1. What is the function of a summation junction of a neuron? What is threshold activation function?

summation junction that aggregates all the weighted inputs and then passes the result to the activation function. The activation function is a threshold function that gives out 1 as the output if the sum of the weighted inputs is equal to or above the threshold value and 0 otherwise.

So let’s say we have n inputs = { X1, X2, X3, …. , Xn }

And we have n weights for each= {W1, W2, W3, …., W4}

So the summation of weighted inputs X.W = X1.W1 + X2.W2 + X3.W3 +....+ Xn.Wn

If X ≥ ø(threshold value)

Output = 1

Else

Output = 0

1. What is a step function? What is the difference of step function with threshold 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

f(x)=0, {if x}<0

1. Explain the McCulloch–Pitts model of neuron.

The McCulloch-Pitts neural model, which was the earliest ANN model, has only two types of inputs — Excitatory and Inhibitory. The excitatory inputs have weights of positive magnitude and the inhibitory weights have weights of negative magnitude. The inputs of the McCulloch-Pitts neuron could be either 0 or 1. It has a threshold function as an activation function. So, the output signal *yout* is 1 if the input *ysum* is greater than or equal to a given threshold value, else 0

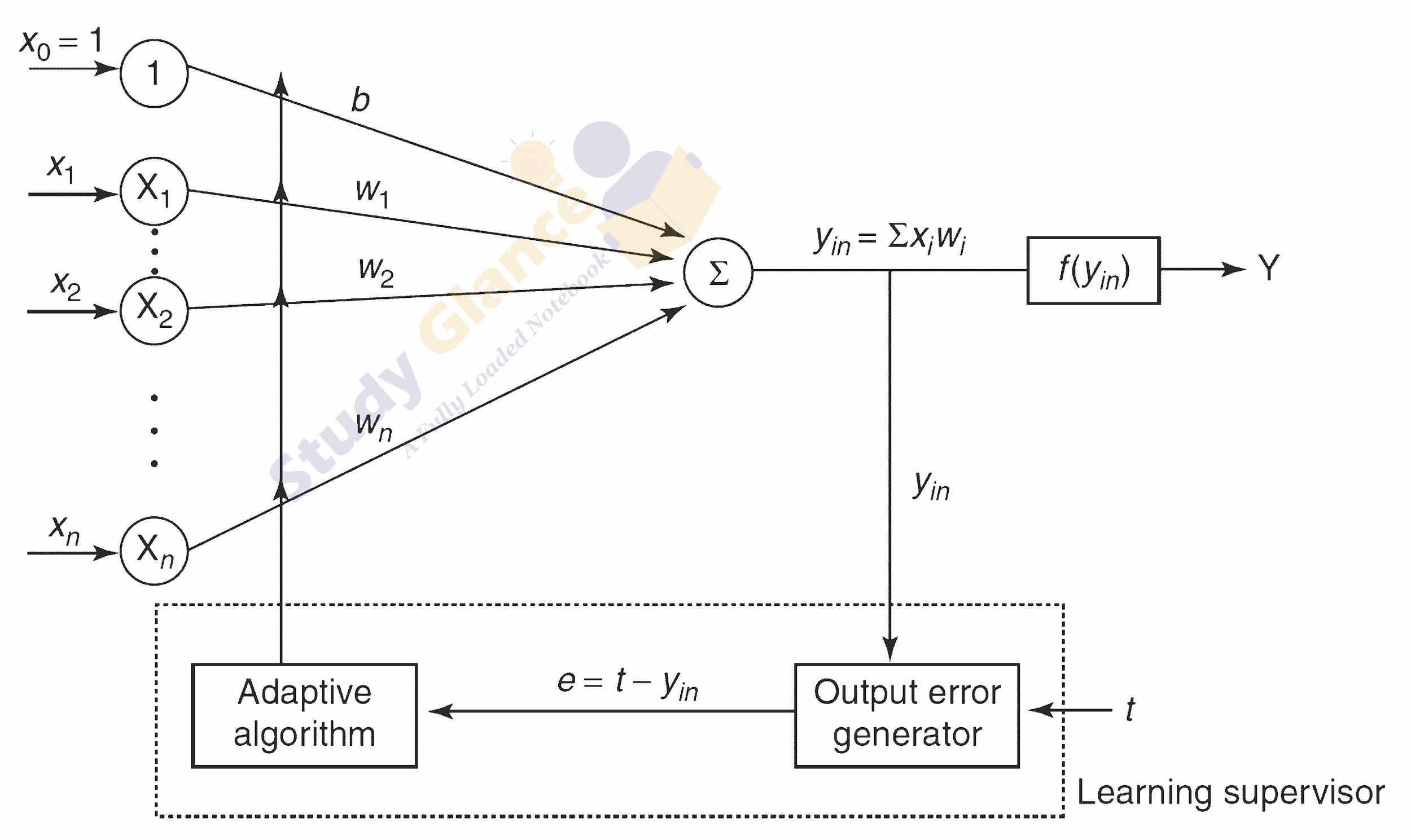
The McCulloch-Pitts neural model, which was the earliest ANN model, has only two types of inputs — Excitatory and Inhibitory. The excitatory inputs have weights of positive magnitude and the inhibitory weights have weights of negative magnitude. The inputs of the McCulloch-Pitts neuron could be either 0 or 1. It has a threshold function as an activation function. So, the output signal *yout* is 1 if the input *ysum* is greater than or equal to a given threshold value, else 0

