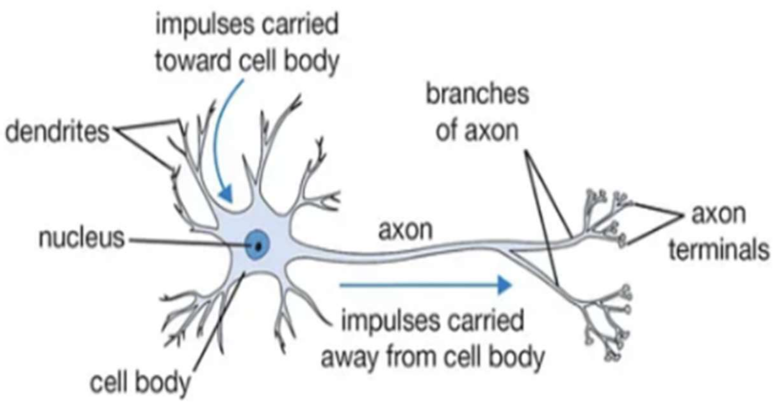
NEURAL NETWORKS

Neural networks are a key technology in the field of artificial intelligence (AI), specifically falling under the umbrella of deep learning. A neural network, also known as an artificial neural network (ANN) is a powerful machine learning model inspired by the structure and function of the human brain. These networks consist of interconnected nodes, or neurons, arranged in layers that collaborate to solve complex problems.

Biological Neuron vs Artificial Neuron

Biological neurons, the building blocks of the nervous system, exhibit complex structures comprising dendrites, cell body (soma), axon, and synaptic terminals. Each component plays a crucial role in the transmission and processing of signals.

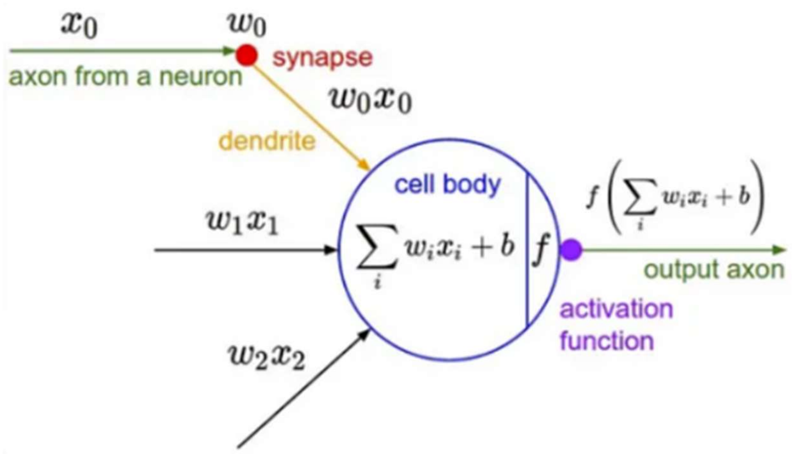


Biological Neuron

The biological neuron's intricate network of connections via dendrites facilitates the transmission of electrical or electro-chemical signals to other neurons. These signals are processed by the neuron and subsequently relayed to additional neurons through axons. The varying thickness of dendrites signifies the relative importance of different inputs.

Artificial neurons are simplified abstractions of biological neurons, are foundational units within artificial neural networks (ANNs), facilitating the development of computational models with learning and decision-making capabilities akin to biological systems.

Each artificial neuron receives input signals that are multiplied by randomly assigned weights and then combined with a fixed bias value unique to its layer. The connections between neurons are represented by weights that determines the strength of the connection. This aggregated input is processed within the neuron, with the bias ensuring a non-zero output or enhancing the system's response if the weighted sum is zero. Remarkably, the bias consistently maintains a weight and input value of '1'.



Artificial Neuron

The resulting sum can range from 0 to infinity, necessitating the establishment of a threshold to achieve the desired response. Activation functions, such as Sigmoid, ReLU, Tanh, and Linear, are applied to this sum to determine the neuron's final output, tailored to the characteristics of the input values.

