Unit 5

Recommender Systems / Recommendation Systems:-

In today's world, recommender systems have become essential. They have made our lives easier and more comfortable, from suggesting books, movies, and songs to recommending what to buy next. According to reports, Netflix states that algorithmically-generated recommendations influence 80% of its viewership.

But how do these systems work? What are the different types of recommender systems available? This article aims to introduce the various recommender systems and their strengths and weaknesses. What's important is that we discuss our personal experience of working with them.

What are recommender systems?

Recommender systems, also known as recommendation systems, are machine learning algorithms that use data to recommend items or content to users based on their preferences, past behavior, or their combination. These systems can recommend various items, such as movies, books, music, products, etc.

Different approaches to building recommender systems include collaborative filtering, content-based filtering, demographic-based filtering, utility-based filtering, knowledge-based filtering, and hybrid approaches. The ultimate goal of recommender systems is to help users find items they will likely enjoy and increase user engagement with the application or platform.

Making recommendations involves two main stages:

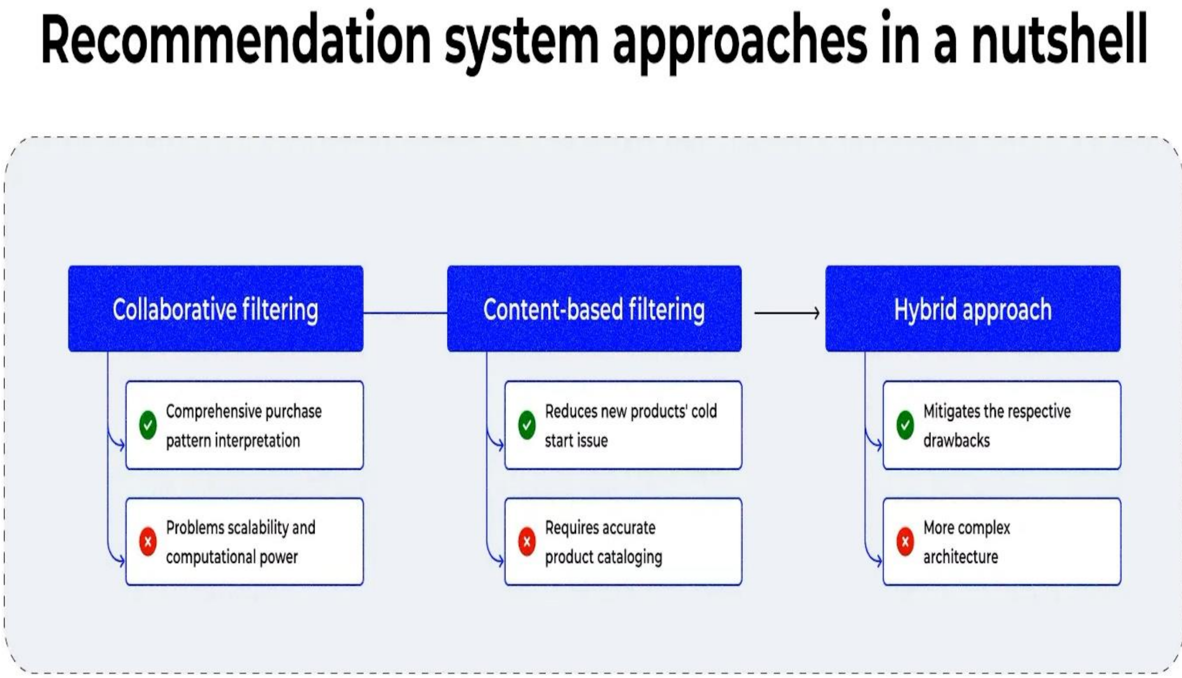
- Candidate generation — creating a subset of products the user might be interested in.
- Scoring — reducing and sorting a candidate list.

Large enterprises such as Google, Instagram, Spotify, Amazon, Reddit, and Netflix frequently employ them to boost engagement with their platform and users. By collecting user data, Amazon utilizes recommendations to propose products to diverse users. For instance, Spotify suggests songs similar to the ones you've often played or liked to keep you using their service to stream music.

What types of recommender systems exist?

There are three primary sub-categories that most recommendation engines fall into, which depend on the method used to choose and suggest products or services that cater to the individual needs of each customer.

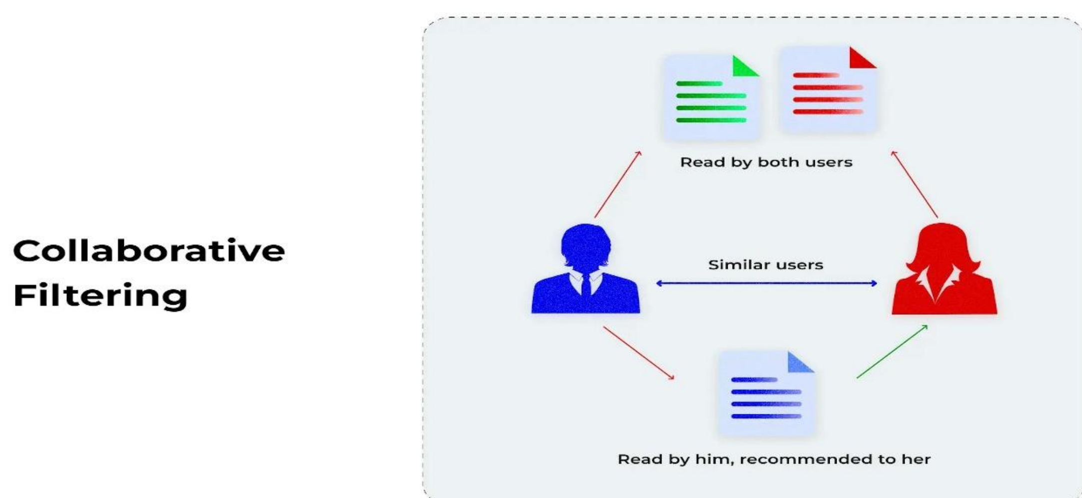
- Collaborative filtering;
- Content-based filtering;
- Hybrid recommendation systems.



Let's look at each of them separately.

Collaborative filtering

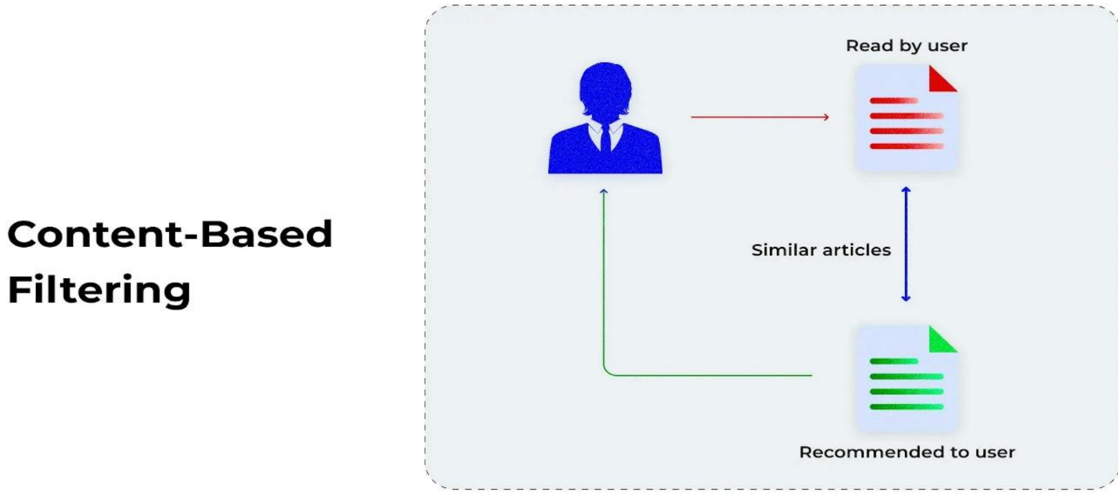
In this scenario, the emphasis is placed on customers and their experiences with the online platform and their opinions on products instead of the features of the items themselves. As a result, recommendation systems falling under this category leverage machine learning algorithms to gather feedback from users and comprehend what they prefer. This enables the system to suggest products other users with similar preferences purchase.



PROS	CONS
Broader exposure to many different products.	The cold-start problem is where new users or items have no or scant ratings or interactions, which the system may find challenging to provide suitable recommendations for.
No need for domain knowledge or understanding of item content.	Data Sparsity is where data is not enough to model the user's preferences accurately, which may result in poor recommendations.
Captures the change in user's preferences over time.	Scalability issue, where growing the number of users and items exponentially increases the size of the user-item matrix.
Enhances user experience by recommending popular items among similar users.	The problem with privacy since users have to disclose their rating or interaction data to the system.
Easy creation and use of the model.	Limited diversity in recommendations since it tends to recommend popular items with high ratings.

Content-based filtering

Content-based filtering in recommender systems recommends items to users based on their previous actions or preferences. It analyzes item metadata to identify items with similar characteristics to those that the user has interacted with before. This approach examines the characteristics of the items users have expressed an interest in to recommend similar items, unlike collaborative filtering, which finds similarities among users. Content-based filtering is widely used in e-commerce, news feeds, music, and movie recommendations.

