

## What is a neural network?

Neural Networks are computational models that mimic the complex functions of the human brain. The neural networks consist of interconnected nodes or neurons that process and learn from data, enabling tasks such as pattern recognition and decision making in machine learning. The article explores more about neural networks, their working, architecture and more.

## Table of Content

- [Evolution of Neural Networks](#)
- [What are Neural Networks?](#)
- [How does Neural Networks work?](#)
- [Learning of a Neural Network](#)
- [Types of Neural Networks](#)
- [Simple Implementation of a Neural Network](#)

## Evolution of Neural Networks

Since the 1940s, there have been a number of noteworthy advancements in the field of neural networks:

- **1940s-1950s: Early Concepts**  
Neural networks began with the introduction of the first mathematical model of artificial neurons by McCulloch and Pitts. But computational constraints made progress difficult.
- **1960s-1970s: Perceptrons**  
This era is defined by the work of Rosenblatt on perceptrons. [Perceptrons](#) are single-layer networks whose applicability was limited to issues that could be solved linearly separately.
- **1980s: Backpropagation and Connectionism**  
[Multi-layer network](#) training was made possible by Rumelhart, Hinton, and Williams' invention of the backpropagation method. With its emphasis on learning through interconnected nodes, connectionism gained appeal.
- **1990s: Boom and Winter**  
With applications in image identification, finance, and other fields, neural networks saw a boom. Neural network research did, however, experience a "winter" due to exorbitant computational costs and inflated expectations.
- **2000s: Resurgence and Deep Learning**  
Larger datasets, innovative structures, and enhanced processing capability spurred a comeback. [Deep learning](#) has shown amazing effectiveness in a number of disciplines by utilizing numerous layers.

- **2010s-Present: Deep Learning Dominance**

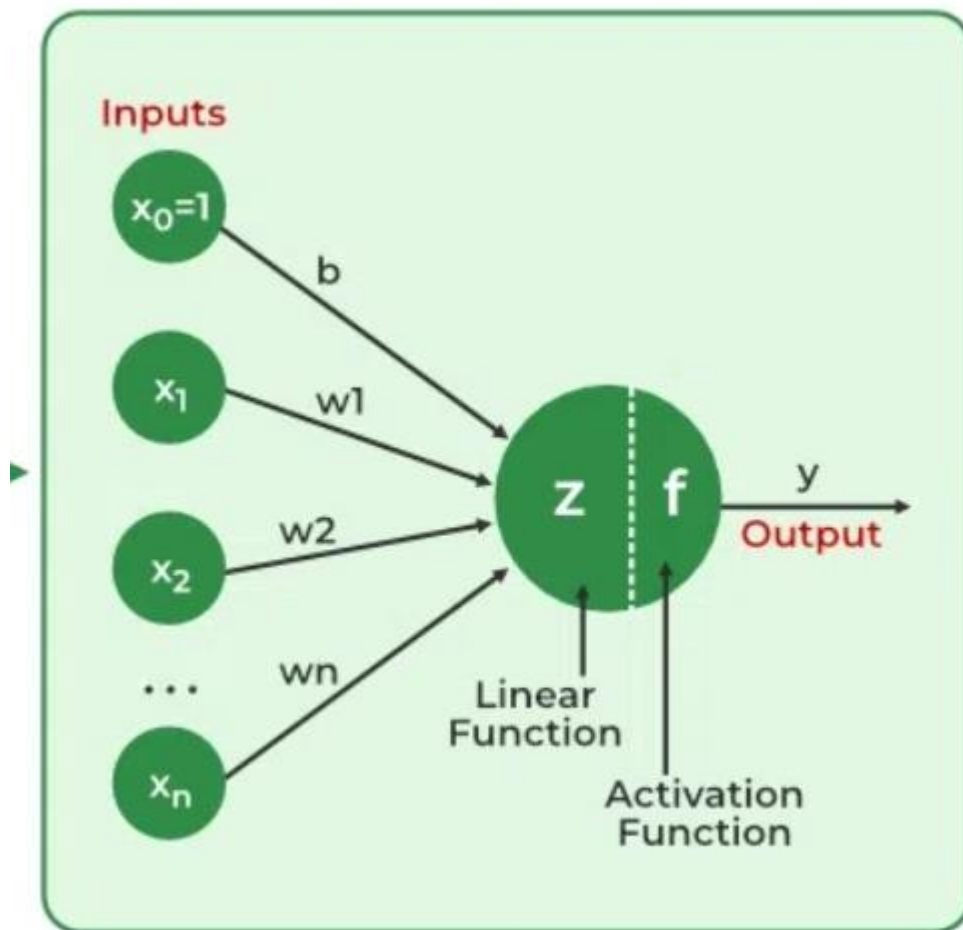
Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), two deep learning architectures, dominated machine learning. Their power was demonstrated by innovations in gaming, picture recognition, and natural language processing.

