SUBJECT: Neural Networks and Soft Computing(NNSC)

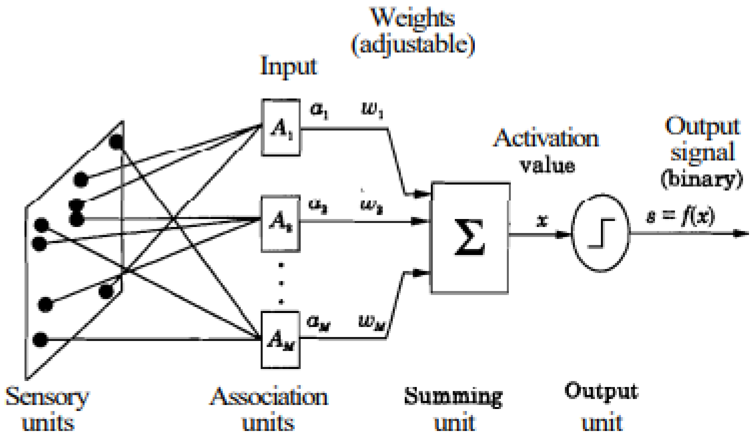
Regulation: R20 **A.Y.:**2023-24

Name of the Faculty: M.SUMAN **Year & Sem:** IV-I

UNIT- 2

1. Define Perceptron? Explain in brief about Perceptron Model?[7] - Understand-**Model Question**

The Rosenblatt's perceptron model (Figure 1.5) for an artificial neuron consists of outputs from sensory units to a fixed set of association units, the outputs of which are fed to an MP neuron



The association units perform predetermined manipulations on their inputs. The main deviation from the MP model is that learning (i.e., adjustment of weights) is incorporated in the operation of the imit. The desired or target output (b) is compared with the actual binary output (s), and the error (delta) is used to adjust the weights. The following equations describe the operation of the perceptron model of a neuron:

Activation: $$x = \sum_{i=1}^{M} w_i a_i - \theta$$

Output signal: $s = f(x)$

Error: $6 = b - s$

Weight change: $Aw_i = 76 a_i$

where η is the learning rate parameter.

2. Discuss the MC-Culloch and the Adaline Neron Models[7]-Understand-**Model Question**

McCulloch-Pitts Model

In McCulloch-Pitts (MP) model (Figure 1.2) the activation (x) is given by a weighted sum of its M input values (a_i) and a bias term (8). The output signal (s) is typically a nonlinear function flx) of the activation value x. The following equations describe the operation of an MP model:

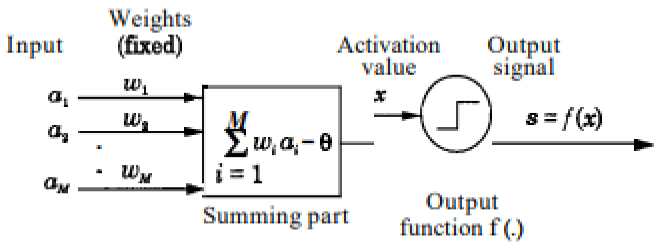
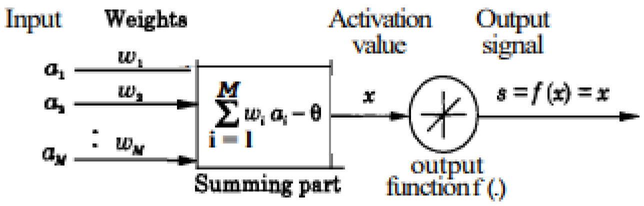


Figure 1.2 McCulloch-Pitts model of a neuron.

ADAptive LINear Element (ADALINE) is a computing model proposed by Widrow and is shown in Figure



Widrow's Adaline model is that, in the Adaline the analog activation value (x) is compared with the target output (b). In other words, the output is a linear fundion of the activation value (x). The equations that describe the operation of an Adaline are as follows:

Activation:	$x = \sum_{i=1}^{M} w_i a_i - \theta$
Output signal:	$s = f(x) = x$
Error:	$6 = b - s = b - x$
Weight change:	$\Delta w_i = \eta \delta a_i$

3. What are the various types of neuron activation functions? Explain with a neat sketch.[7] - Analyze-**Model Question**

The neuron output is calculated by processing the weighted sum (or) NET value through the activation function nothing but activation functions are mainly used to find the output of a neuron. The following are the different activation functions.

