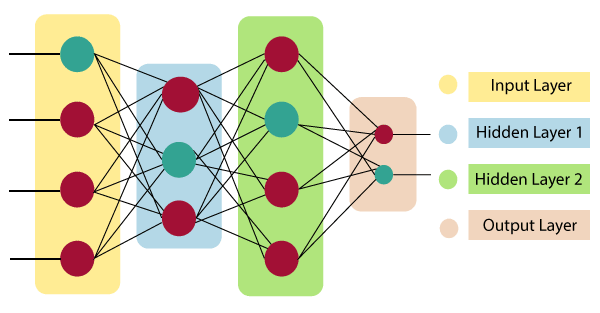
**FSDS MAY BATCH 2022(DL Assignment -2)**

**Submitted by: Shubham Tiwari.**

Q1: Describe the structure of an artificial neuron. How is it similar to a biological neuron? What are its main components?

Ans: [**Artificial Neural Network**](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/)**:** Artificial Neural Network (ANN) is a type of neural network which 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.

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.

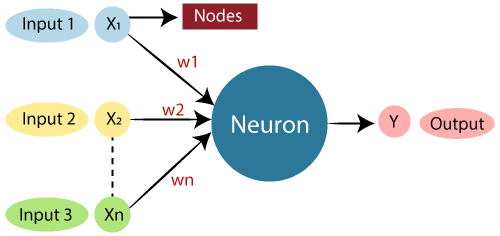
**Output Layer:**

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.

What is Artificial Neural Network

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.



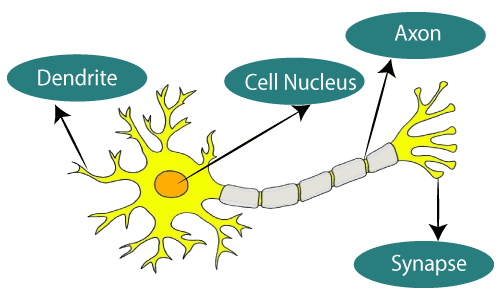
**Some advantages of ANN :**

* 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.

**Some disadvantages of ANN :**

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

2. **Biological Neural Network** : Biological Neural Network (BNN) is a structure that consists of Synapse, dendrites, cell body, and axon. In this neural network, the processing is carried out by neurons. Dendrites receive signals from other neurons, Soma sums all the incoming signals and axon transmits the signals to other cells.



**Some advantages of BNN** :

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

**Some disadvantages of BNN :**

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

Q2: What are the different types of activation functions popularly used? Explain each of them.

Ans: **The activation function** decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. The purpose of the activation function is to introduce non-linearity into the output of a neuron. There are various types of activation functions such as :

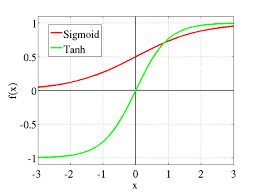
**1) Sigmoid activation function.**



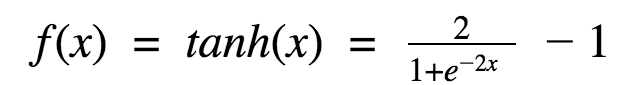
* It is a function which is plotted as **‘S’** shaped graph.
* Usually used in output layer of a binary classification, where result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be 1 if value is greater than 0.5 and 0 otherwise.
* The only disadvantage is that it is prone to vanishing gradient problem.

2)**Tanh activation function.**

* The activation that works almost always better than sigmoid function is Tanh function also knows as Tangent Hyperbolic function. It’s actually mathematically shifted version of the sigmoid function. Both are similar and can be derived from each other.
* In forward propogation the value of tanh ranges from -1 to 1 while the derivate ranges from 0 to 1.

****

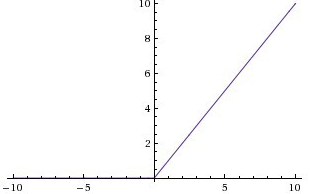
* The **advantage** is that it is zero-centred i.e efficient updation happening.
* The **disadvantage** is that time complexity with respect to finding out the derivative and vanishing gradient problem still exist for a very deep layered neural network.



