Soft Computing :

↳ Introduced by Lotfi Zadeh

Goal :- able to emulate human brain as closely as possible.
i.e. self learning.

Combination of Genetic Algorithms, Neural Network & fuzzy logic along with evolutionary computing

Hard Computing

$\boxed{\text{Precise Model}}$

- Symbolic logic Reasoning (traditional AI)
- Traditional Numeric Reasoning modeling & Search

Soft Computing

$\boxed{\text{Approximate Models}}$

- Approximate Reasoning
- functional Approximation & Randomized Search

classmate
Date
Page

→ Artificial Neural Network (ANN)
Artificial

(*) Neural Network is a processing device either an algorithm or an actual NW whose design was inspired by design and functioning of animal brain & components thereof.

→ have ability to learn by example flexible & powerful.

→ no need to device an algo, to perform specific task.

(2) Human Brain :-
how it works is a mystery
→ most basic element is neuron which does not regenerate.
→ provide us ability to
  L think
  L learn (apply experiences)
  L Remember.

(*) Artificial Neural Network
defined as an information processing model i.e. inspired by how biological nervous system works, such as the brain process information.

→ replicates the most basic function of brain
→ learn by example

(*) Advantages of Neural N/W

1) Adaptive learning
 L ability to learn how to do task based on given data.

2) Self Organization
 L Can create its own representation of information it's receives during learning time

3) Real time Operation
 L computations may be carried out in parallel.
 L special H/W needed

4) Fault tolerance via redundant information coding

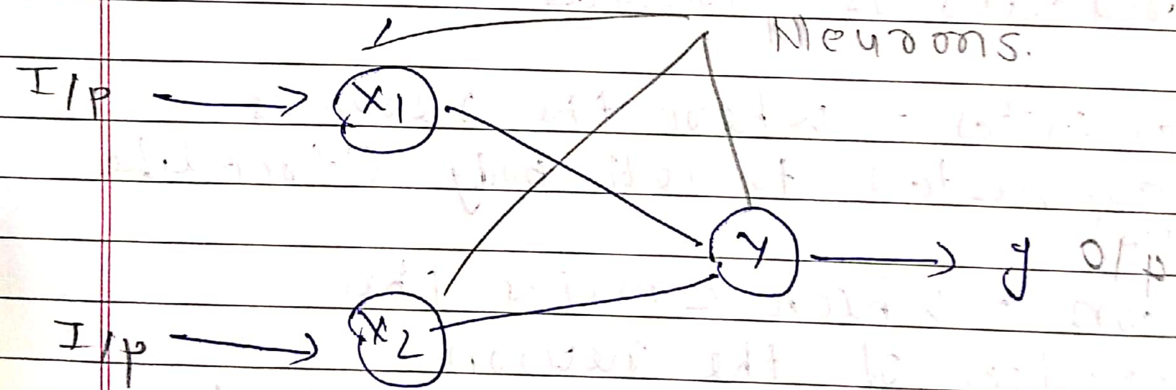
⊛ Objective of Neural Network

To develop a computational device for modeling the brain to perform various computational task at a faster rate than a traditional system.

⊛ Neuron :- Processing element in ANN

→ Each neuron has an internal state of its own, called the activation or activity level of neuron, which a function of i/p the neuron receives

→ The activation signal of neuron can be transmitted to other neurons. (one signal at a time)



