Introduction to ANN

Definition :→

A Structure (network) composed of a number of inter-connected units (artificial neurons). Each unit has an input/output (I/o) characteristics and implements a local computation or function.

- The output of any unit is determined by It's I/o characteristics, its interconnection to other units, and external inputs.

OR

- A set of processing units when assembled in a closely interconnected network, offers a superisingly rich structure exhibiting some features of the biological neural network. Such a structure is called an artificial neural network (ANN). Since ANNs are implemented on computers, it is worth comparing the processing capabilities of a computer with those of the brain.

Usefulness and Capability :-

1. Nonlinearity →

linearity — If there is a system where we give a set of inputs and we expect some output out of it. in that case, we call the system, give a set of i/p and we expect some output out of it. In that case, we call the system to be linear, if the relation between the output and the input can be best

best described in terms of a simple linear equation. If there are let say 4 i/p and 1 o/p, then if the o/p is a linear combination of all 4 i/p, then the system is linear.

In real life most of the problems are Non-linear in nature.

- ANN is interconnection of non-linear neurons.
- Non linearity is distributed throughout

2. Input - output mapping.
3. Adaptivity ⟶ can adapt the free parameters to changes in the surrounding environment.
4. Evidential Response ⟶ Decision with a measure of "confidance".
5. Fault Tolarance ⟶ Graceful degradation.
6. VLSI Implementability ⟶
7. Neurobiological analogy.

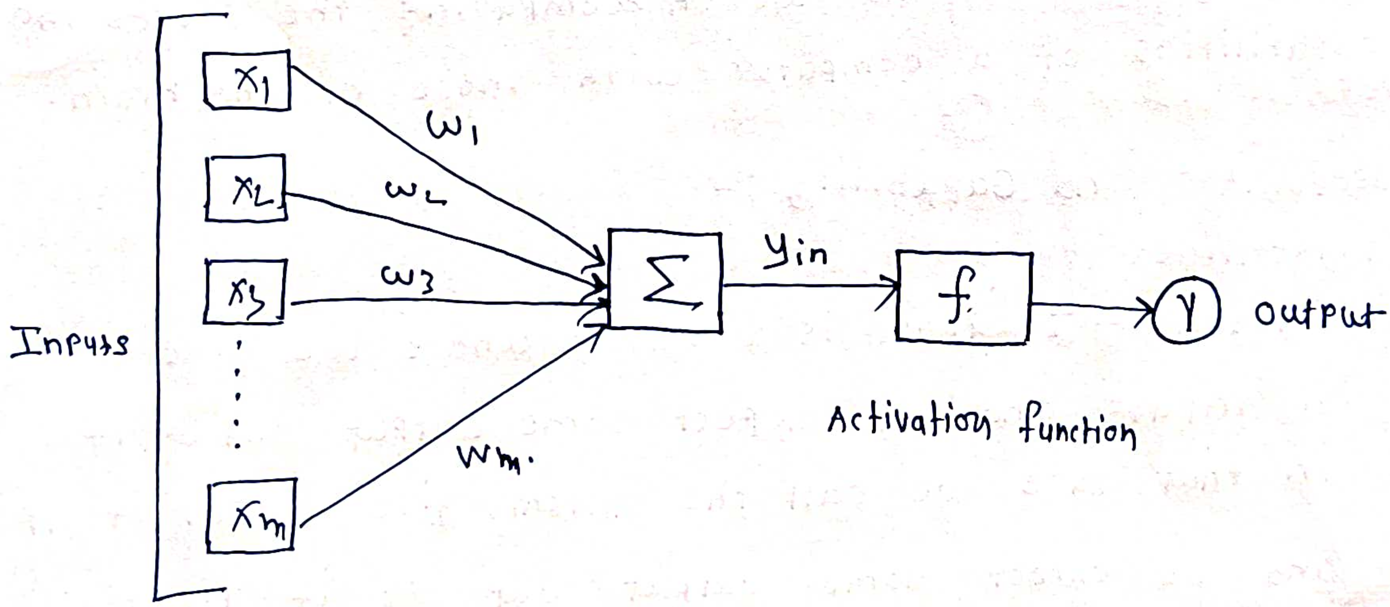


fig: ANN Basic Structure

History of Neural Network :-

Key development

McCulloch and pitts (1943)
- Model of neuron
- Logic operations
- Lack of learning

Hebb (1949)
- synaptic modifications
- Hebb's learning law

Minsky (1954)
- Learning Machines.

Rosenblatt (1958)
- Perceptron learning and convergence.
- Pattern classification
- Linear separability constraint

widrow and Hoff (1960)
- Adaline - LMS learning
- Adaptive signal processing

Minsky and Papert (1969)
- Perceptron - Mutilayer perceptron (MLP)
- Hard problems
- No learning for MLP

Webbos (1974)
- Error backpropogation

Hopfield (1982)
- Energy analysis Ackley, Hinton and ~~williams (1986)~~ Sejnowski (1985)

other significant contribution.

- von Neumann (1946) - General purpose electronic computer.
- Nobert weiner (1948) - Cybernetics
- Shannon (1948) - Information Theory
- Ashby (1952) - Desigh for a Brain.
- Gabor (1954) - Nonlinear adaptive filter.
- Uttley (1956) - Theoretical machine.
- Caianiello (1961) - statistical theory and learning.
- Minsky (1961) - Artificial intelligence
- Steinbuch (1961) - Learnmatrix
- Minsky and Selfridge (1961) - credit assignment problem.
- Nilsson (1965) - Learning Machine
- Amari (1967) - Mathematical Solution ~~for~~ to credit assignment.
- kohonen (1971) - Associative memories
- willshaw (1971) - Self-organization and generalization
- Malsburg (1973) - Self-organization
- Tikhonov (1973) - Regularization theory.
- Little (1974) - Ising model and neural network
- Grossberg (1976) - Adaptive resonance theory
- Anderson (1977) - Brain state in-box model
- Little and shaw (1978) - stochastic law for NN, spin glasses.
- Fukushima (1980) - Neocognitron
- kohonen (1982) - Feature mapping Barto.