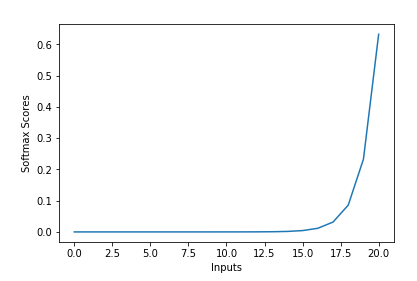
**3) Relu-Activation function.**

Relu = **max**(o , x)

* It is going to solve vanishing gradient problem but there is a issue that if the derivative is zero then it will lead to dead neuron.
* It is not zero-centric.
* It is a fast process compared to tanh and sigmoid function and is linear.



**4) Softmax Activation function.**



* The softmax function is also a type of sigmoid function but is handy when we are trying to handle multi- class classification problems.

**5) Leaky Relu and parametric Relu.**

* It prevents dead neuron i.e the derivative will never become zero.

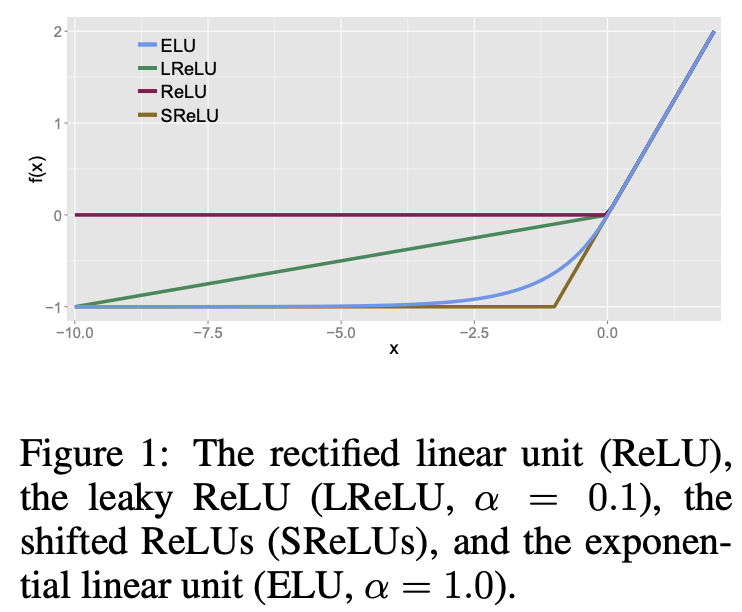
F(x)= max(0.01x , x)

* It is non zero centric but it will take time to find global minima.
* The difference between leaky relu and parametric Relu is that in parametric Relu 0.01 is a kind of hyperparameter.

F(x) =max (αx , x )

**6) ELU (Exponential Linear units).**

* + It is zero-centred ,preventing dead neuron and is linear.
  + There is one problem that equation is computational so it is time consuming.
  + Vanishing gradient will never occur.



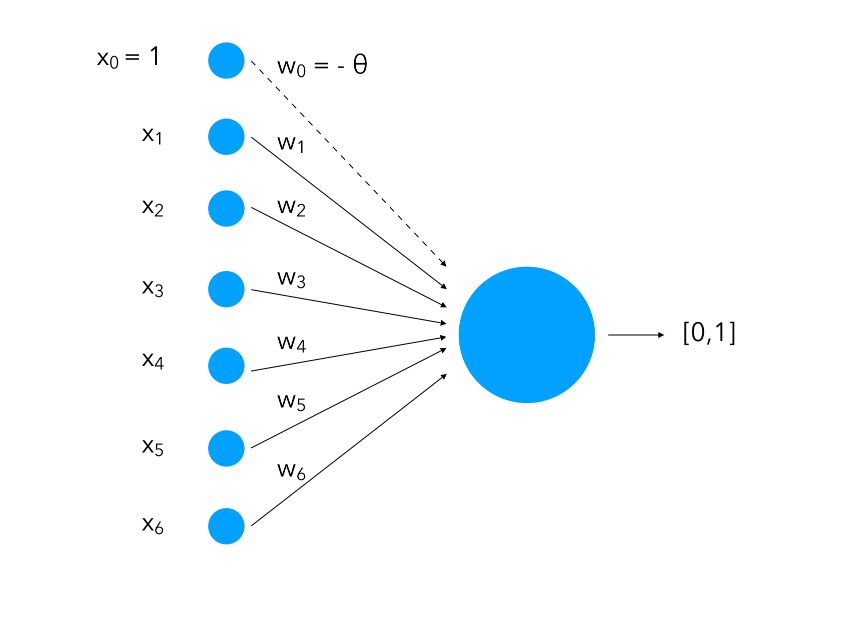
Q3: Explain, in details, Rosenblatt’s perceptron model. How can a set of data be classified using a simple perceptron?