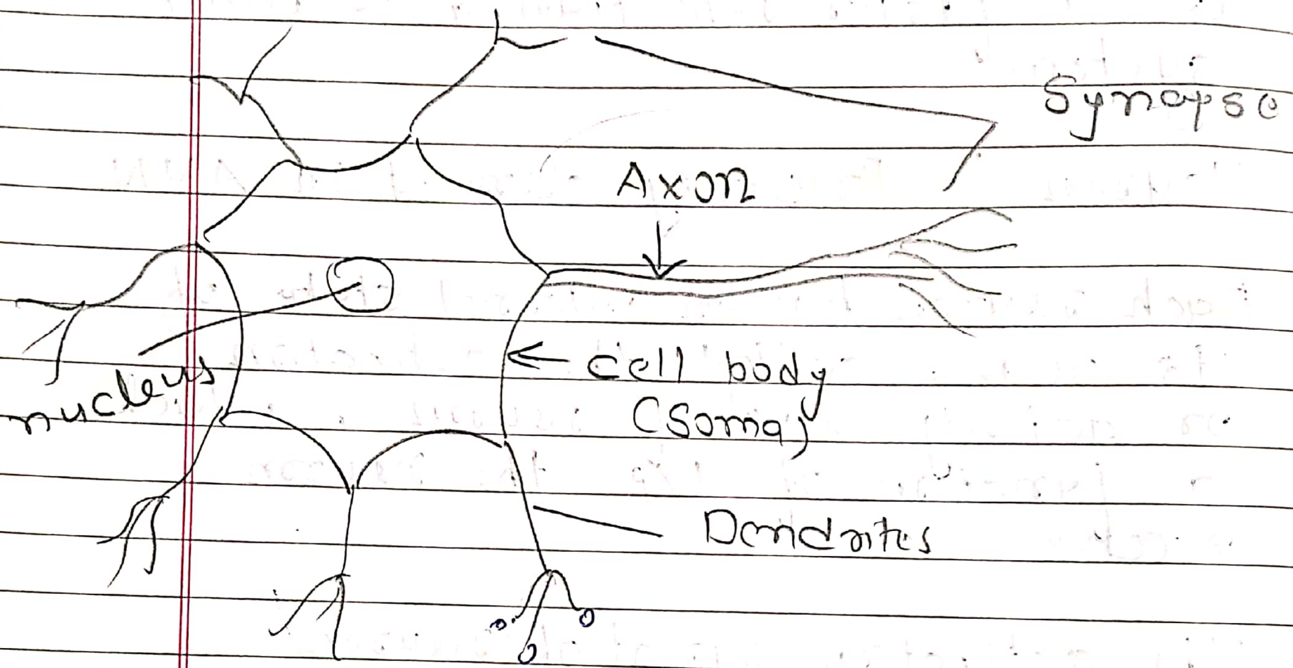
$$y_{in} = x_1 w_1 + x_2 w_2$$

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

O/p = function (net i/p calculated)

(*) Biological Neural n/w

Brain consists of approximately 10^{11} neurons, with numerous interconnections



1) Soma / Cell body – where the cell nucleus is located

2) Dendrites – where the nerve is connected to cell body (tree like n/w)

3) Axon – which carries the impulses of the neuron
→ single, long connection extending the cell body ~~after~~ ~~carrying~~

4) The end of the Axon & Dendrites ~~split~~ splits in to fine strands
⟹ ~~[illegible]~~ Synapse

It is found that each strand terminates into a small bulb like organ called Synapse.

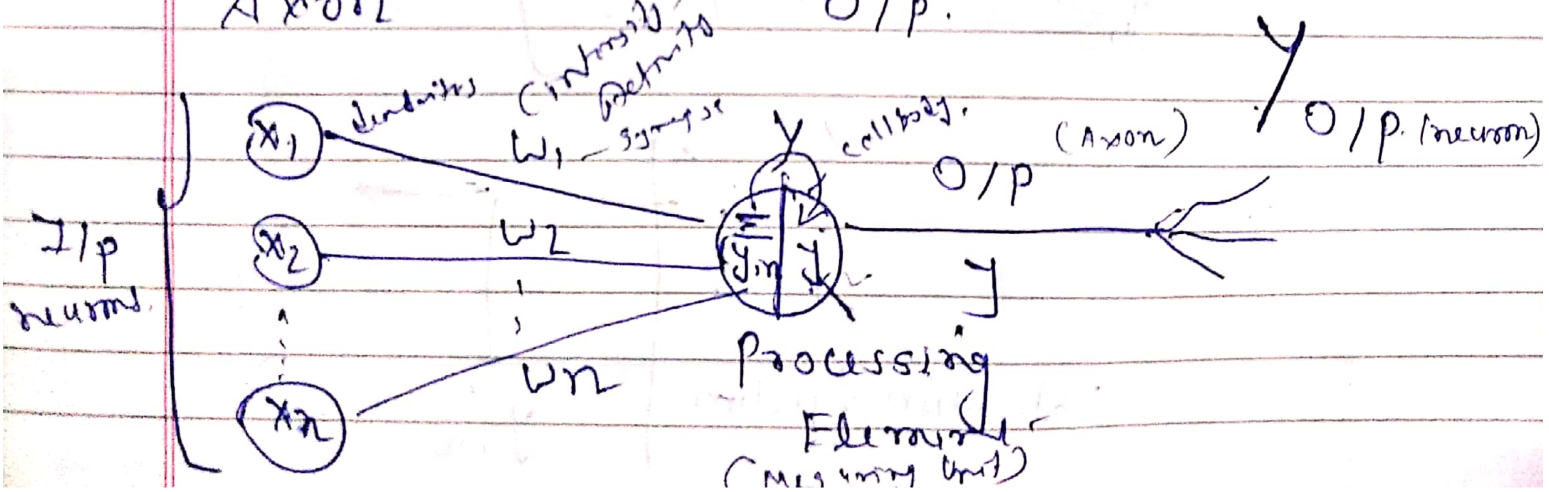
→ through Synapse neuron can introduce signals to near by neurony.

