MODULE - 1

Soft computing deals with imprecision, inaccurate and partial truth.

Application

- face recognition
- finger print recognition
- smart air conditioner

Different computing techniques

* neural network / artificial neural network (ANN)
* fuzzy logic (FL)
* Genetic algorithm (GA)

→ Difference between hard computing & Soft computing

Hard Computing	Soft computing
• Everything is planned. ie. we are aware of the outcomes.	• It deals with imprecision, inaccurate and partial truth.
• It requires program to be written.	• It can evolve its own programs.
• It uses 2-valued logic [yes/no, 0/1]	• It uses multi-valued / fuzzy logic.
• It is strictly sequencial	• It ~~is~~ allow parallel computing.
• It produces precise answers	• It produce approximate answers.
• eg:- normal AC	• eg:- Smart AC

Soft computing developed in the year 1981 by Lotfi A zadeh.

Fuzzy logic (FL)

It process data by allowing partial set memberrship rather than crisp set membership (normal set)

Boolean logic allows only 0 or 1. whereas fuzzy logic allows value b/w 0 & 1. (both 0,1 can include)

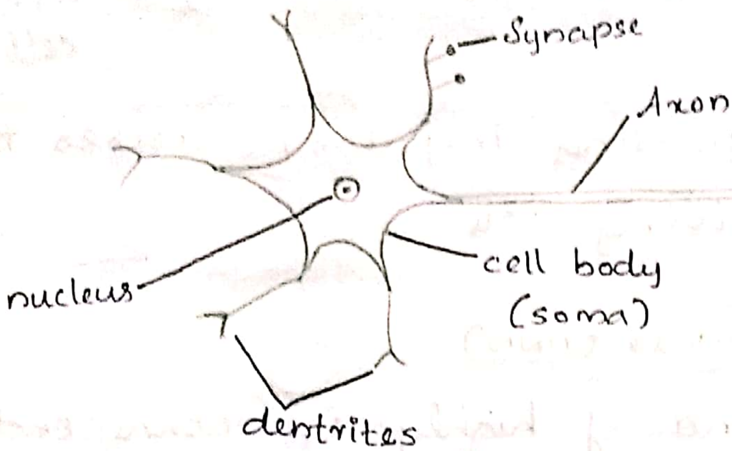
- FL provides a simple way to arrive at a definite conclusion based up on vague, ambigous, imprecise or missing input information.

Genetic algorithm

We create an initial population of individual representing possible solutions to a problem that we are trying to solve. Each of these individuals has certain characteristics that make them more or less fit as members of the population. The more fit members will have a higher probability of mating & producing offset that have a significant chance of retaining the desirable characteristics of their parents than the less fit members. In this way, we can find out the optimal solution.

Aatificial Neural Network (ANN)

Biological neuron



Biological neuron consist of cell body, nucleus, dentrites

Dentrites are tree like network made of neeve fiber connected to the cell body. The axon is a single long connection extending from the cell body & carrying signals from the neurons. The end of the axon is split into fine strands. Each strands terminale into a small bulb like organ called synapse.

It is through the synapse, the neuron introduces its signal to the other nearby neurons. The electric impulses are passed b/w the synapse & the dentrites.

After firing a cell has to wait for a period of time called the refractory period before it can fire again.

The synapse are 2 types.

i) inhibitory

ii) Excitatory

i) inhibitory :-

Synapse are said to be inhibitory if they let passes the impulse hinder the firing of receiving cell.

ii) Excitatory :-

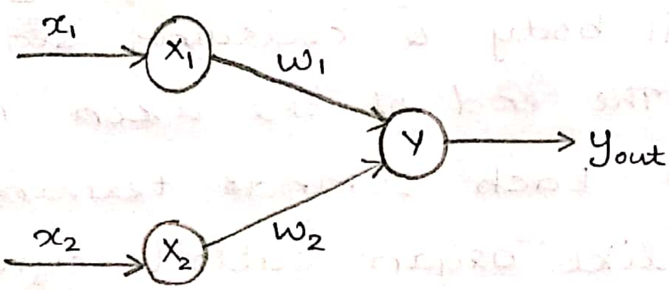
If they let passing impulses causes the firing of the receiving cell.

7/8/19

Artificial neural network (ANN)

ANN process large no. of highly interconnected processing elements called nodes or units or neurons which usually operates inparallel & are configured in regular architecture. Each neuron is connected to another by a connection link.

