Artificial Neural Networks

1) Introduction to Artificial Neural Networks

ANN and Applications, ANN usefulness and capabilities, Equivalent electrical model, Artificial Neural Model and Linear Regression, Gradient Descent Algorithm, Non-linear activation units and learning Mechanism (Hebbian, Competitive, Boltzman), classification of Synaptic modification.

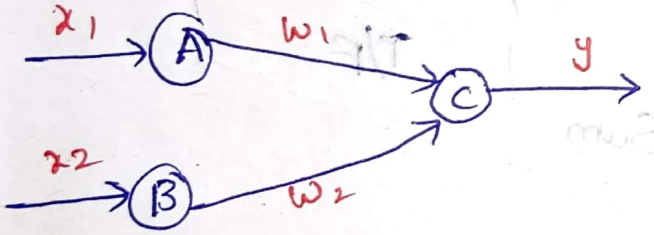
ANN :-

→ To develop computational device that mimics the functionality of human brain to perform various tasks faster than traditional system.

→ Efficient information processing system which resembles the characteristics of biological neural n/w.

→ In case of Human brain, we have neuron.
In case of ANN we have nodes.
nodes are connected to another node through connecting link.

Ex:-



A, B are i/p ~~conn~~ nodes it collects ~~i/p through x_1, x_2~~.
information through x_1 and x_2.

~~Sm~~ =
w_1 and w_2 = corresponding weights
y = output

ANN Usefulness and Capabilities :-

1. Non-linearity :-

Interconnection of non-linear neurons.

→ Non-linearity is distributed throughout.

2. Input - Output mapping :-

i/p - o/p mapping referer to "Learing with a teacher

3. Adaptivity :-

These can adapt the free parameters to the changes in the surrounding environment.

4. Evidential Response :-

Decision with a measure of confidence.

5. Fault Tolerance :- Graceful degradation.

* Comparison between ~~neuron~~ Biological Neuron and Artificial Neuron (Brain Vs Computer):

1. Speed :-

The cycle time of execution in the ANN is of few ~~non~~ nanoseconds whereas in case of biological neuron it is of a few milliseconds.

The artificial neuron modeled using a computer is more faster.

2. Processing :-

The biological neuron can perform massive parallel operations simultaneously.

→ The artificial neuron can also perform several parallel operation simultaneously, but in general the artificial neuron network process is faster than that of brain.

3) Size and Complexity :-

The total number of neurons in the brain is about 10^{11} and the total number of interconnection is about 10^{15}. The complexity of the brain is comparatively higher i.e, the computational work takes place 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 choosen 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 location.

→ In an artificial neuron, the continous loading of new information may sometimes overload the memory location.

→ 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 retrived.

→ Owing to these facts, the adaptability is more toward an artificial neuron.

5) Control Mechanism :-

In an artificial neuron modeled using a computer, there is a control unit present in Control Processing unit, which can transfer and control precise scalar values from unit to unit, but there is no such control unit for monitoring the brain.

→ The strength of a neuron in the brain depends on the active chemicals present and whenever neuron connections are strong or weak as a result of structure layer rather than individual synapses.

→ The control mechanism of a artificial neuron is very simple compared to that of a biological neuron.

6) Tolerance :-

The biological neuron possesses fault tolerant capability where as the artificial neuron has no fault tolerance.

2.3 Basic Models of Artificial Neural Network

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.

2.3.1 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 the geometry of their interconnections are essential for an ANN. The point where the connection originates and terminates should be noted, and the function of each processing element in an ANN should be specified.

Besides the simple neuron shown in Figure ??, there exist several other types of neural network connections. The arrangement of neurons to form layers and the connection pattern formed within and between layers is called 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.

Figures 2-6–2-10 depict the five types of neural network architectures. 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. Practically, 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

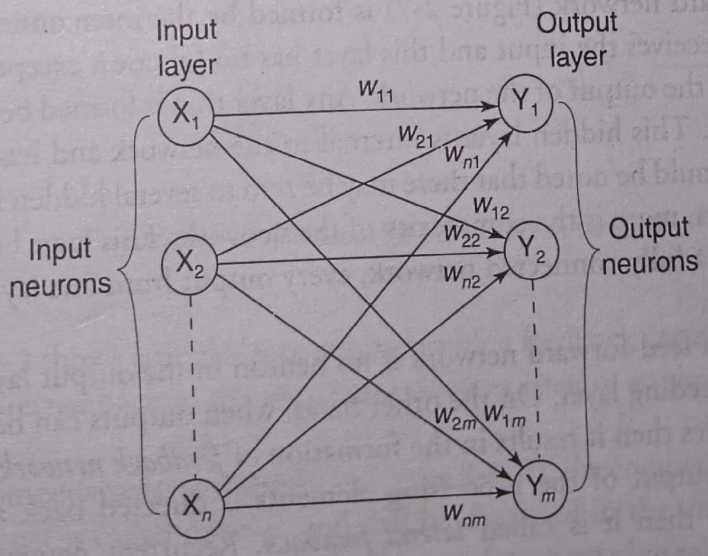


Figure 2-6 Single-layer feed-forward network.