⇒ Electric impulses are passed between synapse & Dendrites

if electrical potencial inside the body reaches a threshold then the receiving cells fires an impulse or action of fixed strength & duration, which sent through the axon to the synaptic junction of the other cells.

⊗ Artificial Neuron:-

Biological Neuron	Artificial Neuron
Cell	Neuron.
Dendrites	(Synapse) Weights or interconnection
Soma	Net input
Axon	O/p.



net input

$$y_{in} = x_1 w_1 + x_2 w_2 + \dots + x_n w_n$$

$$= \sum_{i=1}^{n} x_i w_i$$

where i is the processing element.

→ The activation function is to apply over processing elements to calculate the o/p

→ The weights represents strength of synapse connecting the i/p & o/p neuron

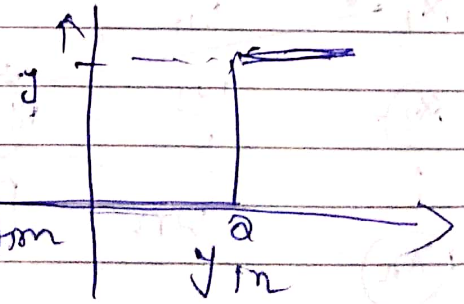
→ A positive weight corresponds to excitatory synaplse = w

A negative weight corresponds to inhibitatory synapse = w

$y_{in} \geq \theta \Rightarrow y = 1$

$y_{in} < \theta \Rightarrow y = 0$

θ - Threshold



Activation function

ANN - Artificial Neural Network

BNN - Biological Neural Network

(BN)

Comparision between biological Neuron & Artificial Neuron. (AN)

① **Speed** -

Biological Neuron is slower than AN because of massive parallel architecture.

BNN - 10^{-3} Seconds

ANN - 10^{-9} seconds

② **Processing** :-

BNN is faster than ANN because of parallel processing.

③ **Storage Capasity** :-

BNN - information is stored in synapse strength

ANN - information stored in contiguous memory allocation

(Memory overloading can be a problem)

In BNN new information is added by altering interconnections by adjusting strength without destroying older information.

In ANN information get lost or currepted if connection's disconnected.

(4) Size & Complexity

Total No. of neurons in brain is about 10^{11}

Total no of interconnections: 10^{15}

So. BNN Complexity is higher than ANN

→ Size & Complexity of ANN is based on application

(5) Tolerance :-

(b'ce of $\parallel^{el}$ nature)

BNN has fault tolerance capability while ANN are not fault tolerant

(6) Control Mechanism :-

AN modeled using computer, there is control unit present in CPU.

There is no control unit present in BNN as such an ANN, channels of brain control the informations through chemical actions.

Control mechanism of AN is very simpler than BN

Ex Implement Simple AND function with Neuron.

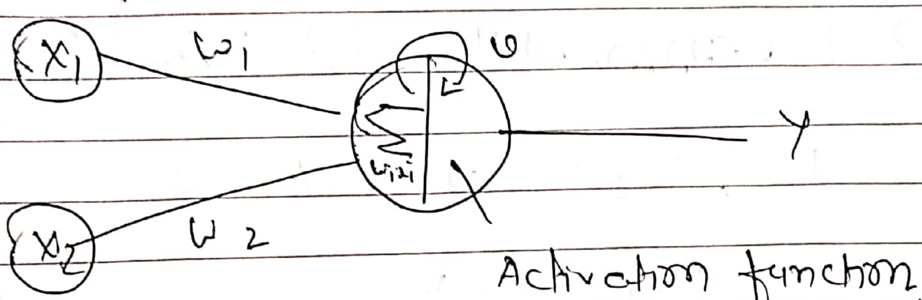
x_1	x_2	y	θ	v
0	0	0	1	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	2