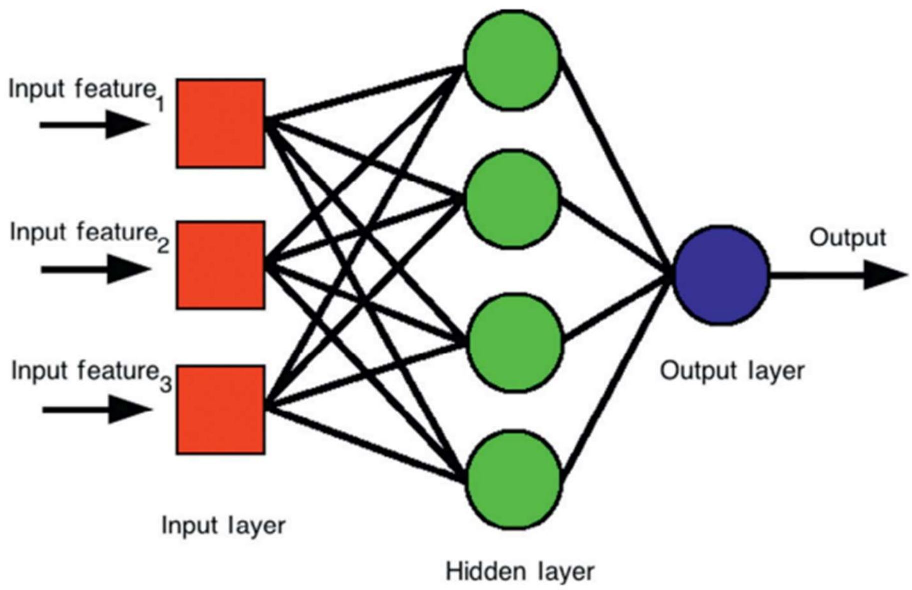
Subsequently, after the output is derived from the final neural network layer, the loss function (comparing input to output) is computed. Through backpropagation, weights are iteratively adjusted to minimize the loss, with the overarching objective being the attainment of optimal weight values.

Architecture of Neural Network

The structure of a neural network comprises layers of interconnected nodes, commonly referred to as neurons or units. These neurons are organized into three main types of layers: an input layer, one or more hidden layers, and an output layer. Let us understand each key element of the neural network in detail:

1. **Input layer:** The input layer is responsible for receiving the initial data or features that are fed into the neural network. Each neuron in the input layer depicts a specific feature or attribute of the input data.

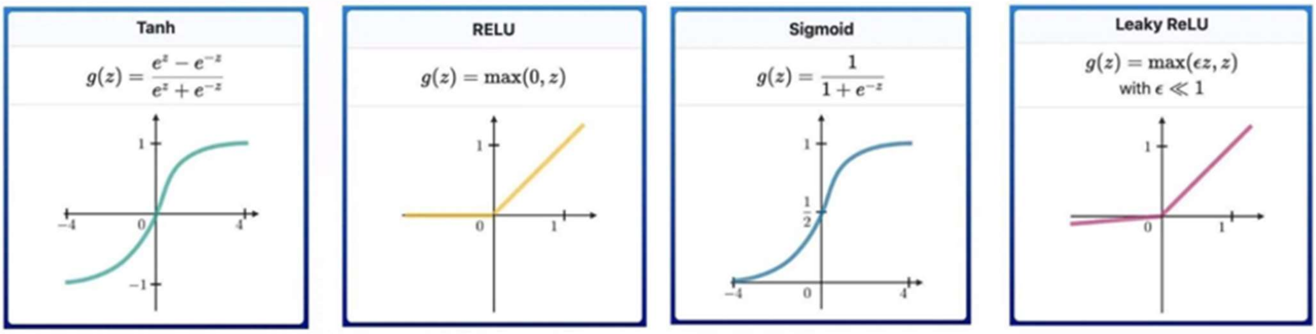
2. **Hidden layers:** Hidden layers are intermediate layers between the input and output layers. They perform complex computations and transformations on the input data. A neural network can have numerous hidden layers, each consisting of numerous neurons or nodes.



3. **Neurons (Nodes):** Neurons or artificial neurons are fundamental units of a neural network. They receive input signals and perform computations to produce an output. Neurons in the hidden and output layers utilize activation functions to introduce non-linearities into the network, allowing it to learn complex patterns.

