# Introduction to Machine Learning

Machine learning is about extracting knowledge from the data. It can be defined as; “Machine learning is a subfield of artificial intelligence, which enables machines to learn from past data or experiences without being explicitly programmed.” As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn.

If you want your program to predict, for example, traffic patterns at a busy intersection (task T), you can run it through a machine learning algorithm with data about past traffic patterns (experience E) and, if it has successfully “learned”, it will then do better at predicting future traffic patterns (performance measure P).

The highly complex nature of many real-world problems, though, often means that inventing specialized algorithms that will solve them perfectly every time is impractical, if not impossible.

## Concepts of Learning

### Supervised

The computer is presented with example inputs and their desired outputs, given by a “teacher”, and the goal is to learn a general function that maps inputs to outputs. The training process continues until the model achieves the desired level of accuracy on the training data. Some real-life examples are:

* Image Classification: You train with images/labels. Then in the future you give a new image expecting that the computer will recognize the new object.
* Market Prediction/Regression: You train the computer with historical market data and ask the computer to predict the new price in the future.

### Unsupervised

In **unsupervised learning** the agent learns patterns in the input without any explicit feedback. No labels are given to the learning algorithm, leaving it on its own to find structure in its input. It is used for clustering population in different groups. Unsupervised learning can be a goal in itself (discovering hidden patterns in data).

* Clustering: You ask the computer to separate similar data into clusters, this is essential in research and science.
* High Dimension Visualization: Use the computer to help us visualize high dimension data.
* Generative Models: After a model captures the probability distribution of your input data, it will be able to generate more data. This can be very useful to make your classifier more robust.

### Semi-supervised learning

Problems where you have a large amount of input data and only some of the data is labeled, are called semi-supervised learning problems. These problems sit in between both supervised and unsupervised learning. For example, a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled.

### Reinforcement Learning

In this method, a computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). The program is provided feedback in terms of rewards and punishments as it navigates its problem space. This method aims at using observations from the interaction with the environment to take actions that would maximize the reward or minimize the risk. Reinforcement learning algorithm (called the agent) continuously learns from the environment in an iterative fashion. In the process, the agent learns from its experiences of the environment until it explores the full range of possible states.

In order to produce intelligent programs (also called agents), reinforcement learning goes through the following steps:

* Input state is observed by the agent.
* Decision making function is used to make the agent perform an action.
* After the action is performed, the agent receives reward or reinforcement from the environment.
* The state-action pair information about the reward is stored.

### Statistical-based Learning: Naive Bayes Model

Naive Bayes Model is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes classifier calculates the probability of an event in the following steps:

* Step 1: Calculate the prior probability for given class labels
* Step 2: Find Likelihood probability with each attribute for each class
* Step 3: Put these value in Bayes Formula and calculate posterior probability.
* Step 4: See which class has a higher probability, given the input belongs to the higher probability class.

# Learning by Genetic Algorithms

A genetic algorithm is an adaptive heuristic search algorithm inspired by "Darwin's theory of evolution in Nature." Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.

This algorithm is important because it solves difficult problems that would take a long time to solve.

It has been used in various real-life applications such as data centers, electronic circuit design, code-breaking, image processing, and artificial creativity.

This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation. In simple words, they simulate “survival of the fittest” among individual of consecutive generation for solving a problem.

### Notion of Natural Selection

The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found.

This notion can be applied for a search problem. We consider a set of solutions for a problem and select the set of best ones out of them.

Before understanding the Genetic algorithm, let's first understand basic terminologies to better understand this algorithm:

* **Population**: Population is the subset of all possible or probable solutions, which can solve the given problem.
* **Chromosomes**: A chromosome is one of the solutions in the population for the given problem, and the collection of gene generate a chromosome.
* **Gene**: A chromosome is divided into a different gene, or it is an element of the chromosome.
* **Allele**: Allele is the value provided to the gene within a particular chromosome.
* **Fitness** **Function**: The fitness function is used to determine the individual's fitness level in the population. It means the ability of an individual to compete with other individuals. In every iteration, individuals are evaluated based on their fitness function.
* **Genetic** **Operators**: In a genetic algorithm, the best individual mate to regenerate offspring better than parents. Here genetic operators play a role in changing the genetic composition of the next generation.
* **Selection**

### Five phases/operators in Genetic Algorithm

**Initialization**

Randomly generate a population with multiple chromosomes. Gene is the smallest unit and can be referred to as a set of characteristics (variables). We aim to join the Genes to obtain the Chromosomes(solution). The chromosome itself represents one candidate solution abstractly. The generation of Chromosome is user-defined (combination of numbers between 0 and 5 or only binary numbers).



**Fitness function**

The fitness function determines how fit an individual is (the ability of an individual to compete with other individuals). Each chromosome is assigned with a fitness score by the fitness function, which represents the goodness of the solution. Let’s say the fitness function is the sum of all the genes. Hence, the chromosome with the maximum sum is the fittest.

**Selection**

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation. Two pairs of individuals (parents) are selected based on their fitness scores for creating the next generation. These will act as parents to generate offspring for the next generation which will naturally inherit the strong features.

Here are some of the methods of parent selection-

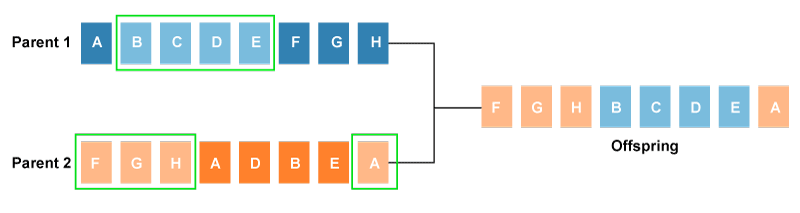
* Roulette Wheel Selection
* Rank Selection
* Steady State Selection
* Tournament Selection
* Elitism Selection
* Crossover
* Mutation

**Reproduction**

After the selection process, the creation of a child occurs in the reproduction step. In this step, the genetic algorithm uses two variation operators that are applied to the parent population. The two operators involved in the reproduction phase are Crossover and Mutation.

**Crossover**

For each pair of parents to be mated, a crossover point is chosen at random from within the genes. Then the crossover operator swaps genetic information of two parents from the current generation to produce a new individual representing the offspring.



The genes of parents are exchanged among themselves until the crossover point is met. These newly generated offspring are added to the population. This process is also called or crossover.