Architecture of a simple ANN



Each connection link is associated with weight which contains information about the input signals. This is used to solve a particular problem.

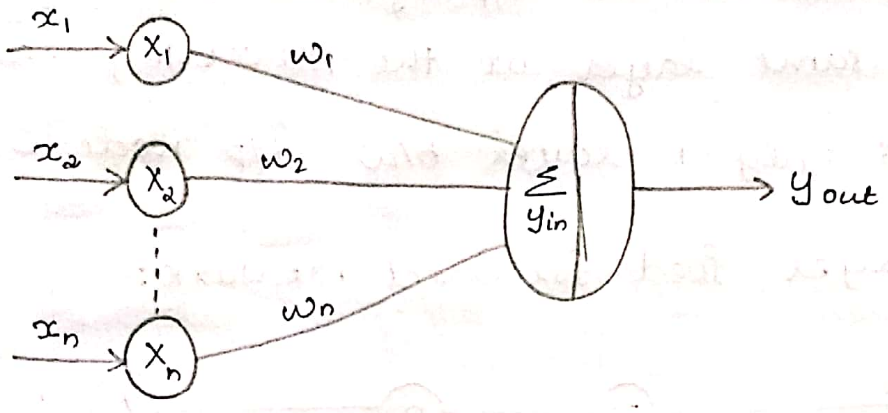
o/p neuron = Y , i/p neuron = X_1, X_2

net i/p = y_{in}

$$\boxed{\begin{aligned} y_{in} &= x_1 w_1 + x_2 w_2 = \sum_{i=1}^{n} x_i w_i \\ y_{out} &= f(y_{in}) \end{aligned}}$$

f → activation function

Mathematical model of ANN



Basic models of ANN

The basic models of ANN are specified by the 3 basic entities

i) The synaptic interconnection

ii) training or learning rule

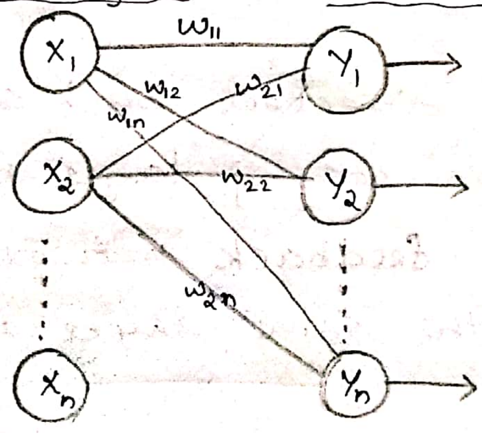
iii) activation function

i) Synaptic interconnection :-

Based on the interconnection, the NN are classified into

- single layer feed-forward network
- multi layer feed-forward network
- Single node with its own feedback
- Single layer recurrent network
- multi layer recurrent network