- Boltzmann machine

Rumelhart, Hinton and williams (1986)

- Generalised delta rule.

Structure and working of Biological Neural Network

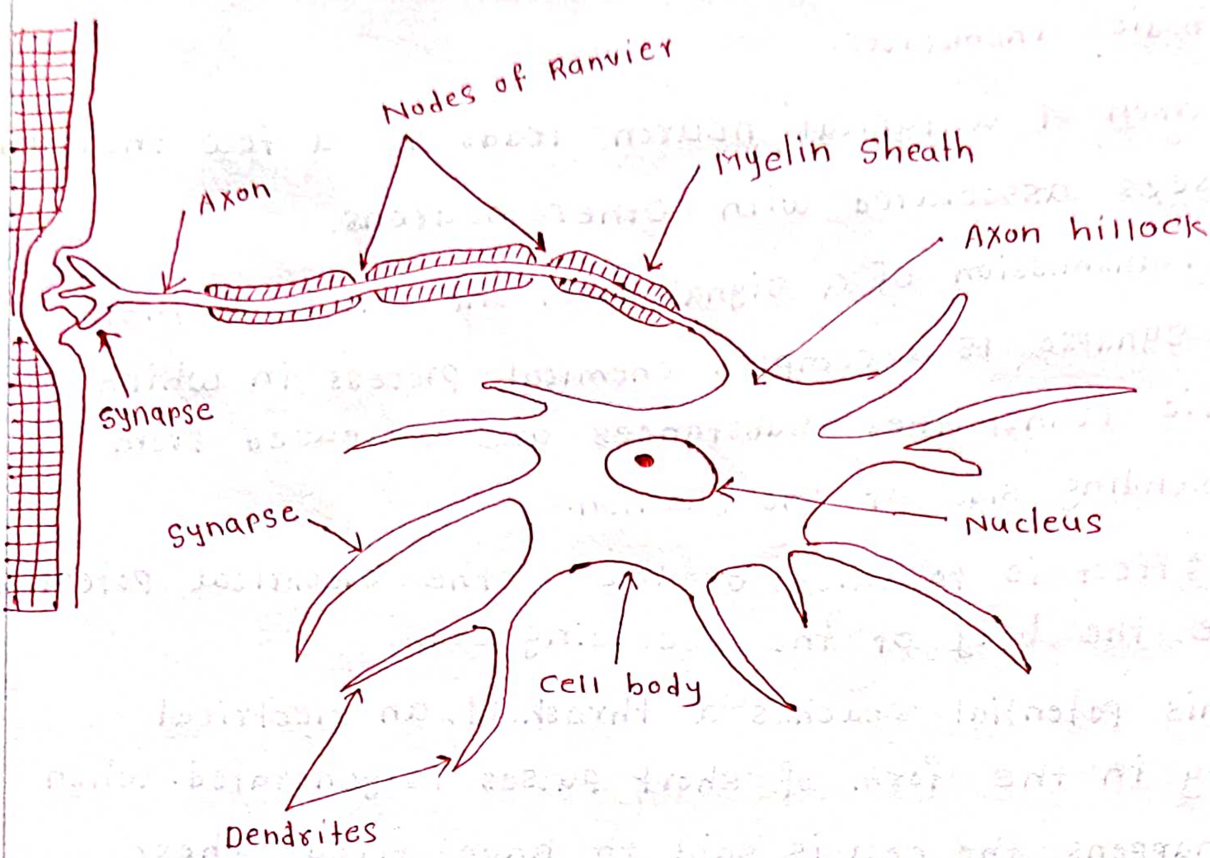


Fig: Schematic diagram of a typical neuron or nerve cell.

- The features of the biological neural network are attributed to its structure and function.
- The fundamental unit of the network is called a neuron or a nerve cell.
- It consist of a cell body or soma where the cell neucleus is located.
- Tree like nerve fibres called dendrites are associated with the cell body. These dendrites receive signals from other neurons.
- Extending from the cell body is a single long fibre called the axon, which eventually branches into strands and substrands connecting to many other neurons at the

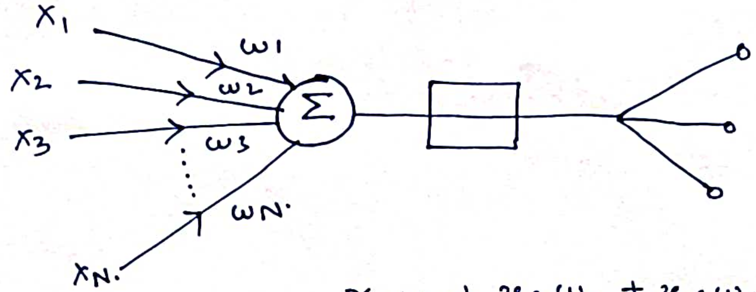
synaptic junctions, or synapses.