Types of crossover styles available:

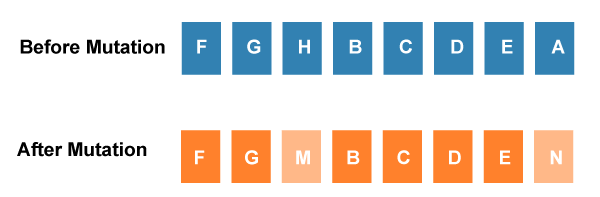
* One point crossover
* Two-point crossover
* Livery crossover
* Inheritable Algorithms crossover

**Mutation**

To avoid the duplicity(crossover generates offspring similar to parents) and to enhance the diversity in offspring we perform mutation. The mutation operator solves this problem by changing the value of some features in the offspring at random.

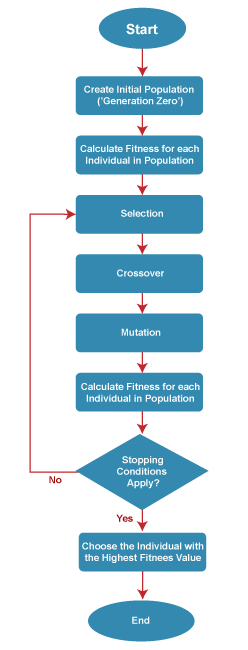
Types of mutation styles available,

* Flip bit mutation
* Gaussian mutation
* Exchange/Swap mutation



**Termination**

After the reproduction phase, a stopping criterion is applied as a base for termination. The algorithm terminates after the threshold fitness solution is reached. It will identify the final solution as the best solution in the population.



**Pseudocode**

START

Generate the initial population

Compute fitness

REPEAT

Selection

Crossover

Mutation

Compute fitness

UNTIL population has converged

STOP

When to apply Genetic Algorithm:

* There are multiple local optima
* The objective function is not smooth (so derivative methods cannot be applied)
* Number of parameters is very large
* Objective function is noisy or stochastic

**Advantages**:

* Concept is easy to understand
* Modular, separate from application
* Answer gets better with time
* Inherently parallel; easily distributed
* Genetic algorithms work on the Chromosome, which is an encoded version of potential solutions’ parameters, rather the parameters themselves.
* Genetic algorithms use fitness score, which is obtained from objective functions, without other derivative or auxiliary information

**Disadvantages**:

* Genetic Algorithms might be costly in computational terms since the evaluation of each individual requires the training of a model.
* These algorithms can take a long time to converge since they have a stochastic nature.

# Genetic Algorithm Learning with Neural Networks:

NNs have helped us solve so many problems. But there’s a huge problem that they still have. Hyperparameters! These are the only values that can not be learned… Until now.

Note: Hyper-parameters are values required by the NN to perform properly, given a problem.

We can use GAs to learn the best hyper-parameters for a NN. Now, we don’t have to worry about “knowing the right hyperparameters” since, they can be learned using a GA. Also, we can use this to learn the parameter’s (weights) of a NN as well.

Since, in a GA, the entities learn the optimum genome for the specified problem, here, the genome of each NN will be its set of hyper-parameters.

## Neural Network (NN):

A neural network is a collection of neurons that take input and, in conjunction with information from other nodes, develop output without programmed rules. Essentially, they solve problems through trial and error.

Neural networks are based on human and animal brains. While neural networks are advanced enough to beat human opponents at games like chess and Go, they lack the cognitive abilities of a human toddler and most animals.

## Artificial Neural Network (ANN):

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.

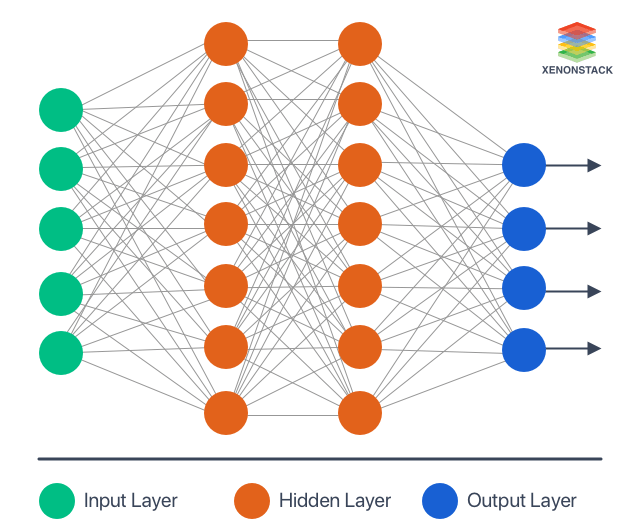
It is based on a Feed-Forward strategy. It is called this because they pass information through the nodes continuously till it reaches the output node. This is also known as the simplest type of neural network.

**Advantages**:

* Ability to learn irrespective of the type of data (Linear or Non-Linear).
* ANN is highly volatile and serves best in financial time series forecasting.

**Disadvantages**:

* The simplest architecture makes it difficult to explain the behavior of the network.
* This network is dependent on hardware.



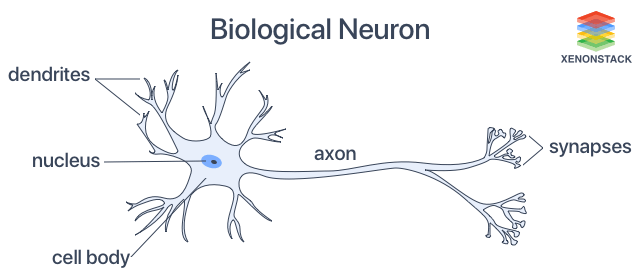
* **Input layer** - It contains those units (Artificial Neurons) which receive input from the outside world on which the network will learn, recognize about, or otherwise process.
* **Output layer** - It contains units that respond to the information about how it learn any task.
* **Hidden layer** - These units are in between input and output layers. The hidden layer's job is to transform the input into something that the output unit can use somehow.

## Deep Learning:

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Models are trained by using a large set of labeled data and neural network architectures that contain many layers. The word “deep” refers to the fact that the circuits are typically organized into many layers, which means that computation paths from inputs to outputs have many steps. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

## Biological Neural Network

Our brain has a large network of interlinked neurons, which act as a highway for information to be transmitted from point A to point B. To send different kinds of information from A to B, the brain activates a different sets of neurons, and so essentially uses a different route to get from A to B. This is how a typical neuron might look like.