### **What are Neural Networks?**

[Neural networks](#) extract identifying features from data, lacking pre-programmed understanding. Network components include neurons, connections, weights, biases, propagation functions, and a learning rule. Neurons receive inputs, governed by thresholds and activation functions. Connections involve weights and biases regulating information transfer. Learning, adjusting weights and biases, occurs in three stages: input computation, output generation, and iterative refinement enhancing the network's proficiency in diverse tasks.

These include:

1. The neural network is simulated by a new environment.
2. Then the free parameters of the neural network are changed as a result of this simulation.
3. The neural network then responds in a new way to the environment because of the changes in its free parameters.



### Importance of Neural Networks

The ability of neural networks to identify patterns, solve intricate puzzles, and adjust to changing surroundings is essential. Their capacity to learn from data has far-reaching effects, ranging from revolutionizing technology like [natural language processing](#) and self-driving automobiles to automating decision-making processes and increasing efficiency in numerous industries. The development of artificial intelligence is largely dependent on neural networks, which also drive innovation and influence the direction of technology.

### How does Neural Networks work?

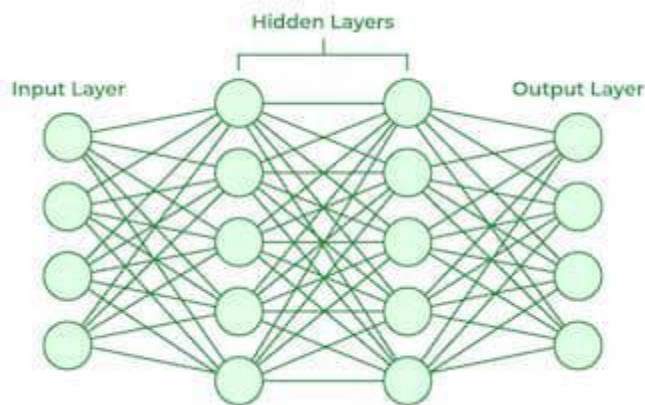
Let's understand with an example of how a neural network works:

Consider a neural network for email classification. The input layer takes features like email content, sender information, and subject. These inputs, multiplied by adjusted weights, pass through hidden layers. The network, through training, learns to recognize patterns indicating whether an email is spam or not. The output layer, with a binary activation function, predicts whether the email is spam (1) or not (0). As the network iteratively refines its weights through backpropagation, it becomes

adept at distinguishing between spam and legitimate emails, showcasing the practicality of neural networks in real-world applications like email filtering.

### Working of a Neural Network

Neural networks are complex systems that mimic some features of the functioning of the human brain. It is composed of an input layer, one or more hidden layers, and an output layer made up of layers of artificial neurons that are coupled. The two stages of the basic process are called backpropagation and [forward propagation](#).



### Forward Propagation

- **Input Layer:** Each feature in the input layer is represented by a node on the network, which receives input data.
- **Weights and Connections:** The weight of each neuronal connection indicates how strong the connection is. Throughout training, these weights are changed.
- **Hidden Layers:** Each hidden layer neuron processes inputs by multiplying them by weights, adding them up, and then passing them through an activation function. By doing this, non-linearity is introduced, enabling the network to recognize intricate patterns.
- **Output:** The final result is produced by repeating the process until the output layer is reached.

### Backpropagation

- **Loss Calculation:** The network's output is evaluated against the real goal values, and a loss function is used to compute the difference. For a regression problem, the [Mean Squared Error](#) (MSE) is commonly used as the cost function.

#### Loss Function:

- **Gradient Descent:** Gradient descent is then used by the network to reduce the loss. To lower the inaccuracy, weights are changed based on the derivative of the loss with respect to each weight.

- **Adjusting weights:** The weights are adjusted at each connection by applying this iterative process, or [backpropagation](#), backward across the network.
- **Training:** During training with different data samples, the entire process of forward propagation, loss calculation, and backpropagation is done iteratively, enabling the network to adapt and learn patterns from the data.
- **Activation Functions:** Model non-linearity is introduced by activation functions like the [rectified linear unit](#) (ReLU) or [sigmoid](#). Their decision on whether to “fire” a neuron is based on the whole weighted input.

## Learning of a Neural Network

### 1. Learning with supervised learning

In [supervised learning](#), the neural network is guided by a teacher who has access to both input-output pairs. The network creates outputs based on inputs without taking into account the surroundings. By comparing these outputs to the teacher-known desired outputs, an error signal is generated. In order to reduce errors, the network’s parameters are changed iteratively and stop when performance is at an acceptable level.