- The recieving junction ends of these junctions on other cells can be found both on the dendrites and on the cell bodies themselves.
- The axon of a typical neuron leads to a few thousand ~~synapseps~~ associated with other neurons.
- The transmission of a signal from one cell to another at a synapse is a complex chemical process in which specific transmitter substances are released from the sending side of the junction.
- The effect is to raise or lower the electrical potential inside the body of the receiving cell.
- If this potential reaches a threshold, an electrical activity in the form of short pulses is generated. when this happens, the cell is said to have fired. These electrical signals of fixed strength and ~~during~~ duration are sent down the axon.
- Generally the electrical activity is confined to the interior of a neuron, whereas the chemical mechanism operates at the synapses.
- The dendrites serve as receptors for signals from other neurons, whereas the purpose of an axon is transmission of the generated neural activity to other nerve cells (inter-neuron) or to muscle fibres (motor neuron).

A third type of neuron, which recieves information from muscles or sensory organs, such as the eye or ear, is called a receptor neuron.

- Axon → Act as the transmission lines for carrying the electrical signals, And they end with synaptic terminals and these are basically used for making connections with other nerve cells.

- Basal dendrite → (Resertive zones)



$x_1w_1 + x_2w_2 + x_3w_3 + \ldots\ldots x_Nw_N$

Fig. Equivalent Electrical model.

Models of Neuron

I. McCulloch - pitts Model

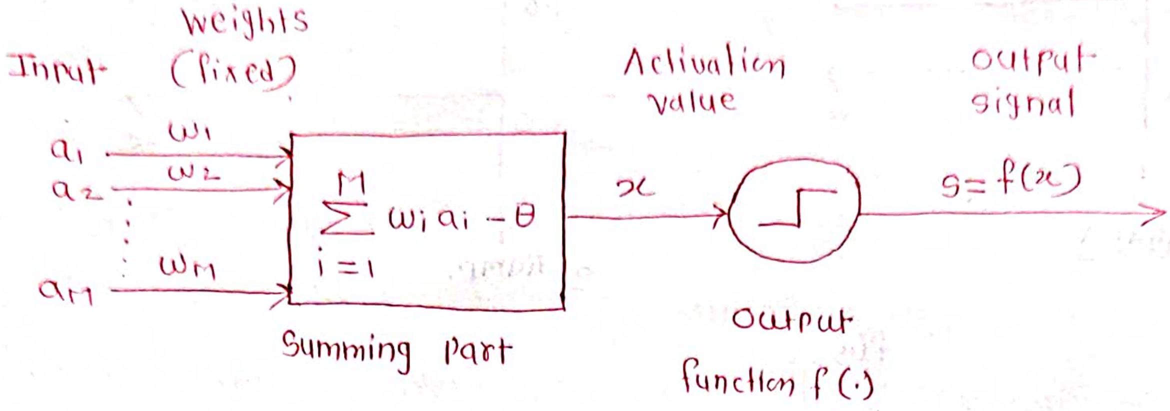


Figure: McCulloch - pitts model of neuron.

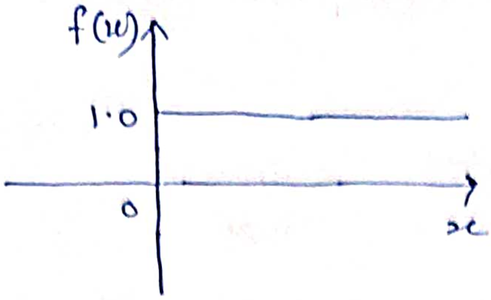
- In Mcculloch - pitts (MP) model the activation (x) is given by a weighted sum of its M input values (a_i) and a bias term (θ).

- The output signal (s) is typically a nonlinear function $f(x)$ of the activation value x. The following equations describe the operation of an MP model.

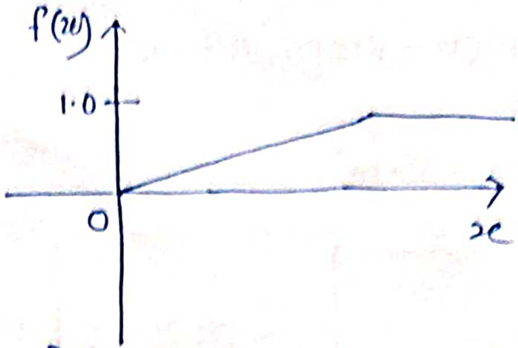
Activation : $x = \sum_{i=1}^{M} w_i a_i - \theta$

output signal : $s = f(x)$

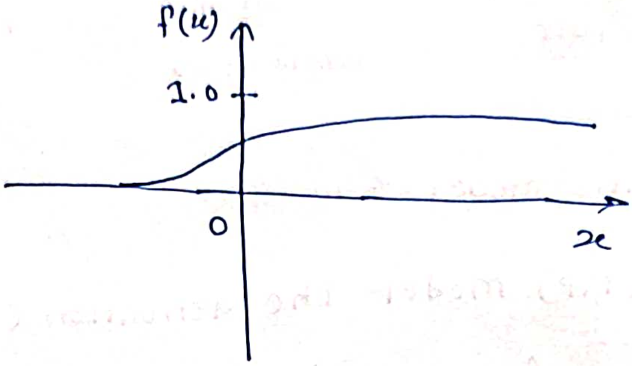
- Three commonly used nonlinear functions (binary, ramp and sigmoid) are shown in figure. Although only the binary function was used in the original MP model.



a) Binary



b) Ramp.