At each neuron, its dendrites receive incoming signals sent by other neurons. If the neuron receives a high enough level of signals within a certain period of time, the neuron sends an electrical pulse into the terminals/synapes. These outgoing signals are then received by other neurons.

The typical nerve cell of the human brain comprises of four parts

* **Function of Dendrite** It receives signals from other neurons.
* **Soma (cell body)** It sums all the incoming signals to generate input.
* **Axon Structure** When the sum reaches a threshold value, the neuron fires, and the signal travels down the axon to the other neurons.
* **Synapses Working** The point of interconnection of one neuron with other neurons. The amount of signal transmitted depends upon the strength (synaptic weights) of the connections.

**Advantages**:

* The synapses are the input processing element.
* It is able to process highly complex parallel inputs.

**Disadvantages**:

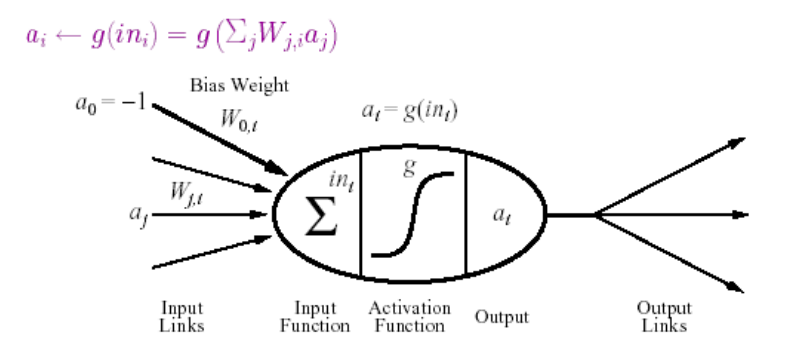
* There is no controlling mechanism.
* Speed of processing is slow being it complex.

## Biological Neural Networks Vs. Artificial Neural Networks (ANN)

|  |  |  |
| --- | --- | --- |
| BASIS FOR COMPARISON | ARTIFICIAL NEURAL NETWORK | BIOLOGICAL NEURAL NETWORK |
| Processing | Sequential and centralised | Parallel and distributed |
| Rate | Artificial neural networks process information in a faster pace. | Biological neurons are slow in processing information. |
| Size | Small | Large |
| Complexity | Incapable to perform complex pattern recognition. | The enormous size and complexity of the connections provide brain a capability of the performing complex tasks. |
| Fault tolerance | Intolerant to the failure. | Implicitly fault tolerant. |
| Control mechanism | Control unit monitors all computing-related activities. | All the processing is centrally controlled. |
| Feedback | Not provided | Provided |

## Mathematical Model of ANN

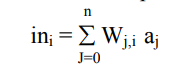
A simple mathematical model of neuron is devised by McCulloch and Pit is given in the figure given below:



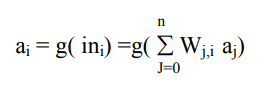
It fires when a linear combination of its inputs exceeds some threshold.

A neural network is composed of nodes (units) connected by directed links A link from unit j to i serve to propagate the activation aj from j to i. Each link has some numeric weight Wj,i associated with it, which determines strength and sign of connection.

Each unit first computes a weighted sum of it’s inputs:



Then it applies activation function g to this sum to derive the output:



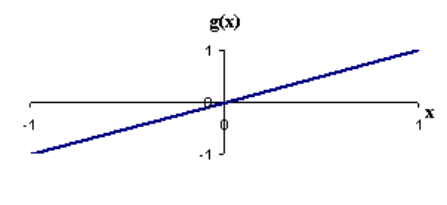
*Here, aj output activation from unit j and Wj,i is the weight on the link j to this node.*

Activation function typically falls into one of three categories:

* Linear
* Threshold (Heaviside function)
* Sigmoid
* Sign

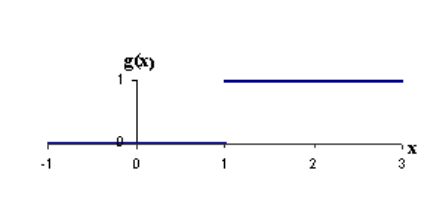
For linear activation functions, the output activity is proportional to the total weighted output.

*g(x) = k x + c, where k and x are constant*



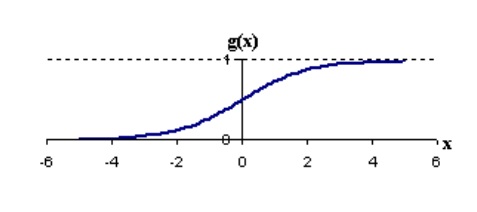
For threshold activation functions, the output are set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

*g(x) = 1 if x>= k = 0 if x < k*



For sigmoid activation functions, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units. It has the advantage of differentiable.

*g(x) = 1/ (1 + e-x)*



## Activation Functions

The neural network activation functions, in general, are used for determining the output of deep learning models, its accuracy, and performance efficiency of the training model that can design or divide a huge scale neural network.

* They basically decide to activate or deactivate neurons to get the desired output.
* It decides in any neural network that given input or receiving information is relevant or it is irrelevant.
* It also performs a nonlinear transformation on the input to get better results on a complex neural network.
* It helps to normalize the output of any input in the range between 1 to -1.
* It must be efficient and it should reduce the computation time because the neural network sometimes trained on millions of data points.

### Need for Activation Functions

1. To make the neural network **non-linear.**

Without activation function, weight and bias would only have a linear transformation, or neural network is just a linear regression model. A linear equation is polynomial of one degree only which is simple to solve but limited in terms of ability to solve complex problems or higher degree polynomials.

But opposite to that, the addition of activation function to neural network executes the non-linear transformation to input and make it capable to solve complex problems such as language translations and image classifications.

1. To make the neural network **differential**.

Activation functions are differentiable due to which they can easily implement back propagations, optimized strategy while performing backpropagations to measure gradient loss functions in the neural networks.

### Types of Activation Functions

1. **Linear or Identity Activation Function**

As you can see the function is a line or linear. Therefore, the output of the functions will not be confined between any range.