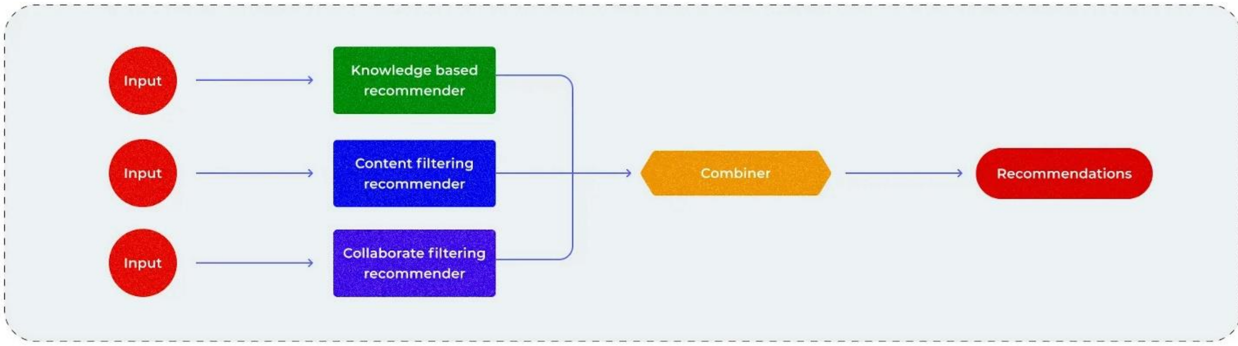


PROS	CONS
Content-based filtering does not require data about other users' preferences, making it easier to implement and potentially more scalable.	Lack of novelty and diversity in recommendations due to the emphasis on characteristics similar to items the user has already interacted with.
It can make recommendations for new items immediately without waiting for users to provide feedback or data.	The requirement of hand-engineered item features or domain knowledge for the feature selection.
It can consider individual users' preferences and the specific features they are looking for in a product, leading to more relevant recommendations.	Difficulty in recommending new or unpopular items that have not been rated by many users yet.
Due to transparency, recommendations can be easier to understand and explain since they are based on specific features and attributes of products rather than more complex relationships between users and items.	The potential for incorrect or insufficient metadata or characteristics could result in inaccurate recommendations.

Hybrid systems

Hybrid recommendation systems combine two or more recommendation strategies in different ways to leverage their complementary strengths.

Hybrid Systems



PROS	CONS
Can lead to improved recommendation accuracy and better coverage of diverse products.	More advanced computing power and intricate architectures are required to combine the mechanisms of both approaches into a unified system.
Can overcome the limitations of individual recommendation strategies and provide more personalized recommendations.	
Can significantly enhance the effectiveness and efficiency of recommendation engines in a variety of applications.	

How powerful are the recommendations?

Recommender systems can be improved by understanding and analyzing relationships between:

- **User and product.** When the user has a preference for a specific product. For example, one Netflix user may prefer thrillers, whereas another likes comedies.
- **Product and product.** Similar items are known as product and product. Similar music or movies, for example.
- **User and user.** The same or different tastes of users concerning the same item. For example, teenagers may differ from adults in terms of the content they consume.

Designing a recommender system with user-item relationships in mind can greatly improve the user experience and increase engagement with the product. An excellent example is YouTube, where personalized recommendations keep users hooked for extended periods. Imagine browsing YouTube without recommendations tailored to your interests; the experience wouldn't be as engaging or enjoyable.

What is the role of machine learning?

While it's fair to say that most people have a basic understanding of recommender systems, many may not be aware of the role that machine learning plays in their advanced functionality. Machine learning (ML), a sub-branch of artificial intelligence (AI), is crucial to creating algorithms that can handle enormous datasets, identify patterns and correlations among multiple variables, and build accurate predictive models.

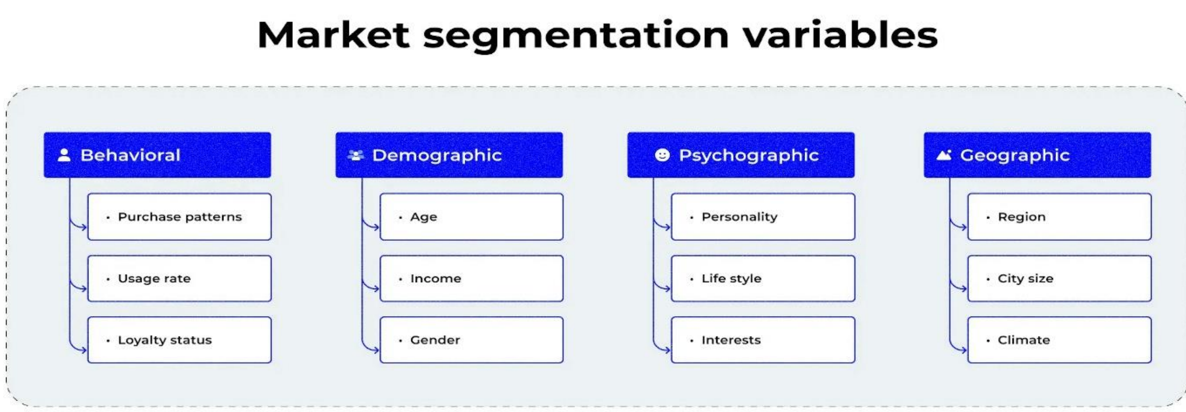
The key differentiator of ML is its ability to learn and improve on results over time based on experience processing vast amounts of data. This can result in models that accurately predict customers' wants and future developments. For example, an ML-powered solution might identify a connection between customer age and preference for one brand. In this way, ML-based recommendations can provide valuable insight into customer behavior and preferences.

The name "learning" in ML is significant, as it enables ML systems and reinforcement learning applications to improve their capabilities through experience continually. As these algorithms process more data, they can identify more relationships among data points and refine their models, improving over time.

How does it work?

Let's see an example to clarify how ML-based recommender systems perform their duties. You can boast solid expertise in the latest movies and TV serials. Furthermore, your knowledge is already framed as you know what you usually watch perfectly. In addition to an extensive range of personal factors, including but not limited to cultural interests, social environment, and profession, numerous other aspects could have influenced your tastes.

But not all platforms have this privilege because when you see it for the first time as you open a new streaming service. Hence, to suggest movies that align with your preferences, the service providers will have to engage with you, gather relevant information, comprehend the kind of customer you are, and make recommendations accordingly. In marketing terms, they have to segment you, namely categorize you into a certain customer archetype or buyer persona according to your characteristics (purchase patterns, interests, gender, etc.), and target you with a suitable movie suggestion.



In digital marketplaces, recommendation systems perform the same functions as traditional sales staff, separating and proposing items to potential buyers. However, while human sellers use their experience and instinct to assess a limited set of factors during a brief interaction with customers, recommendation engines employ ML techniques to assess vast amounts of customer data and a wider array of criteria to achieve classification and targeting. These criteria may include browsing trends, purchasing history, data usage, personal data from user profiles, item evaluations, and device preferences.

Apply this to all users on a particular platform. You can see how a recommendation system can gain a comprehensive understanding of individual buyers, the overall audience, and even the underlying sales trends that would be difficult for a human observer to discern. Moreover, ML algorithms can take into consideration diverse contextual factors that are not always directly linked to customers. For instance, an e-commerce website that utilizes an ML-powered recommendation engine will begin recommending Valentine's Day-related products as the holiday approaches. Similarly, a streaming service could adjust its recommendations for weekend family-friendly movies and documentaries.

Highly-rated recommendation systems on the Web

Recommendation systems have become essential for major digital service providers and e-commerce companies to offer an individualized user experience and increase their sales performance and advertising revenues.

- **Amazon.** This service uses an algorithm to recommend products and search results to users based on strategies like "recommended for you," "bought together," and "recently viewed." Amazon also sends off-site recommendations via email. The recommender engine was implemented in 2011-2012 and contributed to a 29% sales increase in the second fiscal quarter of 2012.
- **Facebook.** This platform employs a deep learning and neural network-based recommendation engine named DLRM (Deep-learning Recommendation Model) to provide suggestions, sort the News Feed, and recommend pages, groups, and products on its Marketplace.
- **LinkedIn.** It has implemented a recommendation system to provide suggestions for job postings, connections, and courses. This system includes LinkedIn Recruiter, a robust HR tool that can gather a list of eligible candidates for open positions and rank them based on their skills, experience, and response probability.
- **Netflix**. This streaming service utilizes a recommendation system to suggest movies to its users. The system considers various useful features such as browsing history and ratings, the popularity and type of movies, seasonal trends, and the similarity between previously viewed content and other available options. The results are then displayed on the home page of Netflix, segregated into horizontal rows of movies.
- **Spotify**. It provides users with 30 new song recommendations every Monday, and these recommendations are mainly determined by an AI system called 'Bandits for Recommendations as Treatments' (BaRT for short).

YouTube. It has introduced a recommendation system that prioritizes specific videos, suggests relevant news, and encourages channel subscriptions. The system utilizes an engine that considers numerous parameters, known as "signals," to identify user preferences more accurately. Such signals include clicks, likes, dislikes, watch time, and shares.

How do you write a recommendation system?

To write a recommendation system, you must follow these general steps:

- **Define the purpose** of your recommendation system and what type of data you will be working with (e.g., movies, products, books, etc.).
- **Gather and preprocess your data**. This may involve cleaning your data, removing duplicates, and structuring it in a format your model can use.
- **Choose a recommendation algorithm**. This will depend on the data type you are working with and the problem you are trying to solve. Collaborative filtering, content-based filtering, and hybrid approaches are common recommendation algorithms.
- **Train and test your model**. Split your data into training and testing sets, and use your training set to train your model. Test your model by measuring its accuracy on the testing set.
- **Evaluate and optimize your model**. Adjust the parameters and settings of your model to improve its accuracy. You may also consider adding additional features or data sources to improve your model.
- **Deploy your model**. Once satisfied with your model, you can deploy it to your application or platform.

Overall, building a recommendation system involves a combination of data processing, machine learning, and software engineering skills.

Importance of Recommendation Systems

Recommender systems are an essential component of current digital platforms, helping to improve user experiences, drive engagement, and provide decision-making tools. These systems serve as information filtering tools, providing users with tailored material or information that is relevant to their taste and interests.
Recommender systems have become essential for organizations since they can significantly boost income by making tailored suggestions that result in improved sales.

- **Faster Decision-making:** Recommender systems increase user tendency to purchase suggested things, boost loyalty and overall happiness, lower transaction costs, and improve decision-making process and quality.
- **Personalized user experience:** Making highly relevant and valuable suggestions, recommender systems improve the user experience.
- **Increase engagement:** Recommendation systems help users interact with a system by providing them material, goods, or services that they are likely to be interested in.