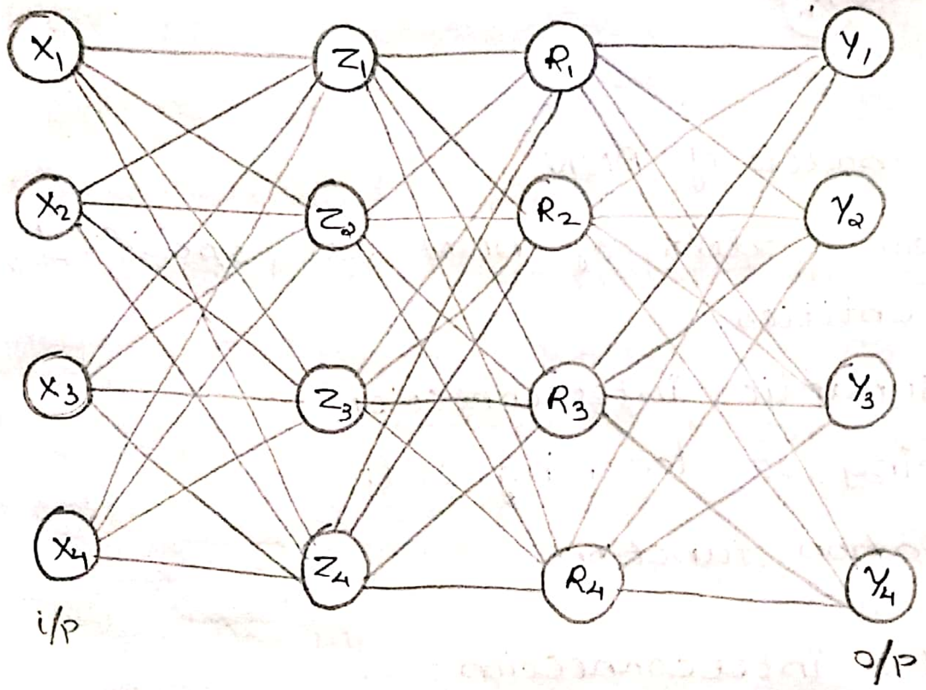
a) Single layer feed forward network :-



A network is said to be feed forward if no neuron in the o/p layer is an i/p to an node in the same layer or the preceding layer

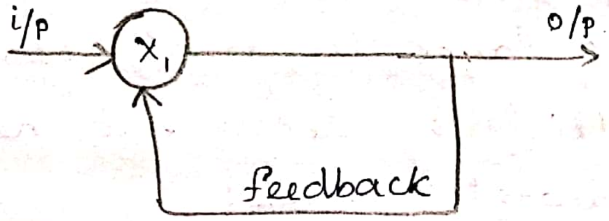
+ There is only 1 layer b/w i/p node & o/p node.

b) multi layer feed - forward network :-



Other than the i/p layer & o/p layer there is hidden layers.

c) Single node with its own feedback :-

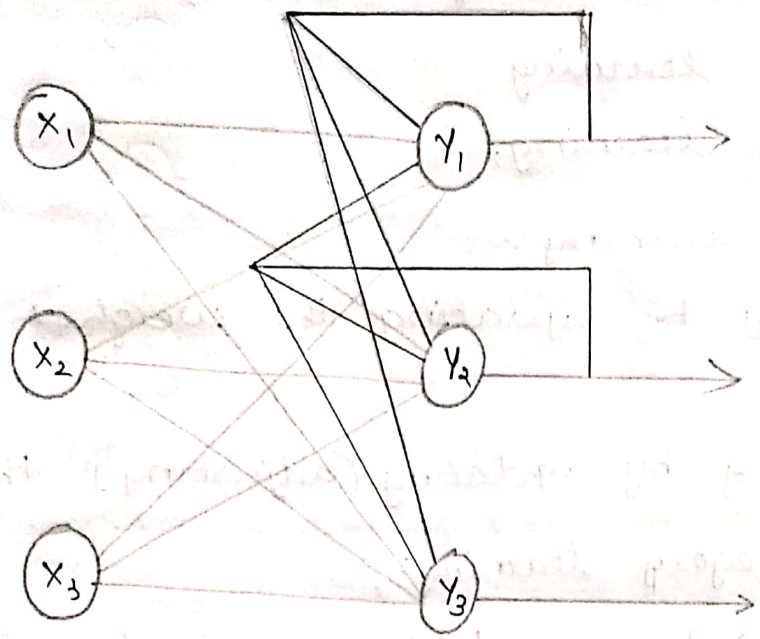


When the o/p are directed back as i/ps to the same or the preceeding layer nodes, then it is called as the feedback network

- feedback given to the same layer is called lateral feedback.