Binary activation function:

In this, the neuron will not produce output if the activation value is less than or equal to zero and it produces output if it is greater than zero. The Binary step function is as shown in Figure .

Mathematically it can be represented as

$$f(x) = 0, x \le 0$$
$$f(x) = 1, x > 0$$

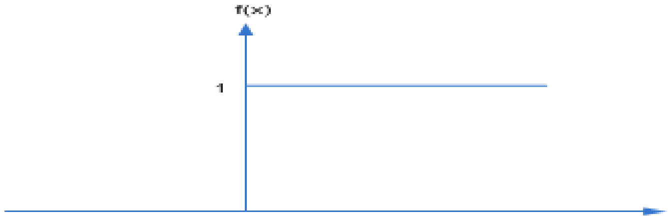


Figure. Binary Activation function

Step Activation function:

The output is zero if the activation value is less than t and it is equal to 1 if the activation value is more than or equal to t. The step activation function is as shown in Figure.

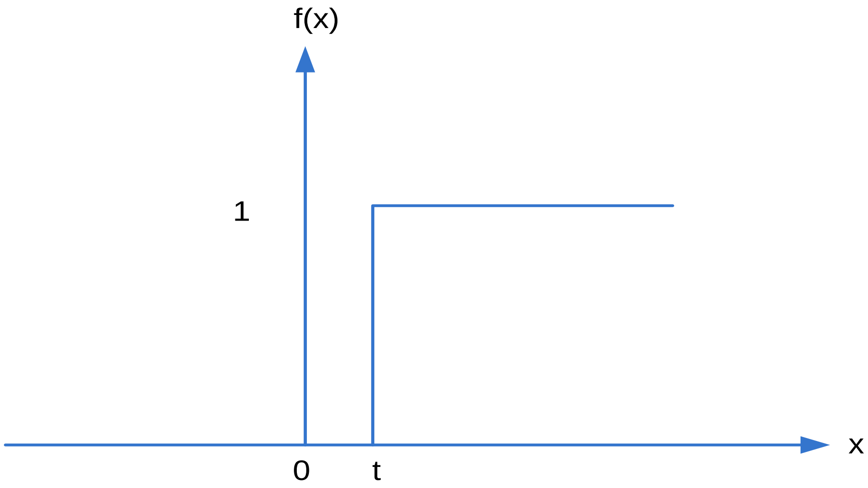


Figure Step activation function

It can be mathematically written as

$$f(x) = 0, x < t$$
$$f(x) = 1, x \geq t$$

Sign Activation Function:

The sign activation function will produce output 1 if the activation value is greater than zero and produces -1 if the activation value is -1. The sign activation function is as depicted in below Figure.

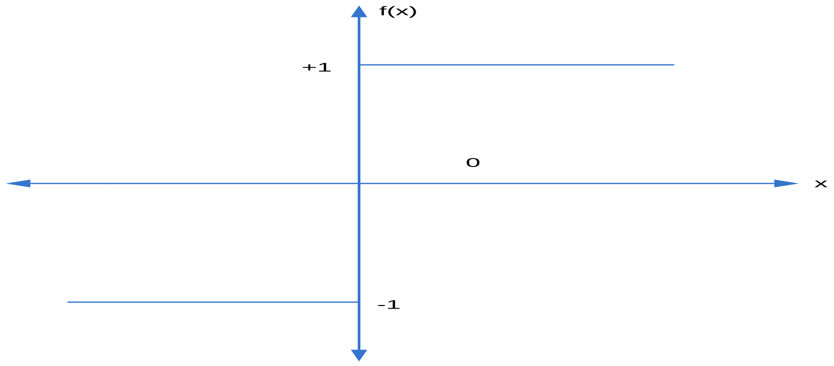


Figure Sign Activation function

Mathematically it can be represented as

$$f(x) = 1, x > 0$$
$$f(x) = -1, x < 0$$

Sigmoidal activation function

The output is obtained by processing the activation function value through the sigmoidal function and it is as depicted in Figure .