4. **Weights and biases:** Weights and biases are adjustable parameters associated with the connections between neurons. Each connection has a weight, which determines the strength or importance of the input signal. Biases, on the other hand, provide an additional tuneable parameter that allows neurons to adjust their activation threshold.

5. **Activation functions:** Activation functions are threshold values that introduce non-linearities into the neural network, enabling it to comprehend complex relationships between inputs and outputs. Common activation functions include sigmoid, tanh, ReLU (Rectified Linear Unit), and SoftMax.



6. **Output layer:** The output layer generates the final predictions or outputs of the neural network. The number of artificial neurons in the output layer depends on the specific problem being solved. For example, in a binary classification task, there may be one neuron representing each class (e.g., "yes" or "no"), while in multi-class classification, there will be multiple neurons representing each class.

7. **Loss function:** The loss function measures the discrepancy between the predicted outputs of the neural network and the true values. It quantifies the network's performance and guides the learning process by providing feedback on its performance.

8. **Backpropagation:** Backpropagation is a learning algorithm used to train a neural network. It involves propagating the error (difference between predicted and actual outputs) backward through the network and adjusting the weights and biases iteratively to minimize the loss function.

9. **Optimization algorithm:** Optimization algorithms, such as **gradient descent,** are employed to update the weights and biases during training. These algorithms determine the direction and magnitude of weight adjustments based on the gradients of the loss function concerning the network parameters.

How do neural networks work?

Let us understand the working function of neural networks through an example. For that, we will consider a neural network that can distinguish between a circle, triangle and square.

1. Data preparation

First, we need a labelled dataset containing images of circles, squares, and triangles. Each image should be labelled with its corresponding shape. It's important to have a diverse and representative dataset to train the neural network effectively.

2. Network architecture

We design our neural network's architecture based on the task's complexity. We can use a neural network with multiple hidden layers in this case. The input layer will take in the image data, and the output layer will have three neurons, each representing one shape (circle, square, or triangle).

3. Input representation

Next, we need to convert the shape images into a suitable format to create the input for the neural network. This can be achieved by representing each image as a **flattened array of pixel values** or by utilizing more advanced techniques, such as Convolutional Neural Networks (CNNs) that can directly process image data. In this case, the choice would be pixelated representation of the image.

4. Activation function and forward propagation

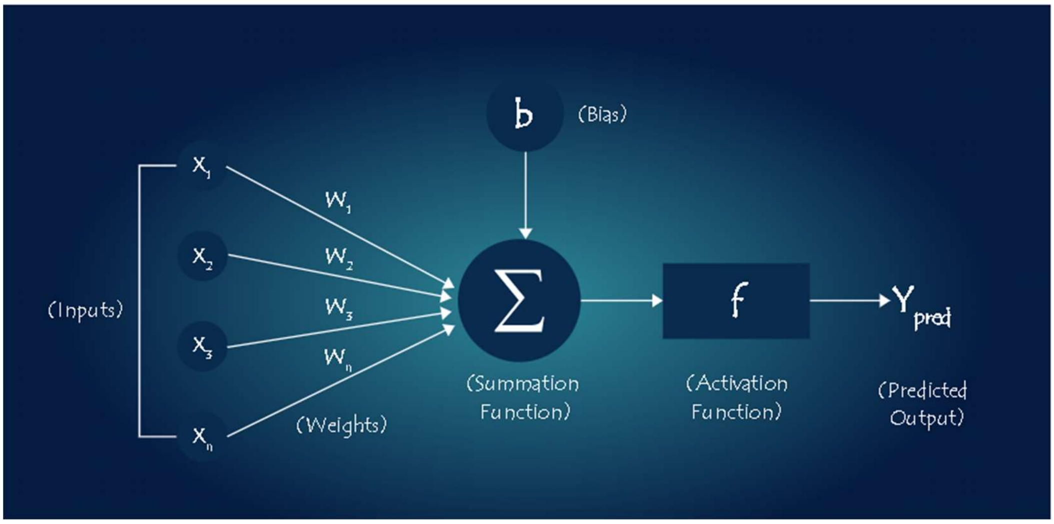
Each pixel is fed as input to each neuron (represented as x_1, x_2, which goes until x_n) of the first layer/input layer. Neurons of one layer are connected to neurons of the next layer through channels, and each channel is assigned a numerical value known as **weight.**

The input values (pixels) are multiplied by the corresponding weights, and their sum is sent as input to the neurons in the hidden layers. Each neuron in the hidden layers also has a numerical value called the **bias** (e.g., B_1, B_2, etc.).