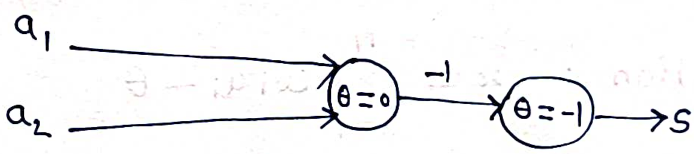


c) Sigmoid

— Networks consisting of MP neurons with binary (on-off) output signals can be configured to perform several logical functions. following fig. shows some examples of logic circuits realized using the MP model.

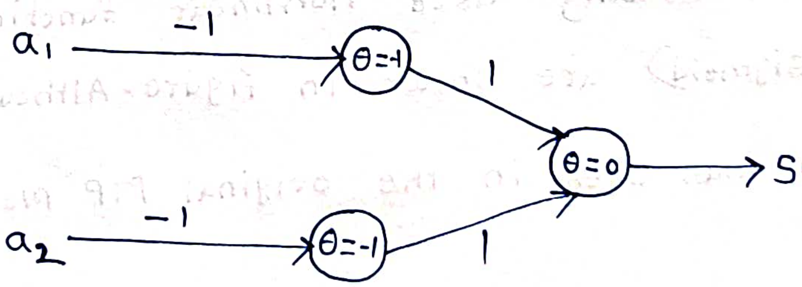
a) NOR gate

a_1	a_2	S
0	0	1
0	1	0
1	0	0
1	1	0



b) NAND gate

a_1	a_2	S
0	0	1
0	1	1
1	0	1
1	1	0



- In this model a binary output function is used with the following logic.

$$f(x) = 1, \quad x > 0$$
$$= 0 \quad x \leq 0$$

- A single input and single output MP neuron with proper weight and threshold gives an output a unit time later. This unit delay property of the MP neuron can be used to build sequential digital circuits.
- In the MP model the weights are fixed. Hence a network using this model does not have the capability of learning.

Perceptron →

- The Rosenblatt's perceptron model 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 unit.
- The desired or target output (b) is compared with the actual binary output (s), and the error (δ) is used to adjust the weights.
- The following equations describe the operation of the perceptron model of the neuron:

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

Output signal : $s = f(x)$

Error : $\delta = b - s$

Weight change : $\Delta w_i = \eta \delta a_i$

Where η is the learning rate parameter.

- There is a perceptron learning law, which gives a step-by-step procedure for adjusting the weights. Whether the weight adjustment coverages or not depends on the nature of desired input-output pairs to be represented by the model.

- The perceptron convergence theorem enables us to determine whether the given pattern pairs are representable or not.

- If the weight values converge, then the corresponding problem is said to be represented by the perceptron network.

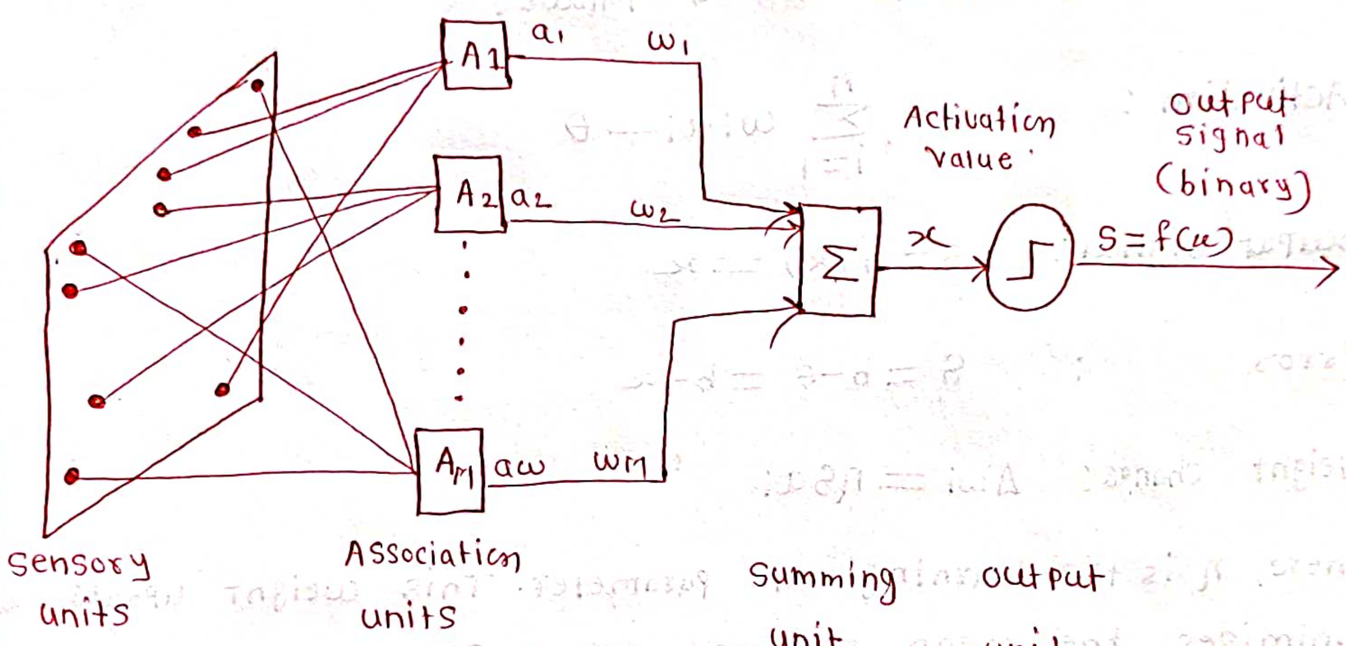


Fig: Rosenblatt's perceptron model of a neuron.