Artificial Neural Network

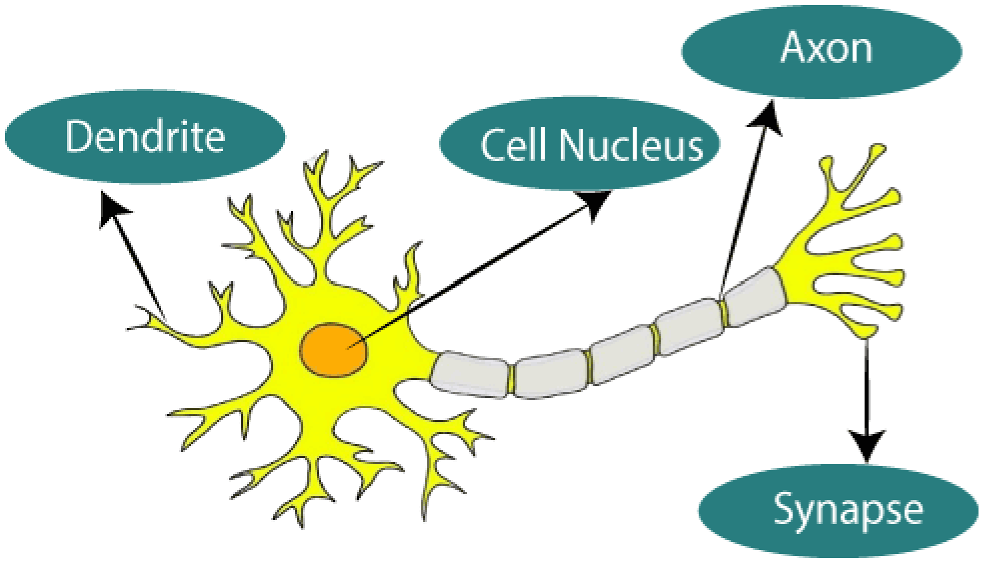
Artificial Neural Network Tutorial provides basic and advanced concepts of ANNs. Our Artificial Neural Network tutorial is developed for beginners as well as professions.

The term "Artificial neural network" refers to a biologically inspired sub-field of artificial intelligence modeled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes.

Artificial neural network tutorial covers all the aspects related to the artificial neural network. In this tutorial, we will discuss ANNs, Adaptive resonance theory, Kohonen self-organizing map, Building blocks, unsupervised learning, Genetic algorithm, etc.

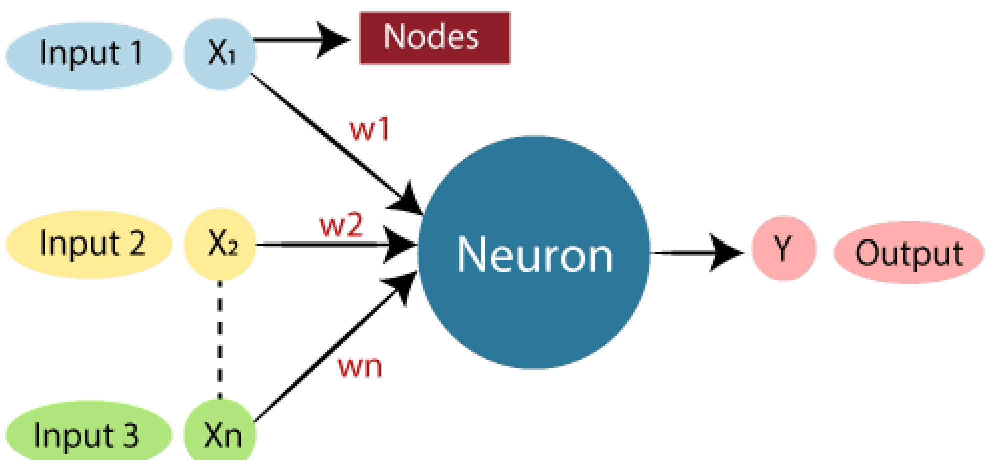
What is Artificial Neural Network?

The term "**Artificial Neural Network**" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.



The given figure illustrates the typical diagram of Biological Neural Network.

The typical Artificial Neural Network looks something like the given figure.



Dendrites from Biological Neural Network represent inputs in Artificial Neural Networks, cell nucleus represents Nodes, synapse represents Weights, and Axon represents Output.

Relationship between Biological neural network and artificial neural network:

Biological Neural Network	Artificial Neural Network

Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

An **Artificial Neural Network** in the field of **Artificial intelligence** where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

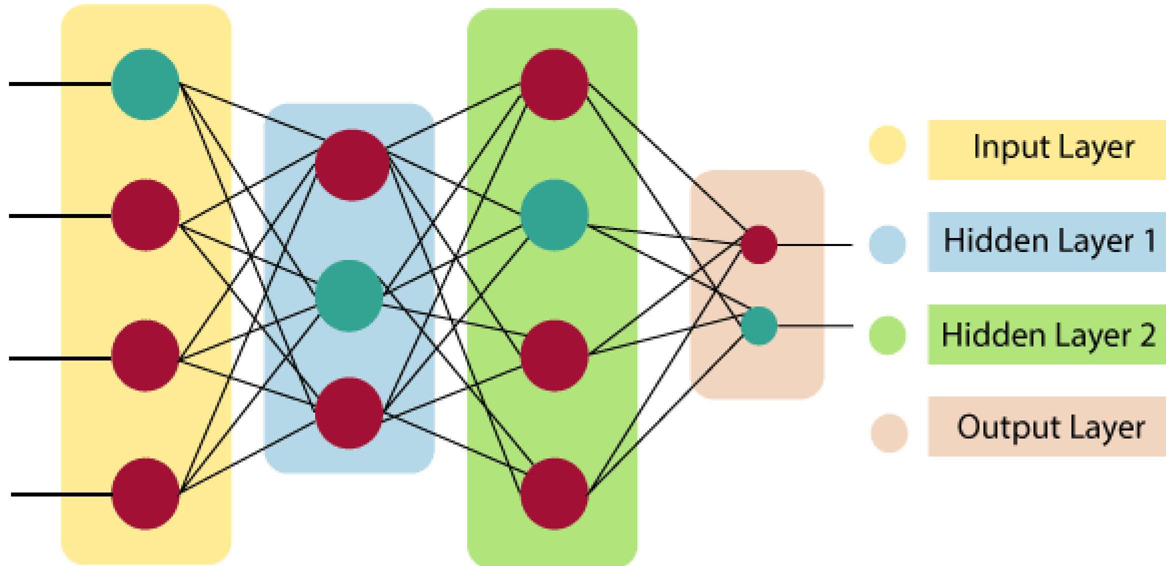
There are around 1000 billion neurons in the human brain. Each neuron has an association point somewhere in the range of 1,000 and 100,000. In the human brain, data is stored in such a manner as to be distributed, and we can extract more than one piece of this data when necessary from our memory parallelly. We can say that the human brain is made up of incredibly amazing parallel processors.

We can understand the artificial neural network with an example, consider an example of a digital logic gate that takes an input and gives an output. "OR" gate, which takes two inputs. If one or both the inputs are "On," then we get "On" in output. If both the inputs are "Off," then we get "Off" in output. Here the output depends upon input. Our brain does not perform the same task. The outputs to inputs relationship keep changing because of the neurons in our brain, which are "learning."

The architecture of an artificial neural network:

To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Lets us look at various types of layers available in an artificial neural network.

Artificial Neural Network primarily consists of three layers:



Input Layer:

As the name suggests, it accepts inputs in several different formats provided by the programmer.

Hidden Layer:

The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

Output Layer:

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} Wi * Xi + b$$

It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.

Advantages of Artificial Neural Network (ANN)

Parallel processing capability:

Artificial neural networks have a numerical value that can perform more than one task simultaneously.

Storing data on the entire network:

Data that is used in traditional programming is stored on the whole network, not on a database. The disappearance of a couple of pieces of data in one place doesn't prevent the network from working.

Capability to work with incomplete knowledge:

After ANN training, the information may produce output even with inadequate data. The loss of performance here relies upon the significance of missing data.

Having a memory distribution:

For ANN is to be able to adapt, it is important to determine the examples and to encourage the network according to the desired output by demonstrating these examples to the network. The succession of the network is directly proportional to the chosen instances, and if the event can't appear to the network in all its aspects, it can produce false output.

Having fault tolerance:

Extortion of one or more cells of ANN does not prohibit it from generating output, and this feature makes the network fault-tolerance.

Disadvantages of Artificial Neural Network:

Assurance of proper network structure:

There is no particular guideline for determining the structure of artificial neural networks. The appropriate network structure is accomplished through experience, trial, and error.

Unrecognized behavior of the network:

It is the most significant issue of ANN. When ANN produces a testing solution, it does not provide insight concerning why and how. It decreases trust in the network.

Hardware dependence:

Artificial neural networks need processors with parallel processing power, as per their structure. Therefore, the realization of the equipment is dependent.

Difficulty of showing the issue to the network:

ANNs can work with numerical data. Problems must be converted into numerical values before being introduced to ANN. The presentation mechanism to be resolved here will directly impact the performance of the network. It relies on the user's abilities.

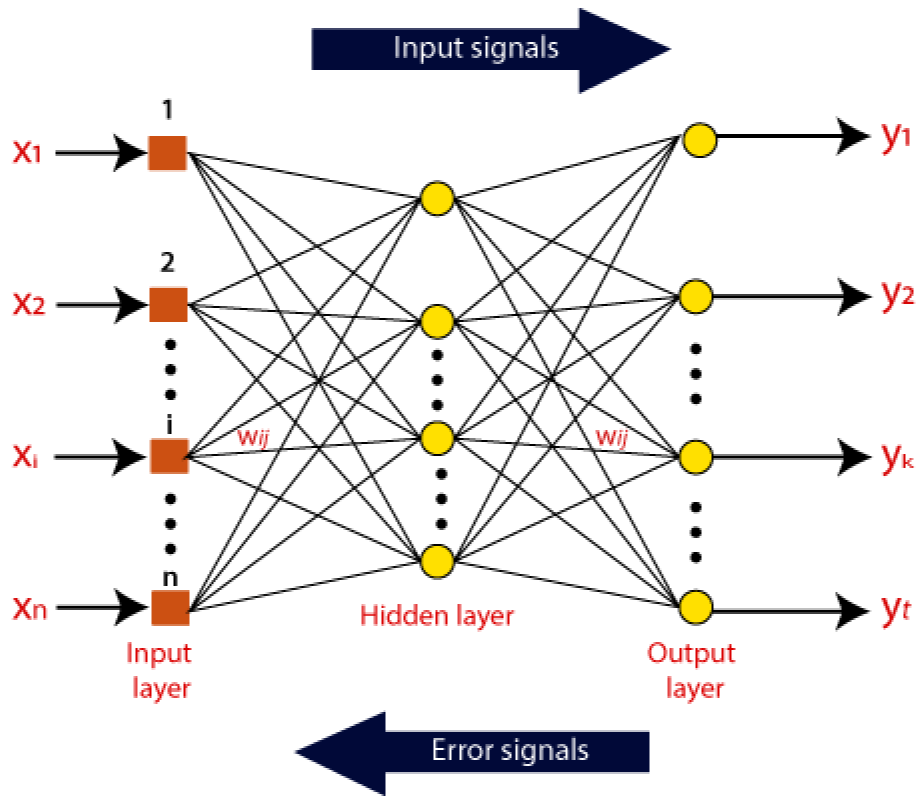
The duration of the network is unknown:

The network is reduced to a specific value of the error, and this value does not give us optimum results.