Ans: **Rosenblatt perceptron** is a binary single neuron model. The inputs integration is implemented through the addition of the weighted inputs that have fixed weights obtained during the training stage. If the result of this addition is larger than a given threshold θ the neuron fires. When the neuron fires its output is set to 1, otherwise it’s set to 0.

This model implements the functioning of a single neuron that can solve linear classification problems through very simple learning algorithms. Rosenblatt Perceptrons are considered as the first generation of neural networks (the network is only compound of one neuron ). This simple single neuron model has the main limitation of not being able to solve non-linear separable problems.

The Rosenblatt’s Perceptron was designed to overcome most issues of the McCulloch-Pitts neuron :

* it can process non-boolean inputs.
* and it can assign different weights to each input automatically.
* the threshold θ is computed automatically.



We attach to each input a weight ( wi) and notice how we add an input of value 1 with a weight of −θ−. This is called bias. What we are doing is instead of having only the inputs and the weight and compare them to a threshold, we also learn the threshold as a weight for a standard input of value 1.

The inputs can be seen as neurons and will be called the **input layer**. Altogether, these neurons and the function (which we’ll cover in a minute) form a **perceptron**.

How do we make classification using a perceptron then?

y=1 if ∑iwixi≥0 else y=0

One limitation remains: the inputs need to be linearly separable since we split the input space into two halves.

Q4: Use a simple perceptron with weights w0 , w1 , and w2  as −1, 2, and 1, respectively, to classify data points (3, 4); (5, 2); (1, −3); (−8, −3); (−3, 0).

Ans:xxxxxxxxxxxxxx

Q5: Explain the basic structure of a multi-layer perceptron. Explain how it can solve the XOR problem.

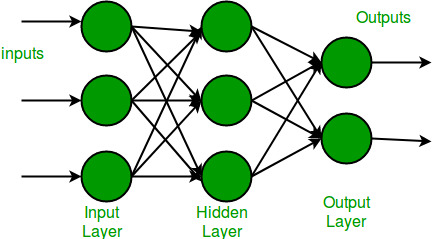
Ans: **Multi-layer Perceptron .**

Multi-layer perception is also known as MLP. It is fully connected dense layers, which transform any input dimension to the desired dimension. A multi-layer perception is a neural network that has multiple layers. To create a neural network we combine neurons together so that the outputs of some neurons are inputs of other neurons.

A gentle introduction to **neural networks and TensorFlow** can be found here:

* [Neural Networks](https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/)
* [Introduction to TensorFlow](https://www.geeksforgeeks.org/introduction-to-tensorflow/)

A multi-layer perceptron has one input layer and for each input, there is one neuron(or node), it has one output layer with a single node for each output and it can have any number of hidden layers and each hidden layer can have any number of nodes. A schematic diagram of a Multi-Layer Perceptron (MLP) is depicted below.