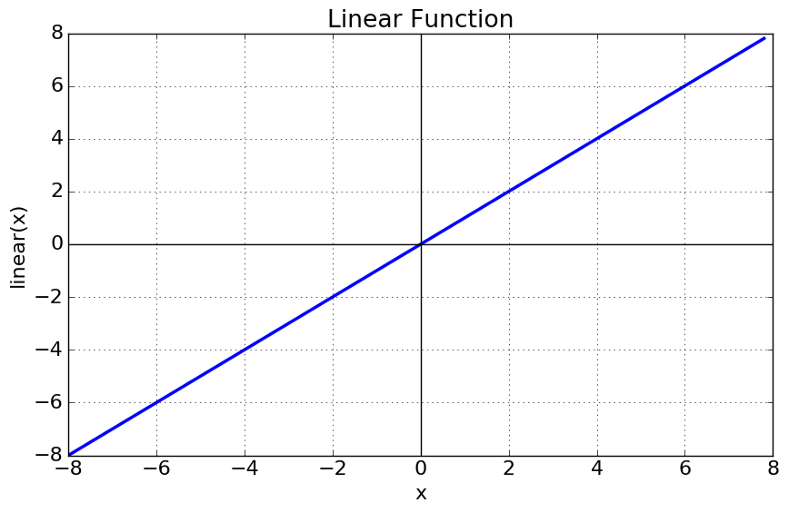


Fig: Linear Activation Function

**Equation** : f(x) = x

**Range** : (-infinity to infinity)

It doesn’t help with the complexity or various parameters of usual data that is fed to the neural networks.

1. **Non-linear Activation Function**

The Nonlinear Activation Functions are the most used activation functions. Nonlinearity helps to makes the graph look something like this



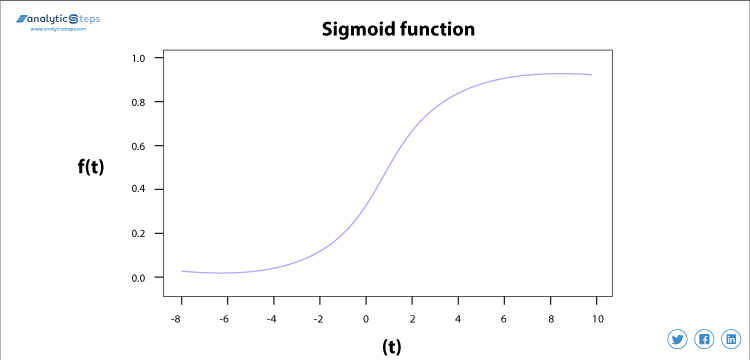
Fig: Non-linear Activation Function

It makes it easy for the model to generalize or adapt with variety of data and to differentiate between the output.

The Nonlinear Activation Functions are mainly divided on the basis of their range or curves-

1. **Sigmoid or Logistic Activation Function**

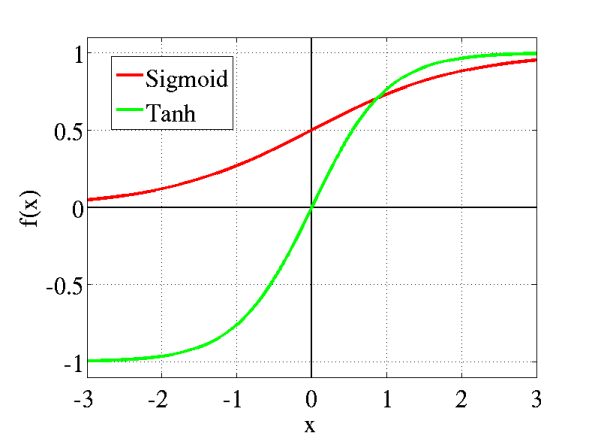
The Sigmoid Function curve looks like a S-shape.



* The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output.
* The logistic sigmoid function can cause a neural network to get stuck at the training time.
* The softmax function is a more generalized logistic activation function which is used for multiclass classification.
* The function is differentiable. That means, we can find the slope of the sigmoid curve at any two points.
* The function is monotonic but function’s derivative is not.

1. **Tanh or hyperbolic tangent Activation Function**

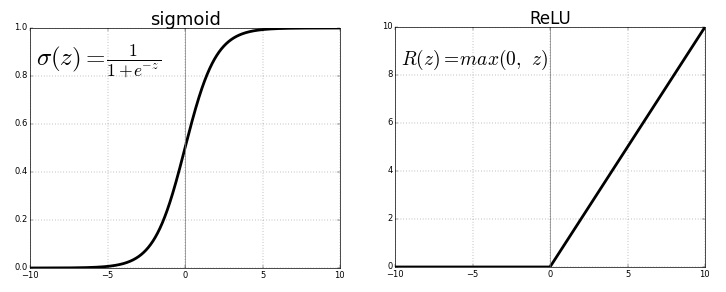
tanh is also like logistic sigmoid but better.



* The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped).
* The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph.
* The tanh function is mainly used classification between two classes.
* The function is differentiable.
* The function is monotonic but function’s derivative is not.

1. **ReLU (Rectified Linear Unit) Activation Function**

The ReLU is the most used activation function in the world right now. Since, it is used in almost all the convolutional neural networks or deep learning.



* Chiefly implemented in hidden layers of Neural network.
* **Equation :-** A(x) = max(0,x). It gives an output x if x is positive and 0 otherwise.
* Value Range :- [0, inf)
* **Nature :-** non-linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.
* **Uses :-** ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. At a time only a few neurons are activated making the network sparse making it efficient and easy for computation.
* Learns much faster than sigmoid and Tanh function.
* The function and its derivative both are monotonic.

## Types of ANN:

### Feed-forward:

This neural network is one of the simplest forms of ANN, where the data or the input travels in one direction. The data passes through the input nodes and exit on the output nodes. In simple words, it is the simplest form of neural network as information is only processed in one direction. While the data may pass through multiple hidden nodes, it always moves in one direction and never backwards.

* May or may not have hidden layers.
* Number of layers depends on the complexity of the function.
* It has uni-directional forward propagation but no backward propagation.
* Weights are static here.
* An activation function is fed by inputs which are multiplied by weights.
* Classifying activation function or step activation function is used.