Science artificial neural networks that have steeped into the world in the mid-20th century are exponentially developing. In the present time, we have investigated the pros of artificial neural networks and the issues encountered in the course of their utilization. It should not be overlooked that the cons of ANN networks, which are a flourishing science branch, are eliminated individually, and their pros are increasing day by day. It means that artificial neural networks will turn into an irreplaceable part of our lives progressively important.

How do artificial neural networks work?

Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.



Afterward, each of the input is multiplied by its corresponding weights (these weights are the details utilized by the artificial neural networks to solve a specific problem). In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network. All the weighted inputs are summarized inside the computing unit.

If the weighted sum is equal to zero, then bias is added to make the output non-zero or something else to scale up to the system's response. Bias has the same input, and weight equals to 1. Here the total of weighted inputs can be in the range of 0 to positive infinity. Here, to keep the response in the limits of the desired value, a certain maximum value is benchmarked, and the total of weighted inputs is passed through the activation function.

The activation function refers to the set of transfer functions used to achieve the desired output. There is a different kind of the activation function, but primarily either linear or non-linear sets of functions. Some of the commonly used sets of activation functions are the Binary, linear, and Tan hyperbolic sigmoidal activation functions. Let us take a look at each of them in details:

Binary:

In binary activation function, the output is either a one or a 0. Here, to accomplish this, there is a threshold value set up. If the net weighted input of neurons is more than 1, then the final output of the activation function is returned as one or else the output is returned as 0.

Sigmoidal Hyperbolic:

The Sigmoidal Hyperbola function is generally seen as an "**S**" shaped curve. Here the tan hyperbolic function is used to approximate output from the actual net input. The function is defined as:

F(x) = (1/1 + exp(-????x))

Where ???? is considered the Steepness parameter.

Types of Artificial Neural Network:

There are various types of Artificial Neural Networks (ANN) depending upon the human brain neuron and network functions, an artificial neural network similarly performs tasks. The majority of the artificial neural networks will have some similarities with a more complex biological partner and are very effective at their expected tasks. For example, segmentation or classification.

Feedback ANN:
In this type of ANN, the output returns into the network to accomplish the best-evolved results internally. As per the **University of Massachusetts**, Lowell Centre for Atmospheric Research. The feedback networks feed information back into itself and are well suited to solve optimization issues. The Internal system error corrections utilize feedback ANNs.

Feed-Forward ANN:
A feed-forward network is a basic neural network comprising of an input layer, an output layer, and at least one layer of a neuron. Through assessment of its output by reviewing its input, the intensity of the network can be noticed based on group behavior of the associated neurons, and the output is decided. The primary advantage of this network is that it figures out how to evaluate and recognize input patterns.

Prerequisite

No specific expertise is needed as a prerequisite before starting this tutorial.

Audience

Our Artificial Neural Network Tutorial is developed for beginners as well as professionals, to help them understand the basic concept of ANNs.

Problems

We assure you that you will not find any problem in this Artificial Neural Network tutorial. But if there is any problem or mistake, please post the problem in the contact form so that we can further improve it.

Multiple Choices on Artificial Neural Networks

1. The activation function of the artificial neural network serves what main purpose?

a. The focused goal to Calculate the weighted sum of the inputs
b. To design non-linearity into the network
c. When conducting the back propagation to be able to update the weights.
d. To re-scale, it back to the original range

Answer: b) To supply non-linearity into the network

Explanation: They are applied to introduce non-linearity into the network so that the network can be able to learn various forms of patterns. They decide the output level of a neuron by using the weighted sum of the inputs passing through its control.

2.Which of the following is not an example of an artificial neural network? This model is:

a. Feedforward neural network
b. Recurrent neural network
c. Convolutional neural network
d. Logistic regression

Answer: d) Logistic regression

Explanation: Logistic regression is actually a model used in classification thus it cannot in anyway be considered as a neural network.

---

3. What is the function of an algorithm known as backpropagation in the construction of neural networks?

a. For the initial setting of weights of the network
b. To compute the value at the output of the network
c. For modifying the weights of the network with an aim to reduce the error
d. To scale down the input data

Answer: c) For the weights of network applied corrections aimed at minimizing an error

Explanation: Backpropagation is an algorithm that calculates the error from the input which has been processed through the network and then goes back through the connections to modify weights accordingly.

---

4. What layer of neural network take the input data?

a. A layer that is concealed
b. The final layer that gives the results
c. The first layer where input is taken
d. A layer that does not connect to other layers, it only takes the signal and ‘activates’ it.

Answer: c) Input layer

Explanation: The input layer is the outermost layer in a neural network and is responsible for taking the input data which is to be analysed.

---

5. The explicit bias term in a neuron is used hence to introduce an explicit bias to the activation function employed in the neuron.

a. To make the relationship between inputs and weights non-linear
b. To make the _ of the activation function
c. To standardize the input data
d. To alter the weights

Answer: b) To replace the activation function

Explanation: The bias term makes the activation function to be flexible in learning other patterns since it shifts the curve up or down.

---

6. Which activation function we use in the last layer for classification problem? There are four activation functions, including

a. ReLU
b. Sigmoid
c. Tanh and
d. Linear.

Answer: b) Sigmoid

Explanation: The sigmoid activation function is utilized in the output layer of the feed forward networks particularly in classification problems since the output value ranges between 0 and 1 which may be seen as a probable distribution of the input patterns.

---

7. What can be considered as the primary distinction between supervised and unsupervised learning in neural nets?

a. Supervised learning: it involves training the algorithm on data sets which are labelled while unsupervised learning: does not involve training of data set having labels.
b. Supervised learning is applied when the result, which needs to be predicted, is categorical while unsupervised learning is applied when the result is continuous.
c. Supervised learning is believed to be sophisticated than unsupervised learning.
d. Supervised learning takes less time more than unsupervised learning.

Answer: a) Supervised learning work on labelled data set while unsupervised learning does not work on labelled data set.

Explanation: Supervised learning on the other hand, involves learning where the input values are well defined and pointed out while the output values. In case of unsupervised learning the network looks for different relationships inside the data and has no tags for it.

---

8. What is the architecture of the neural network for the aims of image recognition?

a. RNN
b. CNN
c. Perceptron
d. Autoencoder

Answer: b) Convolutional neural network

Explanation: Once trained, CNNs are the best to use for special types of data such as images, for example when classifying an image or detecting an object in an image.

---

9. What is exactly the vanishing gradient problem in the deep neural networks?

a. If the gradients are too big
b. If the gradients are too small
c. If weights are too big
d. If the weights are too small

Answer: b) When the gradients become too small

Explanation: This is because during the back propagation the gradients become very much small and thus leading to loss of gradient problem.

---

10. Which technique is favourable in avoiding overfitting in the case of neural network?

a. Regularization

b. Dropout
c. Early stopping
d. All of them

Answer: d) All of the above

Explanation: Quite a few methods are applied to control over fitting; among them, we have regularization techniques, dropout, and early stopping.

---

11. What is a hidden layer of a neural network and what does it look like? The decision rule:

a. For the sake of input data
b. For the output data
c. For the desirable intricate nonlinearity
d. For new weights.

Answer: c) For the purpose of concept acquisition that involves learning of complex patterns and features.

Explanation: In a neural network there are always one or more hidden layers which are responsible for extracting hard and comprehensive features from the input data as used in the predictions or classifications.

---

12. Which activation function do we often apply in the hidden layers of the artificial neural network?

a. ReLU
b. Sigmoid
c. Tanh
d. Linear

Answer: a) ReLU

Explanation: The rectified linear unit (ReLU) non-linearity is used universally as the activation function in the hidden layer of neural networks for their mathematical efficiency and non-sensitive nature to the vanishing gradient issue.

---

13. What is the principal drawback of Neural Networks? It is for the following reasons:

a. Neural networks are computationally expensive
b. Neural networks are not easy to interpret
c. Large amounts of data are required for neural networks
d. All the above

Answer: d) All the above.

Explanation: That’s why, training deep networks in neural networks can be a big computationally demanding. They also can be ambiguous, because it is rather complicated to understand the inner structure of the network. Also, neural networks are data intensive, and as such may demand big data to have an accurate training.

Perceptron in Machine Learning

In Machine Learning and Artificial Intelligence, Perceptron is the most commonly used term for all folks. It is the primary step to learn Machine Learning and Deep Learning technologies, which consists of a set of weights, input values or scores, and a threshold. ***Perceptron is a building block of an Artificial Neural Network***. Initially, in the mid of 19th century, **Mr. Frank Rosenblatt** invented the Perceptron for performing certain calculations to detect input data capabilities or business intelligence. Perceptron is a linear Machine Learning algorithm used for supervised learning for various binary classifiers. This algorithm enables neurons to learn elements and processes them one by one during preparation. In this tutorial, "Perceptron in Machine Learning," we will discuss in-depth knowledge of Perceptron and its basic functions in brief. Let's start with the basic introduction of Perceptron.

What is the Perceptron model in Machine Learning?

Perceptron is Machine Learning algorithm for supervised learning of various binary classification tasks. Further, ***Perceptron is also understood as an Artificial Neuron or neural network unit that helps to detect certain input data computations in business intelligence***.

