Module I

Introduction to Soft Computing Artificial neural networks - biological neurons, Basic models of artificial neural networks – Connections, Learning, Activation Functions, McCulloch and Pitts Neuron, Hebb network.

Introduction to Soft Computing

Neural-Networks

The neural networks have the ability to learn by example which makes them very flexible and powerful.

For neural networks, there is no need to devise an algorithm to perform a specific task that is, there is no need to understand the internal mechanisms of that task. These networks are also well suited for real time systems because of their fast response and computational times which are because of their parallel architecture.

Artificial Neural Network: Definition

An artificial neural network (ANN) may be defined as an information processing model that is inspired by the way biological nervous systems, such as the brain, process information. This model tries to replicate only the most basic functions of the brain.

An ANN is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems.

Advantages of Neural Networks

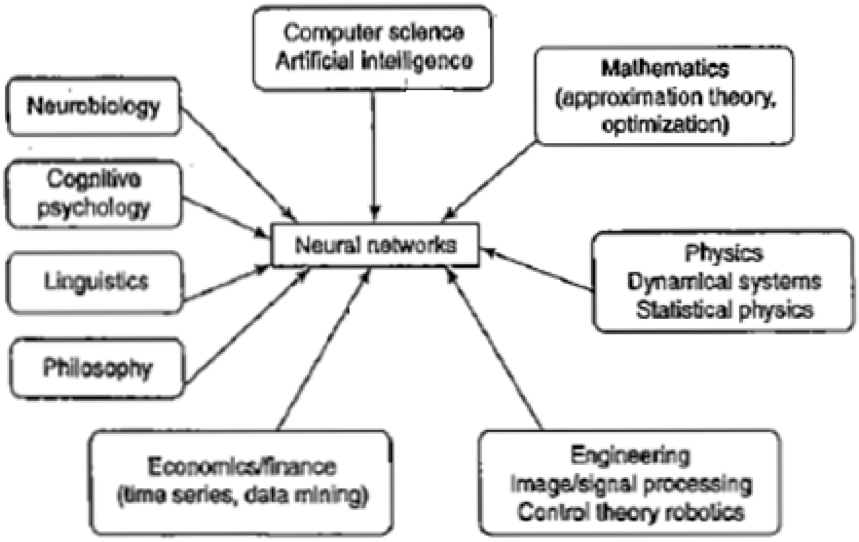
l. ***Adaptive learning:*** An ANN *is* endowed with the ability m learn how to do tasks based on the data given for training or initial experience.

2. ***Self-organization:*** An ANN can create its own organization or representation of the information it receives during learning time.

3. ***Real-time operation:*** ANN computations may be carried out in parallel. Special hardware devices are being designed and manufactured to take advantage of this capability of ANNs.

4. ***Fault tolerance via redundant information coding:*** Partial destruction of a neural network leads to the corresponding degradation of performance. However, some capabilities may be retained even after major network damage.

Neural networks can be viewed from a multi-disciplinary point of view as shown in Figure below:



<u>Application Scope of Neural Networks</u>

1. ***<u>Air traffic control</u>*** could be automated with the location, altitude, direction and speed of each radar blip taken as input to the network. The output would be the air traffic controller's instruction in response to each blip.

2. ***<u>Animal behavior, predator/prey relationships</u>*** **and** ***<u>population cycles</u>*** may be suitable for analysis by neural networks.

3. ***<u>Appraisal and valuation</u>*** of property, buildings, automobiles, machinery, etc. should be an easy task for a neural network.

4. ***<u>Betting</u>*** **<u>on horse races, stock markets, sporting events</u>**, etc. could be based on neural network predictions.

5. ***<u>Criminal sentencing</u>*** could be predicted using a large sample of crime details as input and the resulting sentences as output.

6. ***Complex physical and chemical processes*** that may involve the interaction of numerous (possibly unknown) mathematical formulas could be ·modeled heuristically using a neural network.

7. ***Data mining, cleaning and validation*** could be achieved by determining which records suspiciously diverge from the pattern of their peers.

8. ***Direct mail advertisers*** could use neural network analysis of their databases to decide which customers should be targeted, and avoid wasting money on unlikely targets.

9. ***Echo patterns*** from sonar, radar, seismic and magnetic instruments could be used to predict their targets.

10. ***Econometric modeling*** based on neural networks should be more realistic than older models based on classical statistics.

11. ***Employee hiring*** could be optimized if the neural networks were able to predict which job applicant would *show* the best job performance.

12. ***Expert consultants*** could package their intuitive expertise into a neural network to automate their services.

13. ***Fraud detection*** regarding credit cards, insurance or taxes could be automated using a neural network analysis of past incidents.

14. ***Handwriting and typewriting*** could be recognized by imposing a grid over the writing, then each square of the grid becomes an input to the neural network. This is called "**Optical Character Recognition**."

15. ***Lake water levels*** could be predicted based upon precipitation patterns and river/dam flows.

16. ***Machinery control*** could be automated by capturing the actions of experienced machine operators into a neural network.

17. ***Medical diagnosis*** is an ideal application for neural networks.

18. ***Medical research*** relies heavily on classical statistics to analyze research data. Perhaps a neural network should be included in the researcher's tool kit.

19. ***Music composition*** has been tried using neural networks. The network *is* trained to recognize patterns in the pitch and tempo of certain music, and then the network writes its own music.

20. ***Photos and fingerprints*** could be recognized by imposing a fine grid over the photo. Each square of the grid becomes an input to me neural network.

21. ***Chemical formulations*** could be optimized based on the predicted outcome of a formula change.

22. ***Retail inventories*** could be optimized by predicting demand based on past patterns.

23. ***River water levels*** could be predicted based on upstream reports, and time and location of each report.

24. ***Scheduling of buses, airplanes and elevators*** could be optimized by predicting demand.

25. ***Staff scheduling*** requirements for restaurants, retail stores, police stations, banks, etc., could be predicted based on the customer flow, day of week, paydays, holidays, weather, season, ere.

26. ***Strategies* for games, business and war** can be captured by analyzing the expert player's response to given stimuli.

27. ***Traffic flows*** could be predicted so that signal timing could be optimized.

28. ***Voice recognition*** could be obtained by analyzing the audio oscilloscope pattern, much like a stock market graph.

29. ***Weather prediction*** may be possible. Inputs would include weather reports from surrounding areas. Output(s) would be the future weather in specific areas based on the input information.

Fuzzy Logic

Fuzzy logic is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded microcontrollers to large, networked, multichannel PC or workstation based data acquisition and control systems. It can be implemented in hardware, software or a combination of both.

FL Provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FLs approach to control problems mimics how a person would make decisions, only much faster.

Genetic Algorithm

Genetic algorithms are adaptive computational procedures modeled on the mechanics of natural generic systems. They express their ability by efficiently exploiting the historical information to speculate on new offspring with expected improved performance.

GAs are executed iteratively on a set of coded solutions, called population, with three basic operators: selection/reproduction, crossover and mutation.