The bias is added to the input sum, and the resulting value is passed through a threshold function called the **activation function**. The activation function determines whether the particular neuron will be activated or not.

The activated neurons transmit data to the neurons in the next layer through the channels, propagating the data through the network. This process is known as **forward propagation.**

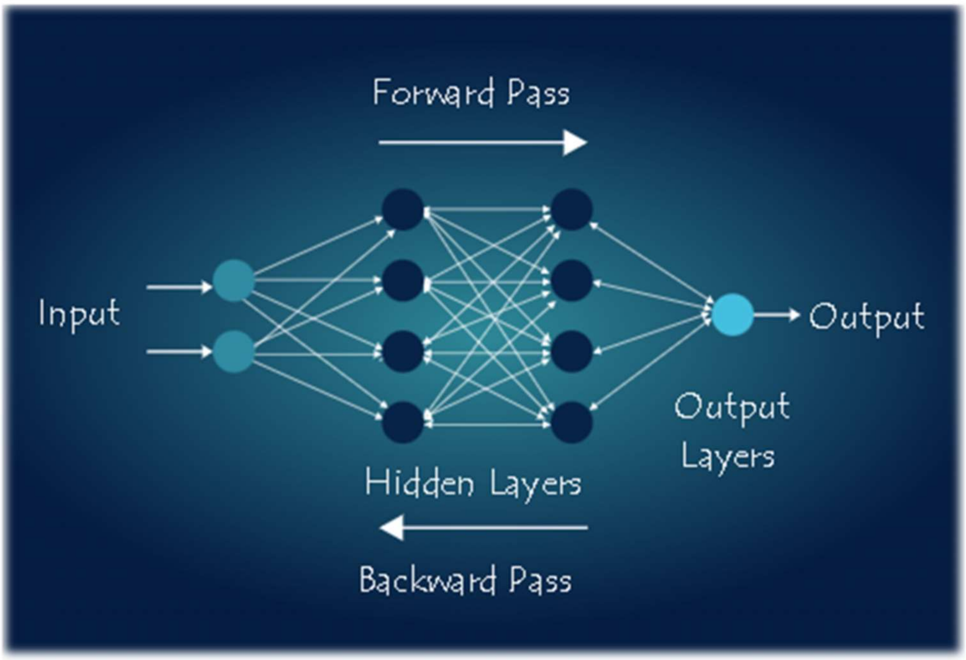
The neuron with the highest value is activated in the output layer and determines the final output. The values in the output layer represent probabilities. To obtain these probabilities, we commonly use the SoftMax activation function. SoftMax takes a vector of real numbers as input and transforms them into a probability distribution over multiple classes, ensuring that they sum up to 1. This allows us to interpret the neural network output as the predicted probabilities for each class of shapes, such as circles, squares, or triangles.



5. Loss function

In the output layer, if the neuron associated with the square receives the highest probability, it will be considered the predicted output. However, if the input image were actually a circle, the neural network's prediction would be incorrect. The loss function helps the neural network to calculate how well the model has performed on the task at hand.

A **loss function**, alternatively known as an objective function or a cost function, is a mathematical function that measures the discrepancy between the predicted output of a machine learning model and the true or desired output. It quantifies how much the model's predictions deviate from the actual values. We need a suitable loss function to train the neural network that quantifies the difference between the predicted probabilities and the true labels.



6. Backpropagation and optimization

Backpropagation, short for "backward propagation of errors," is a fundamental algorithm used in training neural networks. It enables the network to learn from its mistakes and adjust its internal parameters (weights and biases) to improve its predictions.

In backward propagation, the network compares the predicted output with the true output and calculates the gradient of the loss function concerning the network's parameters. This gradient indicates the direction and magnitude of the changes needed to minimize the loss.