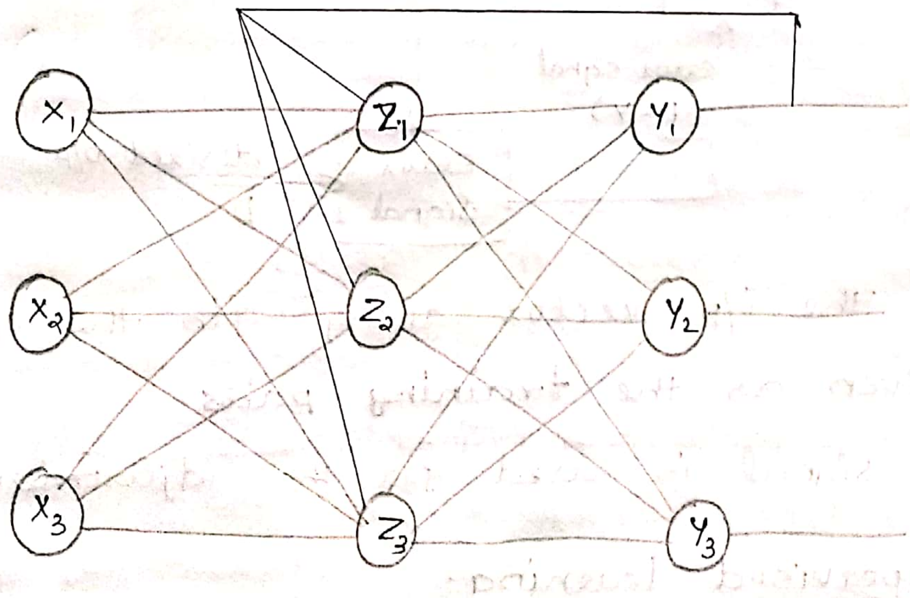
Scanned by CamScanner

d) Single layer recurrent network :-



Recurrent networks ~~are~~ feedback network with closed loop.

e) Multilayer recurrent network:-



Lateral Inhibition ~~s~~ structure

In this structure, each processing neurons receives 2 different classes of i/ps.

i) excitatory i/p from nearby processing element.

ii) inhibitory i/p from distantly located processing element.

ii) Learning :-

There are 2 kinds of learning.

a) parameter learning
b) structural learning.

a) Parameter learning :-

learning by updating the weights.

b) Structural

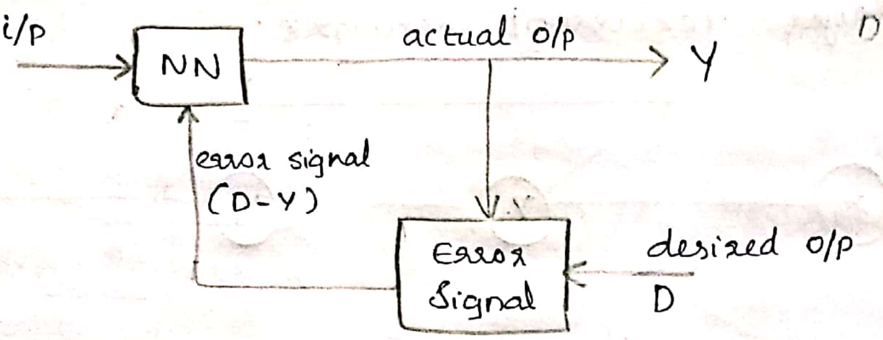
learning by updating (adjusting) the structure

Another category learning :-

a) Supervised . b) unsupervised

a) Supervised learning :-

learning with the help of supervisor.



The i/p vector along with the o/p vector is given as the **training** pairs.

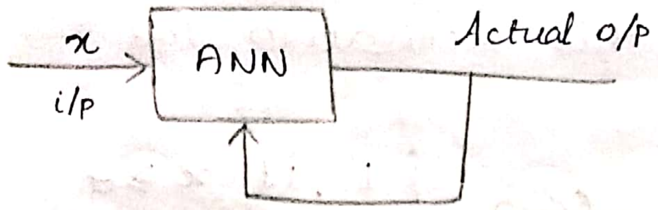
Error signal is used for the adjustment of weight.

b) unsupervised learning :-

Learning without the help of supervisor/teacher

- In this, the i/p 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 member belongs.

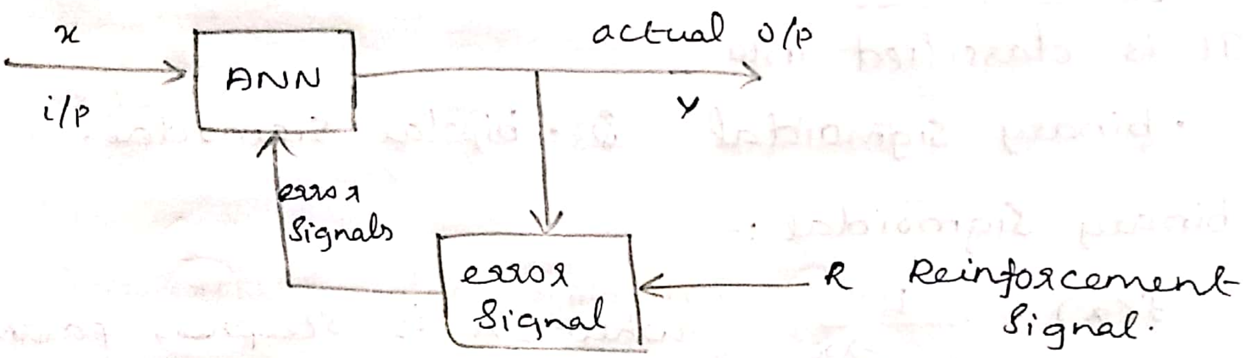
Scanned by CamScanner



In the training process, the network receives the i/p pattern & organises to form clusters. When a new i/p pattern is applied, the NN gives an o/p response indicating the class to which the i/p pattern belongs.

