3.1 Example:

Given a two-input neuron with the following parameters: $b = 1.2$, $\mathbf{W} = [3 \;\; 2]$ and $\mathbf{p} = [-5 \;\; 6]^{\mathrm{T}}$, calculate the neuron output for the following transfer functions:

1. A symmetrical hard limit transfer function.
2. A saturating linear transfer function.
3. A hyperbolic tangent sigmoid transfer function.

Solution:

First calculate the net input :

$$n = \mathbf{W}\mathbf{p} + b = [3 \;\; 2]\begin{bmatrix} -5 \\ 6 \end{bmatrix} + (1.2) = -1.8$$

Now find the outputs for each of the transfer functions.

1. $f(n) = \begin{cases} -1 & n < 0 \\ 1 & n \geq 0 \end{cases}$

$$a = f(-1.8) = -1$$

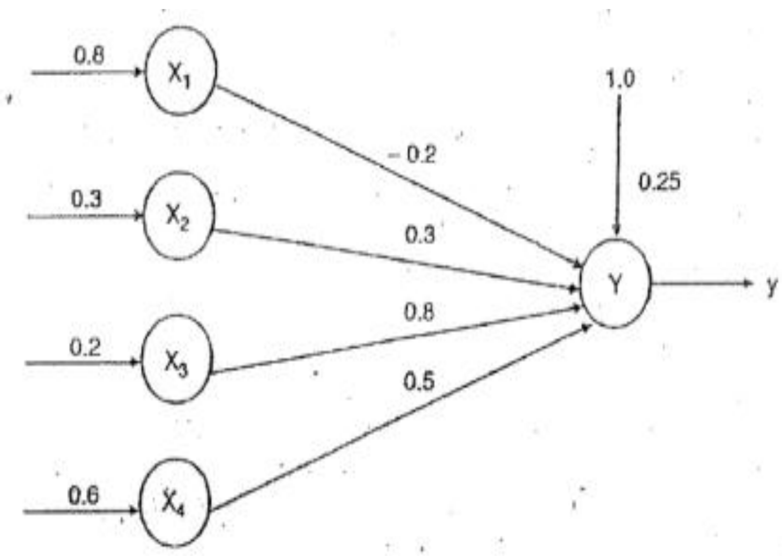
2. $f(n) = \begin{cases} 0 & n < 0 \\ n & 0 \leq n \leq 1 \\ 1 & n > 1 \end{cases}$

$$a = f(-1.8) = 0$$

3. $f(n) = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$

$$a = f(-1.8) = -0.9468$$

Example 1.1 For the network shown in Figure 1.9, find the output of the neuron Y when the activation function is



1- **Logistic (Log-Sigmoid) Transfer Function:**

2-**liner function**

Solution:

Net input to neuron Y is:

$$y_{in} = (0.8\times[-0.2]) + (0.3\times0.3) + (0.2\times0.8) + (0.6\times0.5) + 0.25$$
$$= 0.64$$

1- **Logistic (Log-Sigmoid) Transfer Function:**

$$y = f(y_{in}) = \frac{1}{1+e^{-0.64}} = 0.6548$$

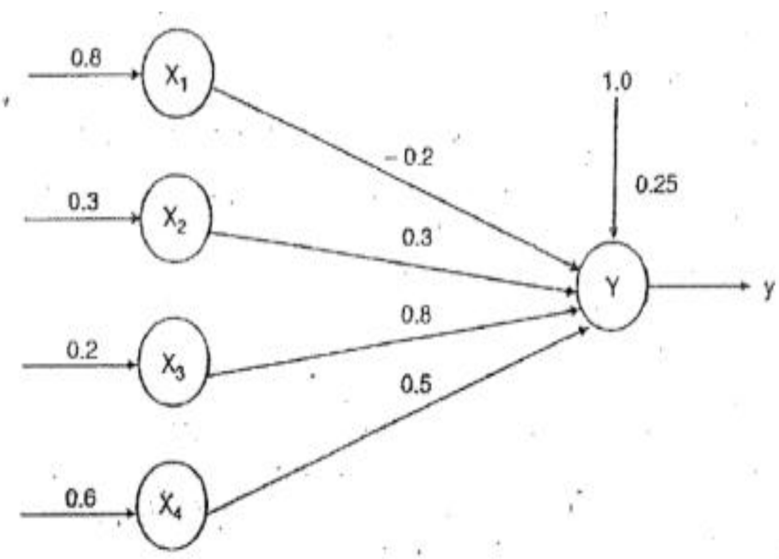
2-**liner function**

$$y = f(y_{in}) =$$

0.64

H.W

Example 1.1 For the network shown in Figure 1.9, find the output of the neuron Y when the activation function is