Optimization algorithms like Stochastic Gradient Descent (SGD) or more advanced techniques such as Adam are commonly used for this purpose.

7. Training

We repeat the process of forward propagation, loss calculation, backpropagation, and weight updates for multiple epochs. During training, the neural network learns to recognize patterns and features that differentiate circles, squares, and triangles.

8. Evaluation

After training, we evaluate the neural network's performance on a separate test set. We feed the test images through the network, and the predicted class is compared with the true labels to measure accuracy or other relevant metrics.

9. Prediction

Once the neural network is trained and evaluated, we can utilize it to make predictions on new, unseen shape images. The network takes in the image as input, performs forward propagation, and outputs the predicted probabilities for each shape class. The class with the most probability can be considered the predicted shape for the input image.

By training on a diverse dataset and optimizing the network's architecture and training parameters, the neural network can learn to differentiate between circles, squares, and triangles based on the patterns and features it discovers during training.

<u>How to assess the performance of a neural network?</u>

The performance of neural networks is assessed using several model evaluation metrics. Here are some commonly used metrics:

1. **Accuracy:** Accuracy measures the proportion of correctly classified instances from the total number of instances in the dataset. It is a simple and intuitive metric, but it may not be suitable for imbalanced datasets where the classes are not equally represented.
2. **Precision:** Precision measures the proportion of correctly predicted positive instances out of all positive ones. It focuses on the accuracy of positive predictions and helps evaluate the model's ability to avoid false positives.
3. **Recall:** Recall, also referred to as sensitivity or the true positive rate, quantifies the proportion of accurately predicted positive instances out of the total number of actual positive instances. It focuses on the model's ability to identify all positive instances and avoid false negatives.
4. **F1 Score:** The F1 score is the harmonic mean of precision and recall. It provides a balanced evaluation metric that considers both precision and recall. It is specifically beneficial when there is an imbalance between positive and negative instances.
5. **ROC Curve:** The Receiver Operating Characteristic (ROC) curve offers a visual representation of the connection between the true positive rate (TPR) and the false positive rate (FPR) across various classification thresholds. By illustrating the trade-off between sensitivity (the ability to identify positives correctly) and specificity (the ability to identify negatives correctly), the ROC curve aids in determining an optimal threshold for classification.
6. **Area Under the ROC Curve (AUC-ROC):** The AUC-ROC is another metric that quantifies the overall performance of a binary classification model. It represents the probability that a randomly chosen positive instance will be ranked higher than a randomly chosen negative instance. A higher AUC-ROC value indicates better model performance.
7. **Mean Squared Error (MSE):** MSE is a commonly used loss function for regression tasks. It measures the average squared difference between the predicted and actual values. Lower MSE values indicate better model performance.

8. **Mean Absolute Error (MAE):** MAE is another loss function for regression tasks. It measures the average absolute difference between the predicted and actual values. MAE provides a more interpretable metric than MSE, as squared differences do not influence it.
9. **R-squared (R2) Score:** The R-squared score quantifies the fraction of the target variable's variance that the input variables can explain. It operates as an indicator of how well the model fits the data. A value of 1 signifies a perfect fit, whereas a value of 0 suggests that the model fails to capture any information from the input.

When evaluating a neural network model, it is vital to consider the specific requirements of the task at hand and choose the appropriate evaluation metrics accordingly. The selection of metrics should align with the objectives and priorities of the problem domain.

Types of Neural Network Architecture

Neural networks are often characterized by their depth, which refers to the number of layers between input and output, commonly known as hidden layers. This association with depth is why the term "neural network" is nearly interchangeable with "deep learning." Additionally, neural networks can be defined by the number of hidden nodes or the configuration of input and output layers per node. Variations in neural network design allow for different methods of information propagation between layers, both forward and backward. These variations facilitate a wide range of applications and tasks, contributing to the flexibility and adaptability of neural network architectures.

- **Feed-forward neural networks**

A feedforward neural network is composed of multiple layers of artificial neurons. It comprises an input layer, one or more hidden layers, and an output layer. An input layer transmits information to a hidden layer, which in turn transmits information to an output layer, facilitating a one-way flow of information. Each neuron in the network receives inputs, applies weights and biases, passes the weighted sum through an activation function, and forwards the result to the next layer. FNNs are primarily used for classification, regression, and pattern recognition tasks.