Adaline:

- ADAptive LINear Element (ADALINE) is a computing model proposed by widrow.

- The main distinction between the Rosenblatt's perceptron model and the widrow's Adaline model is that, in the Adaline the analog activation value (x) is compared with the target output (b).

- In the other words, the output is a linear function of the activation value (x). The equations that describe the operation of an Adaline are as follows:

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

output signal: $s = f(x) = x$

Error : $\delta = b - s = b - x$

Weight change: $\Delta w_i = \eta \delta a_i$

- where, η is the learning rate parameter. This weight update rule minimises the mean squared error δ^2, averaged over all inputs. Hence it is called Least mean squared [LMS] error learning law.

- This law is derived using the negative gradient of the error surface in the weight space. Hence it is also known as a gradient descent algorithm.

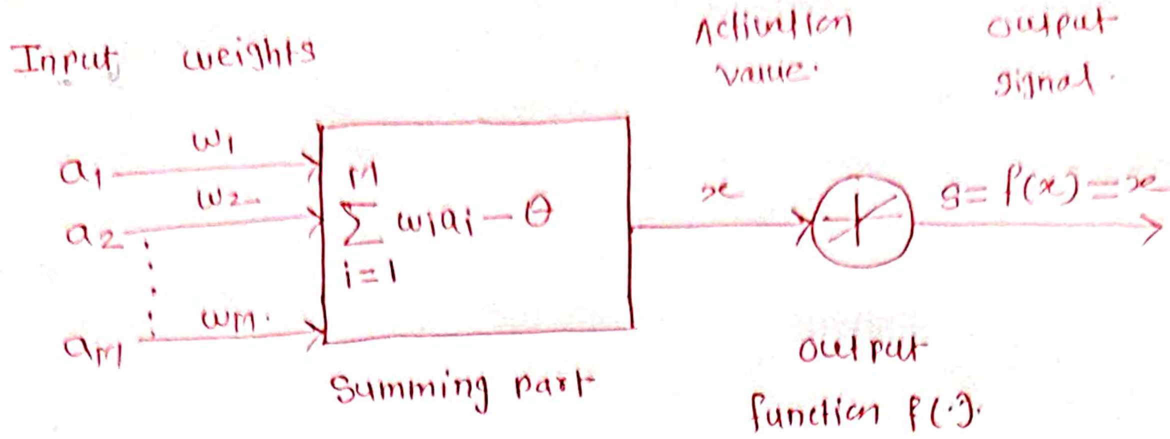
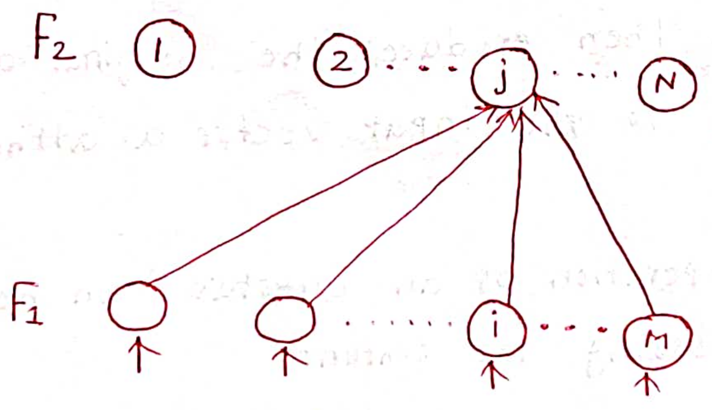


Fig: Widrow's Adaline model of a neuron.

Topology of Neural network Architecture :→

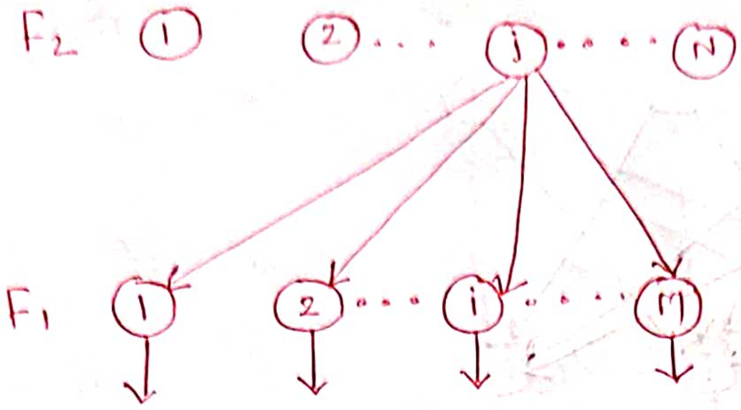
- Artificial neural networks are useful only when processing units are organised in a suitable manner to accomplish a given pattern recognition task.
- The arrangement of the processing units, connections, and pattern input/output is referred to as topology.
- Artificial neural networks are normally organized into layers of processing units. The units of a layer are similar in the sense that they all have the same activation dynamics and output function.
- Connections can be made either from the units of one layer to the units of another layer or among the units within the layer, or both interlayer and intralayer connection.
- Further, the connections across the layers and among the units within a layer can be organised either in a feedforword manner or in a feedback manner. In feedback network the same processing unit may be visited more than once.
- Let us consider two layers F_1 and F_2 with M and N processing units respectively.
- ~~By Providing~~ ~~Conn~~