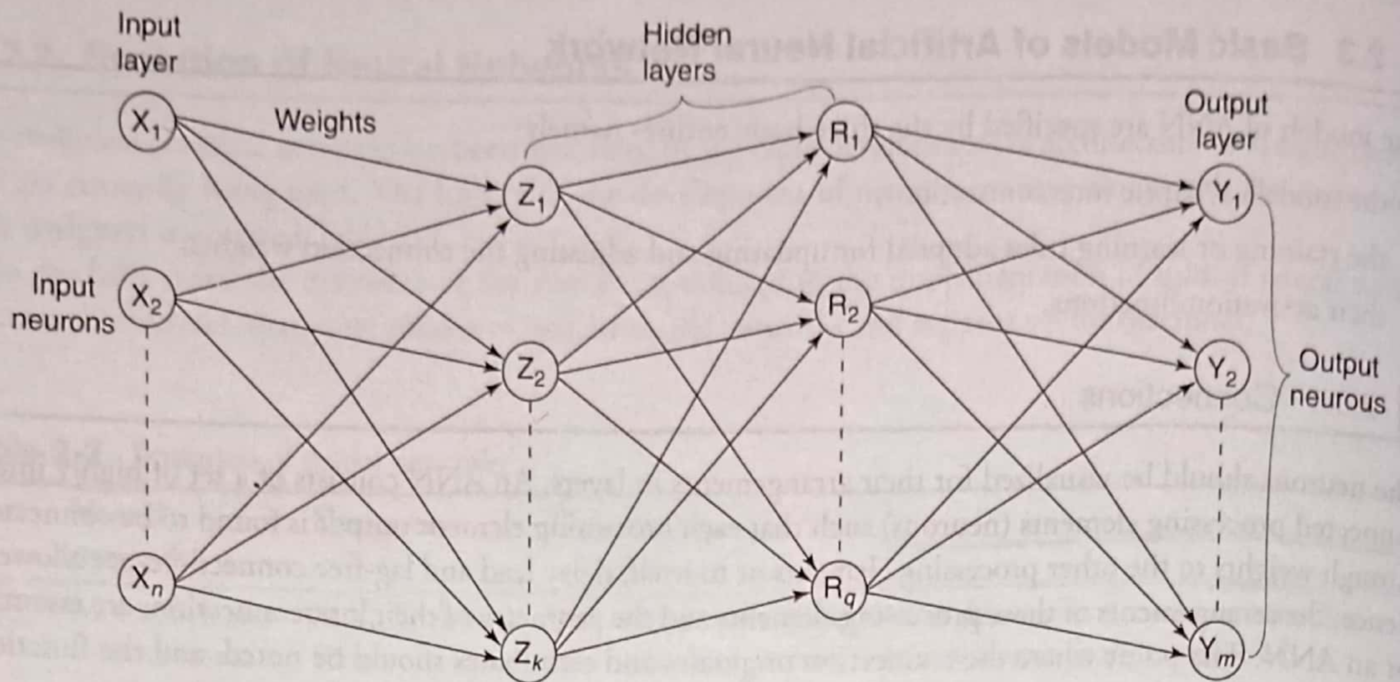


Figure 2-7 Multilayer feed-forward network.

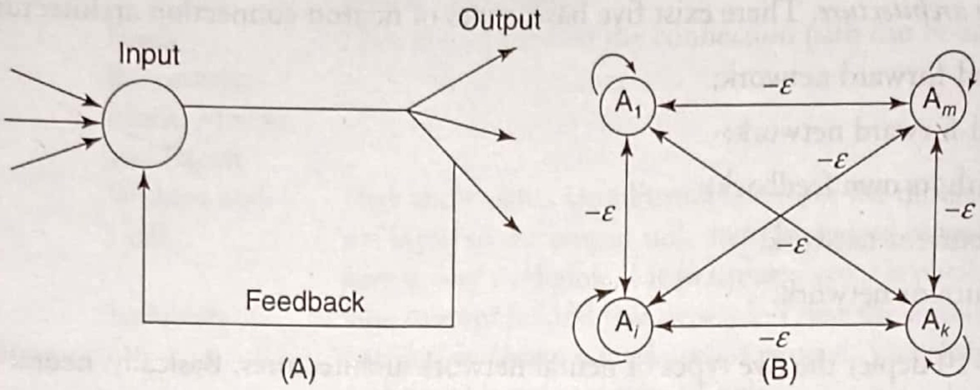


Figure 2-8 (A) Single node with own feedback. (B) Competitive nets.

nodes with various weights, resulting in a series of outputs, one per node. Thus, a single-layer *feed-forward network* is formed.