1- **Threshold (Hard Limit) Transfer Function:**

2- **Radial Basis Transfer Function:**

3-liner function

4- hyberbolic function

5- **Symmetric Hard Limit Transfer Function**

6- **Positive Linear Transfer Function:**

7- **Logistic (Log-Sigmoid) Transfer Function:**

Exampl2 -2

If the net input to an output neuron is 0.64 calculate its output when the activation function

1- Threshold (Hard Limit) Transfer Function:

2- Radial Basis Transfer Function:

3-liner function

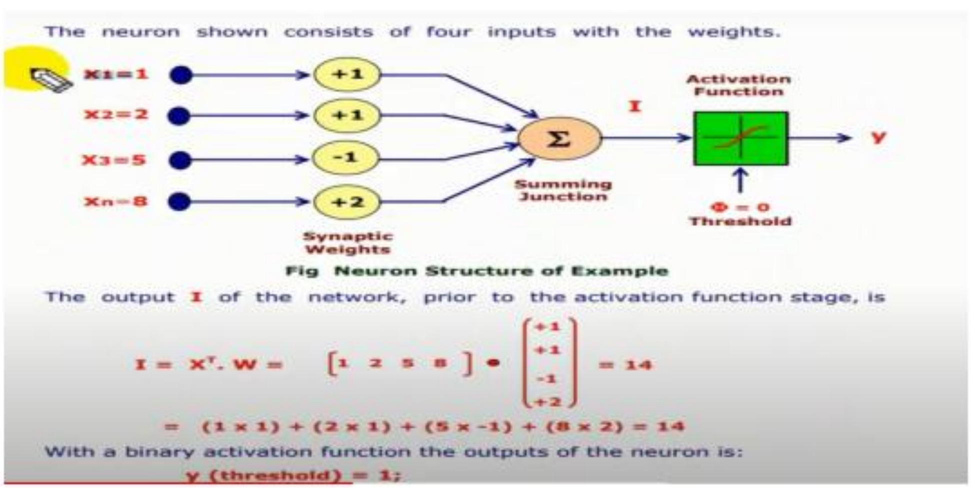
4- hyberbolic function

5- Symmetric Hard Limit Transfer Function

6-Positive Linear Transfer Function:

7- Logistic (Log-Sigmoid) Transfer Function:

Exampl3 -3



```
input        weight
-----        --------

x1=1         w1=1
x2=2         w2=1
x3=5         w3=-1
x4=8         w4=2


net=1*1+2*1+5*-1+8*2

net=14

binry threshold

y=1
```

```
input        weight
-----        --------

x1=1         w1=1
x2=2         w2=1
x3=5         w3=-1
x4=8         w4=2


net=1*1+2*1+5*-1+8*2

net=14

bibolar threshold

y=1
```

- **Example :**

The neuron shown consists of four inputs with the weights.

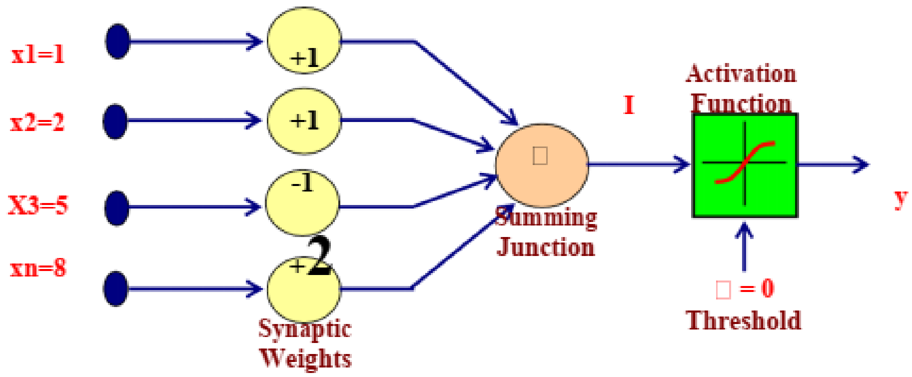


Fig Neuron Structure of Example

The output **I** of the network, prior to the activation function stage, is

$$I = X^T . W = \begin{bmatrix} 1 & 2 & 5 & 8 \end{bmatrix} \bullet \begin{bmatrix} +1 \\ +1 \\ -1 \\ +2 \end{bmatrix} = 14$$