a) Instar and outstar



a) Instar

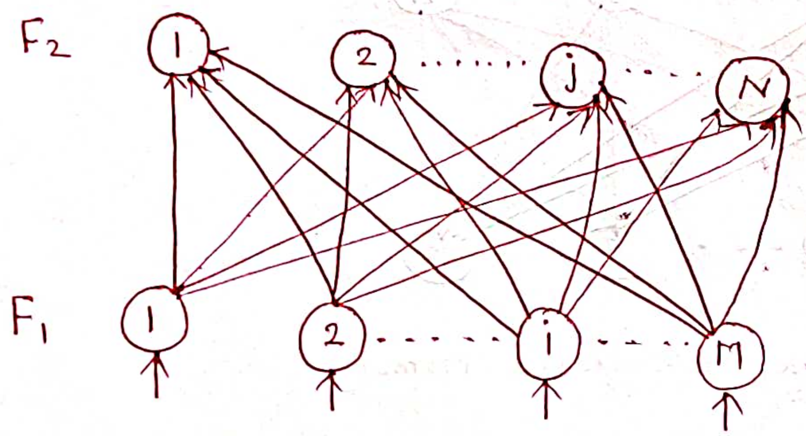
- By providing connections to the j^{th} unit in the F_2 layer from all the units in the F_1 layer. we get one network structure, called as Instar neural network. and outstar neural network. which have fan-in and fan-out geometries respectively.
- During learning, the normalized weight vector $w_j = (w_{j1}, w_{j2}, \ldots\ldots w_{jm})^T$ in instar approaches the normalized input vector, when an input vector $a = (a_1, a_2, \ldots\ldots\ldots a_m)^T$ is presented at the f_1 layer. Thus the activation $w_j^T a = \sum_{i=1}^{m} w_{ji} a_i$ of the jth unit of F_2 will be activated to the maximum extent.
- Operation of instar can be viewed as content addressing the memory.
- In case of of an outstar, during learning, the weight vector for the connections from the jth unit in F_2 approaches the activity pattern in f_1, when an input vector a is presented at f_1.
- During recall, whenever the unit j is activated, the signal pattern $(s_j w_{1j}, s_j w_{2j}, \ldots\ldots\ldots s_j w_{mj})$ will be transmitted to f_1. where s_j is the output of the jth unit.
- This pattern then produces the original activity pattern corresponding to the input vector a, although the input is absent.
- Thus the operation of an outstar can be viewed as memory addressing the content.



b) outstar

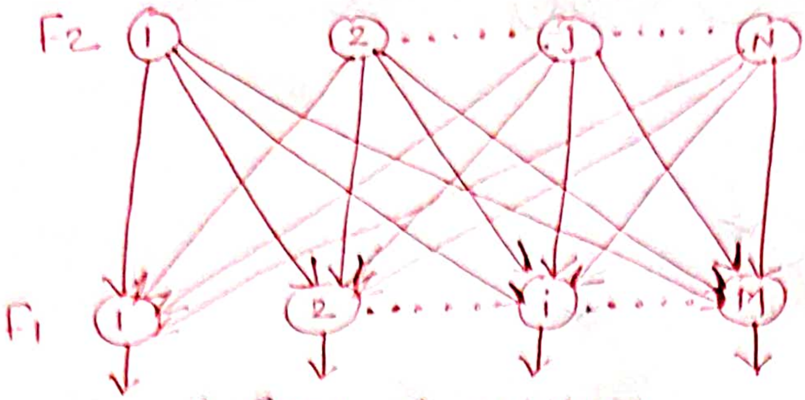
b) Group of instars & Group of outstars

- When all the connections from the units in f_1 to f_2 are made as in the following fig. we obtain a heteroassociation network. This network can be viewed as a group of instars, if the flow is from f_1 to f_2.



c) Group of instars

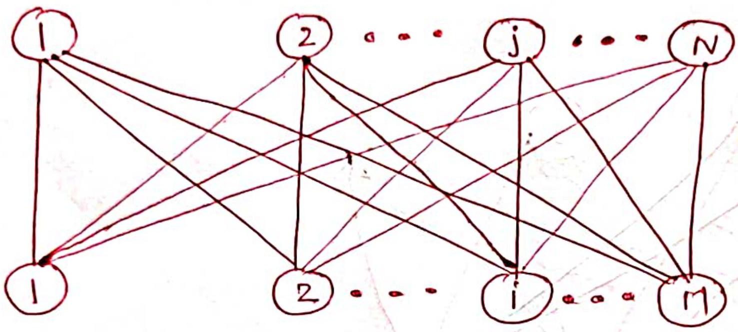
- If the flow is from F_2 to f_1 then the network can be viewed as a group of outstars.



d> Group of outstars

c> Bidirectional associative memory.

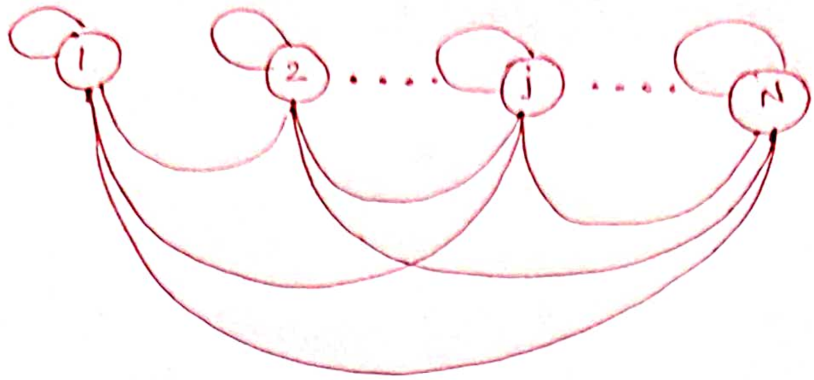
- when the flow is bidirectional, we get a bidirectional associative memory. where either of the layers can be used as input / output.



e> Bidirectional associative memory.

d> Autoassociative memory.