### 2. Learning with Unsupervised learning

Equivalent output variables are absent in [unsupervised learning](#). Its main goal is to comprehend incoming data’s (X) underlying structure. No instructor is present to offer advice. Modeling data patterns and relationships is the intended outcome instead. Words like regression and classification are related to supervised learning, whereas unsupervised learning is associated with clustering and association.

### 3. Learning with Reinforcement Learning

Through interaction with the environment and feedback in the form of rewards or penalties, the network gains knowledge. Finding a policy or strategy that optimizes cumulative rewards over time is the goal for the network. This kind is frequently utilized in gaming and decision-making applications.

## Types of Neural Networks

There are *seven* types of neural networks that can be used.

- **Feedforward Networks:** A [feedforward neural network](#) is a simple artificial neural network architecture in which data moves from input to output in a single direction. It has input, hidden, and output layers; feedback loops are absent. Its straightforward architecture makes it appropriate for a number of applications, such as regression and pattern recognition.
- **Multilayer Perceptron (MLP):** [MLP](#) is a type of feedforward neural network with three or more layers, including an input layer, one or more hidden layers, and an output layer. It uses nonlinear activation functions.

- **Convolutional Neural Network (CNN):** A [Convolutional Neural Network](#) (CNN) is a specialized artificial neural network designed for image processing. It employs convolutional layers to automatically learn hierarchical features from input images, enabling effective image recognition and classification. CNNs have revolutionized computer vision and are pivotal in tasks like object detection and image analysis.
- **Recurrent Neural Network (RNN):** An artificial neural network type intended for sequential data processing is called a [Recurrent Neural Network](#) (RNN). It is appropriate for applications where contextual dependencies are critical, such as time series prediction and natural language processing, since it makes use of feedback loops, which enable information to survive within the network.
- **Long Short-Term Memory (LSTM):** [LSTM](#) is a type of RNN that is designed to overcome the vanishing gradient problem in training RNNs. It uses memory cells and gates to selectively read, write, and erase information.

### Simple Implementation of a Neural Network

- Python3

```
import numpy as np

# array of any amount of numbers. n = m
X = np.array([[1, 2, 3],
              [3, 4, 1],
              [2, 5, 3]])

# multiplication
y = np.array([.5, .3, .2])

# transpose of y
y = y.T

# sigma value
sigm = 2

# find the delta
delt = np.random.random((3, 3)) - 1

for j in range(100):

    # find matrix 1. 100 layers.
    m1 = (y - (1/(1 + np.exp(-(np.dot((1/(1 + np.exp(
        -(np.dot(X, sigm))))), delt)))))) * ((1/(
        1 + np.exp(-(np.dot((1/(1 + np.exp(
        -(np.dot(X, sigm))))), delt)))) * (1-(1/(
        1 + np.exp(-(np.dot((1/(1 + np.exp(
```

```

        -(np.dot(X, sigm))))), delt))))))

# find matrix 2
m2 = m1.dot(delt.T) * ((1/(1 + np.exp(-(np.dot(X, sigm))))))
    * (1-(1/(1 + np.exp(-(np.dot(X, sigm))))))
# find delta
delt = delt + (1/(1 + np.exp(-(np.dot(X, sigm))))).T.dot(m1)

# find sigma
sigm = sigm + (X.T.dot(m2))

# print output from the matrix
print(1/(1 + np.exp(-(np.dot(X, sigm))))))

```

#### Output:

```

[[0.99999325 0.99999375 0.99999352]
 [0.99999988 0.99999989 0.99999988]
 [1.      1.      1.      ]]

```

### Advantages of Neural Networks

Neural networks are widely used in many different applications because of their many benefits:

- **Adaptability:** Neural networks are useful for activities where the link between inputs and outputs is complex or not well defined because they can adapt to new situations and learn from data.
- **Pattern Recognition:** Their proficiency in pattern recognition renders them efficacious in tasks like as audio and image identification, natural language processing, and other intricate data patterns.
- **Parallel Processing:** Because neural networks are capable of parallel processing by nature, they can process numerous jobs at once, which speeds up and improves the efficiency of computations.
- **Non-Linearity:** Neural networks are able to model and comprehend complicated relationships in data by virtue of the non-linear activation functions found in neurons, which overcome the drawbacks of linear models.

### Disadvantages of Neural Networks

Neural networks, while powerful, are not without drawbacks and difficulties:

- **Computational Intensity:** Large neural network training can be a laborious and computationally demanding process that demands a lot of computing power.
- **Black box Nature:** As “black box” models, neural networks pose a problem in important applications since it is difficult to understand how they make decisions.

- **Overfitting:** Overfitting is a phenomenon in which neural networks commit training material to memory rather than identifying patterns in the data. Although regularization approaches help to alleviate this, the problem still exists.
- **Need for Large datasets:** For efficient training, neural networks frequently need sizable, labeled datasets; otherwise, their performance may suffer from incomplete or skewed data.