Conclusion

$v > 1 \Rightarrow y = 1$

$v \leq 1 \Rightarrow y = 0$

- Assume $w_1 = w_2 = 1$



$v = \sum x_i w_i + b$

$v > \theta \Rightarrow y = 1$

$v \leq \theta \Rightarrow y = 0$

→ Activation function

$v > \theta \Rightarrow y = 1$

$v \leq \theta \Rightarrow y = 0$

→ $v = \sum x_i w_i$

① for $x_1 = 0$, $x_2 = 0$, $y = 0$

$v = x_1 w_1 + x_2 w_2$

$= (0 \times 1) + (0 \times 1) = 0$ $\quad 0 \leq \theta$

$y = 0$

now $v \leq \theta \Rightarrow y = 0$

$\theta >= 0$ let's take $\boxed{\theta = 1}$

0.2 . 0.3 . . .

(2) $x_1 = 0$, $x_2 = 1$, $y = 0$

$$\begin{aligned} v &= x_1 w_1 + x_2 w_2 \\ &= (0 \times 1) + (1 \times 1) \\ &= 1 \end{aligned}$$

$y = 0$

$$\begin{aligned} v &\leq \theta \Rightarrow y = 0 \\ 1 &\leq \theta \Rightarrow y = 0 \end{aligned}$$

$\theta >= 1$. let take $\boxed{\theta = 1}$

(3) $x_1 = 1$, $x_2 = 0$, $y = 0$

$$\begin{aligned} v &= x_1 w_1 + x_2 w_2 \\ &= (1 \times 1) + (0 \times 1) \\ &= 1 \end{aligned}$$

$v = 1$ so $\boxed{\theta >= 1}$ same as (2)

(4) $x_1 = 1$, $x_2 = 1$, $y = 1$

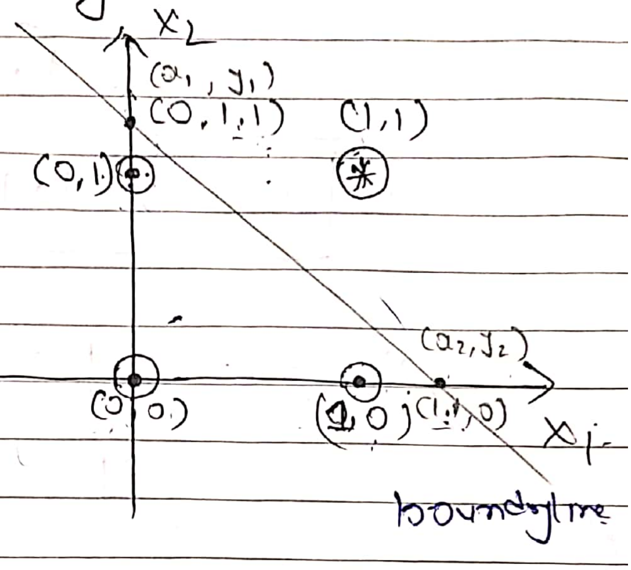
$$\begin{aligned} v &= x_1 w_1 + x_2 w_2 \\ &= (1 \times 1) + (1 \times 1) \\ &= 2 \end{aligned}$$

$$\left.\begin{aligned} v > \theta &\Rightarrow y = 1 \\ 2 > \theta &\Rightarrow y = 1 \end{aligned}\right\} \underline{\theta < 2}$$

take $\boxed{\theta = 1}$

Ex-2 Using linear Separability concept obtain the response for AND function (take binary i/ps)