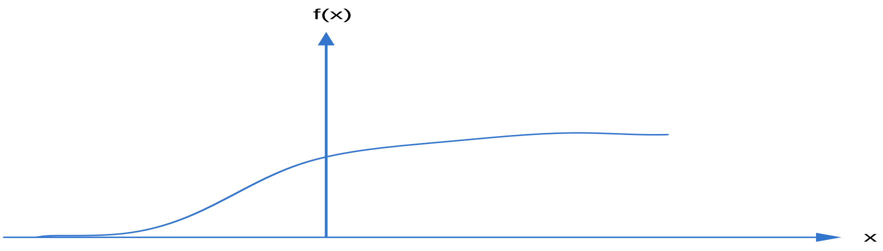


Figure Sigmoidal activation function

Mathematically it can be represented as

$$f(x) = \frac{1}{1 + e^{-x}}$$

4. Explain the following architectures of the Neural Networks[14] - Remember-**Model Question**
   i. Single Layer Feed Forward Neural Networks
   ii. Multi Layer Feed Forward Neural Networks
   iii. Feed Back Neural Networks
5. Discuss the various learning rules of the Artificial Neural Networks[7] - Understand-**Model Question**

There is mainly three different types of architectures of artificial neural networks such as a single layer, multilayer, and feedback neural networks.

Single-layer Neural Networks

This form of neural network will have input and the output layers. The input layer neurons receive the input and neurons in the output layer produce the output. The weighted links interconnect the layers and transfers the information between the layers. Despite the two layers, the computations are performed only at output layer. The single-layer NN is as depicted in Figure 4.9. These neural networks are having some limitations such as only applicable to linear separable problems.

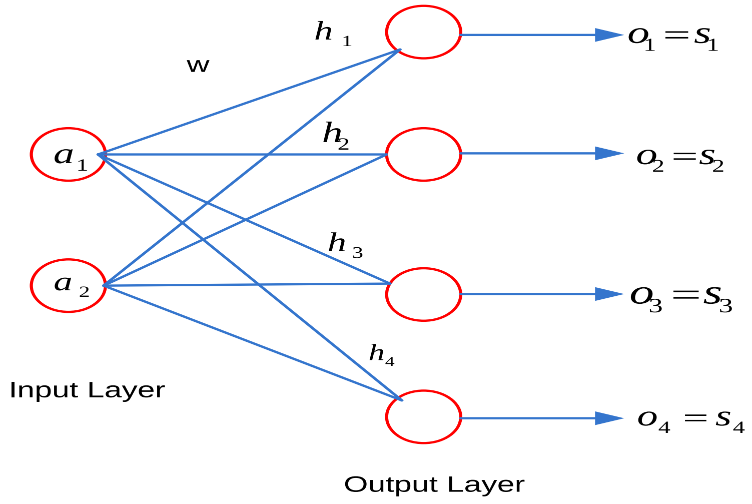


Figure Single layer neural networks

Multi Layer Neural Network

It consists of input, hidden, and output layers . The input layer only supplies the input and no computations are performed at this layer. Weighted connections interconnect the input and hidden layers. Once the input is applied hidden layer, the weighted sum or activation value will be calculated. All these activation values are passed through the activation functions to evaluate the output of all hidden layer neurons. The outputs of the hidden layer neurons are given as input to the output layer.

The hidden and output layers are also interconnected by weighted connections. Again weighted sum or activation values will be calculated at each neuron in the output layer and these values will be processed through the activation value to determine the output at each neuron of the output layer. Even though these networks can be applied to both linearly separable and nonlinear separable problems but they are suffering from storage-related problems. It is as depicted in Figure.