- **Reinforcement learning :-**

It is similar to the supervised learning.



Here only the critic information is available rather than the exact information (desired o/p)

iii) Activation function :- [AF]

It is a fn applied to net i/p [y_{in}] in order to get the o/p [y_{out}]

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

- The different types of activation fns are :

a) identity fn :-

It is a linear fn defined as $f(x) = x$ for all x.

The o/p is same as the i/p.

- The i/p layer uses this activation fn.

b) binary step fn :-

It is defined as $f(x) = \begin{cases} 1, & \text{if } x \geq \theta \\ 0, & \text{if } x < \theta \end{cases}$

. where x is the i/p & θ is the threshold. (limit)

c) bipolar step fn :-

The o/p can be either ~~0~~ 1 or -1. and it is defined as $f(x) = \begin{cases} 1, & \text{if } x \geq \theta \\ -1, & \text{if } x < \theta \end{cases}$

d) Sigmoidal fn :-

It is classified into

- binary sigmoidal & • bipolar sigmoidal.

• binary sigmoidal :-

$$f(x) = \frac{1}{1+e^{-\lambda x}}$$, where λ is steepness parameter

The value ~~range~~ of this fn ranges from 0 to 1.

• bipolar sigmoidal :-

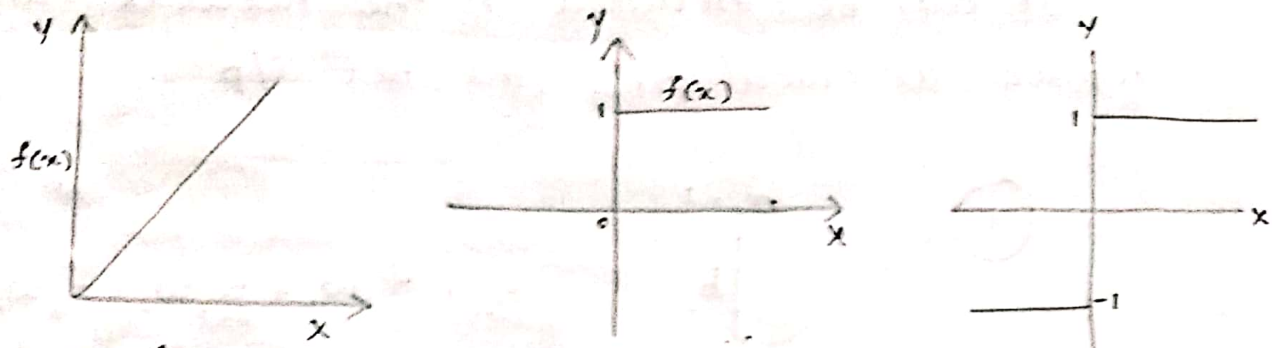
It is defined as $f(x) = \frac{2}{1+e^{-\lambda x}} - 1$

The value of this fn ranges from -1 to 1.