They use only the payoff (objective function) information and probabilistic transition rules for moving to the next iteration. They are different from most of the normal optimization and search procedures in the following four ways:

1. GAs work with the coding of the parameter set, not with the parameter themselves;
2. *GAs* work simultaneously with multiple points, not a single point;
3. GAs search via sampling (a blind search) using only the payoff information;
4. GAs search using stochastic operators, not deterministic rules.

Hybrid Systems

Hybrid systems can be classified into three different systems:

- Neuro fuzzy hybrid system
- Neuron generic hybrid system
- Fuzzy genetic hybrid systems

Neuro Fuzzy Hybrid Systems

A neuro fuzzy hybrid system is a fuzzy system that uses a learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples.

1. It can handle any kind of information (numeric, linguistic, logical, etc.).
2. It can manage imprecise, partial, vague or imperfect information.
3. It can resolve conflicts by collaboration and aggregation.
4. It has self-learning, self-organizing and self-tuning capabilities.
5. It doesn't need prior knowledge of relationships of data.
6. It can mimic human decision-making process.
7. It makes computation fast by using fuzzy number operations.

Neuro Genetic Hybrid Systems

Genetic algorithms {GAs) have been increasingly applied in ANN design in several ways: topology optimization, genetic training algorithms and control parameter optimization.

In **topology optimization**, GA is used to select a topology for the ANN which in turn is trained using some training scheme, most commonly back propagation.

In **genetic training algorithms**, the learning of an ANN is formu1ated as a weight optimization problem, usually using the inverse mean squared error as a fitness measure.

Many of the **control parameters** such as learning rate, momentum rate, tolerance level, etc., can also be optimized using *GAs*.

Fuzzy Genetic Hybrid Systems

The optimization abilities of GAs are used to develop the best set of rules to be used by a fuzzy inference engine, and to optimize the choice of membership functions. A particular use of GAs is in fuzzy classification systems, where an object is classified on the basis of the linguistic values of the object attributes.

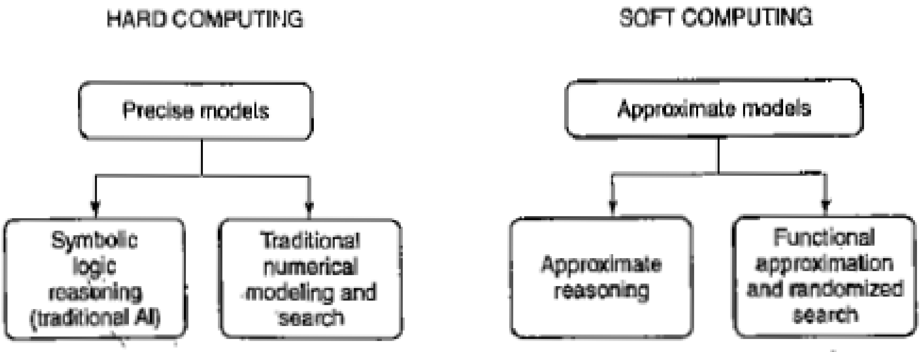
Soft Computing

The two major problem-solving technologies include:

1. Hard computing
2. Soft computing.

Hard computing deals with precise models where accurate solutions are achieved quickly.

Soft computing deals with approximate models and gives solution to complex problems. The two problem-solving technologies are shown in Figure below:



Soft computing uses a combination of GAs, neural networks and FL. An important thing about the constituents of soft computing is that they are complementary, not competitive, offering their own advantages and techniques to partnerships to allow solutions to otherwise unsolvable problems.

Artificial Neural Network

Neural networks are those information processing systems, which are constructed and implemented to model the human brain.

Objective

The main objective of the neural network is to develop a computational device for modeling the brain to perform various computational tasks at a faster rate than the traditional systems.

Tasks

Artificial neural networks perform various tasks such as

- pattern matching and classification
- optimization function
- approximation
- vector quantization
- data clustering.

These tasks are very difficult for traditional Computers. Therefore, for implementation of artificial networks high speed digital computers are used.

Artificial Neural Network

An artificial neural network (ANN) is an efficient information processing system which resembles in characteristics with a biological neural network.

ANNs possess large number of highly interconnected processing elements called ***nodes*** **or** ***units*** **or** ***neurons***.

Each neuron is connected with the other by a connection link.

Each connection link is associated with weights which contain information about the input signal.

This information is used by the neuron net to solve a particular problem.

ANNs' collective behavior is characterized by their ability to learn. They have the capability to model networks of original neurons as found in the brain. Thus, the ANN processing elements are called *neurons* or *artificial neurons.*

Basic operation of a neural net

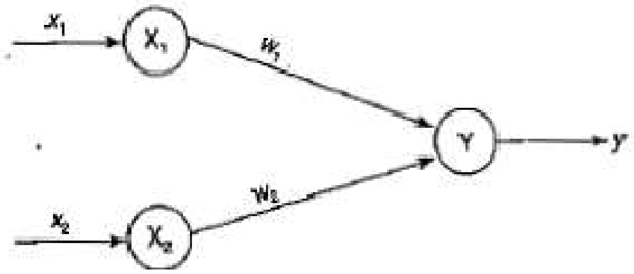
Each neuron has an internal stare of its own. This internal state is called *activation* or *activity* level of neuron, which is the function of the inputs the neuron receives. The activation signal of a neuron is transmitted to other neurons.

A neuron can send only one signal at a time, which can be transmitted to several ocher neurons.

To depict the basic operation of a neural net, consider a set of neurons, say X_1 and X_2, transmitting signals to another neuron, Y.

Here X_1, and X_2 are input neurons, which transmit signals, and Y is the output neuron, which receives signals.

Input neurons X_1, and $X_2$are connected to the output neuron Y, over a weighted interconnection links (W_1, and W_2) as shown in Figure.



For the above simple neuron net architecture, the net input has to be calculated in the following way:

$$y_{in} = +\mathbf{x_1 w_1} + \mathbf{x_2 w_2}$$

x_1 and x_2 →activations of the input neurons X_1, and X_2, i.e., the output of input signals.

The output *y* of the output neuron Y can be obtained by applying activations over the net input, i.e., the function of the net input:

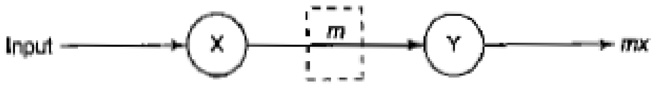
$$y = \mathbf{f(y_{in})}$$

Output= Function (net input calculated)

The function to be applied over the net input is called *activation function.*

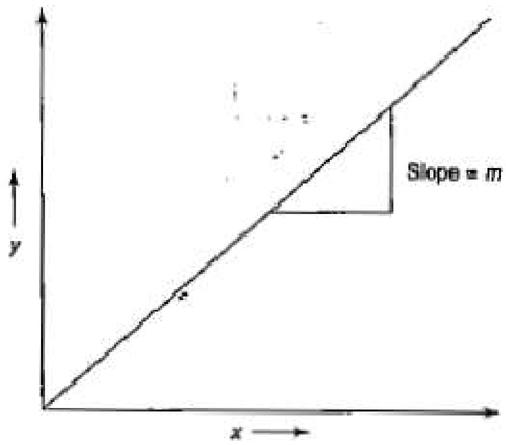
Linear straight line equation

The above calculation of the net input is similar to the calculation of output of a pure linear straight line equation (y = *mx).* The neural net of a pure linear equation is as shown in Figure:



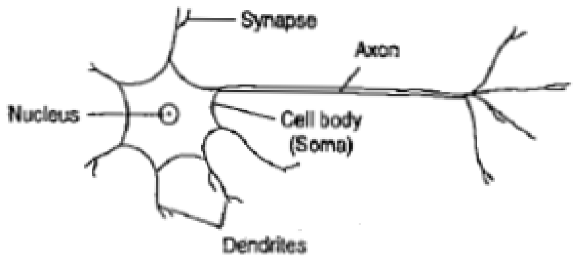
Here, to obtain the output *y,* the slope *m* is directly multiplied with the input signal. This is a linear equation.