1. Explain the ADALINE network model.

Adaline which stands for Adaptive Linear Neuron, is a network having a single linear unit. It was developed by Widrow and Hoff in 1960. Some important points about Adaline are as follows −

* It uses bipolar activation function.
* Adaline neuron can be trained using Delta rule or Least Mean Square(LMS) rule or widrow-hoff rule
* The net input is compared with the target value to compute the error signal.
* on the basis of adaptive training algoritham weights are adjusted

The basic structure of Adaline is similar to perceptron having an extra feedback loop with the help of which the actual output is compared with the desired/target output. After comparison on the basis of training algorithm, the weights and bias will be updated.



1. What is the constraint of a simple perceptron? Why it may fail with a real-world data set?

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.

1. What is linearly inseparable problem? What is the role of the hidden layer?

Simple problems, such as AND, OR etc are linearly separable. Even a simple problem such as XOR is not linearly separable. When you consider real world problems where data points are high dimensional, unlike the simple problems such as AND, OR, it is not possible to have a look at all the possible datapoints beforehand, hence, linear separability seems more like a property of the training dataset. For example, consider the spam vs non-spam e-mail classification problem. You cannot say without looking at the training data whether this problem is linearly separable or not. Also, even if this problem is linearly separable for one particular set of training examples, the same problem might not be linearly separable if you take a different set of training examples.

Hidden layers are the ones that are actually responsible for the excellent performance and complexity of neural networks. They perform multiple functions at the same time such as data transformation, automatic feature creation, etc

1. Explain XOR problem in case of a simple perceptron.

n XOR gate invites two input values and outputs a single value. The input values can be either 0 (false) or 1 (true) and the output value can be either 0 (false) and 1 (true). An XOR behaves in a pretty special way:

If either of the two inputs are true, then the output is true.

If both of the two inputs are either true or false, then the output is false.

This pattern shows in an Exclusive OR boolean operator (a.k.a XOR): the operator only returns true only if exclusively the first OR the second one is true. If both inputs are true or false, that is not exclusive because that can also satisfy the AND boolean operator: if both inputs are true, then it not only satisfies the OR operator, but also the AND operator. In other words, that case is not Exclusive. Hope this makes sense.

Why can’t raw perceptron learn XOR

So, let’s start simple: assume we only have one step function node and our perceptron should invite 2 inputs, which can be either 0 or 1. Let the two inputs be x1 and x2.

For every input, we have a weight. Let the two weights be w1 and w2.

Let’s take a look at the insightful cases: when both inputs are 0

That means that if the sum is 0, then the step function should yield 0. So the threshold should be

t>0

Then what about when both inputs are 1?

Since the weighted sum of our perception is always

x1∗w1+x2∗w2

Since x1 = 1 and x2 = 1

Our sum is w1+w2

which w1+w2<t

Okay, cool, but what if only one of the inputs is 1?

Let’s say x1 = 1 and x2 = 0

Then it’s 1∗w1+0∗w2

Let’s say x1 = 0 and x2 = 1

Then by the same convention, the sum is w2

By the XOR condition,

w1>t

and

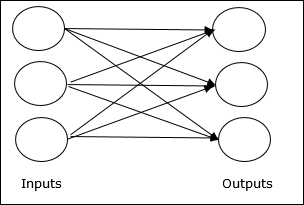
w2>t

If both weights should be greater than t, then how could w1+w2<t

So we can see that the XOR values cannot be fit into a simple two dimensional planeDesign a multilayer perceptron to implement A XOR B.

1. Explain the single-layer feed forward architecture of ANN.

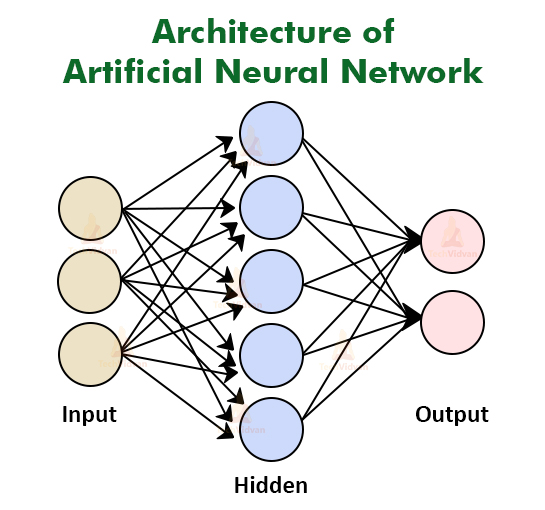
Single layer feedforward network − The concept is of feedforward ANN having only one weighted layer. In other words, we can say the input layer is fully connected to the output layer.