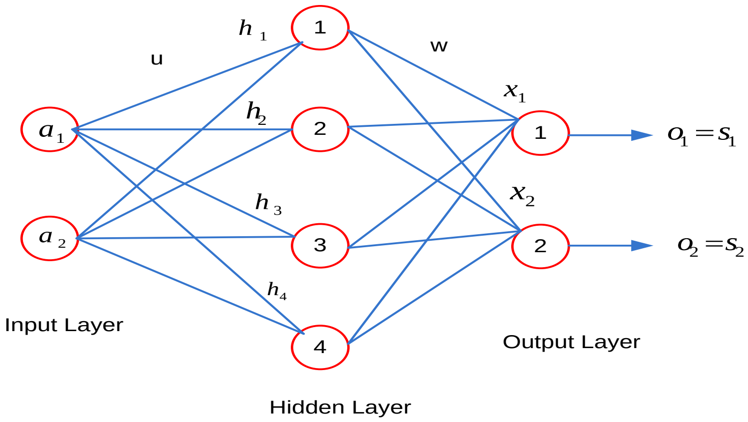


Figure Architecture of Multilayer Neural Networks

Feed Back or Recurrent Neural Networks

The artificial neural networks whose output is fed back to the input are called feedback or recurrent networks. In these types of networks, the present output not only depends upon the input but also on the previous output. Examples of such types of problems are stock market price, weather forecasting, etc.

A scientist called John Hopfield had made important contributions to feedback networks. So these feedback networks are popular with his name and also called the Hopfield network. Two different Hopfield networks are there one is discrete and the other is a continuous Hopfield neural network. If the output is determined using discrete activation function then it is called a discrete Hopfield network and the continuous activation function like sigmoid is used to determine the output then it is called a continuous Hopfield network. The Architecture of the Hopfield network is as shown in Figure 4.11.

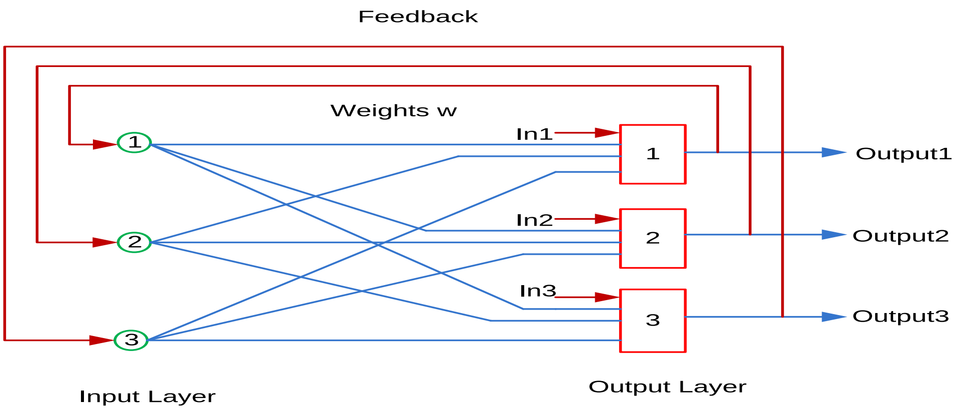


Figure. Feed Back Neural Network

6. How to achieve the performance of Back Propagation Learning? Discuss in [14] detail. - Analysis-**Model Question**

Back Propagation algorithm(BP) provides a systematic method to train the Multi-Layer Neural Networks (MLNN). It is also called as back propagation rule which makes use of the gradient descent method. For the training purpose the BP algorithm uses extended gradient-descent-based method. It is a computationally efficient method for optimizing the weights in a feed-forward network. Training is nothing but modifying the weights to minimize the squared error such that the actual output is nearly equal to target output. According to the BP algorithm, inputs will be applied to hidden layer and activation values are calculated at each and every unit of hidden layer. The output of each hidden layer unit will be calculated by processing the respective activation value through activation function. The outputs of hidden layer neurons is given as inputs to the output layer and again weighted sum or activation values will be calculated at each unit in the output layer which are processed through activation function to find the output. The outputs are compared with the target values and the squared error will be calculated. To minimize the error first the weights connects output and hidden layers will be modified and next the weights connects hidden and output layers will be modified. So while modifying the error the movement is backward so the algorithm is called as BP algorithm.