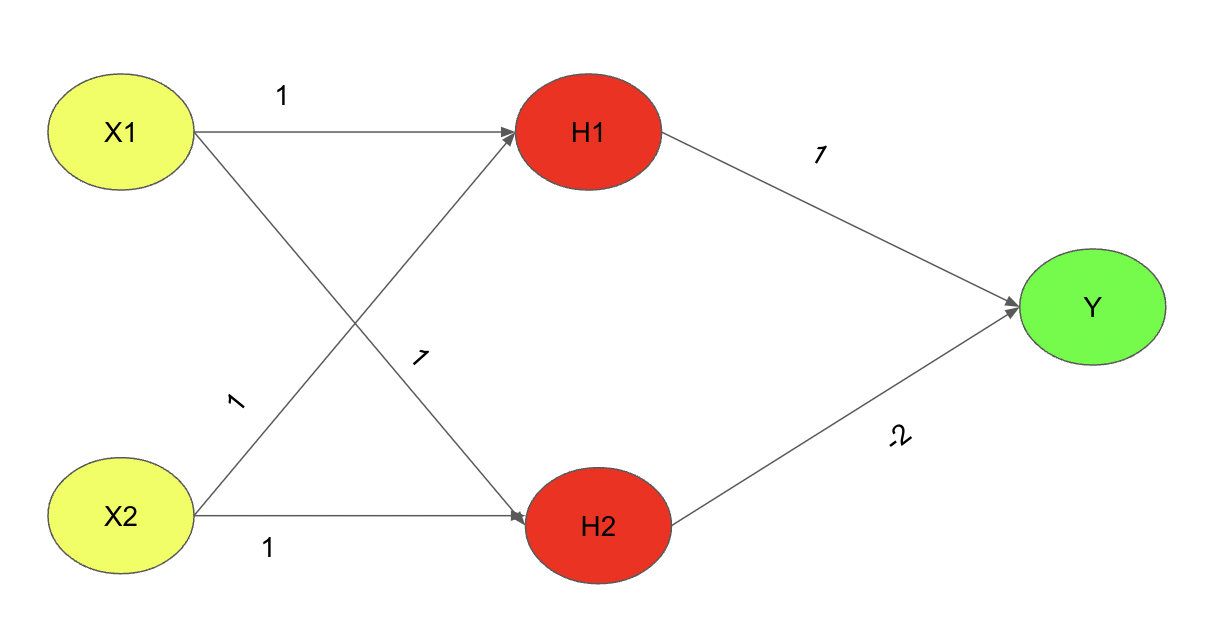


In the multi-layer perceptron diagram above, we can see that there are three inputs and thus three input nodes and the hidden layer has three nodes. The output layer gives two outputs, therefore there are two output nodes. The nodes in the input layer take input and forward it for further process, in the diagram above the nodes in the input layer forwards their output to each of the three nodes in the hidden layer, and in the same way, the hidden layer processes the information and passes it to the output layer.

Every node in the multi-layer perception uses a sigmoid activation function. The sigmoid activation function takes real values as input and converts them to numbers between 0 and 1 using the sigmoid formula.

.

**The XOR problem with neural networks can be solved by** using Multi-Layer Perceptrons or a neural network architecture with an input layer, hidden layer, and output layer. So during the forward propagation through the neural networks, the weights get updated to the corresponding layers and the XOR logic gets executed. The Neural network architecture to solve the XOR problem will be as shown below.



So with this overall architecture and certain weight parameters between each layer, the XOR logic output can be yielded through forward propagation. The overall neural network architecture uses the Relu activation function to ensure the weights updated in each of the processes to be 1 or 0 accordingly where for the positive set of weights the output at the particular neuron will be 1 and for a negative weight updation at the particular neuron will be 0 respectively. So let us understand one output for the first input state

**Example**:  For X1=0 and X2=0 we should get an input of 0. Let us solve it.

Solution: Considering X1=0 and X2=0

H1=RELU(0.1+0.1+0) = 0

H2=RELU(0.1+0.1+0)=0

So now we have obtained the weights that were propagated from the input layer to the hidden layer. So now let us propagate from the hidden layer to the output layer

Y=RELU(0.1+0.(-2))=0

This is how multi-layer neural networks or also known as Multi-Layer perceptrons (MLP) are used to solve the XOR problem and for all other input sets the architecture provided above can be verified and the right outcome for XOR logic can be yielded.

Q6: What is artificial neural network (ANN)? Explain some of the salient highlights in the different architectural options for ANN.

Ans: An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the brain. ANNs, like people, learn by examples. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning largely involves adjustments to the synaptic connections that exist between the neurons. Artificial Neural Networks (ANNs) are a type of machine learning model that are inspired by the structure and function of the human brain. They consist of layers of interconnected “neurons” that process and transmit information.There are several different architectures for ANNs, each with their own strengths and weaknesses. Some of the most common architectures include:

**Feedforward Neural Networks**: This is the simplest type of ANN architecture, where the information flows in one direction from input to output. The layers are fully connected, meaning each neuron in a layer is connected to all the neurons in the next layer.

**Recurrent Neural Networks (RNNs):** These networks have a “memory” component, where information can flow in cycles through the network. This allows the network to process sequences of data, such as time series or speech.

**Convolutional Neural Networks (CNNs):** These networks are designed to process data with a grid-like topology, such as images. The layers consist of convolutional layers, which learn to detect specific features in the data, and pooling layers, which reduce the spatial dimensions of the data.

