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

Ans. an activation function is a function that is added into an artificial neural network in order to help the network learn complex patterns in the data. When comparing with a neuron-based model that is in our brains, the activation function is at the end deciding what is to be fired to the next neuron.

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

Ans. 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. Given below is the graphical representation of step function.

1. Explain the McCulloch–Pitts model of neuron.

Ans. This is simplified model of real neurons, known as Threshold Logic Unit. A set of synapsesc (i.e connections) brings the activations from the other neurons. A processing unit sums the inputs, the applies the non-linear activation funcation (i.e threshold / transfer function).

1. Explain the ADALINE network model.

Ans. MADALINE (Many ADALINE) is a three-layer (input, hidden, output), fully connected, feed-forward artificial neural network architecture for classification that uses ADALINE units in its hidden and output layers, i.e. its activation function is the sign function. The three-layer network uses memistors.

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

Ans. Perceptron networks have several limitations. First, the output values of a perceptron can take on only one of two values (0 or 1) because of the hard-limit transfer function. Second, perceptrons can only classify linearly separable sets of vectors. Perceptrons only represent linearly separable problems. They fail to converge if the training examples are not linearly separable.

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

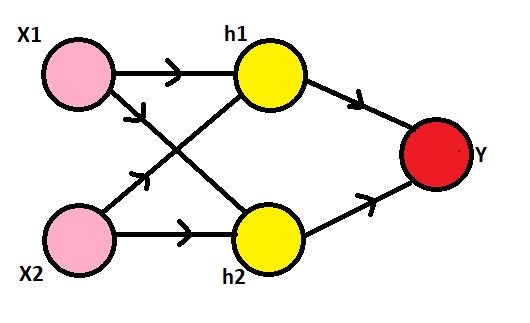
Ans. Clearly not all decision problems are linearly separable: they cannot be solved using a linear decision boundary. Problems like these are termed linearly inseparable. In neural networks, a hidden layer is located between the input and output of the algorithm, in which the function applies weights to the inputs and directs them through an activation function as the output. In short, the hidden layers perform nonlinear transformations of the inputs entered into the network.

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

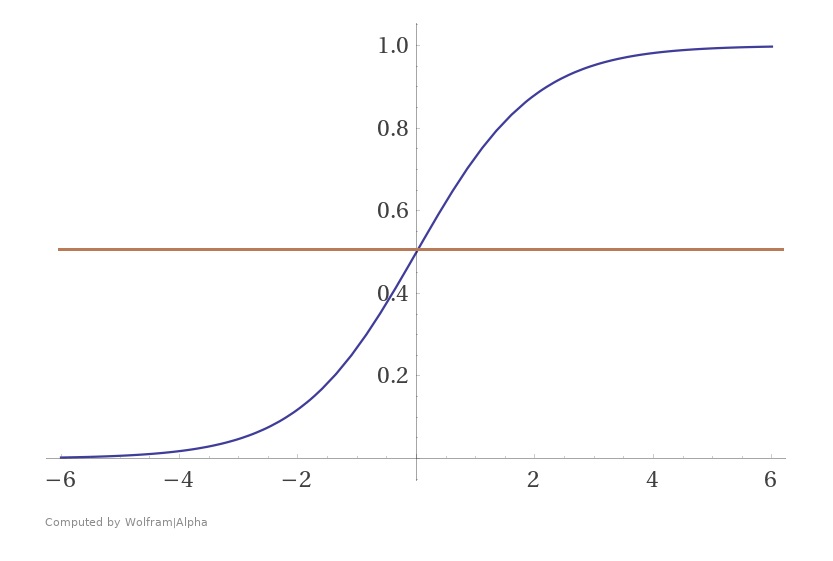
Ans. The XOr problem is that we need to build a Neural Network (a perceptron in our case) to produce the truth table related to the XOr logical operator. This is a binary classification problem. Hence, supervised learning is a better way to solve it.

1. Design a multi-layer perceptron to implement A XOR B.

Ans. The following is a plan for the perceptron.