**Advantages**:

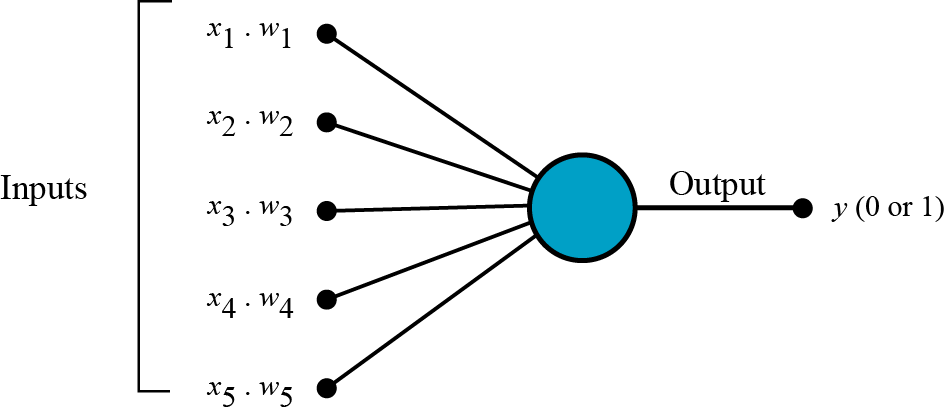
* Less complex, easy to design & maintain
* Fast and speedy [One-way propagation]
* Highly responsive to noisy data

**Disadvantages**:

* Cannot be used for deep learning [due to absence of dense layers and back propagation]

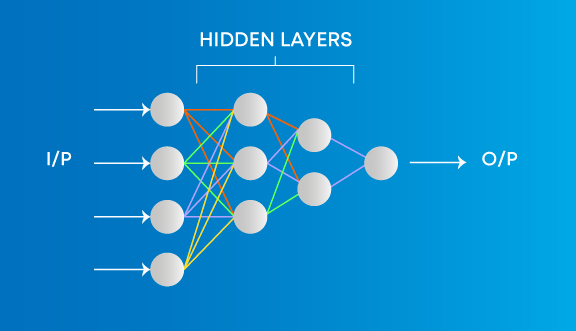
**Single Layer Feedforward Neural Network**

In Single layer feed-forward network, the sum of the products of inputs and weights are calculated and fed to the output. The output is considered if it is above a certain value i.e threshold(usually 0) and the neuron fires with an activated output (usually 1) and if it does not fire, the deactivated value is emitted (usually -1).



A Feed Forward Neural Network is commonly seen in its simplest form as a single layer perceptron. In this model, a series of inputs enter the layer and are multiplied by the weights. Each value is then added together to get a sum of the weighted input values. If the sum of the values is above a specific threshold, usually set at zero, the value produced is often 1, whereas if the sum falls below the threshold, the output value is -1.

**Multi-Layered Feedforward Neural Network**



A multi-layer neural network contains more than one layer of artificial neurons or nodes. They differ widely in design. It is important to note that while single-layer neural networks were useful early in the evolution of AI, the vast majority of networks used today have a multi-layer model. Basically, by adding more hidden layers / more neurons per layer you add more parameters to the model. Hence you allow the model to fit more complex functions.

Examples of Multi layer Neural Network are:

* Convolutional neural networks (CNNs)
* Recurrent neural networks,
* Deep networks and deep belief systems

**Advantages**:

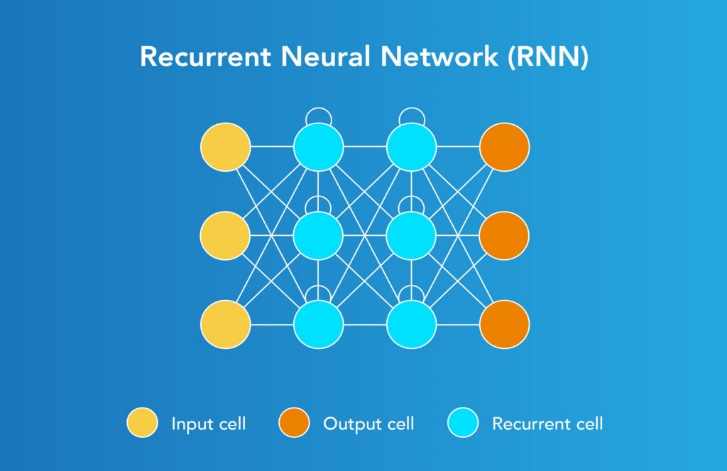
* Used for deep learning [due to the presence of dense fully connected layers and back propagation]

**Disadvantages**:

* Comparatively complex to design and maintain
* Comparatively slow (depends on number of hidden layers)

### Recurrent Neural Network (RNN)

Recurrent neural networks (RNN) are the state of the art algorithm for sequential data and are used by Apple's Siri and and Google's voice search. It is the first algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data.



Designed to save the output of a layer, Recurrent Neural Network is fed back to the input to help in predicting the outcome of the layer. The first layer is typically a feed forward neural network followed by recurrent neural network layer where some information it had in the previous time-step is remembered by a memory function. Forward propagation is implemented in this case. It stores information required for it’s future use. If the prediction is wrong, the learning rate is employed to make small changes. Hence, making it gradually increase towards making the right prediction during the backpropagation.

**Advantages**:

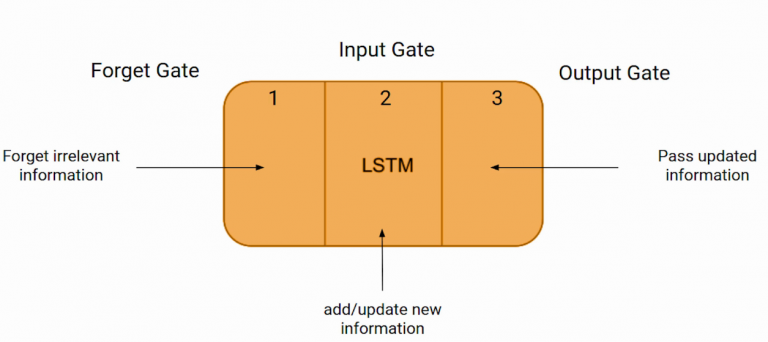
* Model sequential data where each sample can be assumed to be dependent on historical ones is one of the advantage.
* Used with convolution layers to extend the pixel effectiveness.

**Disadvantages**:

* Gradient vanishing and exploding problems
* Training recurrent neural nets could be a difficult task
* Difficult to process long sequential data using ReLU as an activation function.

### Improvement over RNN: LSTM (Long Short-Term Memory) Networks

LSTM networks are a type of RNN that uses special units in addition to standard units.