Thus, when slope and input are linearly varied, the output *is* also linearly varied, as shown in Figure below. This shows that the weight involved in the ANN is equivalent to the slope of the linear straightline.



Biological Neural Network

A schematic diagram of a biological neuron is shown in Figure below:



The biological neuron depicted in Figure, consists of three main parts:

1. *Soma* or *cell body*- where the cell nucleus is located.

2. *Dendrites*- where the nerve is connected to the cell body.

3. *Axon*- which carries the impulses of the neuron.

Dendrites are tree-like networks made of nerve fiber connected to the cell body.

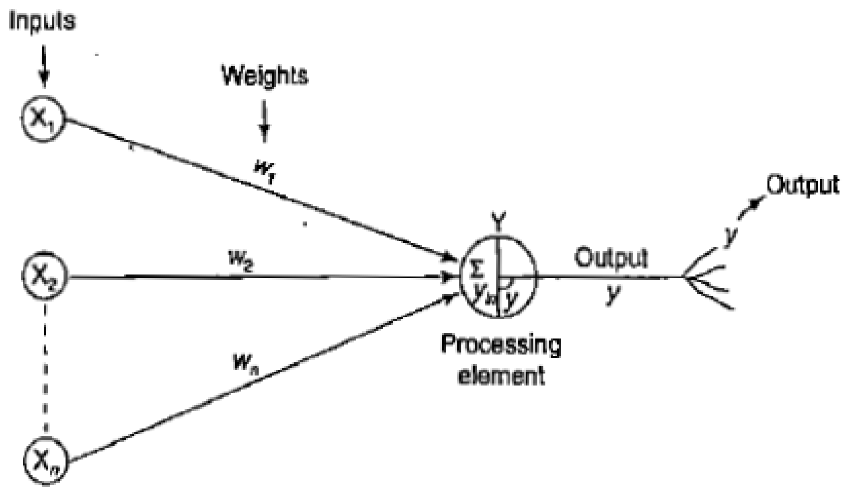
An axon is a single, long connection extending from the cell body and carrying signals from the neuron. The end of the axon splits into fine strands. It is found that each strand terminates into a small bulb like organ called synapse. It is through synapse that the neuron introduces its signals to other nearby neurons. The receiving ends of these synapses on the nearby neurons can be found both on the dendrites and on the cell body. There are approximately 10^4 synapses per neuron in the human brain.

Electric impulses are passed between the synapse and the dendrites. This type of signal transmission involves a chemical process in which specific transmitter substances are released from the sending side of the junction. This result in increase or decrease in the electric potential inside the body of the receiving cell.

If the electric potential reaches a threshold then the receiving cell fires and a *pulse* or *action potential* of fixed strength and duration *is* sent out through the axon to the synaptic junctions of the other cells. After firing, a cell has to wait for a period of time called the refractory period before it can fire again.

The synapses are said to be *inhibitory* if they let passing impulses hinder the firing of the receiving cell or *excitatory* if they let passing impulses cause the firing of the receiving cell.

The Figure below shows a mathematical representation of the chemical processing taking place in an artificial neuron.



In this model, the net input is elucidated as

$$y_{in} = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n = \sum_{i=1}^{n} x_i w_i$$

where i represents the ith processing element. The activation function is applied over it to calculate the output. The weight represents the strength of synapse connecting the input and the output neurons.

A positive weight corresponds to an excitatory synapse, and a negative weight corresponds to an inhibitory synapse.

The terms associated with the biological neuron and their counterparts in artificial neuron are in Table below.

Biological neuron	Artificial neuron
Cell	Neuron
Dendrites	Weights or interconnections
Soma	Net input
Axon	Output

Brain vs. Computer - Comparison between Biological Neuron and Artificial Neuron

1. Speed:

The cycle time of execution in the ANN is of **few nanoseconds** whereas in the case of biological neuron it is of a **few milliseconds**. Hence, the artificial neuron modeled using a computer is faster.

2. Processing:

Basically, the biological neuron can perform massive parallel operations simultaneously. The artificial neuron can also perform several parallel operations simultaneously, **but,** in general, the artificial neuron network process is **faster than that of the brain**.

3. Size and complexity:

The total number of neurons in the brain is about 10^{11} and the total number of interconnections is about 10^{15}. Hence, it can be noted that the complexity of the brain is comparatively higher, i.e. the computational work takes places not only in the brain cell body, but also in axon, synapse, etc. On the other hand, the size and complexity of an ANN is based on the chosen application and the network designer. The size and complexity of a biological neuron is more than that of an artificial neuron.

4. *Storage capacity (memory):*

The biological neuron stores the information in its interconnections or in synapse strength but in an artificial neuron it is stored in its contiguous memory locations.

In an artificial neuron, the continuous loading of new information may sometimes overload the memory locations. *As* a result, some of the addresses containing older memory locations may be destroyed. But in case of the brain, new information can be added in the interconnections by adjusting the strength without destroying the older information.

A disadvantage related to brain is that sometimes its memory may fail to recollect the stored information whereas in an artificial neuron, once the information is stored in its memory locations, it can be retrieved.

5. Tolerance*:*

The biological neuron assesses fault tolerant capability whereas the artificial neuron has no fault tolerance. The distributed nature of the biological neurons enables to store and retrieve information even when the interconnections in them get disconnected. Thus biological neurons are fault tolerant. But in case of artificial neurons, the information gets corrupted if the network interconnections are disconnected.

Biological neurons can accept redundancies, which is not possible in artificial neurons. Even when some cells die, the human nervous system appears to be performing with the same efficiency.

6. Control mechanism:

In an artificial neuron modeled using a computer, there is a control unit present in Central Processing Unit, which can transfer and control precise scalar values from unit to unit, but there is no such control unit for monitoring in the brain. The strength of a neuron in the brain depends on the active chemicals present and whether neuron connections are strong or weak as a result of structure layer rather than synapses.

However, the ANN possesses simpler interconnections and is free from chemical actions similar to those taking place in brain (biological neuron). Thus, the control mechanism of an artificial neuron is very simple compared to that of a biological neuron.