Perceptron model is also treated as one of the best and simplest types of Artificial Neural networks. However, it is a supervised learning algorithm of binary classifiers. Hence, we can consider it as a single-layer neural network with four main parameters, i.e., **input values, weights and Bias, net sum, and an activation function.**

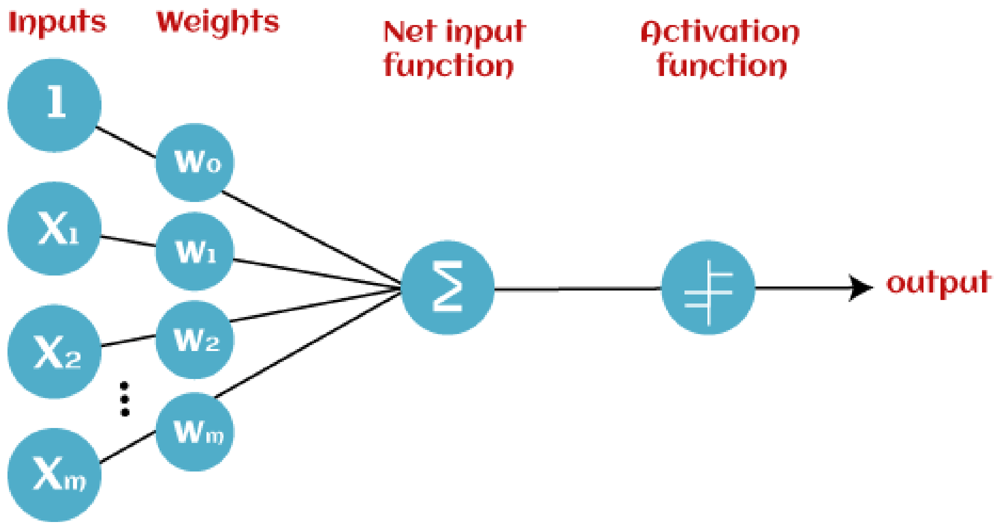
What is Binary classifier in Machine Learning?

In Machine Learning, binary classifiers are defined as the function that helps in deciding whether input data can be represented as vectors of numbers and belongs to some specific class.

Binary classifiers can be considered as linear classifiers. In simple words, we can understand it as a ***classification algorithm that can predict linear predictor function in terms of weight and feature vectors.***

Basic Components of Perceptron

Mr. Frank Rosenblatt invented the perceptron model as a binary classifier which contains three main components. These are as follows:



- **Input Nodes or Input Layer:**

This is the primary component of Perceptron which accepts the initial data into the system for further processing. Each input node contains a real numerical value.

- **Wight and Bias:**

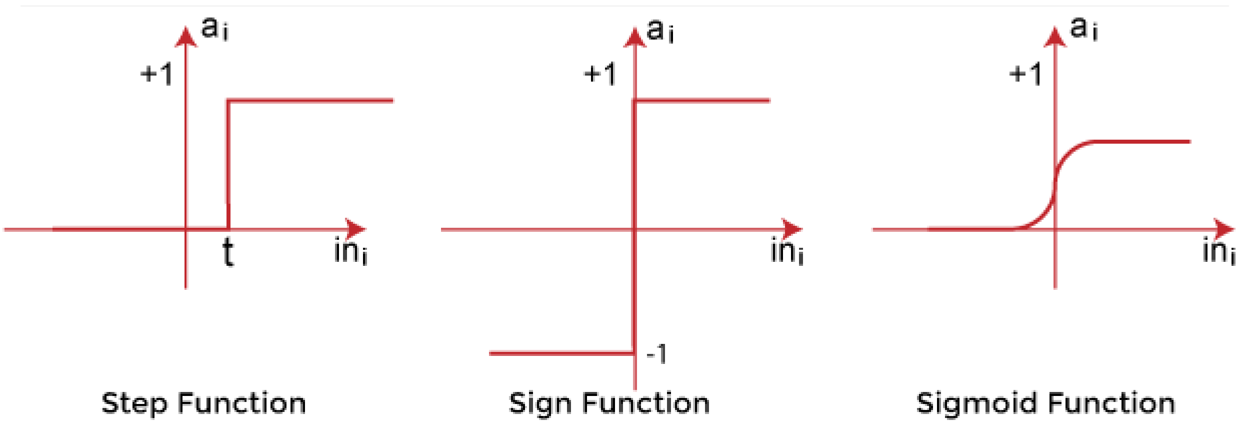
Weight parameter represents the strength of the connection between units. This is another most important parameter of Perceptron components. Weight is directly proportional to the strength of the associated input neuron in deciding the output. Further, Bias can be considered as the line of intercept in a linear equation.

- **Activation Function:**

These are the final and important components that help to determine whether the neuron will fire or not. Activation Function can be considered primarily as a step function.

Types of Activation functions:

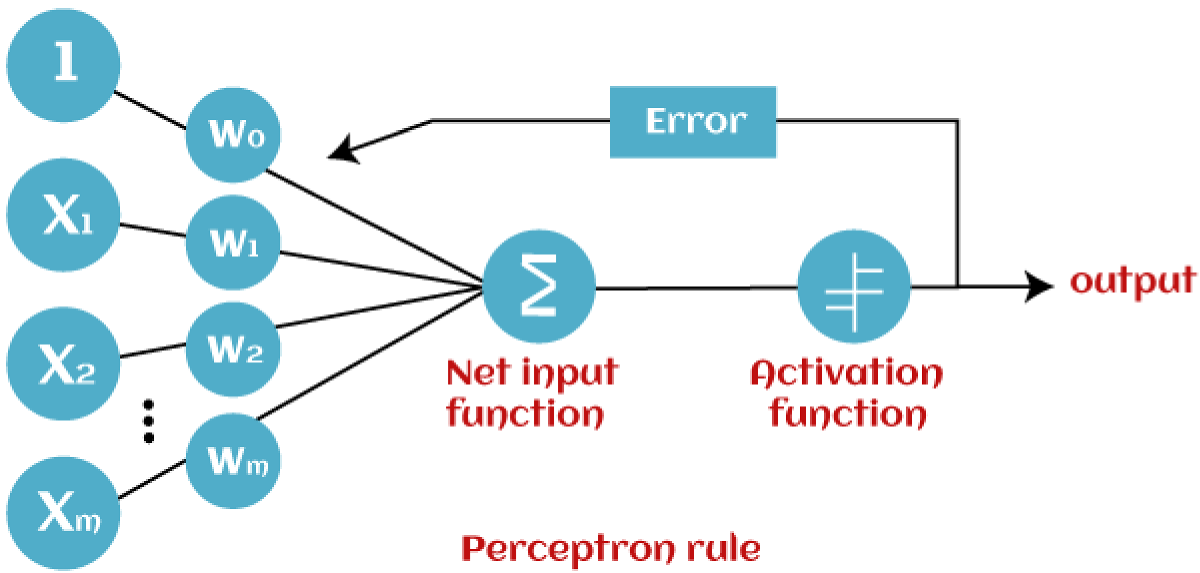
- Sign function
- Step function, and
- Sigmoid function



The data scientist uses the activation function to take a subjective decision based on various problem statements and forms the desired outputs. Activation function may differ (e.g., Sign, Step, and Sigmoid) in perceptron models by checking whether the learning process is slow or has vanishing or exploding gradients.

How does Perceptron work?

In Machine Learning, Perceptron is considered as a single-layer neural network that consists of four main parameters named input values (Input nodes), weights and Bias, net sum, and an activation function. The perceptron model begins with the multiplication of all input values and their weights, then adds these values together to create the weighted sum. Then this weighted sum is applied to the activation function 'f' to obtain the desired output. This activation function is also known as the **step function** and is represented by **'f'**.



This step function or Activation function plays a vital role in ensuring that output is mapped between required values (0,1) or (-1,1). It is important to note that the weight of input is indicative of the strength of a node. Similarly, an input's bias value gives the ability to shift the activation function curve up or down.

Perceptron model works in two important steps as follows:

Step-1

In the first step first, multiply all input values with corresponding weight values and then add them to determine the weighted sum. Mathematically, we can calculate the weighted sum as follows:

∑wi*xi = x1*w1 + x2*w2 +…wn*xn

Add a special term called **bias 'b'** to this weighted sum to improve the model's performance.

∑wi*xi + b

Step-2

In the second step, an activation function is applied with the above-mentioned weighted sum, which gives us output either in binary form or a continuous value as follows:

$Y = f(\sum wi*xi + b)$

Types of Perceptron Models

Based on the layers, Perceptron models are divided into two types. These are as follows:

1. Single-layer Perceptron Model
2. Multi-layer Perceptron model

Single Layer Perceptron Model:
This is one of the easiest Artificial neural networks (ANN) types. A single-layered perceptron model consists feed-forward network and also includes a threshold transfer function inside the model. The main objective of the single-layer perceptron model is to analyze the linearly separable objects with binary outcomes.

In a single layer perceptron model, its algorithms do not contain recorded data, so it begins with inconstantly allocated input for weight parameters. Further, it sums up all inputs (weight). After adding all inputs, if the total sum of all inputs is more than a pre-determined value, the model gets activated and shows the output value as +1.

If the outcome is same as pre-determined or threshold value, then the performance of this model is stated as satisfied, and weight demand does not change. However, this model consists of a few discrepancies triggered when multiple weight inputs values are fed into the model. Hence, to find desired output and minimize errors, some changes should be necessary for the weights input.

"Single-layer perceptron can learn only linearly separable patterns."

Multi-Layered Perceptron Model or Multilayer Network:

Like a single-layer perceptron model, a multi-layer perceptron model also has the same model structure but has a greater number of hidden layers.

The multi-layer perceptron model is also known as the Backpropagation algorithm, which executes in two stages as follows:

- **Forward Stage:** Activation functions start from the input layer in the forward stage and terminate on the output layer.
- **Backward Stage:** In the backward stage, weight and bias values are modified as per the model's requirement. In this stage, the error between actual output and demanded originated backward on the output layer and ended on the input layer.

Hence, a multi-layered perceptron model has considered as multiple artificial neural networks having various layers in which activation function does not remain linear, similar to a single layer perceptron model. Instead of linear, activation function can be executed as sigmoid, TanH, ReLU, etc., for deployment.

A multi-layer perceptron model has greater processing power and can process linear and non-linear patterns. Further, it can also implement logic gates such as AND, OR, XOR, NAND, NOT, XNOR, NOR.