x_1	x_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

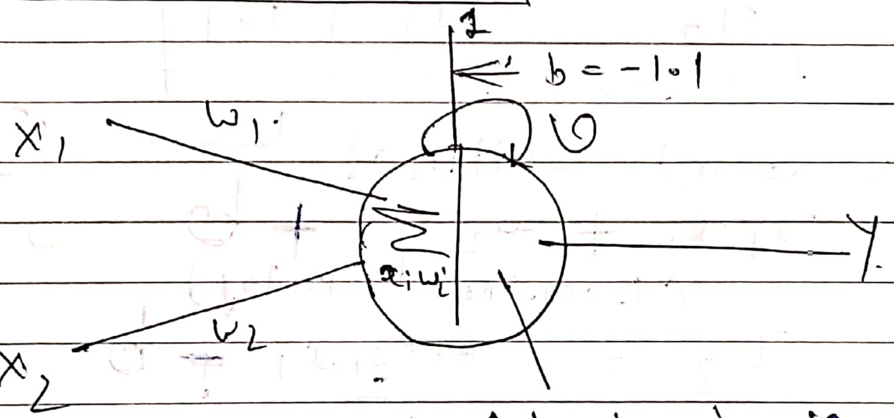


Note :-

no of lines needed to separate pattern = no of neurons needed in NW

$\odot \Rightarrow Y = 0$

$\circledast \Rightarrow Y = 1$



Activation function } $v > 0 \Rightarrow Y = 1$
$v \leq 0 \Rightarrow Y = 0$

line equation

$$\boxed{Y = mx + C}$$

$$m = \frac{y_2 - y_1}{a_2 - a_1}$$

$$= \frac{0 - 1.1}{1.1 - 0}$$

$$\boxed{m = -1}$$

$v = \sum w_i x_i + b > 0$ for positive response

$v = \sum w_i x_i + b < 0$ for negative response

$$y_1 = m x_1 + c$$

$$c = y_1 - m x_1$$

$$= 1.1 - ((-1) \times 0)$$

$$\boxed{c = 1.1}$$

now, $y = mx + c$

$$y = (-1)x + 1.1$$

$$y = -x + 1.1$$

but we have $x = x_1$ & $y = x_2$

So, $\boxed{x_2 = -x_1 + 1.1}$ — (1)

$v = \sum x_i w_i + b$

~~& this can be written as~~

$$\cancel{w_1 x_1 + w_2 x_2 + b = 0}$$

~~(Decision boundary)~~

$$x_2 = -\frac{w_1}{w_2} x_1 + \frac{b}{w_2}$$ — (2)

Activation function

$v \leq 0 \Rightarrow y = 0$

$v > 0 \Rightarrow y = 1$

Compare (1) & (2)

$\boxed{\frac{w_1}{w_2} = 1}$ & $\boxed{b = -1.1}$

i.e. $\boxed{w_1 = 1}$ $\boxed{w_2 = 1}$ $\boxed{b = -1.1}$

Scanned with CamScanner

(0,0) $y_{in} = 0\times1 + 0\times1 - 1.1 < 0 \Rightarrow Y = 0$

(0,1) $y_{in} = 0\times1 + 1 - 1.1 < 0 \Rightarrow Y = 0$

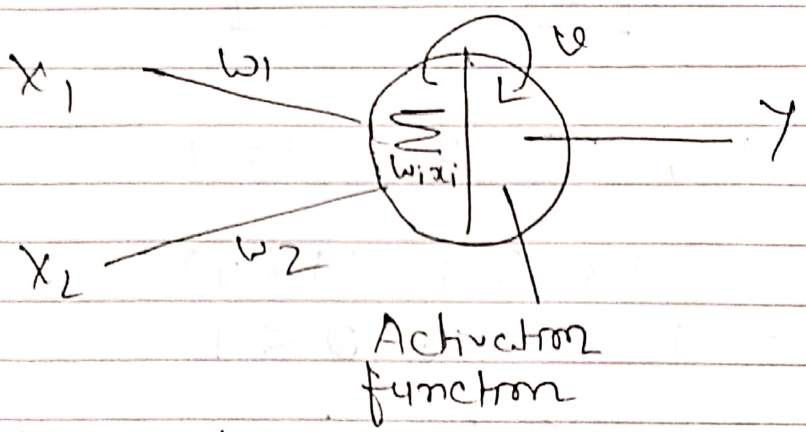
(1,0) $< 0 \Rightarrow Y = 0$

(1,1) $2 - 1.1 > 0 \Rightarrow Y = 1$

Ex. 3 Implement OR function with neuron.

x_1	x_2	Y	v	θ
0	0	0	0	0.2
0	1	1	1	0.2
1	0	1	1	0.2
1	1	1	2	0.2

→ take $w_1 = w_2 = 1$ (Assume)