Characteristics of ANN:

1. It is a neurally implemented mathematical model.
2. There exists a large number of highly interconnected processing elements called neurons in an ANN.
3. The interconnections with their weighted linkages hold the informative knowledge.
4. The input signals arrive at the processing elements through connections and connecting weights.
5. The processing elements of the ANN have the ability to learn, recall and generalize from the given data by suitable assignment or adjustment of weights.
6. The computational power can be demonstrated only by the collective behavior of neurons, and it should be noted that no single neuron carries specific information.

Basic models of artificial neural networks

The models of ANN are specified by the three basic entities namely:

1. The model's synaptic interconnections;
2. The training or learning rules adopted for updating and adjusting the connection weights;
3. Their activation functions.

Connections

The neurons should be visualized for their arrangements in layers. An ANN consists of a set of highly interconnected processing elements (neurons) such that each processing element output is found to be connected through weights to the other processing elements or to itself, delay lead and lag free connections are allowed.

Hence, the arrangements of these processing elements and geometry of their interconnections are essential for an ANN.

The arrangement of neurons to form layers and the connection pattern formed within and between layers is led the *network architecture.*

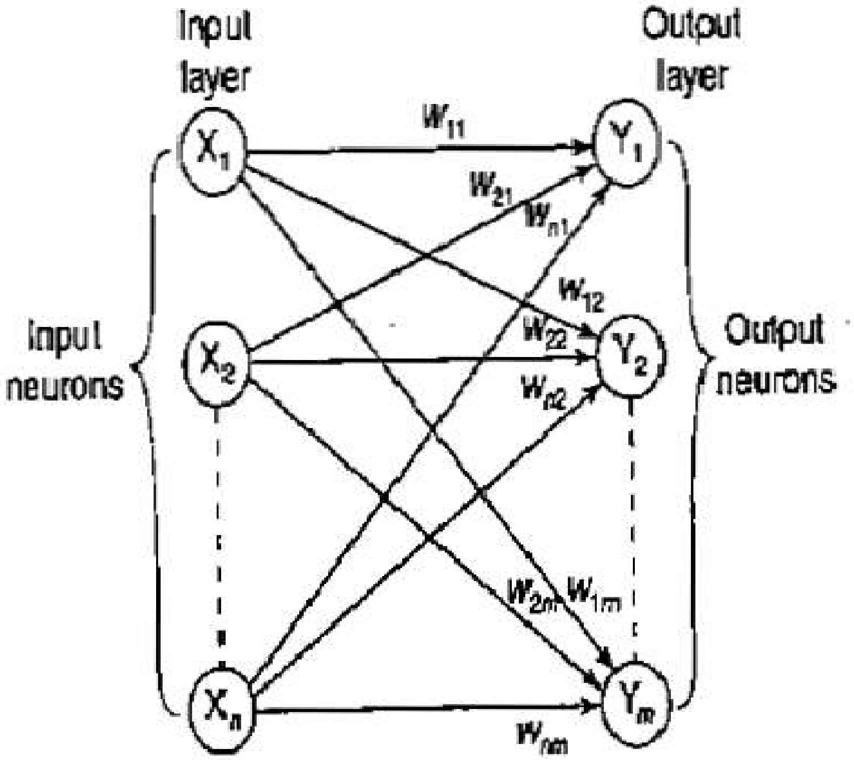
There exist five basic types of neuron connection architectures. They are:

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

Basically, neural nets are classified into single-layer or multilayer neural nets. A layer is formed by taking a processing element and combining it with other processing elements.

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.



2. Multilayer feed-forward network

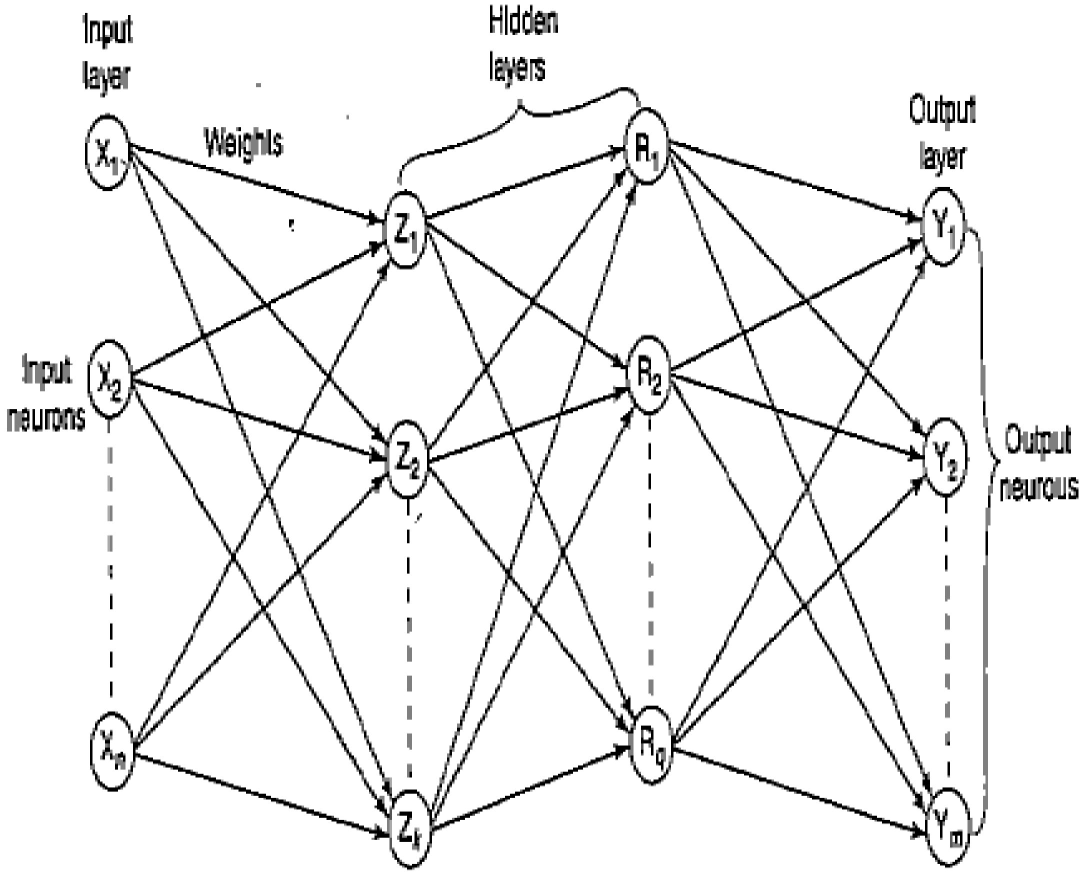
A multilayer feed-forward network is formed *by* the interconnection of several layers.

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.