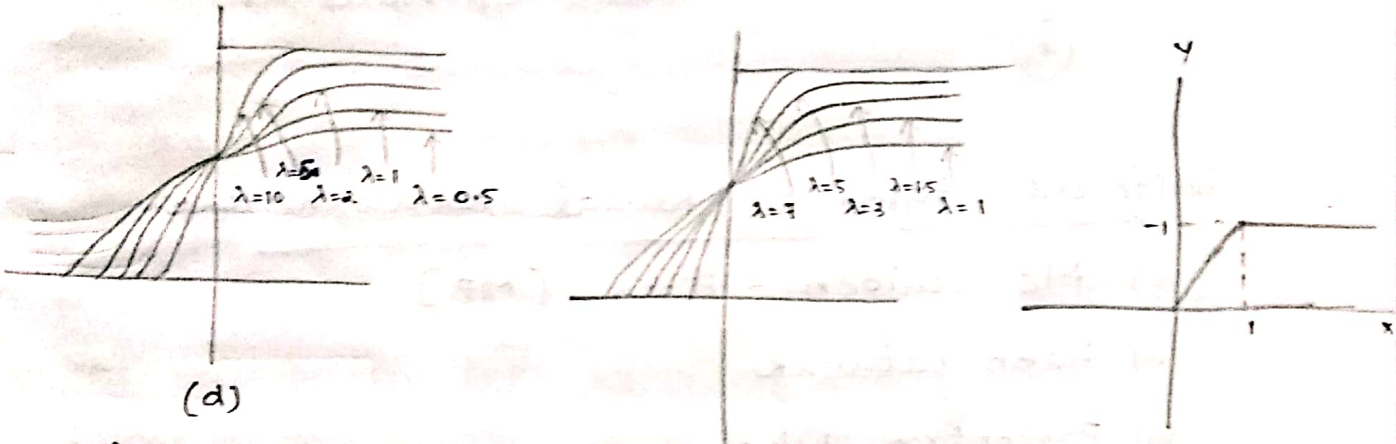
e) Ramp fn :-

It is defined as $f(x) = \begin{cases} 1, & \text{if } x > 1 \\ x, & \text{if } 0 < x < 1 \\ 0, & \text{if } x < 0 \end{cases}$

Graphical representation



(a) (b) (c)



(d) binary sigmoidal

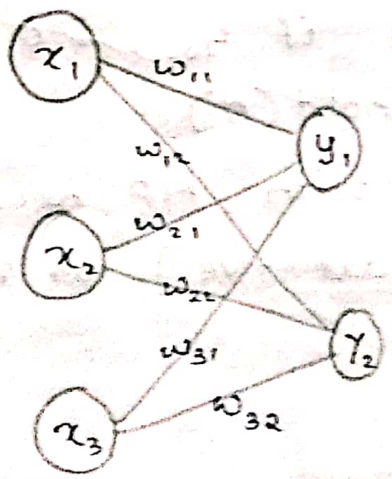
bipolar sigmoidal

(e)

Important terminologies in ANN

i) weight

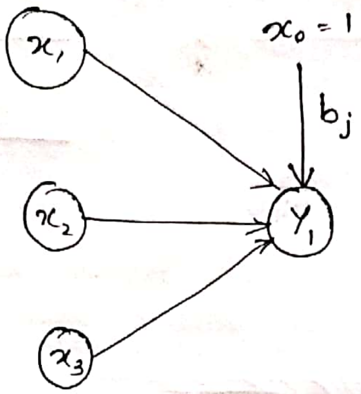
Weight of a NN can be represented as two dimensional matrix



$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{33} \end{bmatrix}$$

ii) <u>Bias</u> :-

It can be included in the network and has impact in calculating the net i/p



$$Y_{in} = x_1 w_1 + x_2 w_2 + x_3 w_3 + b_j$$

where b_j = weight associated with bias

bias :- i/p value = 1

Different types of neural network are

i) Mc Culloch - Pitts [MP]

ii) Hebb network

iii) Perceptron N/w

iv) Adaline

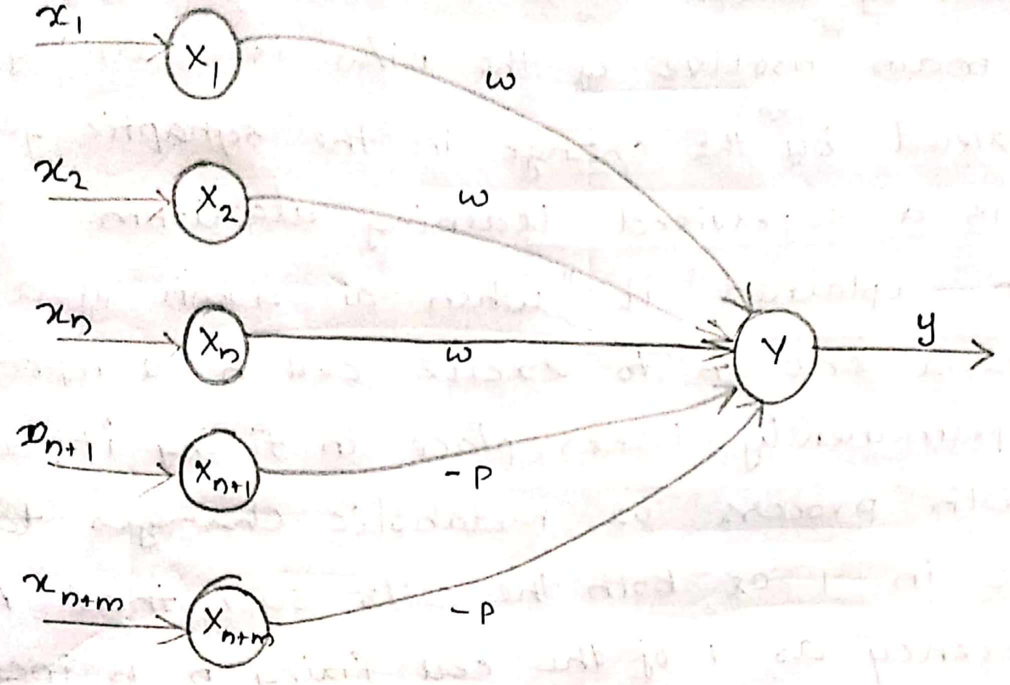
v) Back perceptron.