Conclusion

$v > 0.2 \Rightarrow Y = 1$

$v \le 0.2 \Rightarrow Y = 0$

Activation function

$v > \theta \Rightarrow Y = 1$

$v \le \theta \Rightarrow Y = 0$

(1) $x_1 = 0, \; x_2 = 0, \; Y = 0$

$v = x_1 w_1 + x_2 w_2$

$= (0\times1) + (0\times1)$

$= 0$

So $v \le \theta \Rightarrow Y = 0$

$0 \le \theta \Rightarrow Y = 0$

$\theta ? = 0$ ~~......~~. Take $\boxed{\theta = 0.2}$

(2) $x_1 = 0, \; x_2 = 1, \; Y = 1$

$\vartheta = W_1 x_1 + W_2 x_2$
$= (1 \times 0) + (1 \times 1)$
$= 1$

So $\vartheta > \partial \Rightarrow Y = 1$
$1 > \partial \Rightarrow Y = 1$
$\partial < 1$
$\partial = [0 \text{ to } 0.9]$

take $\boxed{\partial = 0.2}$

(3) $x_1 = 1, \; x_2 = 0, \; Y = 1$

$\vartheta = (1 \times 1) + (1 \times 0)$
$= 1$ i.e. $\partial < 1$

Same as (2). take $\boxed{\partial = 0.2}$

(4) $x_1 = 1, \; x_2 = 1, \; Y = 1$

$\vartheta = (1 \times 1) + (1 \times 1)$
$= 2$

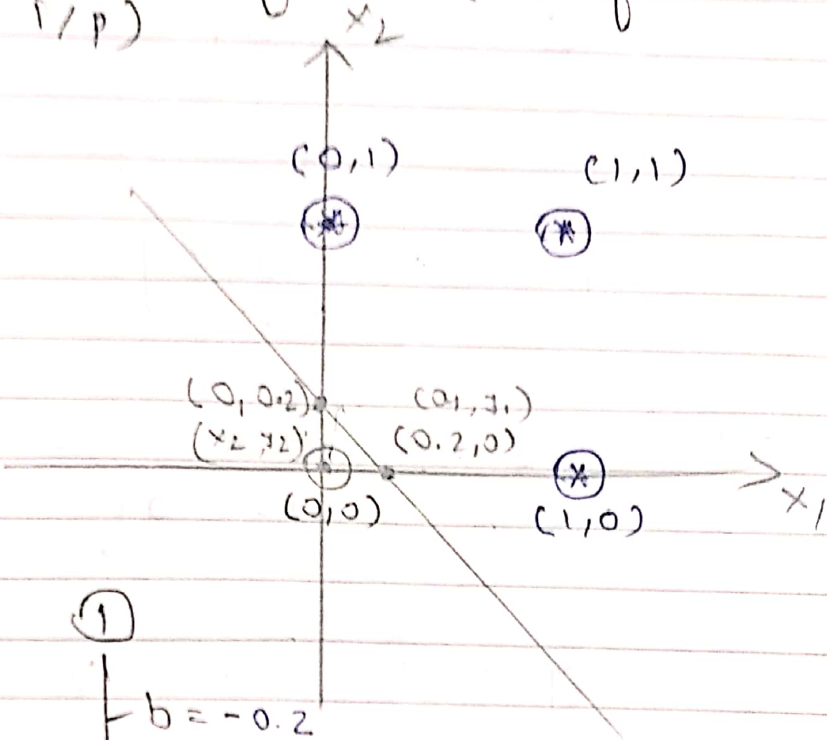
So, $2 > \partial \Rightarrow Y = 1$

$\partial = [0 \ldots 0.9]$

~~take~~ $\boxed{\cancel{\partial = 0.2}}$

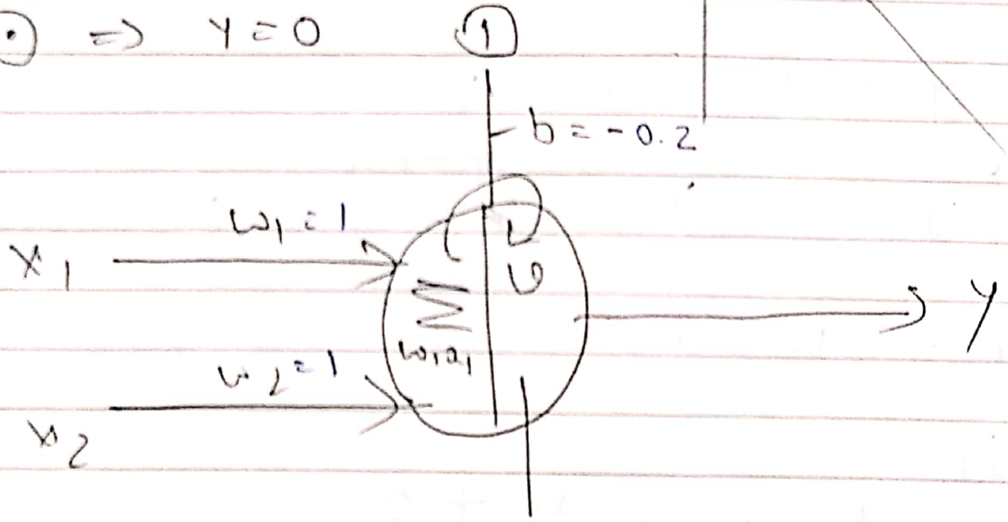
Ex-4 Using linear Separability concept obtain the response for OR function (Use binary i/p)

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



(*) $\Rightarrow$ $y = 1$

(·) $\Rightarrow$ $y = 0$



Activation function $\begin{cases} v > 0 \Rightarrow y = 1 \\ v \leq 0 \Rightarrow y = 0 \end{cases}$

→ equation for line $\Rightarrow$ $y = mx + c$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$= \frac{0.2 - 0}{0 - 0.2} = \boxed{-1}$$

classmate

Date

Page

→ $y_1 = mx_1 + c$

$c = y_1 - mx_1$

$= 0 - [(-1) \times 0.2]$

$$\boxed{c = 0.2}$$

→ $y = mx + c$

put values of m & c

$y = (-1)x + 0.2$

$$\boxed{y = -x + 0.2} \quad \text{--- (1)}$$

→ here we have $y = x_2$ & $x = x_1$

So

$x_2 = -x_1 + 0.2$

also we can write

$w_1x_1 + w_2x_2 + b = 0$ (Decision boundary)

$w_2x_2 = -w_1x_1 - b$

$$\boxed{x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2}} \quad \text{--- (2)}$$