A multilayer feed-forward network (Figure 2-7) 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 the 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. It should be noted that 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. This may, however, provide an efficient output response. In case of a fully connected network, every output from one layer is connected to each and every node in the next layer.

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. On the other hand, 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. Figure 2-8(A) shows a simple recurrent neural network having a single neuron with

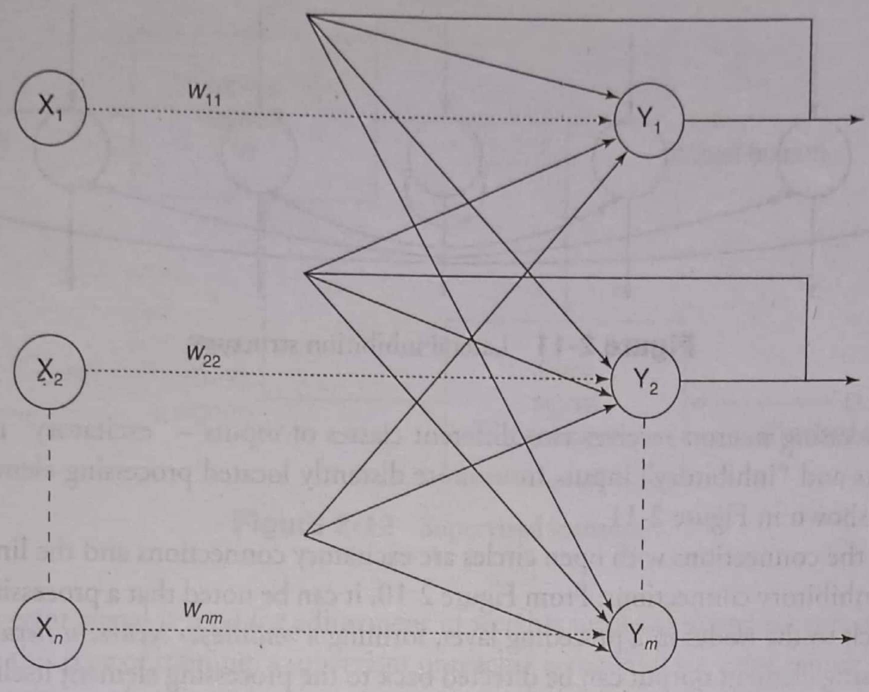


Figure 2-9 Single-layer recurrent network.

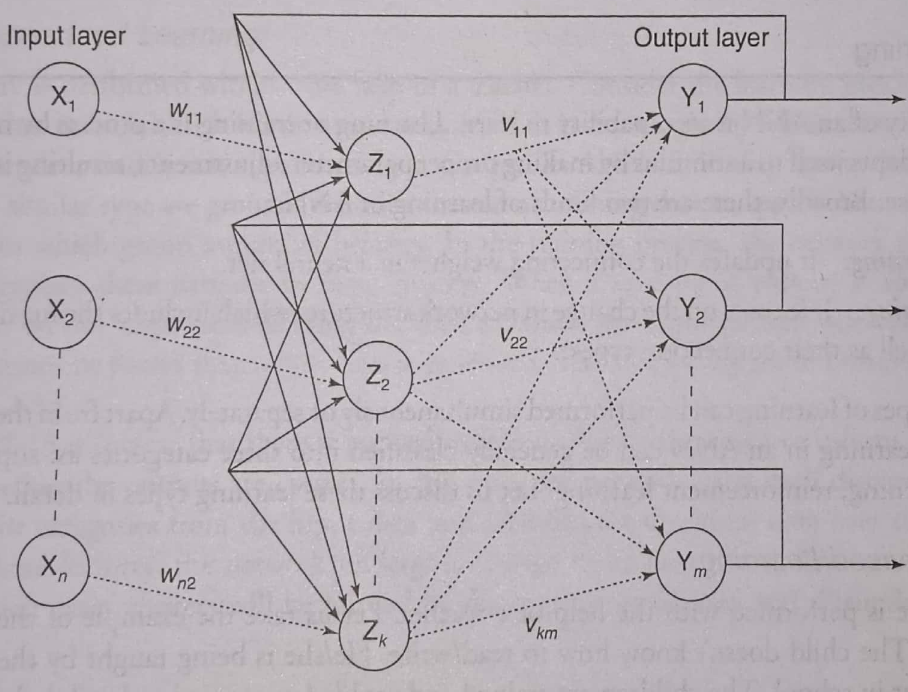


Figure 2-10 Multilayer recurrent network.

feedback to itself. Figure 2-9 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 2-8(B), the competitive interconnections having fixed weights of $-\varepsilon$. This net is called *Maxnet,* and will be discussed in the unsupervised learning network category. Apart from the network architectures discussed so far, there also exists another type of architecture with lateral feedback, which is called the *on-center-off-surround* or *lateral inhibition structure.* In this

2.3.2 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. Broadly, 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 (which includes the number of processing elements as well as their connection types).