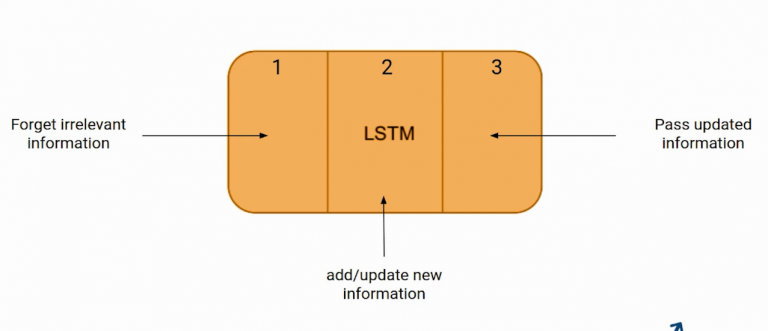
LSTM units includes the following components:

* A ‘memory cell’ that can maintain information in memory for long periods of time.
* A set of gates, used to control when information enters the memory when it’s output, and when it’s forgotten.
* There are three types of gates - Input gate, output gate and forget gate.
  + Input gate decides how many information from the last sample will be kept in memory;
  + the output gate regulates the amount of data passed to the next layer, and
  + forget gates control the tearing rate of memory stored.

**Working**:

Here, the first layer is formed similar to the feed forward neural network with the product of the sum of the weights and the features. The recurrent neural network process starts once this is computed, this means that from one time step to the next each neuron will remember some information it had in the previous time-step.

This makes each neuron act like a memory cell in performing computations. In this process, we need to let the neural network to work on the front propagation and remember what information it needs for later use. Here, if the prediction is wrong we use the learning rate or error correction to make small changes so that it will gradually work towards making the right prediction during the back propagation.



* The first part chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten.
* In the second part, the cell tries to learn new information from the input to this cell.
* At last, in the third part, the cell passes the updated information from the current timestamp to the next timestamp.

**Example**:

Bob is a nice person. Dan, on the Other hand, is evil.

The first sentence is “Bob is a nice person” and the second sentence is “Dan, on the Other hand, is evil”. It is very clear, in the first sentence we are talking about Bob and as soon as we encounter the full stop(.) we started talking about Dan.

As we move from the first sentence to the second sentence, our network should realize that we are no more talking about Bob. Now our subject is Dan. Here, the Forget gate of the network allows it to forget about it. Let’s understand the roles played by these gates in LSTM architecture.

# Learning by Training ANN

**Learning:** One of the powerful features of neural networks is learning. Learning in neural networks is carried out by adjusting the connection weights among neurons. It is similar to a biological nervous system in which learning is carried out by changing synapses connection strengths, among cells.

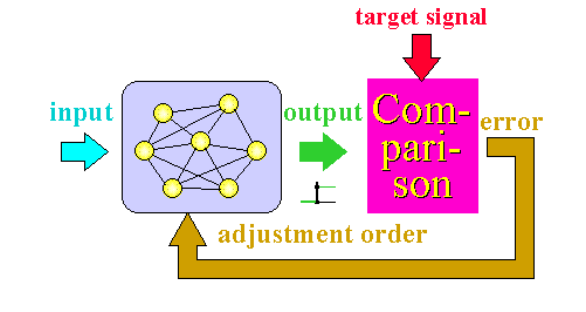
The operation of a neural network is determined by the values of the interconnection weights. There is no algorithm that determines how the weights should be assigned in order to solve specific problems. Hence, the weights are determined by a learning process.

Learning may be classified into two categories:

* Supervised Learning
* Unsupervised Learning

## Supervised Learning

In supervised learning, the network is presented with inputs together with the target (teacher signal) outputs. Then, the neural network tries to produce an output as close as possible to the target signal by adjusting the values of internal weights.



Supervised learning is classified into two categories of algorithms:

* **Classification**: A classification problem is when the output variable is a category, such as “Red” or “blue” , “disease” or “no disease”.
* **Regression:** A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Supervised learning deals with or learns with “labeled” data. This implies that some data is already tagged with the correct answer.

The most common supervised learning method is the “**error correction method**”.

Error correction method is used for networks which their neurons have discrete output functions. Neural networks are trained with this method in order to reduce the error (difference between the network's output and the desired output) to zero.

**Advantages**:

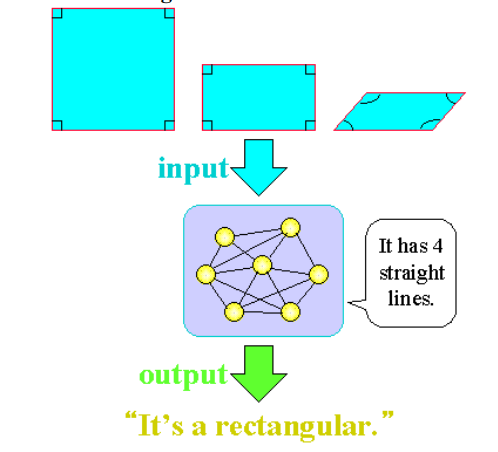
* Supervised learning allows collecting data and produces data output from previous experiences.
* Helps to optimize performance criteria with the help of experience.
* Supervised machine learning helps to solve various types of real-world computation problems.

**Disadvantages**:

* Classifying big data can be challenging.
* Training for supervised learning needs a lot of computation time. So, it requires a lot of time.

## Unsupervised Learning

In unsupervised learning, there is no teacher (target signal) from outside and the network adjusts its weights in response to only the input patterns. A typical example of unsupervised learning is Hebbian learning.



Unsupervised learning algorithms allow you to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning deep learning and reinforcement learning methods.

Unsupervised learning is classified into two categories of algorithms:

* Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
* Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

**Why Supervised Learning?**

* Supervised learning allows you to collect data or produce a data output from the previous experience.
* Helps you to optimize performance criteria using experience
* Supervised machine learning helps you to solve various types of real-world computation problems.