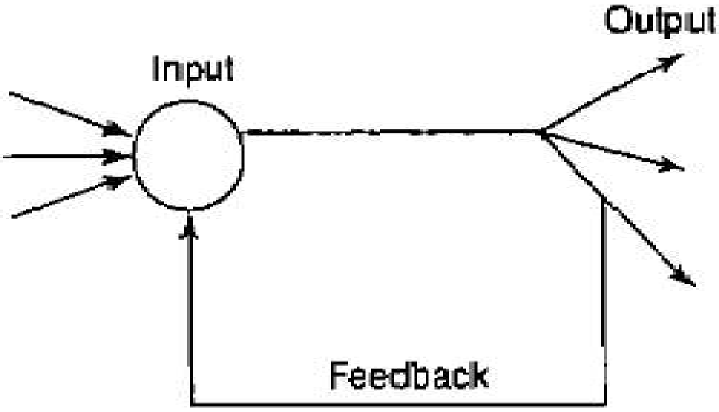
3. Single node with its own feedback

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.

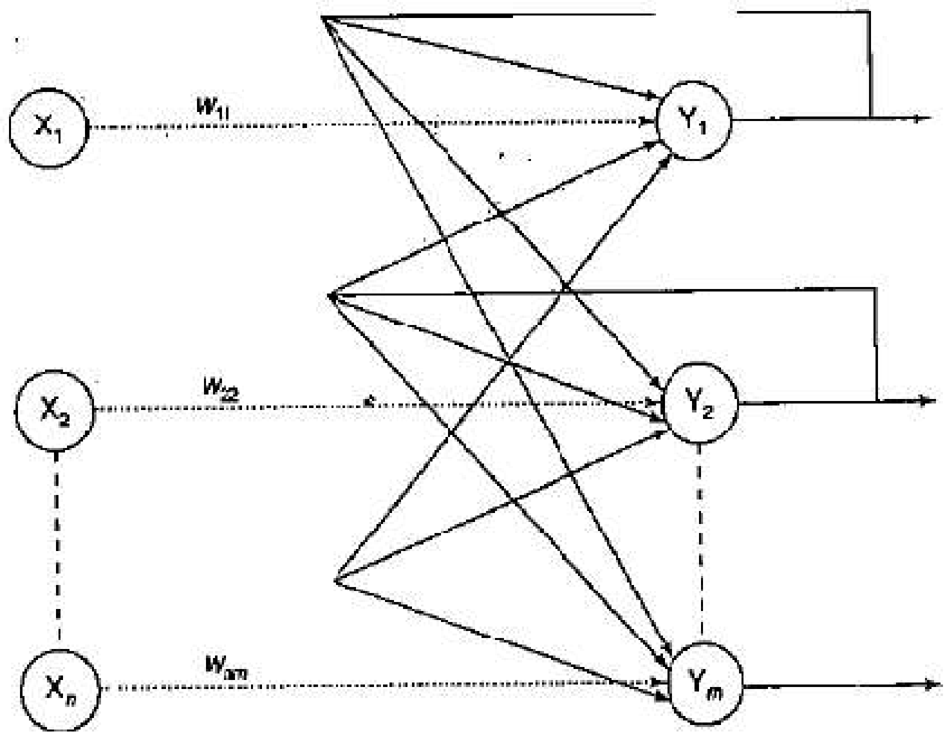
If the feedback of the output of the processing elements is directed back as input to the processing elements in the same layer then it is called ***lateral feedback***. Recurrent networks are feedback networks with closed loop.

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

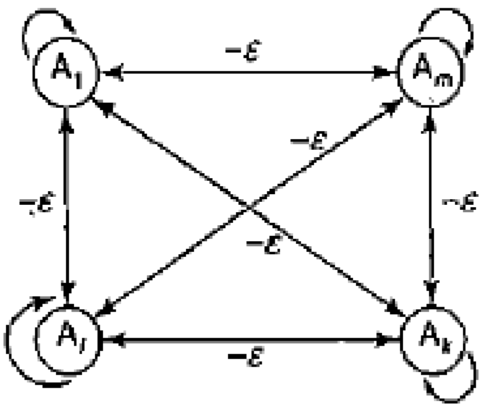


4. single-layer recurrent network

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.

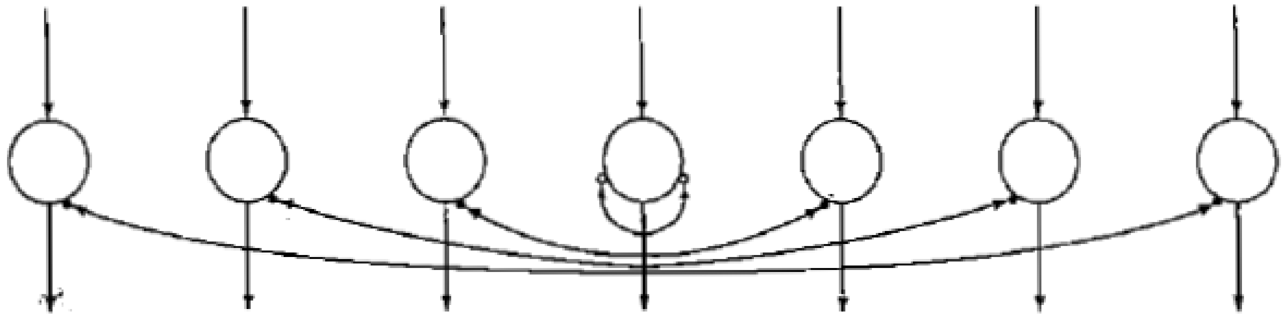


The architecture of a competitive layer is shown in Figure below, the competitive interconnections having fixed weights of *-e*. This net is called *Maxnet*.



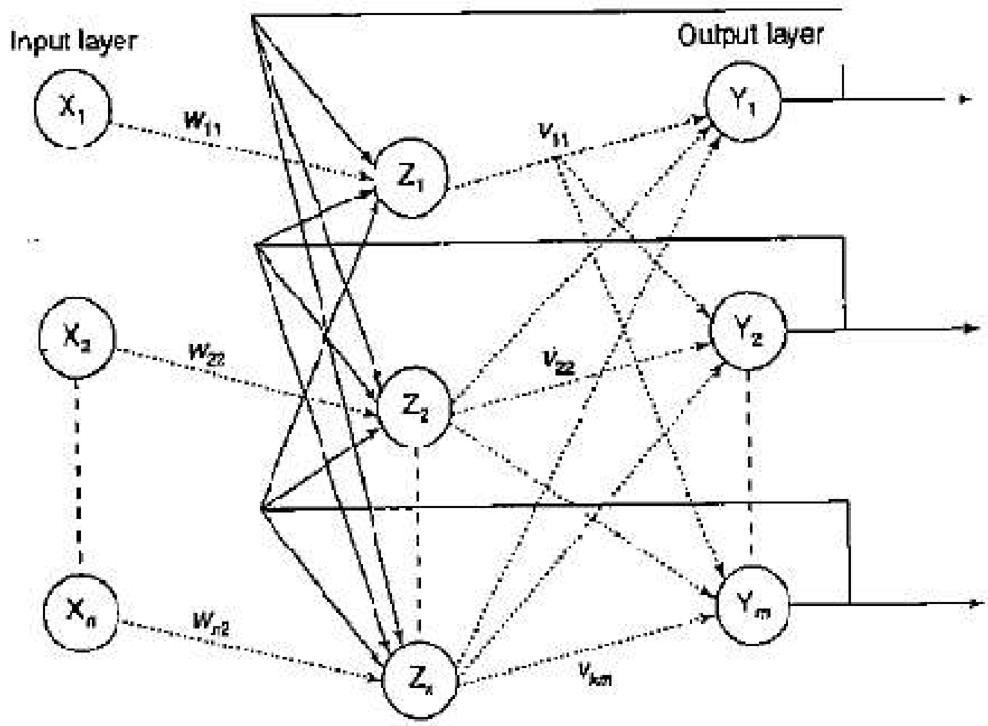
Lateral inhibition structure

In this structure, each processing neuron receives two different classes of inputs- "excitatory" input from nearby processing elements and "inhibitory" inputs from more distantly located processing elements. This type of interconnection is shown in Figure below:



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.