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. Let us discuss these learning types in detail.

2.3.2.1 Supervised Learning

The learning here is performed with the help of a teacher. Let us take the example of the learning process of a small child. The child doesn't know how to read/write. He/she is being taught by the parents at home and by the teacher in school. The children are trained and molded to recognize the alphabets, numerals, etc. Their each and every action is supervised by a teacher. Actually, a child works on the basis of the output that he/she has to produce. All these real-time events involve supervised learning methodology. Similarly, 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 of Figure 2-12 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

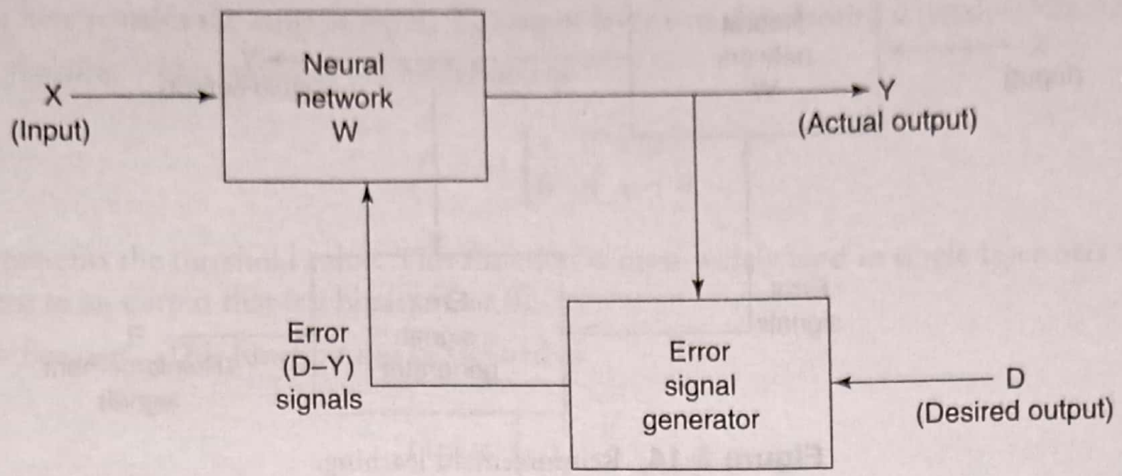


Figure 2-12 Supervised learning.

the network. This error signal is used for adjustment of weights until the actual output matches the desired (target) output. In this type of training, a supervisor or teacher is required for error minimization. Hence, the network trained by this method is said to be using supervised training methodology. In supervised learning, it is assumed that the correct "target" output values are known for each input pattern.

2.3.2.2 *Unsupervised Learning*

The learning here is performed without the help of a teacher. Consider the learning process of a tadpole, it learns by itself, that is, a child fish learns to swim by itself, it is not taught by its mother. Thus, its learning process is independent and is not supervised by a teacher. 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 2-13.

From Figure 2-13 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.

2.3.2.3 *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.

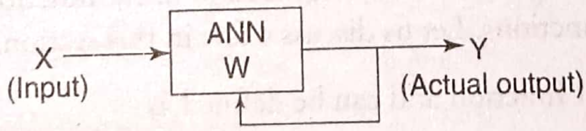


Figure 2-13 Unsupervised learning.