$$= (1 \times 1) + (2 \times 1) + (5 \times -1) + (8 \times 2) = 14$$

With a binary activation function the outputs of the neuron is:

y (threshold) = 1;

2.2 Network Architectures:

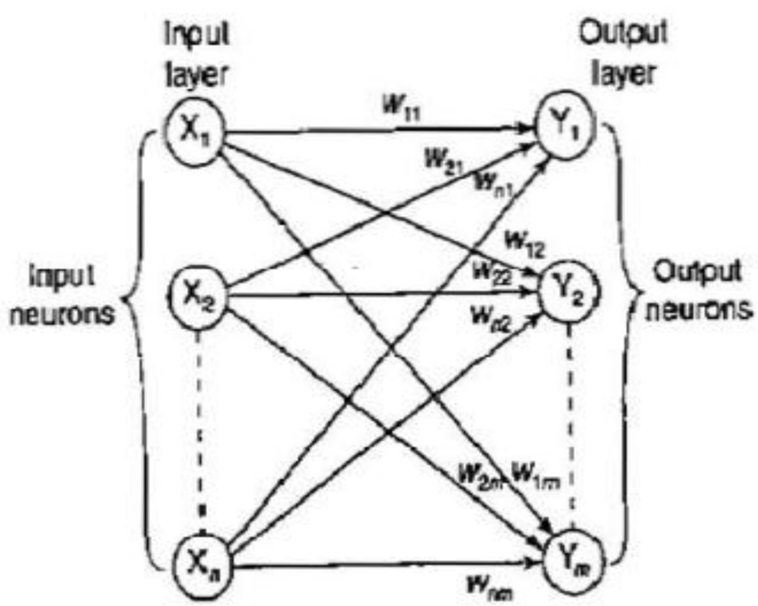
Commonly one neuron, even with many inputs, may not be sufficient. We might need five or ten, operating in parallel, in what we will call a "layer". This concept of a layer is discussed below.

There exist five basic types of neuron connection architectures. They are:

1. single-layer feed-forward network
2. Multilayer feed-forward network
3. Single node with its own feedback
4. single-layer recurrent network
5. Multilayer recurrent network

1. single-layer feed-forward network

A layer implies a stage, going stage by stage, i.e., the input stage and the output stage are linked with each other. These linked interconnections lead to the formation of various network architectures. When a layer of the processing nodes is formed, the inputs can be connected to these nodes with various weights, resulting in a series of outputs, one per node. Thus, a single-layer *feed-forward network* is formed.



Each element of the input vector **p** is connected to each neuron through the weight matrix **W**. Each neuron has a bias b_i , a summer, a transfer function f and an output a_i . Taken together, the outputs form the output vector.

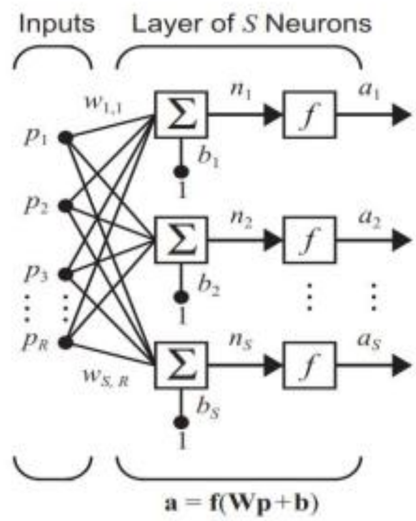


Figure 15: Single Layer of *S* Neurons

The input vector elements enter the network through the weight matrix **W**:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

Fortunately, the S-neuron, R-input, one-layer network also can be drawn in abbreviated notation, as shown in Figure 16.

