NEURAL NETWORK

What is a neural network?

A neural network is a machine learning model that stacks simple "neurons" in layers and learns pattern-recognizing weights and biases from data to map inputs to outputs.

Neural networks are among the most influential algorithms in modern machine learning and artificial intelligence (AI). They underpin breakthroughs in computer vision, natural language processing (NLP), speech recognition and countless real-world applications ranging from forecasting to facial recognition. While today's deep neural networks (DNNs) power systems as complex as transformers and convolutional neural networks (CNNs), the origins of neural networks trace back to simple models such as linear regression and how the human brain digests, processes and decides on the information presented to it.

How do neural networks work?

On a high level, the inspiration for neural networks comes from the biological neurons in the human brain, which communicate through electrical signals. In 1943, Warren McCulloch and Walter Pitts proposed the first mathematical model of a neuron, showing that simple units could perform computation of a function. Later, in 1958, Frank Rosenblatt introduced the perceptron, an algorithm designed to perform pattern recognition. The perceptron is the historical ancestor of today's networks: essentially a linear model with a constrained output. In the following section, we will dive into how neural networks borrow inspiration from the human brains to make decisions and recognize patterns.

A neural network can be understood through a simple example: spam detection. An email is fed into the network, and features such as words or phrases like "prize," "money," "dear" or "win" are used as inputs. The early neurons in the network process the importance of each signal, while later layers combine this information into higher-level cues that capture context and tone. The final layer then computes a probability of whether the email is spam, and if that probability is high enough, the email is flagged. In essence, the network learns how to transform raw features into meaningful patterns and use them to make predictions.This process is powered by two fundamental concepts: weights and biases. Weights act like dials that control how strongly each input feature influences the decision—a word like "prize" may be given more weight than a common word like "hello."

Neural network training

Just like other machine learning algorithms, a neural net requires rigorous training to perform well on testing. To train a network, a single neuron computes:

$z=\sum i=1nwixi+b$

$a=\sigma(z)$

Where:

- *xi = input feature,*
- *wi = weight,*
- *b = bias,*
- *z = weighted sum (linear transformation),*
- *σ = activation function (nonlinear transformation),*
- *a = output,*

σ represents an activation function at the output layer that transforms the linear combination to fit the decision of the function. Using this architecture, the input features X are transformed into an output Y, serving as a predictive machine learning model.

The power of a neural network comes from its ability to learn the right weights and biases from data. This is done by comparing the network's prediction Y^ to the true label Y and measuring the error using a loss function. For example, in classification tasks, the loss might measure how far the predicted probability is from the correct answer.

To minimize this loss, the network uses an algorithm called backpropagation. The neural net trains in four steps:

- *Forward pass: Inputs flow through the network, computing linear combinations, passing through the nonlinear activation function and producing an output prediction.*

- *Error calculation: The loss function measures the difference between prediction and truth.*

- *Backward pass (backpropagation): The error is propagated backward through the network. At each neuron, the algorithm calculates how much each weight and bias contributed to the error using the chain rule of calculus.*
-
- *Weight update: The weights and biases are adjusted slightly in the direction that reduces the error, using an optimization method like gradient descent.*

Types of neural networks

While multilayer perceptrons are the foundation, neural networks have evolved into specialized architectures suited for different domains:

- *Convolutional neural networks (CNNs or convnets): Designed for grid-like data such as images. CNNs excel at image recognition, computer vision and facial recognition thanks to convolutional filters that detect spatial hierarchies of features.*
- *Recurrent neural networks (RNNs): Incorporate feedback loops that allow information to persist across time steps. RNNs are well-suited for speech recognition, time series forecasting and sequential data.*
- *Transformers: A modern architecture that replaced RNNs for many sequence tasks. Transformers leverage attention mechanisms to capture dependencies in natural language processing (NLP) and power state-of-the-art models like GPT.*
- *These variations highlight the versatility of neural networks. Regardless of architecture, all rely on the same principles: artificial neurons, nonlinear activations and optimization algorithms.*

Types of neural networks

While multilayer perceptrons are the foundation, neural networks have evolved into specialized architectures suited for different domains:

- *Convolutional neural networks (CNNs or convnets): Designed for grid-like data such as images. CNNs excel at image recognition, computer vision and facial recognition thanks to convolutional filters that detect spatial hierarchies of features.*
- *Recurrent neural networks (RNNs): Incorporate feedback loops that allow information to persist across time steps. RNNs are well-suited for speech recognition, time series forecasting and sequential data.*
- *Transformers: A modern architecture that replaced RNNs for many sequence tasks. Transformers leverage attention mechanisms to capture dependencies in natural language processing (NLP) and power state-of-the-art models like GPT.*
- *These variations highlight the versatility of neural networks. Regardless of architecture, all rely on the same principles: artificial neurons, nonlinear activations and optimization algorithms.*

Why neural networks matter

Neural networks learn useful internal representations directly from data, capturing nonlinear structure that classical models miss. With sufficient capacity, sound objectives and regularization against overfitting, they scale from small benchmarks to production systems in computer vision, natural language processing, speech recognition, forecasting and more—delivering measurable gains in accuracy and robustness.

Modern deep learning extends these foundations. CNNs specialize in spatial feature extraction for images; RNNs model temporal dependencies in sequences; transformers replace recurrence with attention, aided by residual connections, normalization and efficient parallelism on GPUs.

Despite architectural differences, training remains end-to-end with backpropagation on large datasets, and the core view still holds: $Y=f(X;\sigma)$ *is learned by composing data-dependent transformations with nonlinear activations. Generative AI builds on the same principles at greater scale. Large language models, diffusion models, VAEs and GANs learn distributions over data to synthesize text, images, audio and code.*

The leap from a multilayer perceptron to state-of-the-art generators is primarily one of architecture, data and compute. Understanding activation functions, training requirements and the main types of networks provides a practical bridge from classical neural nets to today's generative systems and clarifies why these models have become central to modern AI.