The architecture of BPNN and the proposed BPNN is as shown in Figure.4.12. The general architecture has input layer with n neurons, hidden layer with p neurons and output layer consists of m number of neurons. The weights between input layer(V_{ij}) and the weights between hidden and output layer (W_{jk}) which are initialized randomly.

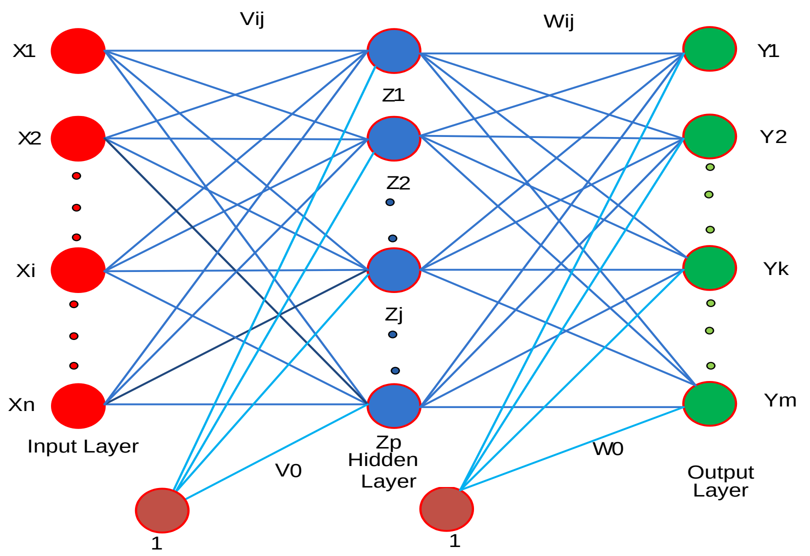


Figure. Architectures of Back propagation Neural Network

The input layer neurons are applied with input vector and the weighted sum and the output at all the neurons in the hidden layer can be calculated as given in Equations

$$Z_{-inj} = Voj + \sum_{i=1}^{n} (X_i \times V_{ij})$$

$$Z_j = f(Z_{-inj}) = \frac{1}{1+e^{Z_{-inj}}}$$

The outputs of hidden layer neurons will be the inputs to the neurons in the output layer. The weighted sum and the output at all the neurons in the output layer can be calculated as given in Equations 4.7 and 4.8.

$$Y_{-ink'} = W_{ok'} + \sum_{j=1}^{p} (Z_{j'} * W_{jk'})$$

$$Y_k = f(Y_{-ink}) = \frac{1}{1+e^{Y_{-ink}}}$$

The output at each neuron is compared with the target output and squared error is calculated. The weights will be varied until the error is minimized below a tolerance value. The weights between hidden layer and the output layer can be calculated as given in equations.

$$W_{jk'}(new) = W_{jk'}(old) + \Delta W_{jk'},$$

$$W_{ok'}(new) = W_{ok'}(old) + \Delta W_{ok}$$

$$\text{Where } \Delta W_{jk} = \eta * \frac{\partial E}{\partial W_{jk}} = \eta * \delta_k * Z_j$$

$$\Delta W_{ok} = \eta * \frac{\partial E}{\partial W_{ok}} = \eta * \delta_k$$

The weights connects the hidden and input layer, bias neuron will be updated as given in Equation.

$$V_{ij}^{\cdot}(new)=V_{ij}^{\cdot}(old)+\Delta V_{ij}$$

$$V_{oj}^{\cdot}(new) = V_{oj}^{\cdot}\ (old) + \Delta V_{oj}$$

Where

$$\Delta V_{ij}=\eta * \delta_j * X_i$$

$$\Delta V_{oj}=alpha*\delta_i \qquad (4.11)$$

4.6.1.1 Methodology

Step1: Initialize all weights to random values

Step2: Apply the deviation in speed as an input to the BPNN.

Step3: Calculate the outputs of the hidden and output layers in the forward pass. For the proposed Neural network the output is change in voltage

Step4: Calculate the squared error which is to be minimized for better training.