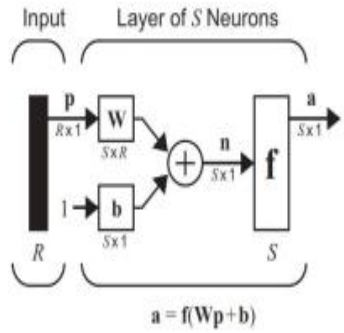


Figure 16: Layer of S Neurons, Abbreviated Notation

2. Multilayer feed-forward network

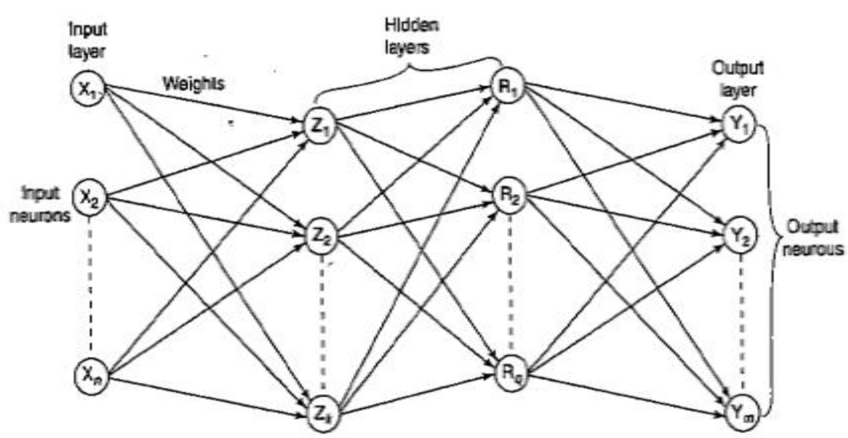
A multilayer feed-forward network is formed *by* the interconnection of several layers.

The input layer is that which receives the input and this layer has no function except buffering the input signal.

The output layer generates the output of the network.

Any layer that is formed between e input and output layers is called **hidden layer**. This hidden layer is internal to the network and has no direct contact with the external environment. There may be zero to several hidden layers in an ANN.

More the number of the hidden layers, more is the complexity of the network.



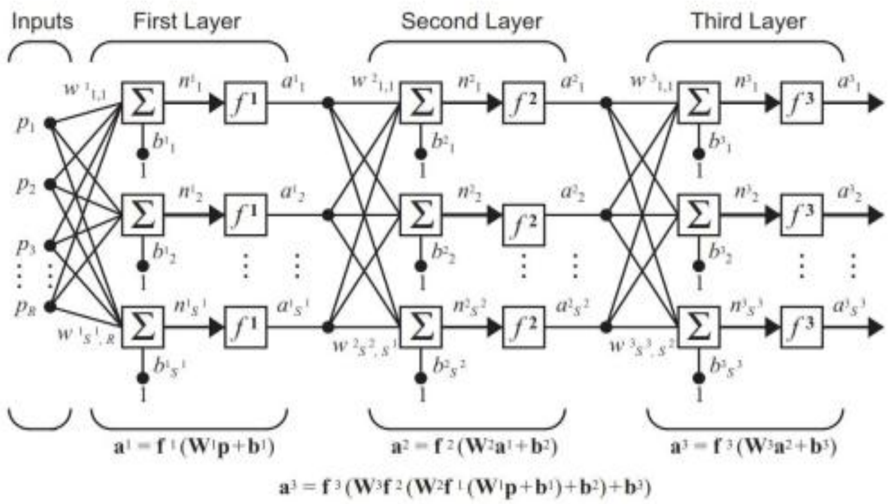


Figure 17: Three-Layer Network

As shown, there are R inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, etc. As noted, different layers can have different numbers of neurons.

The same three-layer network discussed previously also can be drawn using our abbreviated notation, as shown in Figure 18.