- **Recurrent neural networks (RNNs)**

RNNs are designed to process sequential data, where the order and context of data points matter. They introduce feedback connections, allowing information to flow in cycles or loops within the network. RNNs have a memory component that enables them to retain and utilize information from previous steps in the sequence.

This neural network starts with the same front propagation as a feed-forward network but then goes on to remember all processed information to reuse it in the future. If the network's prediction is incorrect, then the system self-learns and continues working toward the correct prediction during backpropagation.

They are widely used for tasks such as **natural language processing, speech recognition and time series analysis.**

- **Long Short-term Memory networks (LSTMs)**

As a type of RNN, LSTM introduces memory cells to address the vanishing gradient problem. These memory cells allow for better retention and utilization of information over long sequences. They incorporate three gates (input, forget, and output gates) to restrain the flow of information and selectively update or forget information from the memory cell.

LSTM networks are particularly effective in tasks involving long-term dependencies, such as language modeling and machine translation.

- **Convolutional neural networks (CNNs)**

CNNs are primarily used for analyzing visual data, such as images or video, but they can also be applied to other grid-like data. They employ specialized layers, such as convolutional layers and pooling layers, to efficiently process spatially structured data. A convolutional layer applies a set of learnable filters to the input, performing convolutions to detect local patterns and spatial relationships. The pooling layers, on the other hand, reduce the dimensionality of the feature maps.

CNNs are known for their ability to automatically extract meaningful features from images, making them suitable for tasks like **object recognition, image classification, and computer vision**. Examples of CNNs include:

- **AlexNet** - Contains multiple convolutional layers designed for image recognition.
- **Visual geometry group (VGG)** - VGG is similar to AlexNet, but has more layers of narrow convolutions.

➢ Generative adversarial networks (GANs)

Generative adversarial networks (GAN) are a type of unsupervised learning where data is generated from patterns that were discovered from the input data. GANs have two main parts that compete against one another:

- **Generator** - creates synthetic data from the learning phase of the model. It will take random datasets and generate a transformed image.
- **Discriminator** - decides whether or not the images produced are fake or genuine.

GANs are used to help predict what the next frame in a video might be, text to image generation, or image to image translation.

➢ Modular neural networks

A modular neural network, also known as modular neural architecture, is a type of neural network structure that is composed of distinct and relatively independent modules. Each module is responsible for handling a specific subtask or aspect of the overall problem. The idea behind modular neural networks is to break down a complicated problem into simpler sub-problems and have specialized modules tackle each.

In a modular neural network, these modules often work in parallel or in a hierarchical manner, where the outputs of one module feed into another. This allows for greater modularity, flexibility, and easier debugging.

Practical applications of neural network

Neural networks have practical applications across various domains thanks to their ability to learn from data and make intelligent predictions. Here are some practical applications of neural networks:

1. Computer vision:

- **Image classification:** Neural networks can classify images into different categories, enabling applications like object recognition, autonomous vehicles, and medical image analysis.
- **Object detection:** It can identify and locate multiple objects within an image, enabling tasks like video surveillance, self-driving cars, and augmented reality.
- **Image generation:** Neural networks can generate realistic images, leading to applications like image synthesis, artistic style transfer, and deepfake technology.

2. Natural Language Processing (NLP):

- **Sentiment analysis**: Neural networks can analyse and classify the sentiment (positive, negative, neutral) expressed in text, resulting in applications like social media monitoring, customer feedback analysis, and opinion mining.
- **Machine translation**: Neural networks make it possible to handle machine translation tasks, facilitating text translation between different languages.
- **Text generation:** Neural networks can generate human-like text, enabling applications like chatbots, language modelling, and content creation.

3. Speech and audio processing:

- **Speech recognition:** Neural networks are used to convert spoken language into written text, resulting in applications like voice assistants, transcription services, and voice-controlled systems.
- **Speaker identification:** Neural networks can identify individuals based on their unique voice characteristics, enabling applications like speaker verification and access control systems.
- **Music generation:** Neural networks can generate original music compositions, aiding creative endeavours and generating personalized soundtracks.