Step5: Update the weights between output and hidden, hidden, and input layers in the backward process.

Step6: Repeat the procedure till the squared error is blow 0.001.

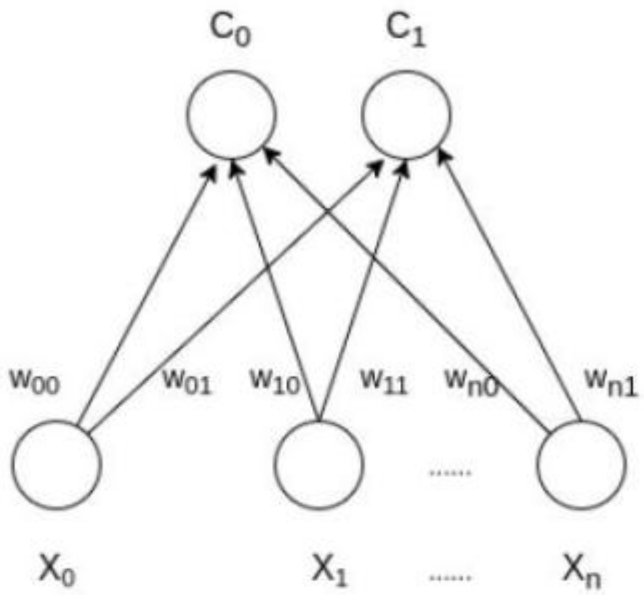
7. What are the limitations of Back Propagation algorithm?[7] - Remember-**Model Question**

The limitations of the BPNN areas given as below.

i. It converges slowly if the learning rate is chosen small and if it chosen large the system may become unstable.
ii. These networks are suffering with local minima problem
iii. The network gives the worse generalization performance if the network is over trained.
iv. It is a gradient based training algorithm which takes longer time for training. Sometimes based on the complexity of the problem it takes days and weeks also.

8. Discuss the Kohnen's self organizing networks[7] -Analysis-**Model Question**

Self Organizing Map (or Kohonen Map or SOM) is a type of Artificial Neural Network which is also inspired by biological models of neural systems from the 1970s. It follows an unsupervised learning approach and trained its network through a competitive learning algorithm. SOM is used for clustering and mapping (or dimensionality reduction) techniques to map multidimensional data onto lower-dimensional which allows people to reduce complex problems for easy interpretation. SOM has two layers, one is the Input layer and the other one is the Output layer. The architecture of the Self Organizing Map with two clusters and n input features of any sample is given below:



How do SOM works?

Let's say an input data of size (m, n) where m is the number of training examples and n is the number of features in each example. First, it initializes the weights of size (n, C) where C is the number of clusters. Then iterating over the input data, for each training example, it updates the winning vector (weight vector with the shortest distance (e.g Euclidean distance) from training example). Weight updation rule is given by : wij = wij(old) + alpha(t) * (xi k - wij(old)) where alpha is a learning rate at time t, j denotes the winning vector, i denotes the i th feature of training example and k denotes the k th training example from the input data. After training the SOM network, trained weights are used for clustering new examples.

A new example falls in the cluster of winning vectors. Algorithm Training: Step 1: Initialize the weights wij random value may be assumed. Initialize the learning rate α. Step 2: Calculate squared Euclidean distance. D(j) = Σ (wij – xi)^2 where i=1 to n and j=1 to m Step 3: Find index J, when D(j) is minimum that will be considered as winning index. Step 4: For each j within a specific neighborhood of j and for all i, calculate the new weight. wij(new)=wij(old) + α[xi – wij(old)] Step 5: Update the learning rule by using : α(t+1) = 0.5 * t Step 6: Test the Stopping Condition

9. Give the architectural description and applications of Hopfield Neural network[14] - Analysis-**Model Question**

Topic name: Applications of NN

10. Discuss the applications of the Artificial Neural Networks[7] - Apply-**Model Question**

    The following are the different applications of ANN are

    i. Optmization
    ii. Forecasting
    iii. Image Prediction
    iv. Data Compression
    v. Image recognition etc.