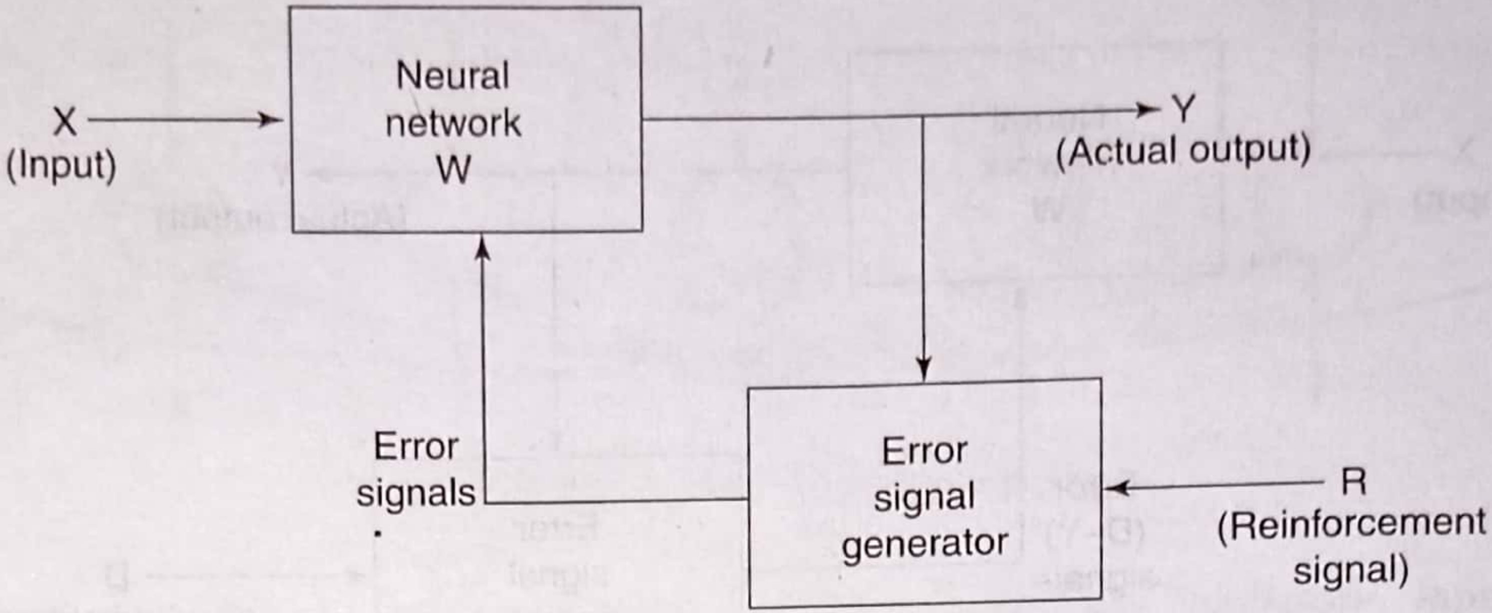


Figure 2-14 Reinforcement learning.

For example, the network might be told that its actual output is only "50% correct" or so. Thus, here only critic information is available, not 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 2-14. The reinforcement learning is a form of supervised learning because the network receives some feedback from its environment. However, the feedback obtained here is only evaluative and not instructive. The external reinforcement signals are processed in the critic signal generator, and the obtained critic signals are sent to the ANN for adjustment of weights properly so as to get better critic feedback in future. The reinforcement learning is also called learning with a critic as opposed to learning with a teacher, which indicates supervised learning.

So, now you've a fair understanding of the three generalized learning rules used in the training process of ANNs.

2.3.3 Activation Functions

To better understand the role of the activation function, let us assume a person is performing some work. To make the work more efficient and to obtain exact output, some force or activation may be given. This activation helps in achieving the exact output. In a similar way, 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 (say f) 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. The nonlinear activation function is used to ensure that a neuron's response is bounded – that is, the actual response of the neuron is conditioned or dampened as a result of large or small activating stimuli and is thus controllable.

Certain nonlinear functions are used to achieve the advantages of a multilayer network from a single-layer network. When a signal is fed through a multilayer network with linear activation functions, the output obtained remains same as that could be obtained using a single-layer network. Due to this reason, nonlinear functions are widely used in multilayer networks compared to linear functions.

There are several activation functions. Let us discuss a few in this section:

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

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

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

2. *Binary step function:* 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. *Bipolar step function:* 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:

- *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.

- *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 sigmoid function range is between −1 and +1. The derivative of this function can be

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

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

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

If the network uses a binary data, it is better to convert it to bipolar form and use the bipolar sigmoidal activation function or hyperbolic tangent function.

5. *Ramp function*: The ramp function is defined as

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

The graphical representations of all the activation functions are shown in Figure 2-15(A)–(F).

Learning Rules in Neural Networks :-

The main objective of a neural network is to learn or train itself.

Learning or training is a process by which a neural network adapts and adjust itself to any stimulus given to it by making proper parametric adjustments in order to generate the desired output or response.

