.ch/75257/1/20200128164027_5e3055eb775f1.pdf) (PDF) from the original on 19 May 2024. Retrieved 30 December 2023.
229. Huang Y (2009). "Advances in Artificial Neural Networks – Methodological Development and Application" (https://doi.org/10.3390%2Falgor2030973). *Algorithms*. **2** (3): 973–1007. doi:10.3390/algor2030973 (https://doi.org/10.3390%2Falgor2030973). ISSN 1999-4893 (https://search.worldcat.org/issn/1999-4893).
230. Kariri E, Louati H, Louati A, Masmoudi F (2023). "Exploring the Advancements and Future Research Directions of Artificial Neural Networks: A Text Mining Approach" (https://doi.org/10.3390%2Fapp13053186). *Applied Sciences*. **13** (5): 3186. doi:10.3390/app13053186 (http

s://doi.org/10.3390%2Fapp13053186). ISSN 2076-3417 (https://search.worldcat.org/issn/2076-3417).
231. Fui-Hoon Nah F, Zheng R, Cai J, Siau K, Chen L (3 July 2023). "Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration" (https://doi.org/10.1080%2F15228053.2023.2233814). *Journal of Information Technology Case and Application Research*. **25** (3): 277–304. doi:10.1080/15228053.2023.2233814 (https://doi.org/10.1080%2F15228053.2023.2233814). ISSN 1522-8053 (https://search.worldcat.org/issn/1522-8053).
232. "DALL-E 2's Failures Are the Most Interesting Thing About It - IEEE Spectrum" (https://spectrum.ieee.org/openai-dall-e-2). *IEEE*. Archived (https://web.archive.org/web/20220715204154/https://spectrum.ieee.org/openai-dall-e-2) from the original on 15 July 2022. Retrieved 9 December 2023.
233. Briot JP (January 2021). "From artificial neural networks to deep learning for music generation: history, concepts and trends" (https://doi.org/10.1007%2Fs00521-020-05399-0). *Neural Computing and Applications*. **33** (1): 39–65. doi:10.1007/s00521-020-05399-0 (https://doi.org/10.1007%2Fs00521-020-05399-0). ISSN 0941-0643 (https://search.worldcat.org/issn/0941-0643).
234. Chow PS (6 July 2020). "Ghost in the (Hollywood) machine: Emergent applications of artificial intelligence in the film industry". *NECSUS_European Journal of Media Studies*. doi:10.25969/MEDIAREP/14307 (https://doi.org/10.25969%2FMEDIAREP%2F14307). ISSN 2213-0217 (https://search.worldcat.org/issn/2213-0217).
235. Yu X, He S, Gao Y, Yang J, Sha L, Zhang Y, et al. (June 2010). "Dynamic difficulty adjustment of game AI for video game Dead-End". *The 3rd International Conference on Information Sciences and Interaction Sciences*. IEEE. pp. 583–587. doi:10.1109/icicis.2010.5534761 (https://doi.org/10.1109%2Ficicis.2010.5534761). ISBN 978-1-4244-7384-7. S2CID 17555595 (https://api.semanticscholar.org/CorpusID:17555595).

Bibliography

- Bhadeshia H. K. D. H. (1999). "Neural Networks in Materials Science" (http://www.msm.cam.ac.uk/phase-trans/abstracts/neural.review.pdf) (PDF). *ISIJ International*. **39** (10): 966–979. doi:10.2355/isijinternational.39.966 (https://doi.org/10.2355%2Fisijinternational.39.966).
- Bishop CM (1995). *Neural networks for pattern recognition*. Clarendon Press. ISBN 978-0-19-853849-3. OCLC 33101074 (https://search.worldcat.org/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 (https://search.worldcat.org/oclc/76538146).
- Cybenko G (2006). "Approximation by Superpositions of a Sigmoidal function" (https://books.google.com/books?id=4RtVAAAAMAAJ&pg=PA303). In van Schuppen JH (ed.). *Mathematics of Control, Signals, and Systems*. Springer International. pp. 303–314. PDF (https://web.archive.org/web/20110719183058/http://actcomm.dartmouth.edu/gvc/papers/approx_by_superposition.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 (https://search.worldcat.org/oclc/35558945).
- Duda RO, Hart PE, Stork DG (2001). *Pattern classification* (2 ed.). Wiley. ISBN 978-0-471-05669-0. OCLC 41347061 (https://search.worldcat.org/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 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.5444). doi:10.1016/S0031-

https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.3444). doi:10.1016/S0031-3203(01)00178-9 (https://doi.org/10.1016%2FS0031-3203%2801%2900178-9).
- Fahlman S, Lebiere C (1991). "The Cascade-Correlation Learning Architecture" (https://web.archive.org/web/20130503184045/http://www.cs.iastate.edu/~honavar/fahlman.pdf) (PDF). Archived from the original (http://www.cs.iastate.edu/~honavar/fahlman.pdf) (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.
- Gurney K (1997). *An introduction to neural networks*. UCL Press. ISBN 978-1-85728-673-1. OCLC 37875698 (https://search.worldcat.org/oclc/37875698).
- Haykin SS (1999). *Neural networks: a comprehensive foundation*. Prentice Hall. ISBN 978-0-13-273350-2. OCLC 38908586 (https://search.worldcat.org/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 (https://search.worldcat.org/oclc/21522159).
- *Information theory, inference, and learning algorithms*. Cambridge University Press. 25 September 2003. Bibcode:2003itil.book.....M (https://ui.adsabs.harvard.edu/abs/2003itil.book.....M). ISBN 978-0-521-64298-9. OCLC 52377690 (https://search.worldcat.org/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 (https://search.worldcat.org/oclc/837524179).
- Lawrence J (1994). *Introduction to neural networks: design, theory and applications*. California Scientific Software. ISBN 978-1-883157-00-5. OCLC 32179420 (https://search.worldcat.org/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 (https://search.worldcat.org/oclc/29877717).
- Maurer H (2021). *Cognitive science: integrative synchronization mechanisms in cognitive neuroarchitectures of the modern connectionism*. CRC Press. doi:10.1201/9781351043526 (https://doi.org/10.1201%2F9781351043526). ISBN 978-1-351-04352-6. S2CID 242963768 (https://api.semanticscholar.org/CorpusID:242963768).
- Ripley BD (2007). *Pattern Recognition and Neural Networks* (https://books.google.com/books?id=m12UR8QmLqoC). Cambridge University Press. ISBN 978-0-521-71770-0.
- Siegelmann H, Sontag ED (1994). "Analog computation via neural networks" (https://doi.org/10.1016%2F0304-3975%2894%2990178-3). *Theoretical Computer Science*. **131** (2): 331–360. doi:10.1016/0304-3975(94)90178-3 (https://doi.org/10.1016%2F0304-3975%2894%2990178-3). S2CID 2456483 (https://api.semanticscholar.org/CorpusID:2456483).
- Smith M (1993). *Neural networks for statistical modeling*. Van Nostrand Reinhold. ISBN 978-0-442-01310-3. OCLC 27145760 (https://search.worldcat.org/oclc/27145760).
- Wasserman PD (1993). *Advanced methods in neural computing*. Van Nostrand Reinhold. ISBN 978-0-442-00461-3. OCLC 27429729 (https://search.worldcat.org/oclc/27429729).
- Wilson H (2018). *Artificial intelligence*. Grey House Publishing. ISBN 978-1-68217-867-6.