Compare (1) & (2)

$u = a_1 w_1 + a_2 w_2 + b$

$\boxed{w_1 = w_2 = 1}$ & $\boxed{b = -0.2}$

Ex:- Implement a function using Neuron to identify following objects are eatable by a bird or not. Take color & shape as i/p.
(Blue) (round)

	Color (Blue = ?)	Shape (round?)	eatable ?
blue berry	1	1	1
Violet	1	0	0
Golf ball	0	1	0
hot dog	0	0	0

1 = Yes
0 = NO

classmate
Date
Page

(*) Macculloch Pitts Neuron
(McP / MP neuron)

→ Threshold with activation function should satisfy following condition

$\boxed{\theta \geq nw - p}$

w = excitatory w
p = inhibitatory w
n = no. of i/p.

$v \geq \theta \Rightarrow y = 1$

$v < \theta \Rightarrow y = 0$

Ex:- AND function using McP neuron (binary data)

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

→ Assume $w_1 = 1$, $w_2 = 1$
(both ~~exci~~ excitatory w)

for (0,0)
$v = x_1 w_1 + x_2 w_2$
$= 0 \times 1 + 0 \times 1$
$= 0$

for (0,1) $= 0 \times 1 + 1 \times 1 = 1$
(1,0) $= 1 \times 0 + 0 \times 1 = 1$
(1,1) $= 1 \times 1 + 1 \times 1 = 2$

here $\theta \geq 2$ then & only the neuron fires.

This can also be obtained by.

$$\theta \geq nw - p$$

$$\theta \geq 2(1) - 0$$

$$\boxed{\theta \geq 2}$$

$$\boxed{\begin{array}{lll} v \geq 2 & \Rightarrow & y = 1 \\ v < 2 & \Rightarrow & y = 0 \end{array}}$$

Activation function.

& $\boxed{w_1 = w_2 = 1}$

Ex-2 Implement AND-NOT function using McP neuron

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

for inputs.

Case I

$w_1 = w_2 = 1$

$(0,0)$, $v = 0 \times 1 + 0 \times 1 = 0$

$(0,1)$, $v = 0 \times 1 + 1 \times 1 = 1$

$(1,0)$, $v = 1 \times 1 + 0 \times 1 = 1$

$(1,1)$, $v = 1 \times 1 + 1 \times 1 = 2$

Assume $w_1 = w_2 = 1$

→ for $w_1, w_2 = 1$ & from calculated net i/p (v), it is not possible to fire the neuron for i/p $(1,0)$ only

→ hence these weights are not suitable

Case – II Assume one weight as excitatory & other as inhibitatory.

i.e. $w_1 = 1$, $w_2 = -1$

now for inputs.