- If the two layers f_1 and f_2 coinside and the weights are symmetric i.e. $w_{ji} = w_{ij}$, $i \neq j$, then we obtain an auto associative memory in which each unit is connected to every other unit and to itself.



f) Autoassociative memory.

Basic Learning Law :-

- Learning laws describe the weight vector for the ith processing unit at time instant $(t+1)$ in terms of the weight vector at time instant (t) as follows:

$$W_i(t+1) = w_i(t) + \Delta w_i(t)$$

where $\Delta w_i(t)$ is the change in the weight vector.

1. Hebb's Law

Here the change in the weight vector is given by.

$$\Delta w_i(t) = \eta f(w_i^T a) a$$

where η is learning rate parameter.
w is the weight.
a is input.

Therefore, the jth component of Δw_i is given by

$$\Delta w_{ij} = \eta f(w_i^T a) a_j$$

$$= \eta s_i a_j \quad \text{for } j = 1, 2, \ldots\ldots M$$

where s_i = output signal of the ith unit.

- The law states that the weight increment is proportional to the product of the input data and the resulting output signal of the unit.
- This law requires weight initialization to small random values around $w_{ij} = 0$ prior to learning.
- This law represents an unsupervised learning.

Perceptron Learning Law :

- Here the change in the ~~vector~~ weight vector is given by

$$\Delta w_i = \eta [b_i - sgn(w_i^T a)] a$$

where, η — learning rate parameter.

$sgn(x)$ — sign of x.

Therefore we have,

$$\Delta w_{ij} = \eta [b_i - sign(w_i^T a)] a_j$$
$$= \eta (b_i - s_i) a_j, \quad \text{for } j = 1, 2 \ldots\ldots M$$

- This law is applicable only for bipolar output functions f(.)
- This is also called discrete perceptron learning law.
- The expression for Δw_{ij} shows that the weights are adjusted only if the actual output s_i is incorrect, since the term in the square brackets is zero for the correct output.
- This is a supervised learning law, as the law required a desired output for each input.
- The weights can be initialized to any random initial values.

3. Delta Learning Law:

- Here change in the weight vector is given by

$$\Delta w_i = \eta [b_i - f(w_i^T a)] \dot{f}(w_i^T a) a$$

where $\dot{f}(x)$ is the derivative with respect to x. Hence,

$$\Delta w_{ij} = \eta [b_i - f(w_i^T a)] \dot{f}(w_i^T a) a_j$$

$$= \eta [b_i - s_i] \dot{f}(x_i) a_j \quad \text{for } j = 1, 2, \ldots\ldots M.$$

- This law is only valid for a differentiable output function, as it depends on derivative of the output function $f(\cdot)$.
- It is a supervised learning law since the change in the weight is based on the error between the desired and the actual output values for a given input
- Weight can be initialized to any random values.

4. Widrow and Hoff LMS learning Law.

- Here the change in the weight vector is given by

$$\Delta w_i = \eta [b_i - w_i^T a] a$$

Hence

$$\Delta w_{ij} = \eta [b_i - w_i^T a] a_j, \text{ for } j = 1, 2, \ldots\ldots M.$$

- This is supervised learning law.
- The change in the weight is made propotional to the negative gradient of the error between the desired output and the continuous activation value, which is also the continuous output signal due to linearity of

the output function.

- It is also known as Least Mean squared (LMS) error learning law.

5. Correlation Learning Law.

- Here the change in the weight vector is given by

$$\Delta w_i = \eta b_i a$$

$$\Delta w_{ij} = \eta b_i a_j, \quad \text{for } j = 1, 2, \ldots M.$$

- Correlation learning is an ~~un~~supervised learning, since it uses the desired output value to adjust the weights.

- The weights are initialized to small random values close to zero. ie $w_{ij} \approx 0$.

Features of ANN :→

1. ANN are extremely powerful computational devices.

2. ANN are modeled on the basis of current braint theories in which information is represented by weight

3. ANNs have massive parallelism which makes them very efficient.

4. They can learn and generalize from training data So there is no need for enormous feats of programming.

5. Storage is fault tolerant i.e. some portion of the neural net can be removed and there will be only a small degradation in the quality of stored data.

6. They are particularly fault tolerant which is equivalent to the "graceful degradation" found in biological systems.

7. Data are naturally stored in the form of associative memory which contrats with conventional memory, in which data are recalled by specifying address of that data.

8) They are very noise tolerant, so they can cope with situations where normal symbolic systems would have difficulty.

9) In practice they can do anything a symbolic logic system can do and more.