i) Mc Culloch - Pitts (MP) NN

- It uses binary step activation fn ie. either the neuron will fire (if $y_{out} = 1$). and neuron will not fire if $y_{out} = 0$ based on the i/p(x) and threshold (θ).
- The weight associated with MP neuron can be excitatory [weights are +ve] or inhibitory (weights are -ve)

Scanned by CamScanner

Model of MP :-



All the excitatory connection weight entering into a particular neuron will have the same weight.

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

θ = threshold.

Based on the value of θ & y_{in}, the network will fire or will not fire.

- For the inhibition to be absolute, the threshold with activation fn should satisfies the condition $\boxed{\theta \geq nw - p}$

  where n = no. of neurons

  w = weight of excitatory neurons.

  P = weight of inhibitory neurons.

- This network is usually used for logic fns. (gates)

Scanned by CamScanner

ii) Hebb network :-

- Founded by Donald Hebb in 1949.
- The main motive of the N/w is learning is performed by the change in the synaptic gap.
- It is a supervised learning algorithm
- Hebb explained it "when an axon of a cell A is near enough to excite cell B, & repeatedly or permanently takes place in firing it, some growth process or metabolic changes takes place in 1 or both the cells such that A's efficiency as 1 of the cell firing B is increased".
- In Hebb learning if 2 interconnected neurons are ON simultaneously, then the weight associated with this neuron can be increased by modification made in their synaptic gap (strength). The weight update in Hebb rule is given by,

$$\boxed{w_{i\,new} = w_{i\,old} + x_i y}$$

- The Hebb rule is more suited for bipolar data. [-1, 1] If binary data is used, the above weight updation formula cannot distinguish the below 2 condition..
  - A training pair in which an i/p unit is ON & target value is OFF
  - A training pair in which both the i/p & the target value are OFF

In this 2 cases, $w_{new} = w_{old}$.

22/8/19

Training algorithm

Step 0: Initialize the weights & bias.

In this N/w usually it is set to 0.

ie. $w_i = 0$ for $i = 1$ to n where n is the no of i/p neurons

Step 1: Step 2 - 4 have to be performed for each i/p training vector & target o/p pair.

ie. $s : t$, where s is the i/p & t is the o/p.

Step 2: i/p unit activations are set. Usually the i/p layer uses the identity activation fn.

ie. $x_i = s_i$ ~~$\&\ y = t$~~, where $i = 1$ to n.

Step 3: O/p units activations are set. ie. $y = t$

Step 4: Weight adjustment & biased adjustment are performed.

ie. $w_{i(new)} = w_{i(old)} + x_i y$ ——— (1)