Advantages of Multi-Layer Perceptron:

- A multi-layered perceptron model can be used to solve complex non-linear problems.
- It works well with both small and large input data.
- It helps us to obtain quick predictions after the training.
- It helps to obtain the same accuracy ratio with large as well as small data.

Disadvantages of Multi-Layer Perceptron:

- In Multi-layer perceptron, computations are difficult and time-consuming.
- In multi-layer Perceptron, it is difficult to predict how much the dependent variable affects each independent variable.
- The model functioning depends on the quality of the training.

Perceptron Function

Perceptron function "f(x)" can be achieved as output by multiplying the input 'x' with the learned weight coefficient 'w'.

Mathematically, we can express it as follows:

f(x)=1; if w.x+b>0

otherwise, f(x)=0

- 'w' represents real-valued weights vector
- 'b' represents the bias
- 'x' represents a vector of input x values.

Characteristics of Perceptron

The perceptron model has the following characteristics.

1. Perceptron is a machine learning algorithm for supervised learning of binary classifiers.
2. In Perceptron, the weight coefficient is automatically learned.
3. Initially, weights are multiplied with input features, and the decision is made whether the neuron is fired or not.
4. The activation function applies a step rule to check whether the weight function is greater than zero.
5. The linear decision boundary is drawn, enabling the distinction between the two linearly separable classes +1 and -1.
6. If the added sum of all input values is more than the threshold value, it must have an output signal; otherwise, no output will be shown.

Limitations of Perceptron Model

A perceptron model has limitations as follows:

- The output of a perceptron can only be a binary number (0 or 1) due to the hard limit transfer function.
- Perceptron can only be used to classify the linearly separable sets of input vectors. If input vectors are non-linear, it is not easy to classify them properly.

Future of Perceptron

The future of the Perceptron model is much bright and significant as it helps to interpret data by building intuitive patterns and applying them in the future. Machine learning is a rapidly growing technology of Artificial Intelligence that is continuously evolving and in the developing phase; hence the future of perceptron technology will continue to support and facilitate analytical behavior in machines that will, in turn, add to the efficiency of computers.

The perceptron model is continuously becoming more advanced and working efficiently on complex problems with the help of artificial neurons.

Conclusion:

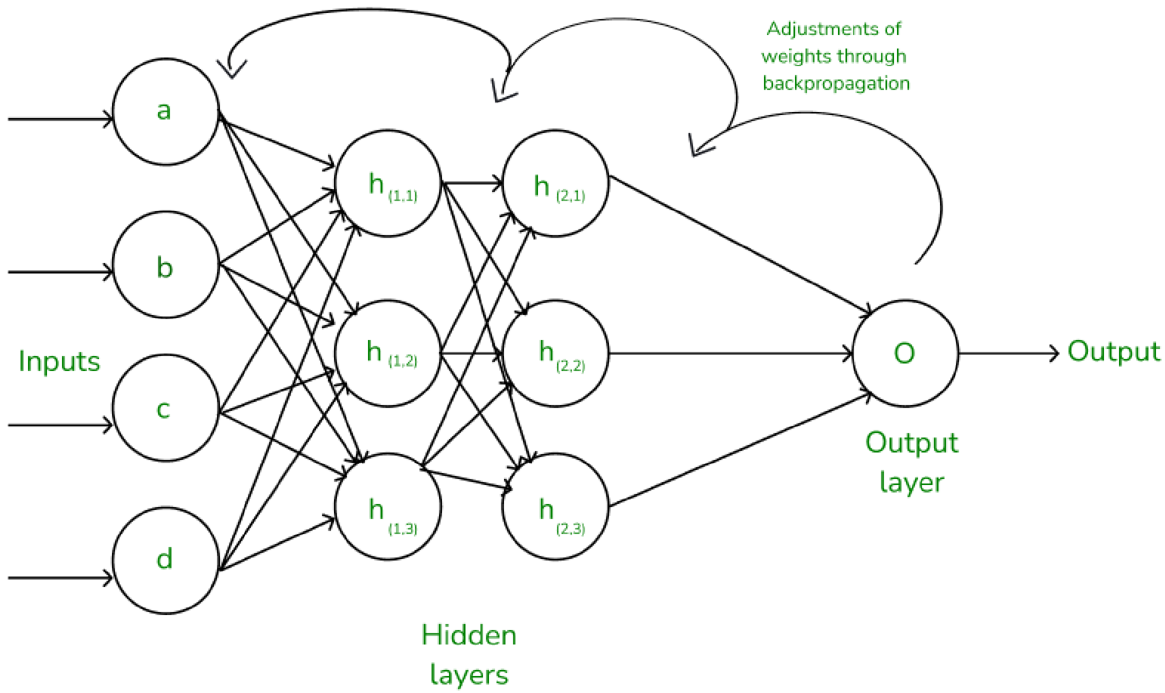
In this article, you have learned how Perceptron models are the simplest type of artificial neural network which carries input and their weights, the sum of all weighted input, and an activation function. Perceptron models are continuously contributing to Artificial Intelligence and Machine Learning, and these models are becoming more advanced. Perceptron enables the computer to work more efficiently on complex problems using various Machine Learning technologies. The Perceptrons are the fundamentals of artificial neural networks, and everyone should have in-depth knowledge of perceptron models to study deep neural networks.

Backpropagation in Neural Network

Backpropagation is also known as "*Backward Propagation of Errors*" and it is a method used to train **neural network** . Its goal is to reduce the difference between the model's predicted output and the actual output by adjusting the weights and biases in the network. In this article we will explore what backpropagation is, why it is crucial in machine learning and how it works.

What is Backpropagation?

Backpropagation is a technique used in deep learning to train artificial neural networks particularly **feed-forward networks**. It works iteratively to adjust weights and bias to minimize the cost function.
In each epoch the model adapts these parameters reducing loss by following the error gradient. Backpropagation often uses optimization algorithms like **gradient descent** or **stochastic gradient descent**. The algorithm computes the gradient using the chain rule from calculus allowing it to effectively navigate complex layers in the neural network to minimize the cost function.



Fig(a) A simple illustration of how the backpropagation works by adjustments of weights

Backpropagation plays a critical role in how neural networks improve over time. Here's why:

1. **Efficient Weight Update**: It computes the gradient of the loss function with respect to each weight using the chain rule making it possible to update weights efficiently.
2. **Scalability**: The backpropagation algorithm scales well to networks with multiple layers and complex architectures making deep learning feasible.
3. **Automated Learning**: With backpropagation the learning process becomes automated and the model can adjust itself to optimize its performance.

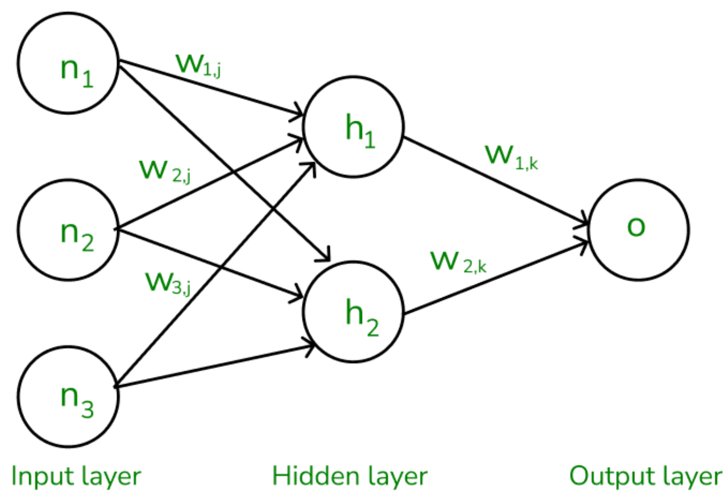
Working of Backpropagation Algorithm

The Backpropagation algorithm involves two main steps: the **Forward Pass** and the **Backward Pass**.

How Does Forward Pass Work?

In **forward pass** the input data is fed into the input layer. These inputs combined with their respective weights are passed to hidden layers. For example in a network with two hidden layers (h1 and h2 as shown in Fig. (a)) the output from h1 serves as the input to h2. Before applying an activation function, a bias is added to the weighted inputs.

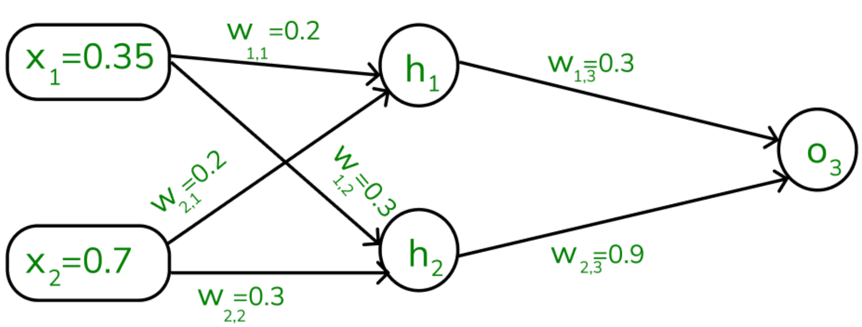
Each hidden layer computes the weighted sum (`a`) of the inputs then applies an activation function like **ReLU (Rectified Linear Unit)** to obtain the output (`o`). The output is passed to the next layer where an activation function such as **softmax** converts the weighted outputs into probabilities for classification.



The forward pass using weights and biases

How Does the Backward Pass Work?
In the backward pass the error (the difference between the predicted and actual output) is propagated back through the network to adjust the weights and biases. One common method for error calculation is the **Mean Squared Error (MSE)** given by:
MSE=(Predicted Output−Actual Output)2*MSE*=(Predicted Output−Actual Output)2
Once the error is calculated the network adjusts weights using **gradients** which are computed with the chain rule. These gradients indicate how much each weight and bias should be adjusted to minimize the error in the next iteration. The backward pass continues layer by layer ensuring that the network learns and improves its performance. The activation function through its derivative plays a crucial role in computing these gradients during backpropagation.

Example of Backpropagation in Machine Learning
Let’s walk through an example of backpropagation in machine learning. Assume the neurons use the sigmoid activation function for the forward and backward pass. The target output is 0.5, and the learning rate is 1.