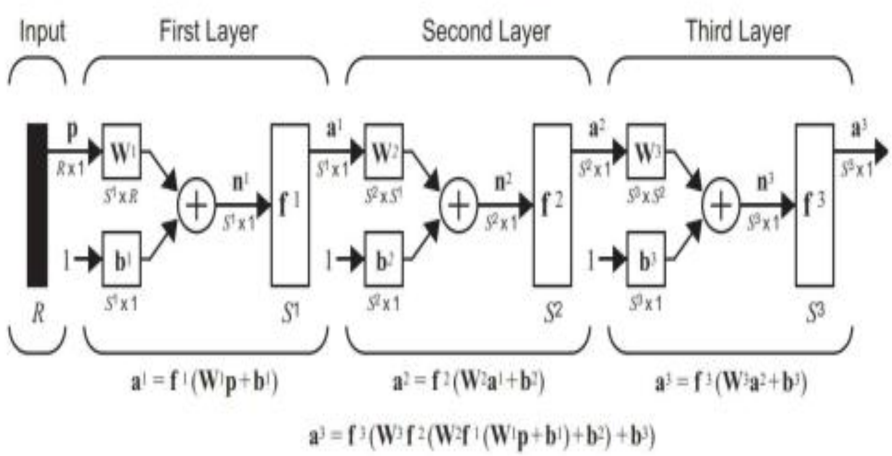


Figure 18: Three-Layer Network, Abbreviated Notation

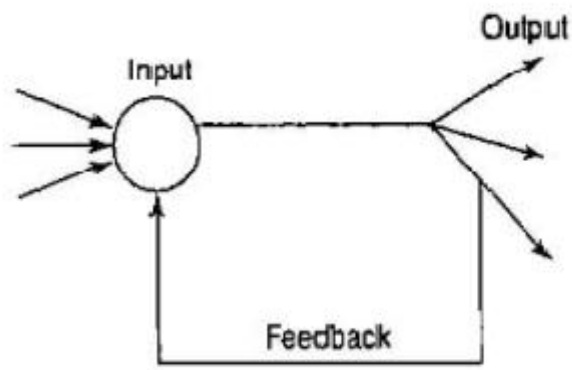
3. Single node with its own feedback

A network is said to be a feed forward network if no neuron in the output layer is an input to a node in the same layer or in the preceding layer.

When outputs can be directed back as inputs to same or preceding layer nodes then it results in the formation of feedback networks.

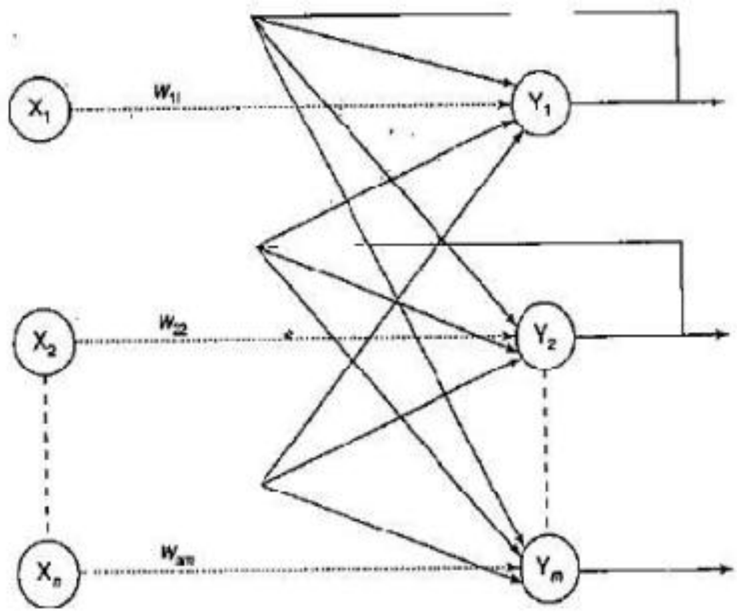
If the feedback of the output of the processing elements is directed back as input to the processing elements in the same layer then it is called ***lateral feedback***. Recurrent networks are feedback networks with closed loop.

The Figure below shows a simple recurrent neural network having a single neuron with feedback to itself.