**Why Unsupervised Learning?**

* Unsupervised machine learning finds all kind of unknown patterns in data.
* Unsupervised methods help you to find features which can be useful for categorization.
* It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.
* It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

|  |  |  |
| --- | --- | --- |
| Parameters | Supervised | Unsupervised |
| Process | In a supervised learning model, input and output variables will be given. | In unsupervised learning model, only input data will be given |
| Input Data | Algorithms are trained using labeled data. | Algorithms are used against data which is not labeled |
| Algorithms Used | Support vector machine, Neural network, Linear and logistics regression, random forest, and Classification trees. | Unsupervised algorithms can be divided into different categories: like Cluster algorithms, K-means, Hierarchical clustering, etc. |
| Computational Complexity | Supervised learning is a simpler method. | Unsupervised learning is computationally complex |
| Use of Data | Supervised learning model uses training data to learn a link between the input and the outputs. | Unsupervised learning does not use output data. |
| Accuracy of Results | Highly accurate and trustworthy method. | Less accurate and trustworthy method. |
| Real Time Learning | Learning method takes place offline. | Learning method takes place in real time. |
| Number of Classes | Number of classes is known. | Number of classes is not known. |
| Main Drawback | Classifying big data can be a real challenge in Supervised Learning. | You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known. |

## Transfer Learning

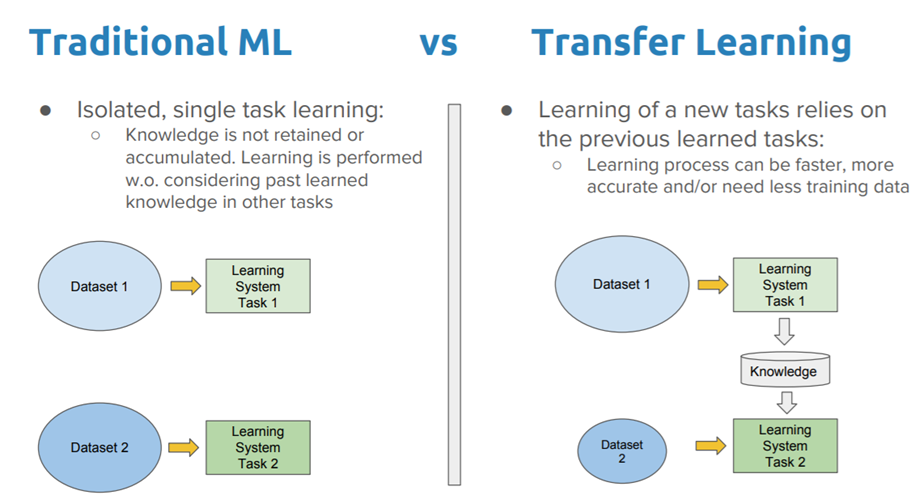
Transfer learning is a machine learning method where a model developed for one task is reused as the starting point for a model on another task.

Two common approaches are as follows:

* Develop Model Approach
* Pre-trained Model Approach

To put it simply—a model trained on one task is repurposed on a second, related task as an optimization that allows rapid progress when modeling the second task.

By applying transfer learning to a new task, one can achieve significantly higher performance than training with only a small amount of data.



## Hebbian Learning

Hebbian Learning Rule, also known as Hebb Learning Rule, was proposed by Donald O Hebb. It is one of the oldest and also most famous and easiest learning rules in neural network.

* It is used for pattern classification.
* It is a single layer neural network, i.e. it has one input layer and one output layer.
* The input layer can have many units, say n. The output layer only has one unit.

Hebbian rule works by updating the weights between neurons in the neural network for each training sample.

The basic principle of Hebbian rule is;

***“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A’s efficiency as one of the cells firing B is increased.”***

From the point of view of artificial neurons and artificial neural networks, Hebb's principle can be described as;

***A method of determining how to alter the weights between model neurons. The weight between two neurons increases if the two neurons activate simultaneously—and reduces if they activate separately. Nodes that tend to be eitherboth positive or both negative at the same time have strong positive weights, while those that tend to be opposite have strong negative weights.***

**Hebb’s Algorithm:**

Step 0: initialize all weights to 0

Step 1: Given a training input, s, with its target output, t, set the activations of the input units: xi = si

Step 2: Set the activation of the output unit to the target value: y = t

Step 3: Adjust the weights:

wi(new) = wi(old) + xiy

Step 4: Adjust the bias (just like the weights):

b(new) = b(old) + y

## Perceptron Learning

The term "Perceptron" was coined by Frank RosenBlatt in 1962 and is used to describe the connection of simple neurons into networks. These networks are simplified versions of the real nervous system where some properties are exaggerated and others are ignored.

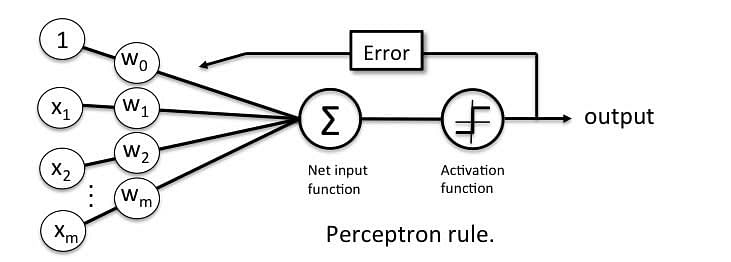
* Perceptron is a linear Machine Learning algorithm used for supervised learning for various binary classifiers.
* This algorithm enables neurons to learn and processes elements in the training set one at a time.
* There are two types of Perceptrons: Single layer and Multilayer.

**Single layer -** Single layer perceptrons can learn only linearly separable patterns

**Multilayer -** Multilayer perceptrons or feedforward neural networks with two or more layers have the greater processing power

**Perceptron Learning Rule**

Perceptron Learning Rule states that the algorithm would automatically learn the optimal weight coefficients. The input features are then multiplied with these weights to determine if a neuron fires or not.



The Perceptron receives multiple input signals, and if the sum of the input signals exceeds a certain threshold, it either outputs a signal or does not return an output. In the context of supervised learning and classification, this can then be used to predict the class of a sample.

**Perceptron Function**

Perceptron is a function that maps its input “x,” which is multiplied with the learned weight coefficient; an output value ”f(x)”is generated.

Perceptron_6.

In the equation given above:

* “w” = vector of real-valued weights
* “b” = bias (an element that adjusts the boundary away from origin without any dependence on the input value)
* “x” = vector of input x values

i.e. “x” = Perceptron_7.

* “m” = number of inputs to the Perceptron

The output can be represented as “1” or “0.” It can also be represented as “1” or “-1” depending on which activation function is used.

**Algorithm:**

1. Initialize weights and threshold.

Set *wi(t), (0 <= i <= n)*, to be the weight *i* at time *t*, and *ø* to be the threshold value in the output node. Set *w0* to be *-ø*, the bias, and *x0* to be always 1.

Set *wi(0)* to small random values, thus initializing the weights and threshold.

1. Present input and desired output

Present input *x0, x1, x2, ..., xn* and desired output *d(t)*

1. Calculate the actual output

*y(t) = g [w0(t)x0(t) + w1(t)x1(t) + .... + wn(t)xn(t)]*

1. Adapts weights

*wi(t+1) = wi(t) + α[d(t) - y(t)]xi(t)* ,

where 0 <= α <= 1 (learning rate) is a positive gain function that controls the adaption rate.