Example (1) of backpropagation sum

Forward Propagation

1. Initial Calculation

The weighted sum at each node is calculated using:

$$a_j = \sum (w_{i,j} * x_i)$$

Where,

- a_j is the weighted sum of all the inputs and weights at each node
- $w_{i,j}$ represents the weights between the ith input and the jth neuron
- x_i represents the value of the ith input

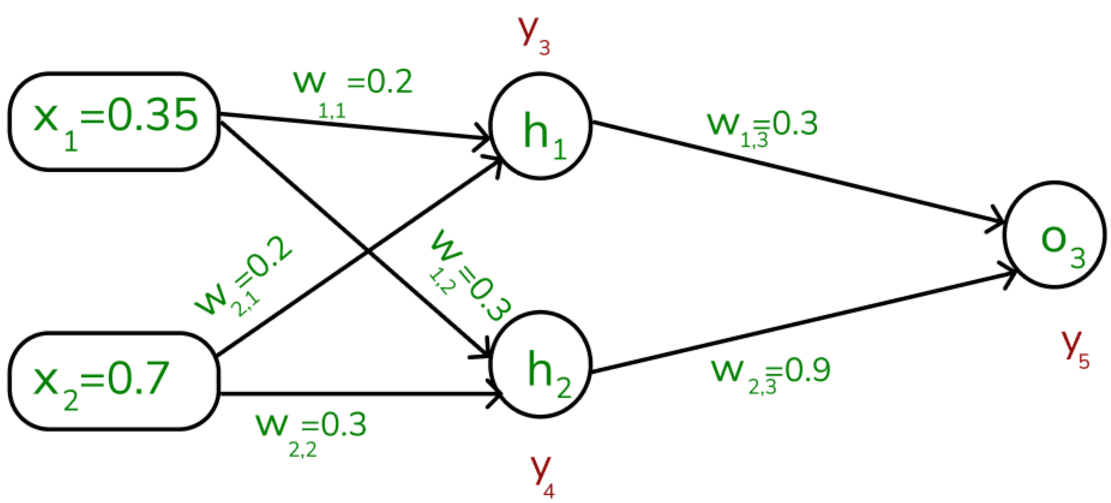
o (output): After applying the activation function to a, we get the output of the neuron:

$$o_j = \text{activation function}(a_j)$$

2. Sigmoid Function

The sigmoid function returns a value between 0 and 1, introducing non-linearity into the model.

$$y_j = 1/1+e^{-a_j}$$



To find the outputs of y3, y4 and y5

3. Computing Outputs

At h1 node

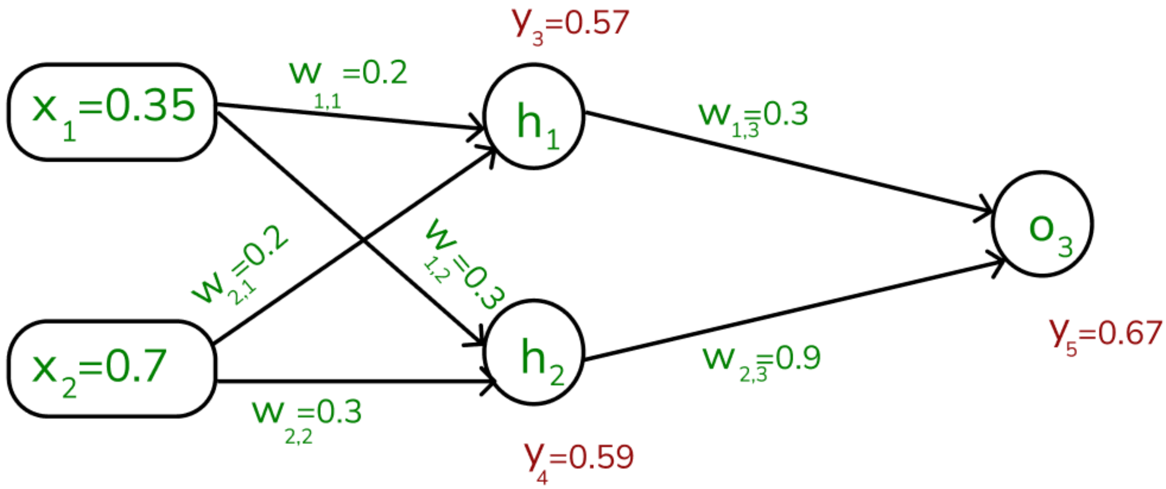
$$a_1 = (w_{1,1}x_1) + (w_{2,1}x_2)$$
$$= (0.2*0.35) + (0.2*0.7)$$
$$= 0.21$$

Once we calculated the a1 value, we can now proceed to find the y3 value:

$$y_j = F(a_j) = 1/1+e^{-a_1}$$
$$y_3 = F(0.21) = 1/1+e^{-0.21}$$
$$y_3 = 0.56$$

Similarly find the values of y4 at **h2** and y5 at O3

$$a_2 = (w_{1,2}*x_1) + (w_{2,2}*x_2) = (0.3*0.35) + (0.3*0.7) = 0.315$$
$$a_3 = (w_{1,3}*y_3) + (w_{2,3}*y_4) = (0.3*0.57) + (0.9*0.59) = 0.702$$
$$y_5 = F(0.702) = 1/1+e^{-0.702} = 0.67$$



Values of y3, y4 and y5

4. Error Calculation

Our actual output is 0.5 but we obtained 0.67. To calculate the error we can use the below formula:

$$Error_j = y_{target} - y_5$$

$$Error = 0.5 - 0.67 = -0.17$$

Using this error value we will be backpropagating.

Backpropagation

1. Calculating Gradients

The change in each weight is calculated as:

$\Delta w_{ij} = \eta \times \delta_j \times O_j$

Where:

- δ_j is the error term for each unit,
- η is the learning rate.

2. Output Unit Error

For O3:

$$\delta_5 = y_5(1 - y_5)(y_{target} - y_5)$$

$$= 0.67(1 - 0.67)(-0.17) = -0.0376$$

3. Hidden Unit Error

For h1:

$$\delta_3 = y_3(1 - y_3)(w_{1,3} \times \delta_5)$$

$$= 0.56(1 - 0.56)(0.3 \times -0.0376) = -0.0027$$

For h2:

$$\delta_4 = y_4(1 - y_4)(w_{2,3} \times \delta_5)$$

$$= 0.59(1 - 0.59)(0.9 \times -0.0376) = -0.0819$$

4. Weight Updates

For the weights from hidden to output layer:

$$\Delta w_{2,3}=1\times(-0.0376)\times0.59=-0.022184$$

New weight:

$$w_{2,3}(new)=-0.022184+0.9=0.877816$$

For weights from input to hidden layer:

$$\Delta w_{1,1}=1\times(-0.0027)\times0.35=0.000945$$

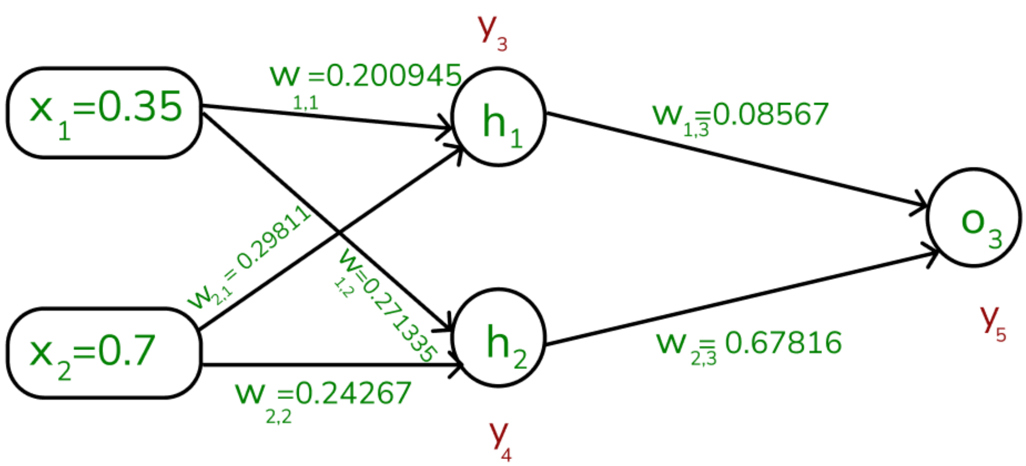
New weight:

$$w_{1,1}(new)=0.000945+0.2=0.200945$$

Similarly other weights are updated:

- $w_{1,2}(new)=0.273225$
- $w_{1,3}(new)=0.086615$
- $w_{2,1}(new)=0.269445$
- $w_{2,2}(new)=0.18534$

The updated weights are illustrated below



Through backward pass the weights are updated

After updating the weights the forward pass is repeated yielding:

- $y_3=0.57$
- $y_4=0.56$
- $y_5=0.61$

Since $y_5=0.61$ is still not the target output the process of calculating the error and backpropagating continues until the desired output is reached.

This process demonstrates how backpropagation iteratively updates weights by minimizing errors until the network accurately predicts the output.

$$\begin{aligned} Error &= y_{target}-y_5 \\ &= 0.5-0.61=-0.11 \end{aligned}$$

This process is said to be continued until the actual output is gained by the neural network.

Advantages of Backpropagation for Neural Network Training

The key benefits of using the backpropagation algorithm are:

1. **Ease of Implementation:** Backpropagation is beginner-friendly requiring no prior neural network knowledge and simplifies programming by adjusting weights with error derivatives.

2. **Simplicity and Flexibility:** Its straightforward design suits a range of tasks from basic feedforward to complex convolutional or recurrent networks.
3. **Efficiency:** Backpropagation accelerates learning by directly updating weights based on error especially in deep networks.
4. **Generalization:** It helps models generalize well to new data improving prediction accuracy on unseen examples.
5. **Scalability:** The algorithm scales efficiently with larger datasets and more complex networks making it ideal for large-scale tasks.

Challenges with Backpropagation

While backpropagation is powerful it does face some challenges:

1. **Vanishing Gradient Problem**: In deep networks the gradients can become very small during backpropagation making it difficult for the network to learn. This is common when using activation functions like sigmoid or tanh.
2. **Exploding Gradients**: The gradients can also become excessively large causing the network to diverge during training.
3. **Overfitting**: If the network is too complex it might memorize the training data instead of learning general patterns.