Learning

The main property of an ANN is its capability to learn. Learning or training is a process by means of which a neural network adapts itself to a stimulus by making proper parameter adjustments resulting in the production of desired response.

There are two kinds of learning in ANNs:

1. **Parameter learning:** It updates the connecting weights in a neural net.

2. **Structure learning**: It focuses on the change in network structure

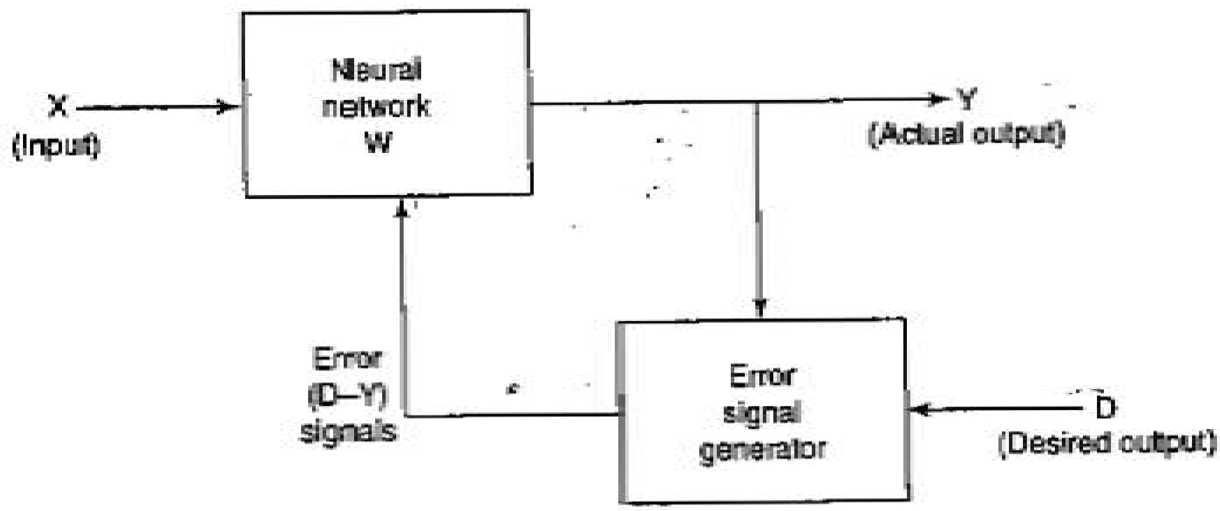
The above two types of learning can be performed simultaneously or separately.

Apart from these two categories of learning, the learning in an ANN can be generally classified into three categories as:

- **Supervised learning**
- **Unsupervised learning**
- **Reinforcement learning**

1) Supervised Learning

In ANNs following the supervised learning, each input vector requires a corresponding target vector, which represents the desired output. The input vector along with the target vector is called ***training pair.*** The network here is informed precisely about what should be emitted as output. The block diagram below depicts the working of a supervised learning network.



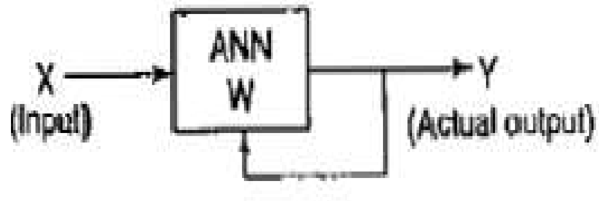
During training, the input vector is presented to the network, which results in an output vector. This output vector is the actual output vector. Then the actual output vector is compared with the desired (target) output vector. If there exists a difference between the two output vectors then an error signal is generated by the network. This error signal is used for adjustment of weights until the actual output matches the desired output. In this type of training, a supervisor is required for error minimization.

2) Unsupervised Learning

In ANNs following unsupervised learning, the input vectors of similar type are grouped without the use of training data to specify how

A member of each group looks or to which group a number belongs. In the training process, the network receives the input patterns and organizes these patterns to form clusters.

When a new input pattern is applied, the neural network gives an output response indicating the class to which the input pattern belongs. If for an input, a pattern class cannot be found then a new class is generated. The block diagram of unsupervised learning is shown in Figure below.



From Figure it is clear that there is no feedback from the environment to inform what the outputs should be or whether the outputs are correct.

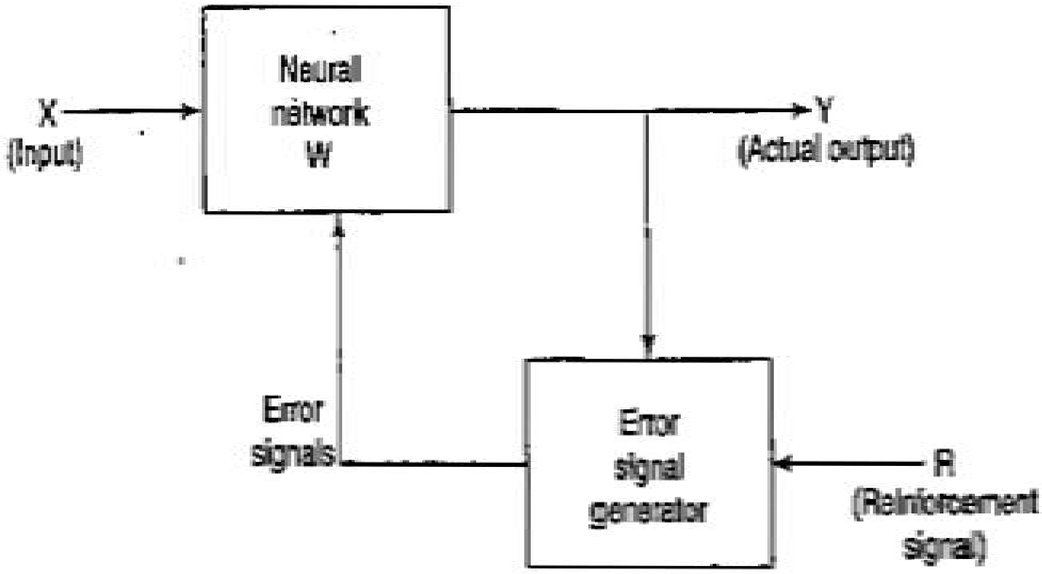
In this case, the network must itself discover patterns, regularities, features or categories from the input data and relations for the input data over the output. While discovering all these features, the network undergoes change in its parameters. This process *is* called *self organizing* in which exact clusters will be formed by discovering similarities and dissimilarities among the objects.