4. Recommender systems:

- **Personalized recommendations:** Neural networks can learn user preferences and make personalized recommendations for products, movies, music, and more, improving user experience and engagement on platforms like e-commerce websites, streaming services, and social media platforms.

5. Financial applications:

- **Stock market prediction**: Neural networks can analyse historical financial data and make predictions about stock market trends, aiding in investment decision-making.
- **Fraud detection:** Neural networks can detect patterns of fraudulent activities in financial transactions, helping to prevent financial fraud and ensure security.

6. Healthcare:

- **Disease diagnosis**: Neural networks can analyse medical images, such as X-rays and MRIs to assist in diagnosing diseases like cancer, eye conditions, and neurological disorders.
- **Drug discovery:** Neural networks can analyse vast amounts of chemical and biological data to aid in discovering and developing new drugs and therapies.
- **Health monitoring:** Neural networks can analyse sensor data from wearable devices to monitor vital signs, detect anomalies, and provide personalized health recommendations.

In addition to these examples, neural networks have a number of practical applications. They have broad applicability across industries and continue to advance AI capabilities in areas where pattern recognition, prediction, and decision-making from complex data are crucial.

Application	Architecture / Algorithm	Activation Function
Process modeling and control	Radial Basis Network	Radial Basis
Machine Diagnostics	Multilayer Perceptron	Tan- Sigmoid Function
Portfolio Management	Classification Supervised Algorithm	Tan- Sigmoid Function
Target Recognition	Modular Neural Network	Tan- Sigmoid Function
Medical Diagnosis	Multilayer Perceptron	Tan- Sigmoid Function
Credit Rating	Logistic Discriminant Analysis with ANN, Support Vector Machine	Logistic function
Targeted Marketing	Back Propagation Algorithm	Logistic function
Voice recognition	Multilayer Perceptron, Deep Neural Networks(Convolutional Neural Networks)	Logistic function
Financial Forecasting	Backpropagation Algorithm	Logistic function
Intelligent searching	Deep Neural Network	Logistic function
Fraud detection	Gradient - Descent Algorithm and Least Mean Square (LMS) algorithm.	Logistic function

Advantages of Artificial Neural Networks

- **Parallel processing abilities.** ANNs have parallel processing abilities, which means the network can perform more than one job at a time.
- **Information storage.** ANNs store information on the entire network, not just in a database. This ensures that even if a small amount of data disappears from one location, the entire network continues to operate.
- **Non-linearity.** The ability to learn and model nonlinear, complex relationships helps model the real-world relationships between input and output.
- **Fault tolerance.** ANNs come with fault tolerance, which means the corruption or fault of one or more cells of the ANN won't stop the generation of output.
- **Gradual corruption.** This means the network slowly degrades over time instead of degrading instantly when a problem occurs.
- **Unrestricted input variables.** No restrictions are placed on the input variables, such as how they should be distributed.
- **Observation-based decisions.** Machine learning means the ANN can learn from events and make decisions based on the observations.
- **Unorganized data processing**. Artificial neural networks are exceptionally good at organizing large amounts of data by processing, sorting and categorizing it.

- **Ability to learn hidden relationships.** ANNs can learn the hidden relationships in data without commanding any fixed relationship. This means ANNs can better model highly volatile data and non-constant variance.
- **Ability to generalize data**. The ability to generalize and infer unseen relationships on unseen data means ANNs can predict the output of unseen data.

Disadvantages of Artificial Neural Networks

- **Lack of rules**. The lack of rules for determining the proper network structure means the appropriate artificial neural network architecture can only be found through trial, error and experience.
- **Hardware dependency.** The requirement of processors with parallel processing abilities makes neural networks dependent on hardware.
- **Numerical translation.** The network works with numerical information, meaning all problems must be translated into numerical values before they can be presented to the ANN.
- **Lack of trust.** The lack of explanation behind probing solutions is one of the biggest disadvantages of ANNs. The inability to explain the why or how behind the solution generates a lack of trust in the network.
- **Inaccurate results.** If not trained properly, ANNs can often produce incomplete or inaccurate results.
- **Black box nature.** Because of their black box AI model, it can be challenging to grasp how neural networks make their predictions or categorize data.