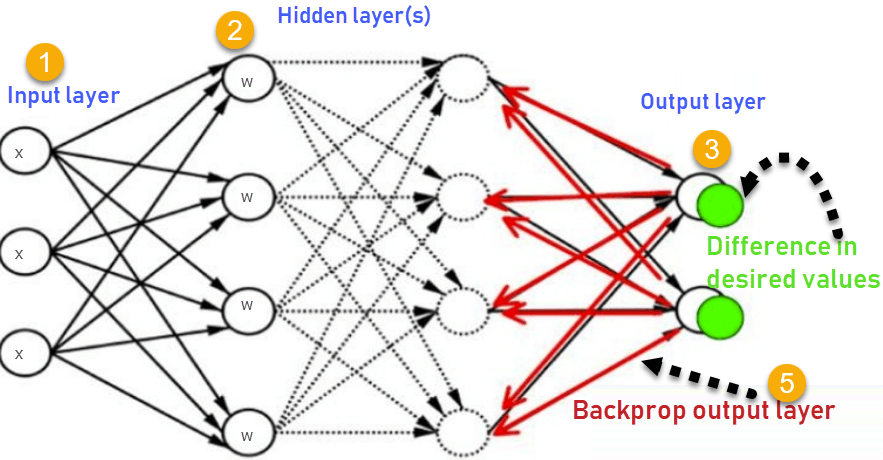
Steps iii. and iv. are repeated until the iteration error is less than a user-specified error threshold or a predetermined number of iterations have been completed.

*Please note that the weights only change if an error is made and hence this is only when learning shall occur.*

**Limitations:**

* 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.

## Back - Propagation Learning

****

Backpropagation in neural network is a short form for “backward propagation of errors”.

As the algorithm's name implies, the errors (and therefore the learning) propagate backwards from the output nodes to the inner nodes. So technically speaking, backpropagation is used to calculate the gradient of the error/loss of the network with respect to all the weights in the network.

Backpropagation is a supervised learning algorithm, for training Multi-layer Perceptrons.

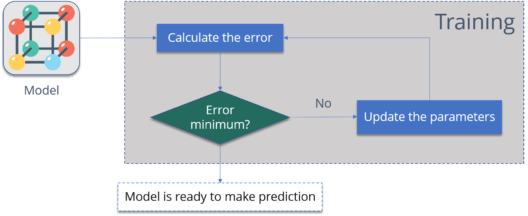
In order for the hidden layer to serve any useful function, multilayer networks must have non-linear activation functions for the multiple layers: a multilayer network using only linear activation functions is equivalent to some single layer, linear network.

**Need for Backpropagation**

While designing a Neural Network, in the beginning, we initialize weights with some random values or any variable for that fact. Now obviously, we are not superhuman. So, it’s not necessary that whatever weight values we have selected will be correct, or it fits our model the best. Even when we select some weight values in the beginning, but our model output can be way different than our actual output i.e. the error value can be huge.

To reduce this huge error value, we need to somehow explain the model to change the parameters (weights), such that error becomes minimum. That means we need to train our model.

One way to train our model is called as Backpropagation. Consider the diagram below:



* Calculate the error – How far is your model output from the actual output.
* Minimum Error – Check whether the error is minimized or not.
* Update the parameters – If the error is huge then, update the parameters (weights and biases). After that again check the error. Repeat the process until the error becomes minimum.
* Model is ready to make a prediction – Once the error becomes minimum, you can feed some inputs to your model and it will produce the output.

**Summary of the backpropagation technique:**

* Present a training sample to the neural network.
* Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
* For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
* Adjust the weights of each neuron to lower the local error.
* Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
* Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.

# Application of Artificial Neural Networks

**Personalized Online Shopping**

The latest artificial intelligence applications use AI-powered algorithms to curate the list of buying recommendations and filterations for the users.

They first collect the user data by going through the user’s most recent search history. Then, these AI algorithms create a list of the products that fit the criteria of being useful or similar, that the users might be interested in looking at and potentially buying in similar and different categories.

**Smart Cars**

From Google’s self-driving car project to Tesla’s “autopilot” feature, it is a matter of time before AI is a standard-issue technology in the automotive industry.

Advanced Deep Learning algorithms can accurately predict what objects in the vehicle’s vicinity are likely to do. The AI system collects data from the vehicle’s radar, cameras, GPS, and cloud services to produce control signals that operate the vehicle. With the evolution of AI, soon enough, fully automated vehicles will be seen on most streets.

**Marketing**

With the growing advancement in AI, in the near future, it may be possible for consumers on the web to buy products by snapping a photo of it. Companies like CamFind and their competitors are experimenting with this idea already.

Big Data and Machine Learning have been the major players in the domain where AI has shined and effectively elevated the various processes involved in handling data.

**Enhanced Images**

AI can help in object identification in images and also enhance the photograph to the maximum extent by identifying the depth, lighting, and scope of the picture, and helping capture every element in as much detail as possible. By using this feature, many apps and cameras let you add a variety of effects in your pictures.

Moreover, Google Photos also uses AI to let users look up photos of particular people in their contact lists or tags.

**Social Media**

The most common use of AI in social media is for face verification and to detect facial features. The machine learning algorithms are used to design your feed based on your interests.

**Surveillance**

AI can also be trained using supervised exercises, developing security algorithms, identification protocols, and much more, to take input from security cameras. Eventually, AI can identify potential threats and warn human security officers to investigate further.

**Agriculture**

The latest artificial intelligence applications in the form of image recognition identify possible defects in the crops through images captured by the user’s smartphone camera. Users are then provided with soil restoration techniques, tips, and other possible solutions to deal with the identified defects.

**Customer Service**

There are many websites now that offer customers the ability to chat with customer support. The more advanced customer service chatbots are able to extract information from the site and present it to you on request. Chatbots are needed to adapt as per the natural language.

**Healthcare**

AI technology boils down to crunch numbers fast and learn from historical data, which is critical in the healthcare industry.

AI has taken a critical step in helping people with looking after patients as well. The automated bots and healthcare applications ensure proper medication and treatment of patients in the facilities.

In certain cases, AI applications have also been known to provide operating assistance to the doctors.

**Smart Homes**

Thermostats and building management systems can help automate building heating and cooling, for instance. In effect, they learn and can predict when to turn your boiler on or off for optimal comfort, whilst factoring in outside weather conditions as well.