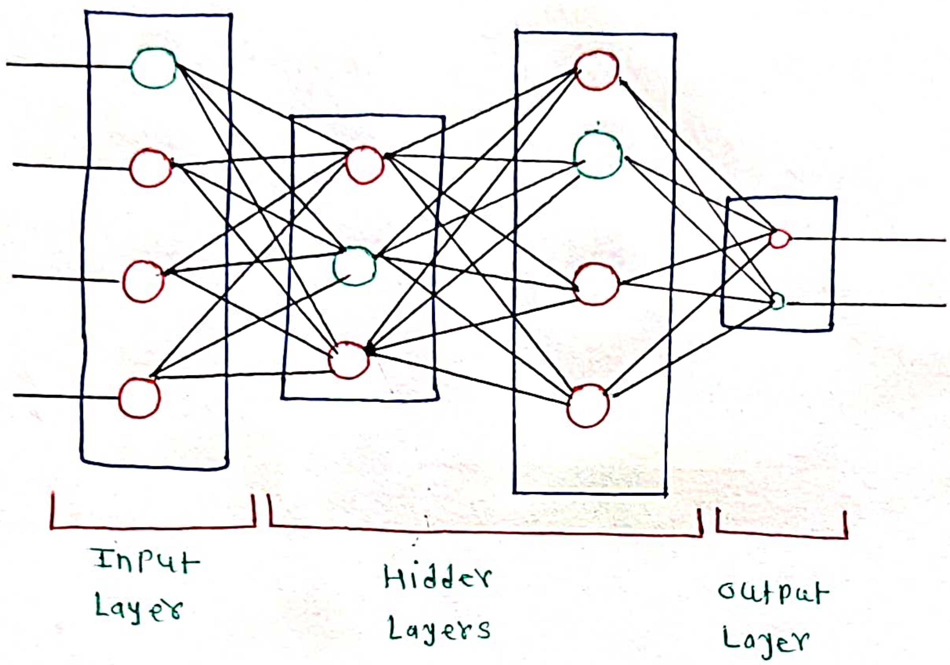
10) Neural networks can extrapolate and intrapolate from their stored information. The neural networks can also be trained. special training teaches the net to look for significant features or relationships of data.

Characteristics of Artificial Neural Network

1. It is neurally implemented mathematical model.
2. It contains huge number of inter connected processing elements called neurons to do all operations.
3. Information stored in the neurons are basically the weighted linkage of neurons.
4. The input signals arrive at the processing element through connections and connecting weights.
5. It has the ability to learn, recall and generalize from the given data by suitable assignment and adjustment of weights.
6. The collective behaviour of the neurons describes its computational power, and no single neuron carries specific information.

Neural net architecture :

- Neural network consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers.
- Artificial Neural Network primarily consists of three layers :



Input Layer :

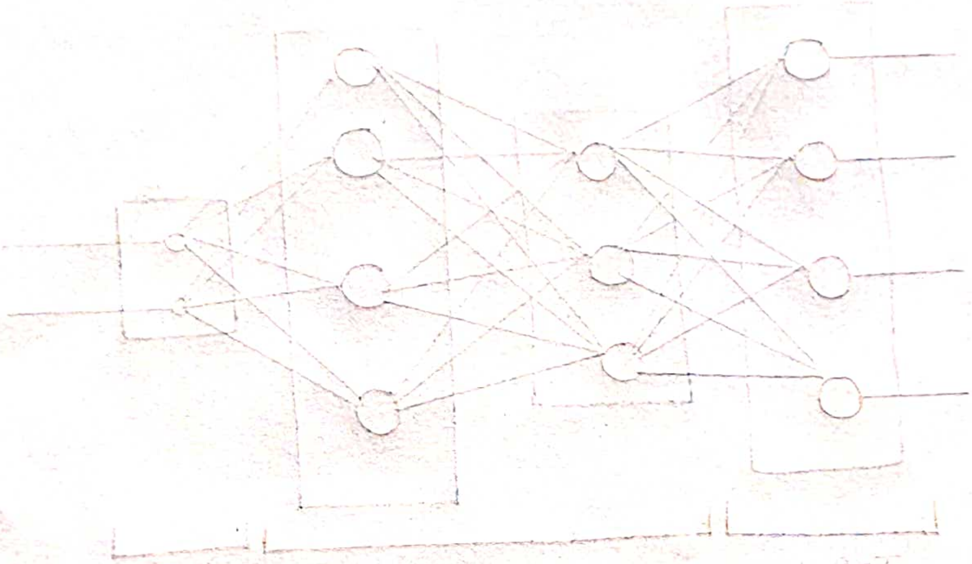
- As the name suggests, it accepts inputs in seveveral different formats provided by the programmer.

Hidden Layer :

- The hidden layer presents in between input and output layers. It performs all the calculations to find hidden features and patterns.

output layer'.

- The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.



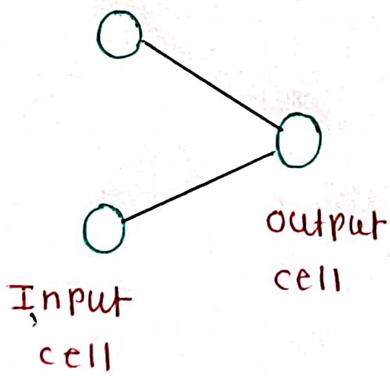
Types of Artificial Neural Network:

A. Perceptron :→

- Proposed by Minsky - papert is one of the simplest and oldest models of neuron.
- It is the smallest unit of neural network that does certain computations to detect features or business intelligence in the input data.
- It accepts weighted inputs, and apply the activation function to obtain the output of the final result.
- perceptron also known as TLU (Threshold logic Unit).
- perceptron is a supervised Learning algorithm that classifies the data into two categories, thus it is a binary classifier.
- A perceptron seprates the input space into two categories by a ~~hyperline~~ hyperplane

Advantage of perceptron :→

Perceptrons can implement Logic Gates like AND, OR, NAND



Disadvantages of perceptron :→

- Perceptrons can only learn linearly seprable problems such as boolean AND problem.
- For Non-linear problems such as booleon ~~AND~~ XOR problem, it does not work.

B. Feed forward Neural Networks

- The simplest form of neural networks where input data travels in one direction only, passing through artificial neural nodes and exiting through output nodes.
- Where hidden layers may or may not be present, input and output layers are present there.