Two Kinds of Learning :-

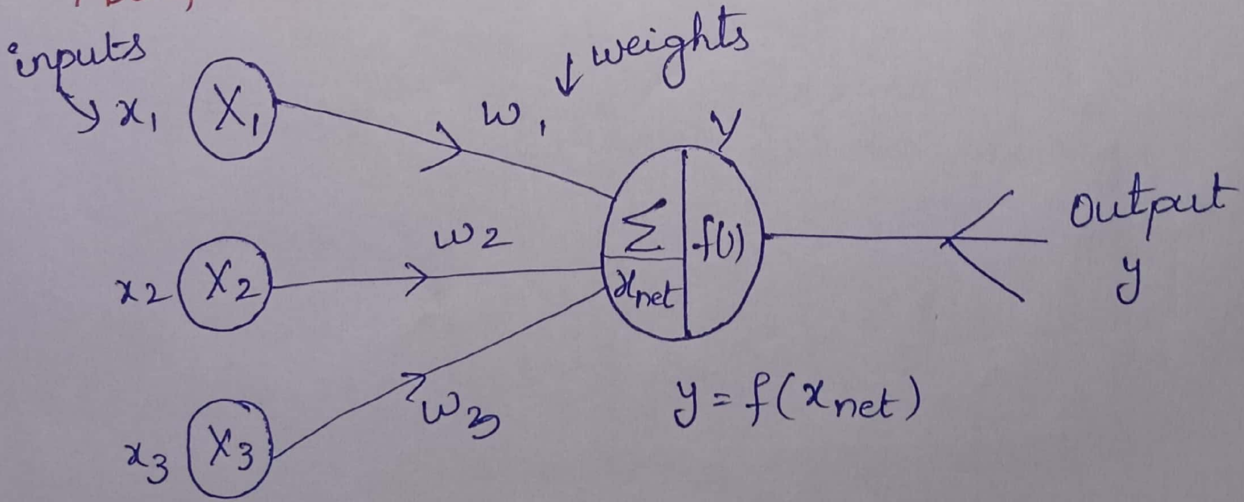
1) **Parameter Learning :-**

It involves changing and updating the connecting weights in the neural network.

2) **Structure Learning :-**

It focusses on changing the structure or architectures of the neural network.

Artificial Neuron :-



Learning Rules (or) Learning Mechanism :-

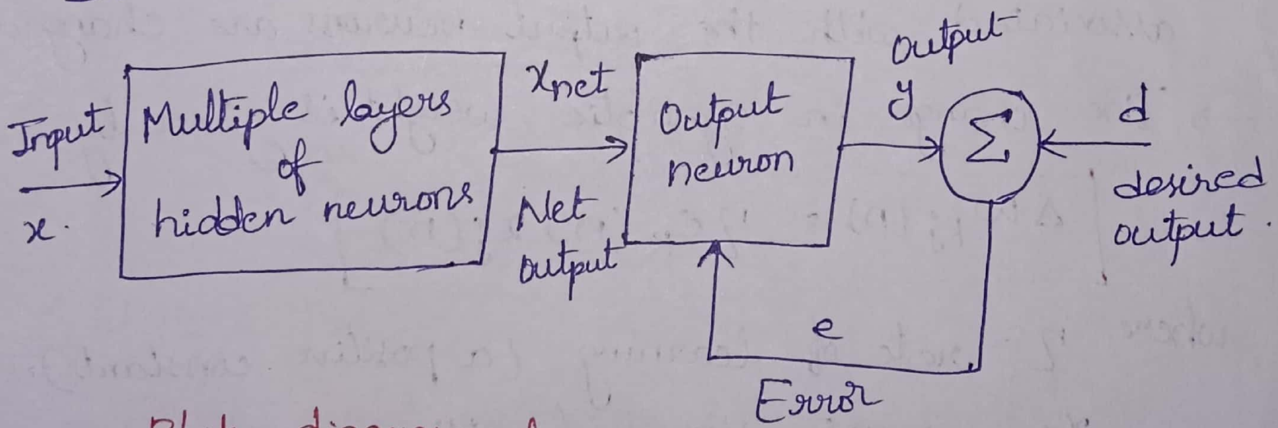
1) Error Correction Learning
2) Memory Based Learning
3) Hebbain Learning
4) Competitive Learning.
5) Boltzmann Learning.

① Error Correction Learning :-

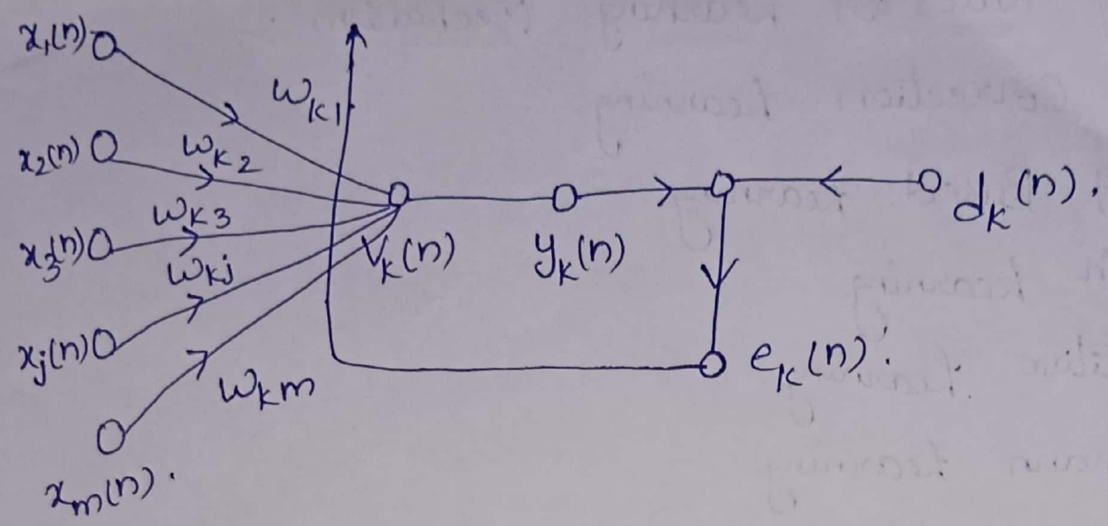
Error Correction Learning consists of :

→ Input and Output layer of neurons
→ Input signal
→ Comparator
→ Desired or Target output

• It is a parameter based learning.