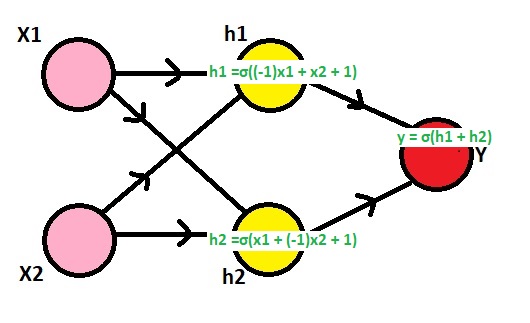
Here, we need to observe that our inputs are 0s and 1s. To make it a XOr gate, we will make the h1 node to perform the (x2 **AND NOT** x1) operation, the h2 node to perform (x1 **AND NOT** x2) operation and the y node to perform (h1 **OR** h2) operation. The NOT gate can be produced for an input a by writing (1-a), the AND gate can be produced for inputs a and b by writing (a.b) and the OR gate can be produced for inputs a and b by writing (a+b). Also, we'll use the sigmoid function as our activation function σ, i.e., σ(x) = 1/(1+e^(-x)) and the threshold for classification would be 0.5, i.e., any x with σ(x)>0.5 will be classified as 1 and others will be classified as 0.



Now, since we have all the information, we can go on to define h1, h2 and y. Using the formulae for AND, NOT and OR gates, we get:

1. h1 = σ((1-x1) + x2) = σ((-1)x1 + x2 + 1)
2. h2 = σ(x1 + (1-x2)) = σ(x1 + (-1)x2 + 1)
3. y = σ(h1 + h2) = σ(h1 + h2 + 0)

Hence, we have built a multi layered perceptron with the following weights and it predicts the output of a XOr logical operator.



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

Ans. In this type of network, we have only two layers input layer and output layer but the input layer does not count because no computation is performed in this layer. The output layer is formed when different weights are applied on input nodes and the cumulative effect per node is taken.

1. Explain the competitive network architecture of ANN.

Ans. A Competitive learning is an artificial neural network learning process where different neurons or processing elements compete on who is allowed to learn to represent the current input.

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

Ans. The Backpropagation algorithm is a supervised learning method for multilayer feed-forward networks from the field of Artificial Neural Networks.

Feed-forward neural networks are inspired by the information processing of one or more neural cells, called a neuron. A neuron accepts input signals via its dendrites, which pass the electrical signal down to the cell body. The axon carries the signal out to synapses, which are the connections of a cell’s axon to other cell’s dendrites.

The principle of the backpropagation approach is to model a given function by modifying internal weightings of input signals to produce an expected output signal. The system is trained using a supervised learning method, where the error between the system’s output and a known expected output is presented to the system and used to modify its internal state.

Technically, the backpropagation algorithm is a method for training the weights in a multilayer feed-forward neural network. As such, it requires a network structure to be defined of one or more layers where one layer is fully connected to the next layer. A standard network structure is one input layer, one hidden layer, and one output layer.

Backpropagation can be used for both classification and regression problems, but we will focus on classification in this tutorial.

In classification problems, best results are achieved when the network has one neuron in the output layer for each class value. For example, a 2-class or binary classification problem with the class values of A and B. These expected outputs would have to be transformed into binary vectors with one column for each class value. Such as [1, 0] and [0, 1] for A and B respectively. This is called a one hot encoding.

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

Ans. The network problem does not immediately corrode. Ability to train machine: Artificial neural networks learn events and make decisions by commenting on similar events. Parallel processing ability: Artificial neural networks have numerical strength that can perform more than one job at the same time.

1. Write short notes on any two of the following:
   * 1. Biological neuron

Biological neuron models, also known as a spiking neuron models, are mathematical descriptions of the properties of certain cells in the nervous system that generate sharp electrical potentials across their cell membrane, roughly one millisecond in duration, called action potentials or spikes . Since spikes are transmitted along the axon and synapses from the sending neuron to many other neurons, spiking neurons are considered to be a major information processing unit of the nervous system. Spiking neuron models can be divided into different categories: the most detailed mathematical models are biophysical neuron models (also called Hodgkin-Huxley models) that describe the membrane voltage as a function of the input current and the activation of ion channels. Mathematically simpler are integrate-and-fire models that describe the membrane voltage as a function of the input current and predict the spike times without a description of the biophysical processes that shape the time course of an action potential. Even more abstract models only predict output spikes (but not membrane voltage) as a function of the stimulation where the stimulation can occur through sensory input or pharmacologically.

* + 1. ReLU function

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

* + 1. Single-layer feed forward ANN
    2. Gradient descent
    3. Recurrent networks