**Autoencoders:** These are neural networks that are used for unsupervised learning. They consist of an encoder that maps the input data to a lower-dimensional representation and a decoder that maps the representation back to the original data.

**Generative Adversarial Networks (GANs):** These are neural networks that are used for generative modeling. They consist of two parts: a generator that learns to generate new data samples, and a discriminator that learns to distinguish between real and generated data.

Q7: Explain the learning process of an ANN. Explain, with example, the challenge in assigning synaptic weights for the interconnection between neurons? How can this challenge be addressed?

Ans: Learning process in ANN mainly depends on four factors, they are:

1. **The number of layers in the network (Single-layered or multi-layered)**
2. **Direction of signal flow (Feedforward or recurrent)**
3. **Number of nodes in layers:**The number of node in the input layer is equal to the number of features of the input data set. The number of output nodes will depend on possible outcomes i.e. the number of classes in case of supervised learning. But the number of layers in the hidden layer is to be chosen by the user. A larger number of nodes in the hidden layer, higher the performance but too many nodes may result in overfitting as well as increased computational expense.
4. **Weight of Interconnected Nodes:**Deciding the value of weights attached with each interconnection between each neuron so that a specific learning problem can be solved correctly is quite a difficult problem by itself. Take an example to understand the problem. Take the example of a**Multi-layered Feed-Forward Network,**we have to train an ANN model using some data, so that it can classify a new data set, sayp\_5(3,-2). Say we have deduced that p\_1=(5,2)   and  p\_2 = (-1,12)   belonging to class C1 while p\_3=(3,-5)   and p\_4 = (-2,-1)  belonging to class C2. We assume the values of synaptic weights w\_0,w\_1,w\_2 as -2, 1/2 and 1/4 respectively. But we will NOT get these weight values for every learning problem. For solving a learning problem with ANN, we can start with a set of values for synaptic weights and keep changing those in multiple iterations. The stopping criterion may be the rate of misclassification < 1% or the maximum numbers of iterations should be less than 25(a threshold value). There may be another problem that, the rate of misclassification may not reduce progressively.

Q8: Explain, in details, the backpropagation algorithm. What are the limitations of this algorithm?

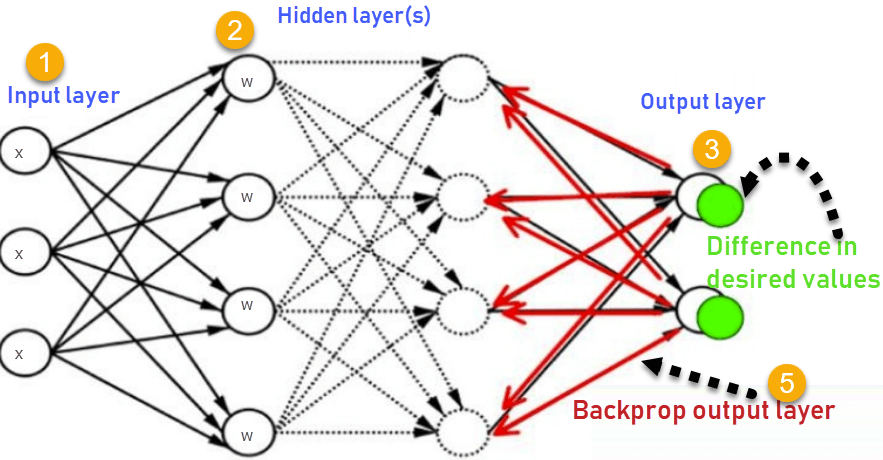
Ans: **Backpropagation** is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization.

Backpropagation in neural network is a short form for “backward propagation of errors.” It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function with respect to all the weights in the network.

**The Back propagation algorithm in neural network computes** the gradient of the loss function for a single weight by the chain rule. It efficiently computes one layer at a time, unlike a native direct computation. It computes the gradient, but it does not define how the gradient is used. It generalizes the computation in the delta rule.

**Steps :**

1. Inputs X, arrive through the preconnected path.
2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error in the outputs.
5. Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.



Two Types of Backpropagation Networks are:

* Static Back-propagation
* Recurrent Backpropagation

**Static back-propagation:**

It is one kind of backpropagation network which produces a mapping of a static input for static output. It is useful to solve static classification issues like optical character recognition.