Backpropagation is a technique that makes neural network learn. By propagating errors backward and adjusting the weights and biases neural networks can gradually improve their predictions. Though it has some limitations like vanishing gradients many techniques like ReLU activation or optimizing learning rates have been developed to address these issues.

Introduction to Deep Learning

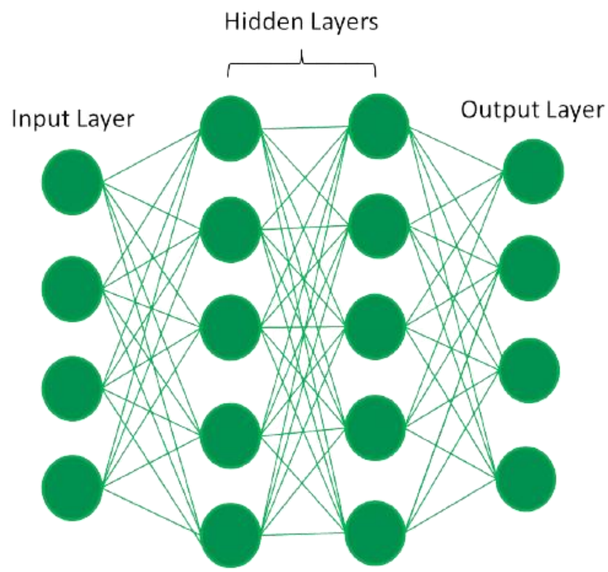
Deep Learning is transforming the way machines understand, learn, and interact with complex data. Deep learning mimics neural networks of the human brain, it enables computers to autonomously uncover patterns and make informed decisions from vast amounts of unstructured data.

Deep Learning leverages artificial neural networks (ANNs) to process and learn from complex data.

How Deep Learning Works?

Neural network consists of layers of interconnected nodes, or neurons, that collaborate to process input data. In a **fully connected deep neural network**, data flows through multiple layers, where each neuron performs nonlinear transformations, allowing the model to learn intricate representations of the data.

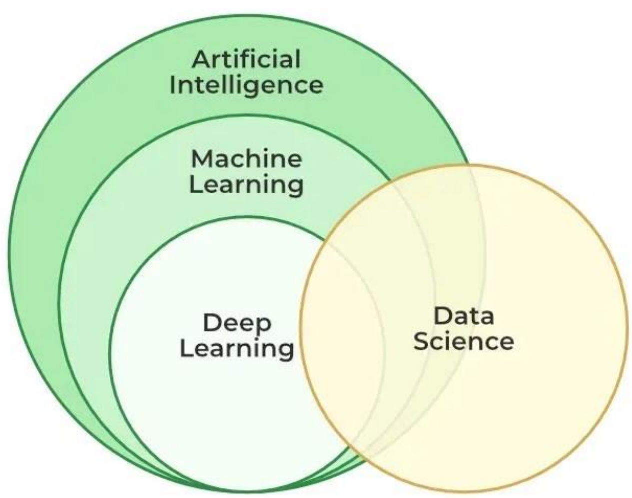
In a deep neural network, the **input layer** receives data, which passes through **hidden layers** that transform the data using nonlinear functions. The final **output layer** generates the model's prediction.



Fully Connected Deep Neural Network

Deep Learning in Machine Learning Paradigms

- **Supervised Learning**: Neural networks learn from labeled data to predict or classify, using algorithms like CNNs and RNNs for tasks such as image recognition and language translation.
- **Unsupervised Learning**: Neural networks identify patterns in unlabeled data, using techniques like Autoencoders and Generative Models for tasks like clustering and anomaly detection.
- **Reinforcement Learning**: An agent learns to make decisions by maximizing rewards, with algorithms like DQN and DDPG applied in areas like robotics and game playing.



Difference between Machine Learning and Deep Learning

Machine learning and Deep Learning both are subsets of artificial intelligence but there are many similarities and differences between them.

Machine Learning	Deep Learning
Apply statistical algorithms to learn the hidden patterns and relationships in the dataset.	Uses artificial neural network architecture to learn the hidden patterns and relationships in the dataset.
Can work on the smaller amount of dataset	Requires the larger volume of dataset compared to machine learning
Better for the low-label task.	Better for complex task like image processing, natural language processing, etc.
Takes less time to train the model.	Takes more time to train the model.
A model is created by relevant features which are manually extracted from images to detect an object in the image.	Relevant features are automatically extracted from images. It is an end-to-end learning process.
Less complex and easy to interpret the result.	More complex, it works like the black box interpretations of the result are not easy.
It can work on the CPU or requires less computing power as compared to deep learning.	It requires a high-performance computer with GPU.

Evolution of Neural Architectures

The journey of deep learning began with the **perceptron**, a single-layer neural network introduced in the 1950s. While innovative, perceptrons could only solve linearly separable problems, failing at more complex tasks like the XOR problem.
This limitation led to the development of **Multi-Layer Perceptrons (MLPs)**. It introduced hidden layers and non-linear activation functions. MLPs, trained using **backpropagation**, could model complex, non-linear relationships, marking a significant leap in neural network capabilities.
This evolution from perceptrons to MLPs laid the groundwork for advanced architectures like CNNs and RNNs, showcasing the power of layered structures in solving real-world problems.

Types of neural networks

1. Feedforward neural networks (FNNs) are the simplest type of ANN, where data flows in one direction from input to output. It is used for basic tasks like classification.
2. Convolutional Neural Networks (CNNs) are specialized for processing grid-like data, such as images. CNNs use convolutional layers to detect spatial hierarchies, making them ideal for computer vision tasks.
3. Recurrent Neural Networks (RNNs) are able to process sequential data, such as time series and natural language. RNNs have loops to retain information over time, enabling applications like language modeling and speech recognition. Variants like LSTMs and GRUs address vanishing gradient issues.
4. Generative Adversarial Networks (GANs) consist of two networks—a generator and a discriminator—that compete to create realistic data. GANs are widely used for image generation, style transfer, and data augmentation.
5. Autoencoders are unsupervised networks that learn efficient data encodings. They compress input data into a latent representation and reconstruct it, useful for dimensionality reduction and anomaly detection.
6. Transformer Networks has revolutionized NLP with self-attention mechanisms. Transformers excel at tasks like translation, text generation, and sentiment analysis, powering models like GPT and BERT.

Deep Learning Applications

1. Computer vision
In computer vision, deep learning models enable machines to identify and understand visual data. Some of the main applications of deep learning in computer vision include:

- **Object detection and recognition:** Deep learning models are used to identify and locate objects within images and videos, making it possible for machines to perform tasks such as self-driving cars, surveillance, and robotics.
- **Image classification:** Deep learning models can be used to classify images into categories such as animals, plants, and buildings. This is used in applications such as medical imaging, quality control, and image retrieval.
- **Image segmentation:** Deep learning models can be used for image segmentation into different regions, making it possible to identify specific features within images.

2. Natural language processing (NLP)
In NLP, deep learning model enable machines to understand and generate human language. Some of the main applications of deep learning in NLP include:

- **Automatic Text Generation:** Deep learning model can learn the corpus of text and new text like summaries, essays can be automatically generated using these trained models.
- **Language translation:** Deep learning models can translate text from one language to another, making it possible to communicate with people from different linguistic backgrounds.
- **Sentiment analysis:** Deep learning models can analyze the sentiment of a piece of text, making it possible to determine whether the text is positive, negative, or neutral.
- **Speech recognition:** Deep learning models can recognize and transcribe spoken words, making it possible to perform tasks such as speech-to-text conversion, voice search, and voice-controlled devices.

3. Reinforcement learning

In reinforcement learning, deep learning works as training agents to take action in an environment to maximize a reward. Some of the main applications of deep learning in reinforcement learning include:

- **Game playing:** Deep reinforcement learning models have been able to beat human experts at games such as Go, Chess, and Atari.
- **Robotics:** Deep reinforcement learning models can be used to train robots to perform complex tasks such as grasping objects, navigation, and manipulation.
- **Control systems:** Deep reinforcement learning models can be used to control complex systems such as power grids, traffic management, and supply chain optimization.

Challenges in Deep Learning

Deep learning has made significant advancements in various fields, but there are still some challenges that need to be addressed. Here are some of the main challenges in deep learning:

1. **Data availability**: It requires large amounts of data to learn from. For using deep learning it's a big concern to gather as much data for training.
2. **Computational Resources**: For training the deep learning model, it is computationally expensive because it requires specialized hardware like GPUs and TPUs.
3. **Time-consuming:** While working on sequential data depending on the computational resource it can take very large even in days or months.
4. I**nterpretability:** Deep learning models are complex, it works like a black box. it is very difficult to interpret the result.
5. **Overfitting:** when the model is trained again and again, it becomes too specialized for the training data, leading to overfitting and poor performance on new data.

Advantages of Deep Learning

1. **High accuracy:** Deep Learning algorithms can achieve state-of-the-art performance in various tasks, such as image recognition and natural language processing.
2. **Automated feature engineering:** Deep Learning algorithms can automatically discover and learn relevant features from data without the need for manual feature engineering.
3. **Scalability:** Deep Learning models can scale to handle large and complex datasets, and can learn from massive amounts of data.
4. **Flexibility:** Deep Learning models can be applied to a wide range of tasks and can handle various types of data, such as images, text, and speech.
5. **Continual improvement:** Deep Learning models can continually improve their performance as more data becomes available.

Disadvantages of Deep Learning

1. **High computational requirements:** Deep Learning AI models require large amounts of data and computational resources to train and optimize.
2. **Requires large amounts of labeled data**: Deep Learning models often require a large amount of labeled data for training, which can be expensive and time- consuming to acquire.
3. **Interpretability:** Deep Learning models can be challenging to interpret, making it difficult to understand how they make decisions.

   Overfitting: Deep Learning models can sometimes overfit to the training data, resulting in poor performance on new and unseen data.
4. **Black-box nature**: Deep Learning models are often treated as black boxes, making it difficult to understand how they work and how they arrived at their predictions.

As we continue to push the boundaries of computational power and dataset sizes, the potential applications of deep learning are limitless. Deep Learning promises to reshape our future, ushering in a new era where machines can learn, adapt, and solve complex problems at a scale and speed previously unimaginable.