$b_{new} = b_{old} + y$; $b_{new} - b_{old} = y$

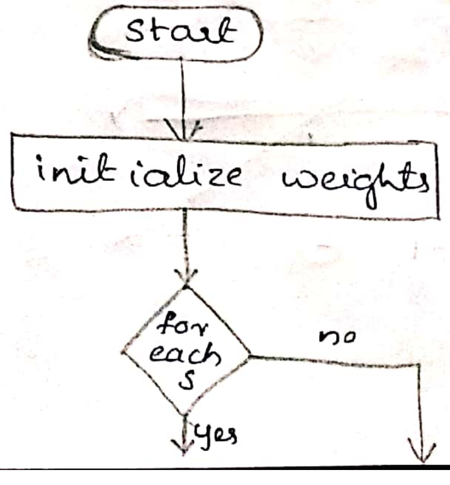
$\Delta b = y$

(1) → $w_{i(new)} - w_{i(old)} = x_i y$

$\Delta w_i = x_i y$

(1) → $w_{i\,new} = w_{i(old)} + \Delta w_i$

flowchart of hebb network :-



Activate i/p units

$x_i = s_i$

↓

Activate o/p units

$y = t$

↓

Weight update

$w_{i(new)} = w_{i(old)} + x_i y$

↓

Bias update

$b_{(new)} = b_{(old)} + y$

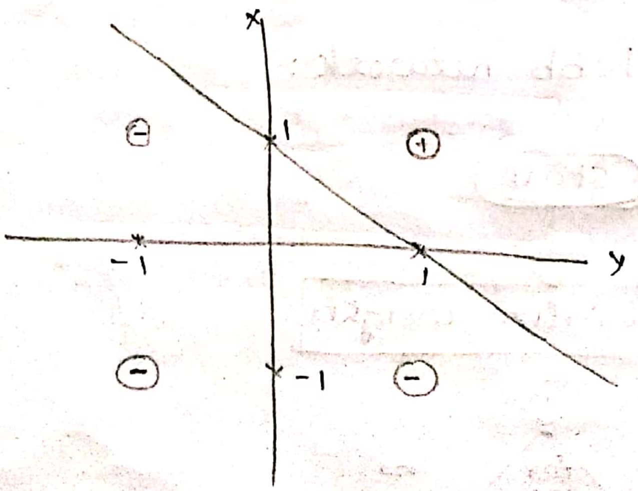
↓

Stop

* Linear separability $(0 = -1)$

AND

x_1	x_2	y		x_1	x_2	y	
0	0	0	⟹	−1	−1	−1	−ve response
0	1	0		−1	1	−1	−ve response
1	0	0		1	−1	−1	−ve response
1	1	1		1	1	1	+ve response

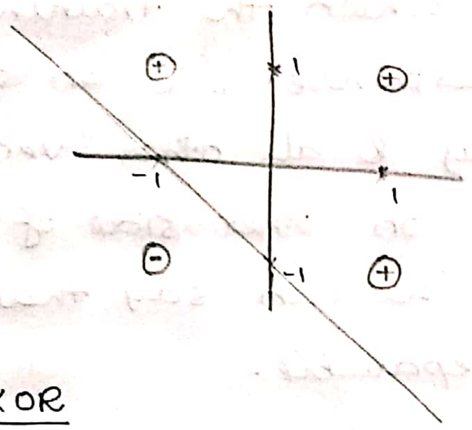


If you can separate the +ve responses & -ve response using a single straight line, then we can say that the fn is linearly separable.

OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

x_1	x_2	y	
-1	-1	-1	-ve
-1	1	1	+ve
1	-1	1	+ve
1	1	1	+ve

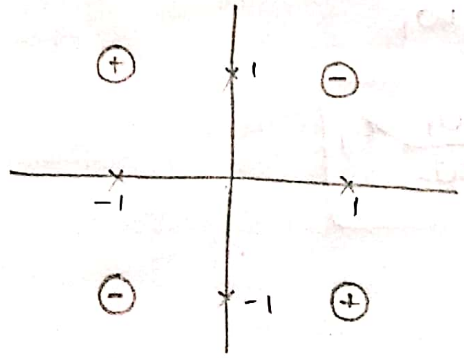


$\therefore$ OR fn is linearly separable.

XOR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

x_1	x_2	y
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1



It cannot be separate -ve responses & +ve responses
So, XOR fn is not linearly separable.

A decision line is drawn to separate +ve & -ve response. The decision line is also called decision making line / decision support line / linear separable line.

- The necessity of linear separability concept is to classify the patterns based upon their o/p response.
- The linear separability of the n/w is based on the decision boundary line. If there exists weights with bias for which the training i/p vector having +ve response lie on one side of the decision boundary & all other vectors having -ve response lie on other side of decision. Then we can say that the problem is linearly separable.
- For +ve response, $y_{in} > 0$
  for -ve response, $y_{in} < 0$

$$f(y_{in}) = \begin{cases} 1, & y_{in} \geq 0 \\ -1, & y_{in} < 0 \end{cases}$$

$\therefore$ equation of separating line, $y_{in} = 0$

$$y_{in} = 0$$

$$x_1 w_1 + x_2 w_2 + b = 0$$

$$x_2 w_2 = -b - x_1 w_1$$

$$\boxed{x_2 = \frac{-b}{w_2} - \frac{w_1}{w_2} . x_1}$$