**Recurrent Backpropagation:**

Recurrent Back propagation in data mining is fed forward until a fixed value is achieved. After that, the error is computed and propagated backward.

The main difference between both of these methods is: that the mapping is rapid in static back-propagation while it is nonstatic in recurrent backpropagation.

**Limitations of the Backpropagation algorithm**:

* It is slow, all previous layers are locked until gradients for the current layer is calculated.
* It suffers from vanishing or exploding gradients problem.
* It suffers from overfitting & underfitting problem.
* It considers predicted value & actual value only to calculate error and to calculate gradients, related to the objective function, partially related to the Backpropagation algorithm.

Q9: Describe, in details, the process of adjusting the interconnection weights in a multi-layer neural network.

Ans: In the Multilayer perceptron, there can more than one linear layer (combinations of neurons). If we take the simple example the three-layer network, first layer will be the input layer and last will be output layer and middle layer will be called hidden layer. We feed our input data into the input layer and take the output from the output layer. We can increase the number of the hidden layer as much as we want, to make the model more complex according to our task.

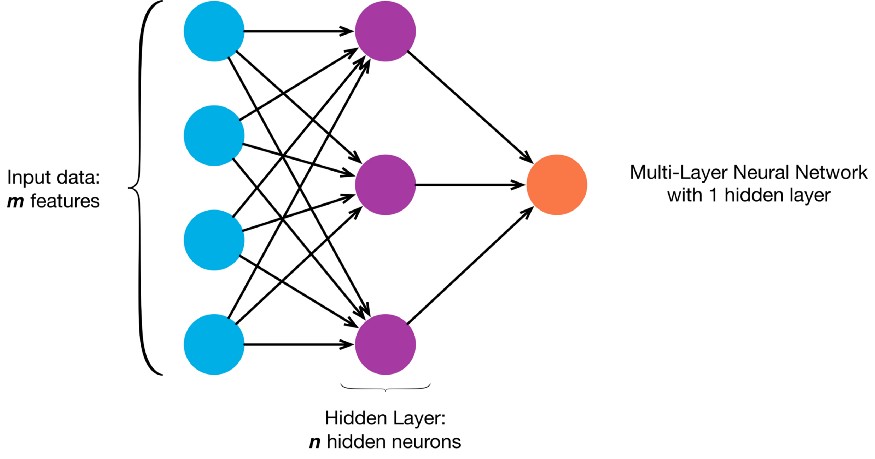
* First, if we have **m**input data (**x1, x2, …, xm**), we call this **m features**
* Secondly, when we multiply each of the m features with a weight (**w1, w2, …, wm**) and sum them all together, this is a **dot product**:

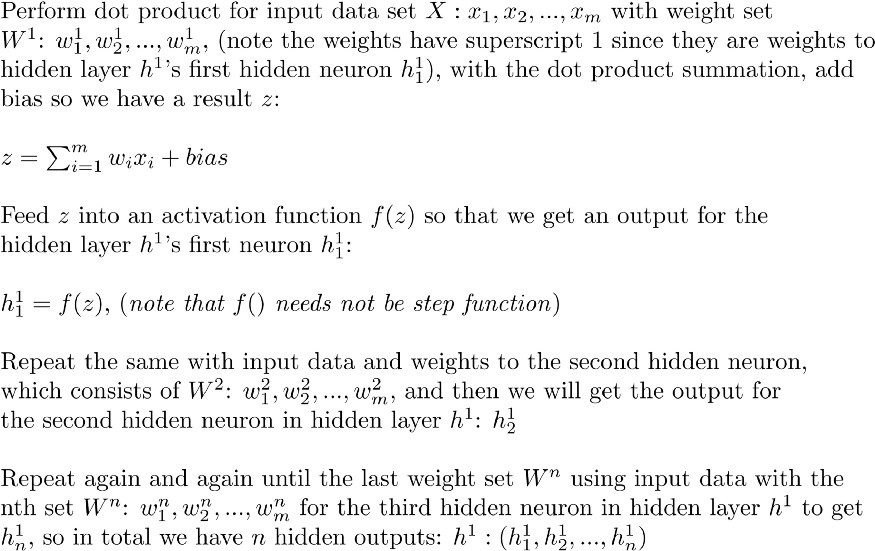


So here are the takeaways for now:

1. With ***m features*** in input ***X,***we need ***m*** weights to perform a dot product
2. With ***n*** hidden neurons in the hidden layer, we need ***n*** sets of weights (***W1, W2, … Wn***) for performing dot products
3. With 1 hidden layer, we perform ***n*** dot products to get the hidden output ***h:***(***h1, h2, …, hn***)
4. Then it’s just like a single-layer perceptron, we use hidden output ***h:***(***h1, h2, …, hn***) as input data that has **n features,**performdot product with 1 set of ***n*** weights (***w1, w2, …, wn***) to get your final output **y**^.

The procedure of how input values are *forward propagated* into the hidden layer, and then from hidden layer to the output is the same Below we provide a description of how this is done, using the following neural network in **Graph** .





Now the hidden layer outputs are calculated, we use them as inputs to calculate the final output.

Q10: What are the steps in the backpropagation algorithm? Why a multi-layer neural network is required?

Ans: The backpropagation algorithm is applicable for multi-layer feed-forward network. It is a supervised learning algorithm which continues adjusting the weights of the connected neurons with an objective to reduce the deviation of the output signal from the target output. This algorithm consists of multiple iterations, **known as epochs.** Each epoch consists of two phases:

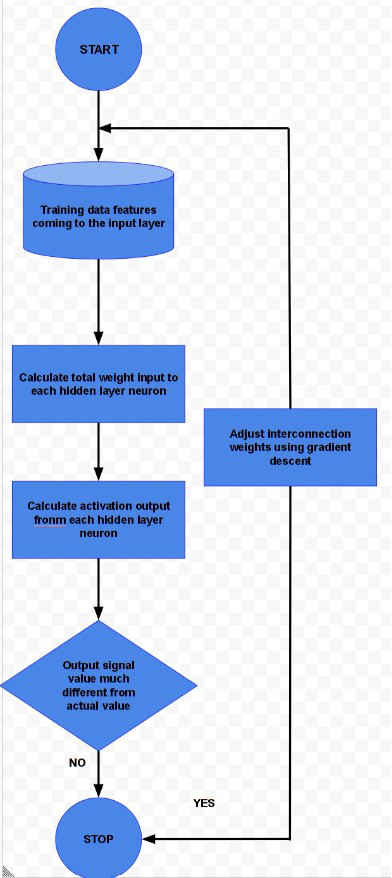
* **Forward Phase:**Signal flow from neurons in the input layer to the neurons in the output layer through the hidden layers. The weights of the interconnections and activation functions are used during the flow. In the output layer, the output signals are generated.
* **Backward Phase:**Signal is compared with the expected value. The computed errors are propagated backwards from the output to the preceding layer. The error propagated back are used to adjust the interconnection weights between the layers.

The below diagram depicts a reasonably simplified version of the back propagation algorithm.

One main part of the algorithm is adjusting the interconnection weights. This is done using a technique termed as Gradient Descent. In simple words, the algorithm calculates the partial derivative of the activation function by each interconnection weight to identify the ‘gradient’ or extent of change of the weight required to minimize the cost function.

**Steps :**

1. Inputs X, arrive through the preconnected path.
2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error in the outputs.
5. Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.



Q11: Write short notes on:

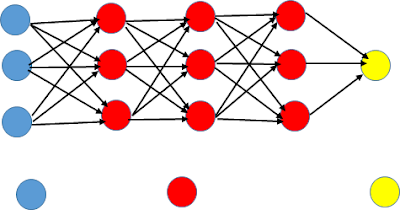
1. Artificial neuron.

Ans: An artificial neuron is a connection point in an artificial neural network. Artificial neural networks, like the human body's biological neural network, have a layered architecture and each network node (connection point) has the capability to process input and forward output to other nodes in the network. Artificial neurons are modeled after the hierarchical arrangement of neurons in biological sensory systems.

2. Multi-layer perceptron.

Ans: **Multi-Layer Perceptron (MLP)**

A multilayer perceptron is a type of feed-forward artificial neural network that generates a set of outputs from a set of inputs. An MLP is a neural network connecting multiple layers in a directed graph, which means that the signal path through the nodes only goes one way. The MLP network consists of input, output, and hidden layers. Each hidden layer consists of numerous perceptron’s which are called hidden layers or hidden unit.