1. Explain the competitive network architecture of ANN.

A neural network consists of three layers. The first layer is the input layer. It contains the input neurons that send information to the hidden layer. The hidden layer performs the computations on input data and transfers the output to the output layer. It includes weight, activation function, cost function.

The connection between neurons is known as weight, which is the numerical values. The weight between neurons determines the learning ability of the neural network. During the learning of artificial neural networks, weight between the neuron changes.



1. Consider a multi-layer feed forward neural network. Enumerate and explain steps in the backpropagation algorithm used to train the network.

Backpropagation Algorithm:

Step 1: Inputs X, arrive through the preconnected path.

Step 2: The input is modeled using true weights W. Weights are usually chosen randomly.

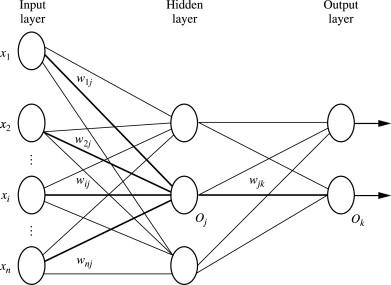
Step 3: Calculate the output of each neuron from the input layer to the hidden layer to the output layer.

Step 4: Calculate the error in the outputs

Backpropagation Error= Actual Output – Desired Output

Step 5: From the output layer, go back to the hidden layer to adjust the weights to reduce the error.

Step 6: Repeat the process until the desired output is achieved.



Parameters :

* x = inputs training vector x=(x1,x2,…………xn).
* t = target vector t=(t1,t2……………tn).
* δk = error at output unit.
* δj = error at hidden layer.
* α = learning rate.
* V0j = bias of hidden unit j.

Training Algorithm :

Step 1: Initialize weight to small random values.

Step 2: While the stepsstopping condition is to be false do step 3 to 10.

Step 3: For each training pair do step 4 to 9 (Feed-Forward).

Step 4: Each input unit receives the signal unit and transmitsthe signal xi signal to all the units.

Step 5 : Each hidden unit Zj (z=1 to a) sums its weighted input signal to calculate its net input

zinj = v0j + Σxivij ( i=1 to n)

Applying activation function zj = f(zinj) and sends this signals to all units in the layer about i.e output units

For each output l=unit yk = (k=1 to m) sums its weighted input signals.

yink = w0k + Σ ziwjk (j=1 to a)

and applies its activation function to calculate the output signals.

yk = f(yink)

Backpropagation Error :

Step 6: Each output unit yk (k=1 to n) receives a target pattern corresponding to an input pattern then error is calculated as:

δk = ( tk – yk ) + yink

Step 7: Each hidden unit Zj (j=1 to a) sums its input from all units in the layer above

δinj = Σ δj wjk

The error information term is calculated as :

δj = δinj + zinj

Updation of weight and bias :

Step 8: Each output unit yk (k=1 to m) updates its bias and weight (j=1 to a). The weight correction term is given by :

Δ wjk = α δk zj

and the bias correction term is given by Δwk = α δk.

therefore wjk(new) = wjk(old) + Δ wjk

w0k(new) = wok(old) + Δ wok

for each hidden unit zj (j=1 to a) update its bias and weights (i=0 to n) the weight connection term

Δ vij = α δj xi

and the bias connection on term

Δ v0j = α δj

Therefore vij(new) = vij(old) + Δvij

v0j(new) = v0j(old) + Δv0j

Step 9: Test the stopping condition. The stopping condition can be the minimization of error, number of epochs

1. What are the advantages and disadvantages of neural networks?

Advantages of Artificial Neural Network

The advantages of the neural network are as follows −

* A neural network can implement tasks that a linear program cannot.
* When an item of the neural network declines, it can continue without some issues by its parallel features.
* A neural network determines and does not require to be reprogrammed.
* It can be executed in any application.

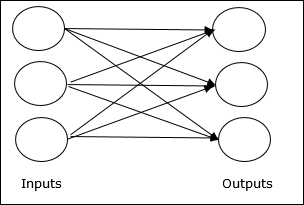
Disadvantages of Artificial Neural Network

The disadvantages of the neural network are as follows −

* The neural network required training to operate.
* The structure of a neural network is disparate from the structure of microprocessors therefore required to be emulated.
* It needed high processing time for big neural networks.

1. Write short notes on any two of the following:
   * 1. Biological neuron
     2. ReLU function
     3. Single-layer feed forward ANN
     4. Gradient descent
     5. Recurrent networks

Single layer feedforward network − The concept is of feedforward ANN having only one weighted layer. In other words, we can say the input layer is fully connected to the output layer.



Recurrent Neural Network(RNN) is a type of [Neural Network](https://www.geeksforgeeks.org/tag/neural-network/) where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is its Hidden state, which remembers some information about a sequence. The state is also referred to as *Memory State* since it remembers the previous input to the network. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