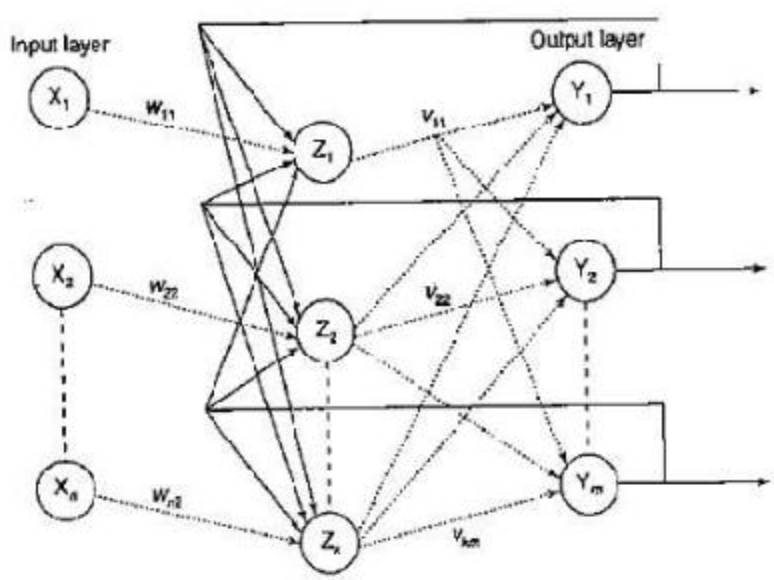
4. single-layer recurrent network

The Figure below shows a single layer network with a feedback connection in which a processing element's output can be directed back to the processing element itself or to the other processing element or to both.



5. Multilayer recurrent network

A processing element output can be directed back to the nodes in a preceding layer, forming a ***multilayer recurrent network***. In these networks, a processing element output can be directed back to the processing element itself and to other processing elements in the same layer.



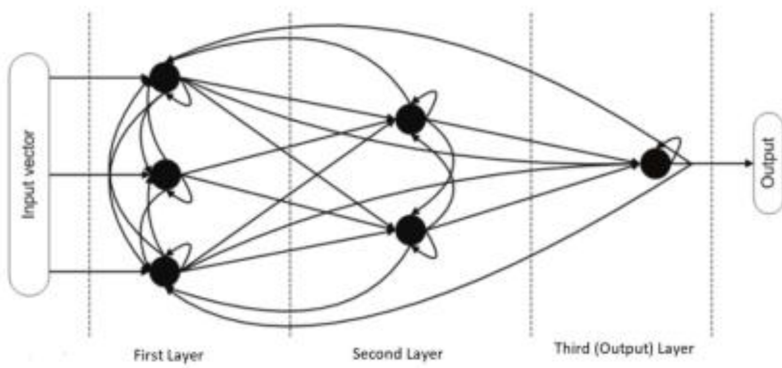


Figure 19: Fully Recurrent Artificial Neural Network
18

Other recurrent artificial neural networks such as Hopfield, Elman, Jordan, bi-directional and other networks are just special cases of recurrent artificial neural networks.

How to Pick an Architecture:

Problem specifications help define the network in the following ways:

1. Number of network inputs = number of problem inputs.

2. Number of neurons in output layer = number of problem outputs.

3. Output layer transfer function choice at least partly determined by problem specification of the outputs.

ANN Learning Methods

تصنف طرائق التعليم (ANN) الى ثلاثة طرائق رئيسية وهي:

1- التعليم المراقب Supervised Learning

2- التعليم غير المراقب Unsupervised Learning

3- التعليم بالتقوية Reinforcement Learning

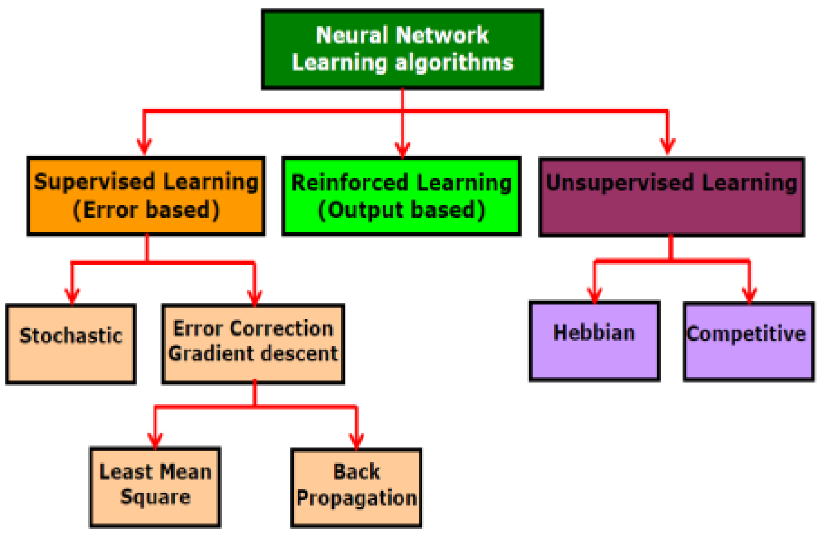
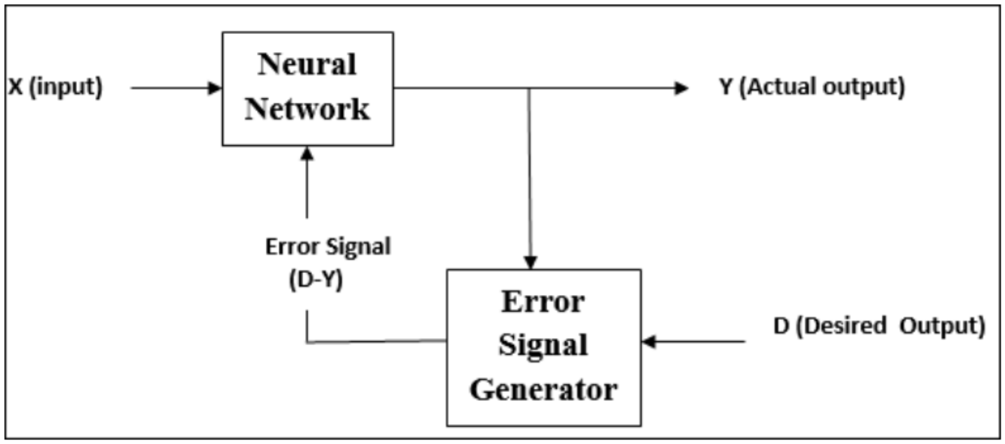
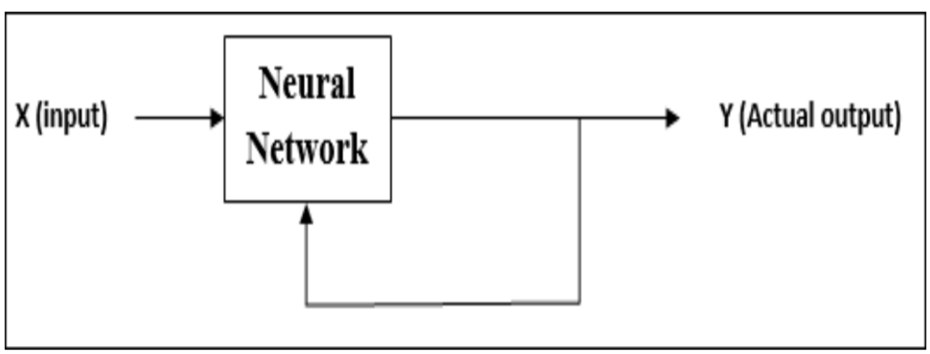


Figure 3: Different Training methods of Artificial Neural Network

Supervised Learning: As the name suggests, this type of learning is done under the supervision of a teacher. This learning process is dependent. During the training of ANN under supervised learning, the input vector is presented to the network, which will give an output vector. This output vector is compared with the desired output vector. An error signal is generated, if there is a difference between the actual output and the desired output vector. On the basis of this error signal, the weights are adjusted until the actual output is matched with the desired output.



Unsupervised Learning: As the name suggests, this type of learning is done without the supervision of a teacher. This learning process is independent. During the training of ANN under unsupervised learning, the input vectors of similar type are combined to form clusters. When a new input pattern is applied, then the neural network gives an output response indicating the class to which the input pattern belongs. There is no feedback from the environment as to what should be the desired output and if it is correct or incorrect. Hence, in this type of learning, the network itself must discover the patterns and features from the input data, and the relation for the input data over the output.



Reinforcement Learning: As the name suggests, this type of learning is used to reinforce or strengthen the network over some critic information. This learning process is similar to supervised learning, however we might have very less information. During the training of network under reinforcement learning, the network receives some feedback from the environment. This makes it somewhat similar to supervised learning. However, the feedback obtained here is evaluative not instructive, which means there is no teacher as in supervised learning. After receiving the feedback, the network performs adjustments of the weights to get better critic information in future.