3) Reinforcement Learning

This learning process is similar to supervised learning. In the case of supervised learning, the correct target output values are known for each input pattern. But, in some cases, less information might be available.

For example, the network might be told that its actual output is only "50% correct" or so. Thus, here only critic information is available, nor the exact information. The learning based on this critic information is called **reinforcement learning** and the feedback sent is called **reinforcement signal**.

The block diagram of reinforcement learning is shown in Figure below:



The reinforcement learning is a form of supervised learning because the network receives some feedback from its environment. The reinforcement learning is also called learning with a critic as opposed to learning with a teacher, which indicates supervised learning.

Activation Functions

The activation function is applied over the net input to calculate the output of an ANN.

The information processing of a processing element can be viewed as consisting of two major parts: input and output.

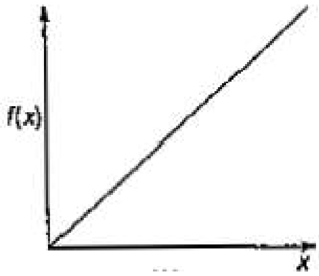
An integration function is associated with the input of a processing element. This function serves to combine activation, information or evidence from an external source or other processing elements into a net input to the processing element.

There are several activation functions. They are

1. Identity function: It is a linear function and can be defined as

$$f(x) = x \text{ for all } x$$

The output here remains the same as input. The input layer uses the identity activation function.

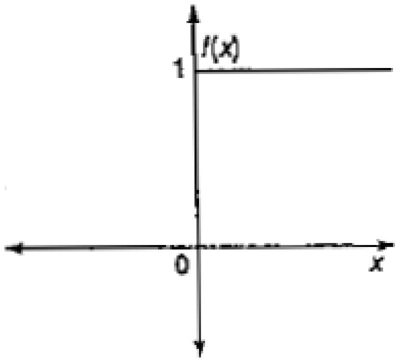


2. <u>Binary step function</u>: This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geqslant \theta \\ 0 & \text{if } x < \theta \end{cases}$$

where **θ** represents the threshold value.

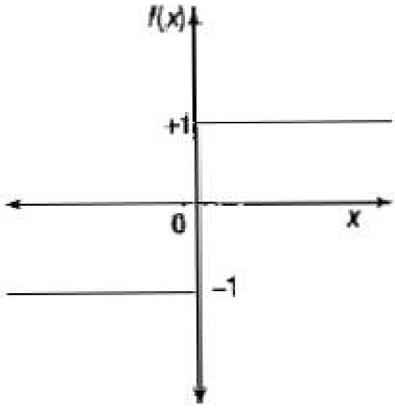
This function is most widely used in single-layer nets to convert the net input to an output that is a binary (1 or 0).



3. <u>Bipolar step function</u>: This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geqslant \theta \\ -1 & \text{if } x < \theta \end{cases}$$

where **θ** represents the threshold value. This function is also used in single-layer nets to convert the net input to an output that is bipolar (+ 1 or -1).



4. Sigmoidal functions: The sigmoidal functions are widely used in back-propagation nets because of the relationship between the value of the functions at a point and the value of the derivative at that point which reduces the computational burden during training.

Sigmoidal functions are of two types: -

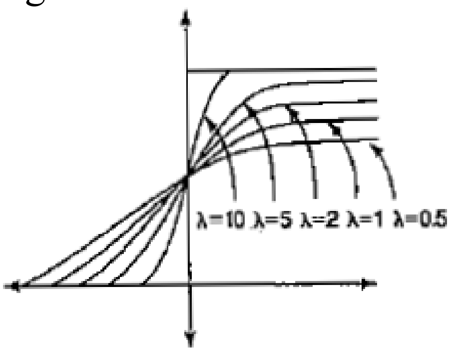
(i) Binary sigmoid function: It is also termed as logistic sigmoid function or unipolar sigmoid function. It can be defined as

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

where λ is the steepness parameter. The derivative of this function is

$$f'(x) = \lambda f(x)[1 - f(x)]$$

Here the range of the sigmoid function is from 0 to 1.

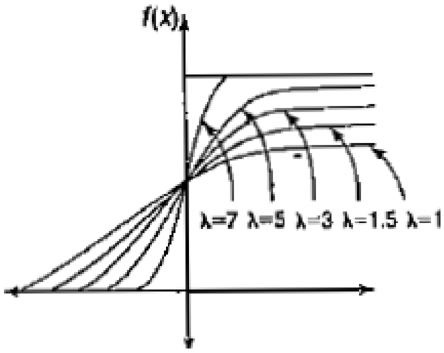


(ii) Bipolar sigmoid function: This function is defined as

$$f(x) = \frac{2}{1 + e^{-\lambda x}} - 1 = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}}$$

where λ is the steepness parameter and the range of the sigmoid function is between -1 and +1. The derivative of this function is

$$f'(x) = \frac{\lambda}{2}[1 + f(x)][1 - f(x)]$$



The bipolar sigmoidal function is closely related to hyperbolic tangent function, which is written as

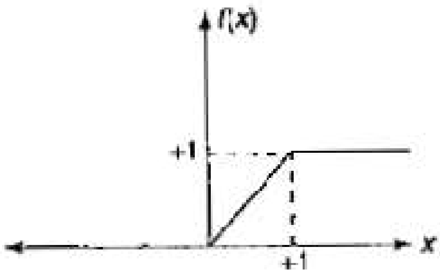
$$h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

The derivative of the hyperbolic tangent function is

$$h'(x) = [1 + h(x)][1 - h(x)]$$

5. Ramp function: The ramp function is defined as

$$f(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x < 0 \end{cases}$$



McCulloch and Pitts Neuron

Theory

The McCulloch-Pitts neuron was the earliest neural network discovered in 1943. It is usually called as M-P neuron.

The M-P neurons are connected by directed weighted paths. It should be noted that the activation of a M-P neuron is binary, that is, at any time step the neuron may fire or may not fire.