Block diagram of Error correction learning.



Error Signal :-

Depending on the output and desired or target input the error signal is given by

$$\boxed{e_k(n) = d_k(n) - y_k(n)}$$

→ The error signal, $e_k(n)$ activates a control mechanism by which synaptic weights associated with the output neurons are changed.

→ The change in synaptic weight is given by

$$\boxed{\Delta w_{kj}(n) = \eta e_k(n) x_j(n)}$$

where η = rate of learning (a positive constant)

x_j = input signal of jth neuron

e_k = error signal.

Δw_{kj} = Change in synaptic weight

Hebbian Learning :-

→ Hebbian learning rule is one of the earliest and the simplest learning rules for the neural networks. It was proposed by Donald Hebb.

→ Hebb proposed that -

* If two interconnected neurons on either side of a synapse are both on or fired or activated at the same time (synchronously), then the synaptic weight between them should be increased.
* If two neurons on either side of a synapse are on or fired or activated at different times (asynchronously), then the weight of that synaptic connection is desired.

→ Such a synapse is called Hebbian Synapse.

Hebbian Learning :-

In the brain, learning is performed by the change in synaptic gap.

→ Hebb explained it : When an axon of cell A is near enough to excite cell B, and repeatedly or permanently takes place in firing it, some growth process or metabolic change takes place in one or both the cells such that A's efficiency, as one of the cell firing B, is increased.

According to the Hebb rule, the weight vector is found to be increase proportionately to the product of the input and the learning signal.

→ 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

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

Flowchart of Training Algorithm :-

→ The training algorithm is used for the calculation and adjustment of weights.

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

→ Till there exists a pair of training input and target output, the training process takes place; else it is stopped.

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: Step 2-4 have to performed for each input training vector and target output pair, $s:t$

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 algorithm process. In step 4, the weight updation formula can also be given in vector form as

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

the change in weight can be expressed as

$$\boxed{\Delta w = xy}$$

As a result,

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

Uses :-

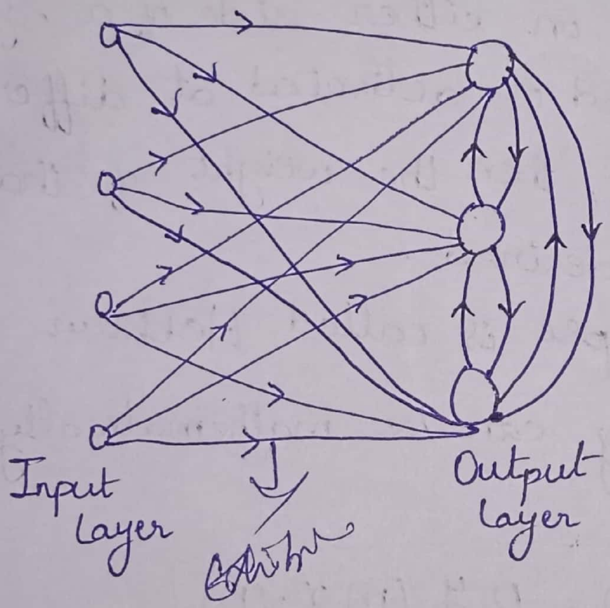
The Hebb rule can be used for Pattern association, pattern categorization, Pattern classification and over a range of other areas.

Competitive Learning :-

→ In competitive learning, the neural network consists of a single layer of output neurons.

→ All the output neurons are full connected to the input neurons.

→ As the name suggests, here all the output neurons compete aganist each other for the right to get fired or activated.