3. Deep learning.

Ans: Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling. It is extremely beneficial to data scientists who are tasked with collecting, analyzing and interpreting large amounts of data; deep learning makes this process faster and easier.

At its simplest, deep learning can be thought of as a way to automate predictive analytics. While traditional machine learning algorithms are linear, deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction.

4. Learning rate

Ans: The learning rate, denoted by the symbol α, is a hyper-parameter used to govern the pace at which an algorithm updates or learns the values of a parameter estimate. In other words,**the learning rate regulates the weights of our**[**neural network**](https://deepchecks.com/glossary/neural-networks/)**concerning the loss gradient**. It indicates how often the neural network refreshes the notions it has learned.

Q12: Write the difference between :-

1. Activation function vs threshold function.

Ans: **An activation function** determines if a neuron should be activated or not activated. This implies that it will use some simple mathematical operations to determine if the neuron’s input to the network is relevant or not relevant in the prediction process.The ability to introduce non-linearity to an artificial neural network and generate output from a collection of input values fed to a layer is the purpose of the activation function.

**Threshold function:** A threshold activation function (or simply the activation function, also known as squashing function) results in an output signal only when an input signal exceeding a specific threshold value comes as an input. It is similar in behaviour to the biological neuron which transmits the signal only when the total input signal meets the firing threshold.

2. Step function vs sigmoid function.

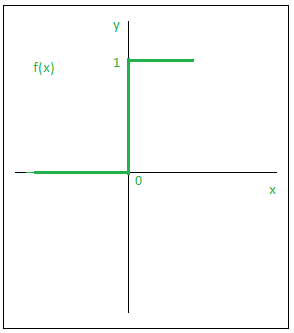
Ans: **Step Function**:

Step Function is one of the simplest kind of activation functions. In this, we consider a threshold value and if the value of net input say y is greater than the threshold then the neuron is activated.

Mathematically,

F(x) =1, if x >=0

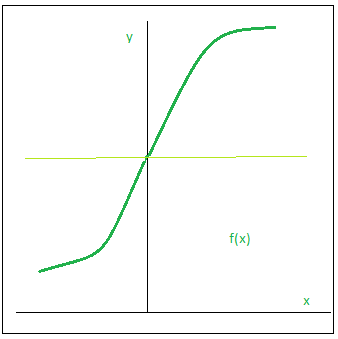




**Sigmoid Function:**

Sigmoid function is a widely used activation function. It is defined as:

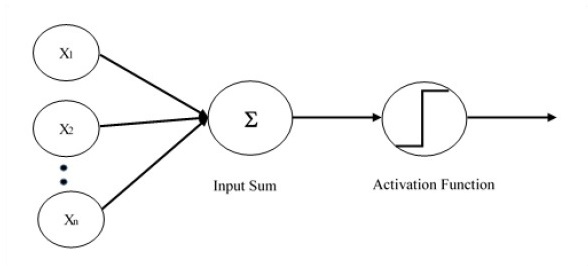




This is a smooth function and is continuously differentiable. The biggest advantage that it has over step and linear function is that it is non-linear. This is an incredibly cool feature of the sigmoid function. This essentially means that when I have multiple neurons having sigmoid function as their activation function – the output is non linear as well. The function ranges from 0-1 having an S shape.

3. Single layer vs multi-layer perceptron

Ans: **Single layer perceptron** is a simple Neural Network which contains only one layer. The single layer computation of perceptron is the calculation of sum of input vector with the value multiplied by corresponding vector weight. The displayed output value will be the input of an activation function.



**The perceptron consists of 4 parts.**

1. Input values or One input layer
2. Weights and Bias
3. Net sum
4. Activation Function

**Single Layer Perceptron has just two layers of input and output. It only has single layer hence the name single layer perceptron. It does not contain Hidden Layers as that of Multilayer perceptron**.

**Multi-Layer Perceptron (MLP)**

A multilayer perceptron is a type of feed-forward artificial neural network that generates a set of outputs from a set of inputs. An MLP is a neural network connecting multiple layers in a directed graph, which means that the signal path through the nodes only goes one way. The MLP network consists of input, output, and hidden layers. Each hidden layer consists of numerous perceptron’s which are called hidden layers or hidden unit.