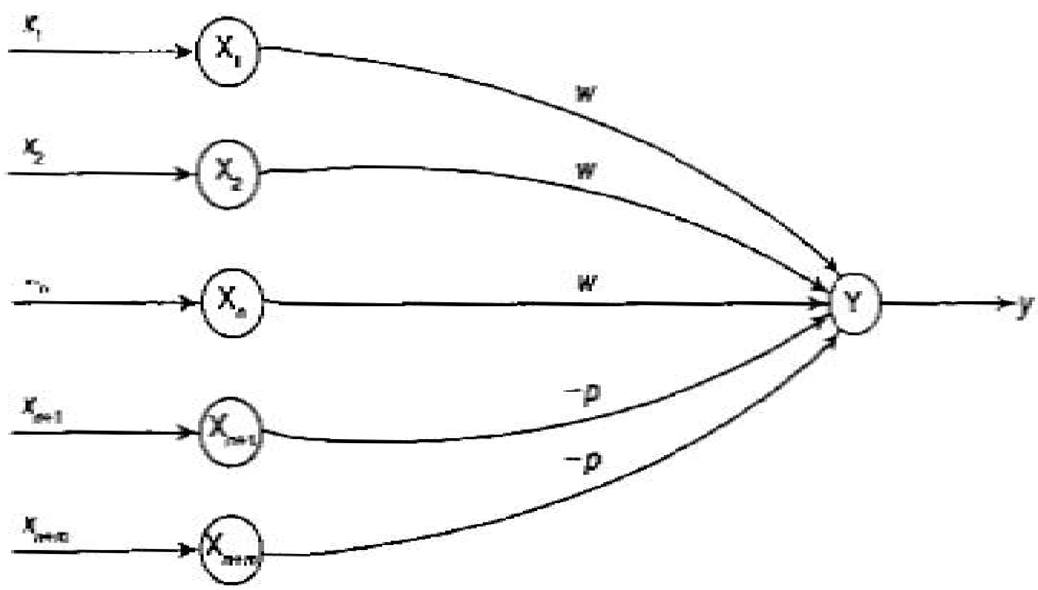
The weights associated with the communication links may be excitatory (weight is positive) or inhibitory (weight is negative). All the excitatory connected weights entering into a particular neuron will have same weights.

Threshold

There is a fixed threshold for each neuron, and if the net input to the neuron is greater than the threshold then the neuron fires. Also, any nonzero inhibitory input would prevent the neuron from firing. The M-P neurons are most widely used in the case of logic functions.

Architecture

A simple M-P neuron is shown in Figure below.



The M-P neuron has both excitatory and inhibitory connections. It is excitatory with weight $(w > 0)$ or inhibitory with weight $-p(p < 0)$.

In Figure, inputs from x_1 ro x_n possess excitatory weighted connections and inputs from x_{n+1} to x_{n+m} possess inhibitory weighted interconnections. Since the firing of the output neuron is based upon the threshold, the activation function here is defined as

$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq \theta \\ 0 & \text{if } y_{in} < \theta \end{cases}$$

For inhibition to be absolute, the threshold with the activation function should satisfy the following condition:

$$\theta > nw - p$$

The output will fire if it receives say k or more excitatory inputs but no inhibitory inputs, where

$$kw \geq \theta > (k-1)w$$

The M-P neuron has no particular training algorithm. An analysis has to be performed to determine the values of the weights and the threshold. Here the weights of the neuron are set along with the threshold to make the neuron "perform a simple logic function.

Hebb network

Theory

According to the Hebb rule, the weight vector is found to increase proportionately to the product of the input and the learning signal. Here the learning signal is equal to the neuron's output.

In Hebb learning, if two interconnected neurons are 'on' simultaneously then the weights associated with these neurons can be increased by the modification made in their synaptic gap (strength). The weight update in Hebb rule is given by

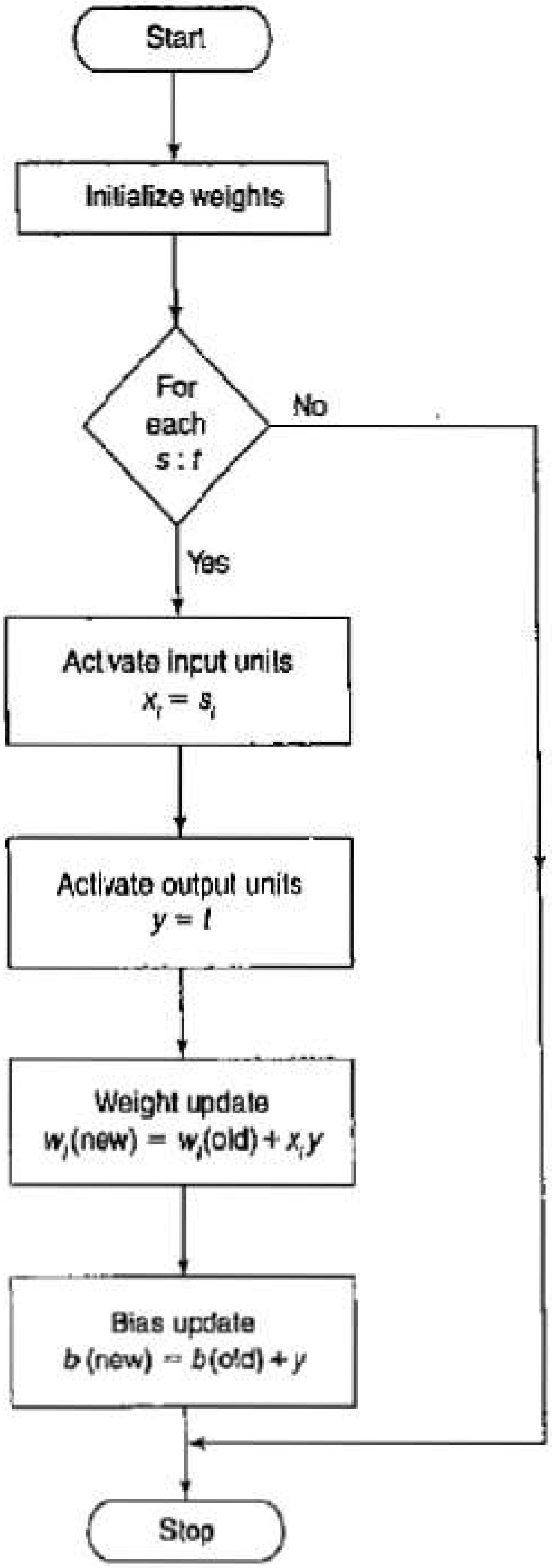
$$w_i(\text{new}) = w_i(\text{old}) + x_i y$$

The Hebb rule is more suited for bipolar data than binary data. If binary data is used, the above weight updation formula cannot distinguish two conditions namely;

1. A training pair in which an input unit is "on" and target value is "off."
2. A training pair in which both the input unit and the target value are "off."

Flowchart of Training Algorithm

The training algorithm is used for the calculation and adjustment of weights. The flowchart for the training algorithm of Hebb network is given in Figure below:



s: *t* refers to each training input and target output pair.

Training Algorithm

The training algorithm of Hebb network is given below:

Step 0: First initialize the weights. Basically in this network they may be set to zero, i.e., $w_i = 0$ for i= 1 to *n* where *"n"* may be the total number of input neurons.

Step 1: Steps 2-4 have to be performed for each input training vector and target output pair, s: r.

Step 2: Input units activations are set. Generally, the activation function of input layer is identity function: $x_i=s_i$; for i=1 to n.

Step 3: Output units activations are set: $y= t$.

Step 4: Weight adjustments and bias adjustments are performed:

$$w_i(\text{new}) = w_i(\text{old}) + x_i y$$

$$b(\text{new}) = b(\text{old}) + y$$

The above five steps complete the algorithmic process. In Step 4, the weight updation formula can also be given in vector form as

$$w(\text{new}) = w_i(\text{old}) + xy$$

Here the change in weight can be expressed as·

$$\Delta w = xy$$

As a result,

$$w(\text{new}) = w(\text{old}) + \Delta w$$

The Hebb rule can be used for pattern association, pattern categorization, pattern classification and over a range of other areas.