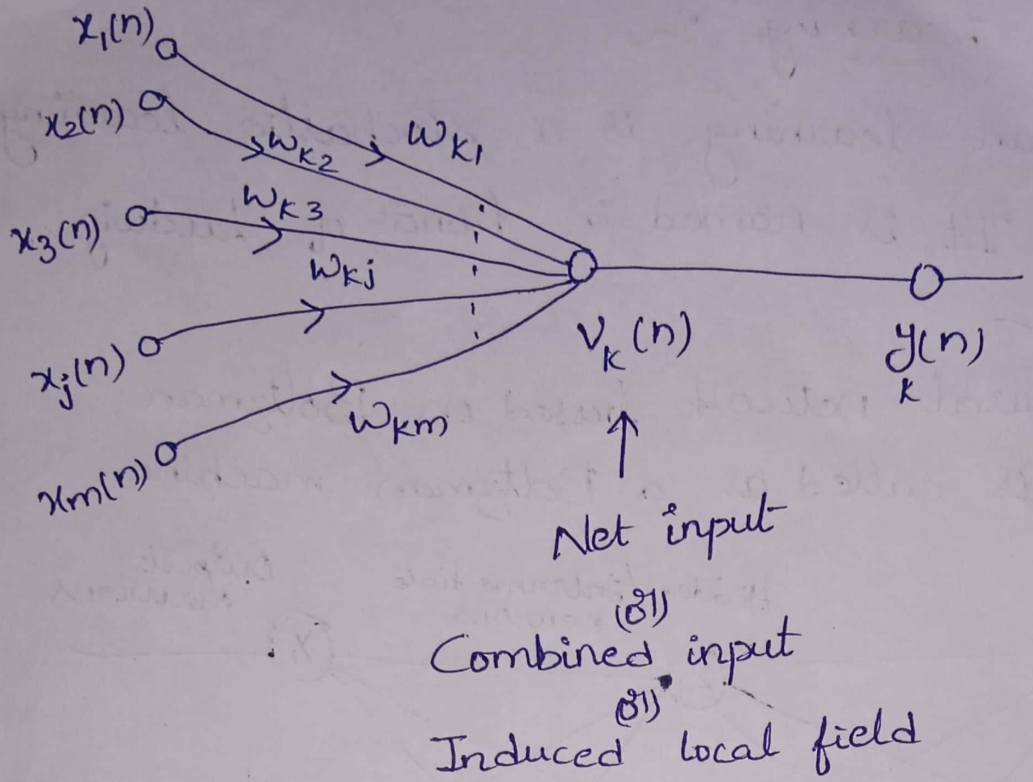


Winning Neuron :-

→ For an output neuron to win the competition, its net input or combined input or induced local field must be maximum, among all the other output neurons.

→ The output of the ~~winne~~ winning neuron is set to 1, while that of the others is set to 0.

Scanned with OKEN Scanner



Mathematically,

$$\boxed{y_k = \begin{cases} 1 & \text{if } v_k > v_j \\ 0 & \text{otherwise,} \end{cases}}$$

where

v_k = net input for k^{th} neuron.

Boltzman Learning :-

→ Boltzman learning is a stochastic learning algorithm. It is named in honor of Ludwig Boltzmann.

→ A neural network based on Boltzman learning is called as a Boltzman machine.

Boltzman Learning :-

→ The neural network consists of input and output layers of neurons with multiple hidden layers.

→ The neurons involved operate in a binary manner being either is ON state denoted by +1 (or) OFF state denoted by -1.

→ It operates by randomly choosing a neuron and flipping its state.

Energy Function of a Boltzmann Machine :-

→ The Boltzmann machine is characterised by an energy function,

$$\boxed{x_j \xrightarrow{W_{kj}} x_k}$$

$$\boxed{E = -\frac{1}{2} \sum_{j} \sum_{k} w_{kj} x_k x_j}$$

where, $j \neq k$

x_k is state of neuron, k.

$$\boxed{x_j \xrightarrow{w_{kj}} x_k}$$

x_j is state of neuron, j

W_{kj} is synaptic weight connecting neurons j and k.

Probability of State Change :-

→ The probability of change of state, P of a neuron is given by

$$P = \frac{1}{1 + \exp(-\Delta E_k / T)}.$$

Operating Conditions :-

→ The neurons can operate in two modes :

1) Clamped Condition :- In which neurons are in fixed state.

2) Free Running Condition :- In which neurons operate freely and can take any state.

1.2 Application Scope of Neural Networks

The neural networks have good scope of being used in the following areas:

1. *Air traffic control* 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. *Animal behavior*, *predator/prey relationships* and *population cycles* may be suitable for analysis by neural networks.
3. *Appraisal and valuation* of property, buildings, automobiles, machinery, etc. should be an easy task for a neural network.

4. *Betting* on horse races, stock markets, sporting events, etc. could be based on neural network predictions.
5. *Criminal sentencing* 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 the neural network.
21. *Recipes and 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, etc.
26. *Strategies* for games, business and war can be captured by analyzing the expert player's response to given stimuli. For example, a football coach must decide whether to kick, pass or run on the last down. The inputs for this decision include score, time, field location, yards to first down, etc.

27. *Traffic flows* could be predicted so that signal timing could be optimized. The neural network could recognize "a weekday morning rush hour during a school holiday" or "a typical winter Sunday morning."

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. Effects such as ocean currents and jet streams could be included.

Today, ANN represents a major extension to computation. Different types of neural networks are available for various applications. They perform operations akin to the human brain though to a limited extent. A rapid increase is expected in our understanding of the ANNs leading to the improved network paradigms and a host of application opportunities.