$(0,0)$, $v = 0 \times 1 + 0 \times (-1) = 0$
$(0,1)$, $v = 0 \times 1 + 1 \times (-1) = -1$
$(1,0)$, $v = 1 \times 1 + 0 \times (-1) = 1$
$(1,1)$, $v = 1 \times 1 + 1(-1) = 0$

from calculated i/p it is possible to get required o/p.

$\theta >= n.w - p$

$\theta >= 2(1) - (1)$

$\theta >= 1$

Activation function

θ $v \geq 1 \Rightarrow y = 1$
$v < 1 \Rightarrow y = 0$

& $\boxed{w_1 = 1}$ $\boxed{w_2 = -1}$

classmate
Date
Page

Ex-3 Impliment XOR function using Mc Culloch Pitts neuron.

for XOR

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

This can not be represented by simle & single logic function

here $y = \underbrace{x_1 \bar{x}_2}_{z_1} + \underbrace{\bar{x}_1 x_2}_{z_2}$

$y = z_1 \text{ OR } z_2$

⇒ First function : $z_1 = x_1 \bar{x}_2$

x_1	x_2	$\bar{x}_2$	$z_1 = x_1 \bar{x}_2$ (AND-NOT)
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

Case 1: Assume $\omega_1 = 1$, $\omega_2 = 1$

for inputs, y_{z_1}

(0,0), $y_{z_1} = 0 \times 1 + 0 \times 1 = 0$

~~(0,1), $y_{z_1} = 0 \times 1 + 1 \times 1 = 1$~~

(1,0), $y_{z_1} = 1 \times 1 + 0 \times 1 = 1$

~~(1,1), $y_{z_1} = 1 \times 1 + 1 \times 1 = 2$~~

→ It is not possible to obtain function z_1 using these weights.

Case II :- take $\omega_1 = 1$, $\omega_2 = -1$

for inputs, υ_{z1}

$(0,0)$ $\upsilon_{z1} = 0 \times 1 + 0 \times (-1) = 0$

$(0,1)$ $\upsilon_{z1} = 0 \times 1 + 1 \times (-1) = -1$

$\boxed{(1,0) \quad \upsilon_{z1} = 1 \times 1 + 0 \times (-1) = 1}$

$(1,1)$ $\upsilon_{z1} = 1 \times 1 + 1(-1) = 0$

$\theta \geq nw - p$

$\theta \geq 2(1) - (1)$

$\boxed{\theta \geq 1}$ for z_1 neuron.

Activation function

$\upsilon \geq 1 \Rightarrow y = 1$

$\upsilon < 1 \Rightarrow y = 0$

classmate
Date
Page

$\Rightarrow$ Second function : $z_2 = \bar{x}_1 x_2$

x_1	$\bar{x}_1$	x_2	$z_2 = \bar{x}_1 x_2$ (NOT-AND)
0	1	0	0
0	1	1	$\boxed{1}$
1	0	0	0
1	0	1	0

Case I take $w_1 = -1$, $w_2 = 1$

for inputs, y_{z_2}

(0,0), $y_{z_2} = -1 \times 0 + 1 \times 0 = 0$

$\boxed{(0,1) \quad y_{z_2} = -1 \times 0 + 1 \times 1 = 1}$

(1,0) $y_{z_2} = -1 \times 1 + 1 \times 0 = -1$

(1,1) $y_{z_2} = -1 \times 1 + 1 \times 1 = 0$

$\theta \geq nw - p$

$\theta \geq 2(1) - 1$

$\boxed{\theta \geq 1}$

Activation function :-

$$\boxed{\begin{aligned} y &\geq 1 \Rightarrow Y = 1 \\ y &< 1 \Rightarrow Y = 0 \end{aligned}}$$

⇒ function 3 = $Y = Z_1 \text{ OR } Z_2$

Z_1	Z_2	Y
0	0	0
0	1	1
1	0	1
0	0	0

Case I :- take '$w_1 = 1$ $w_2 = 1$'

for inputs, v

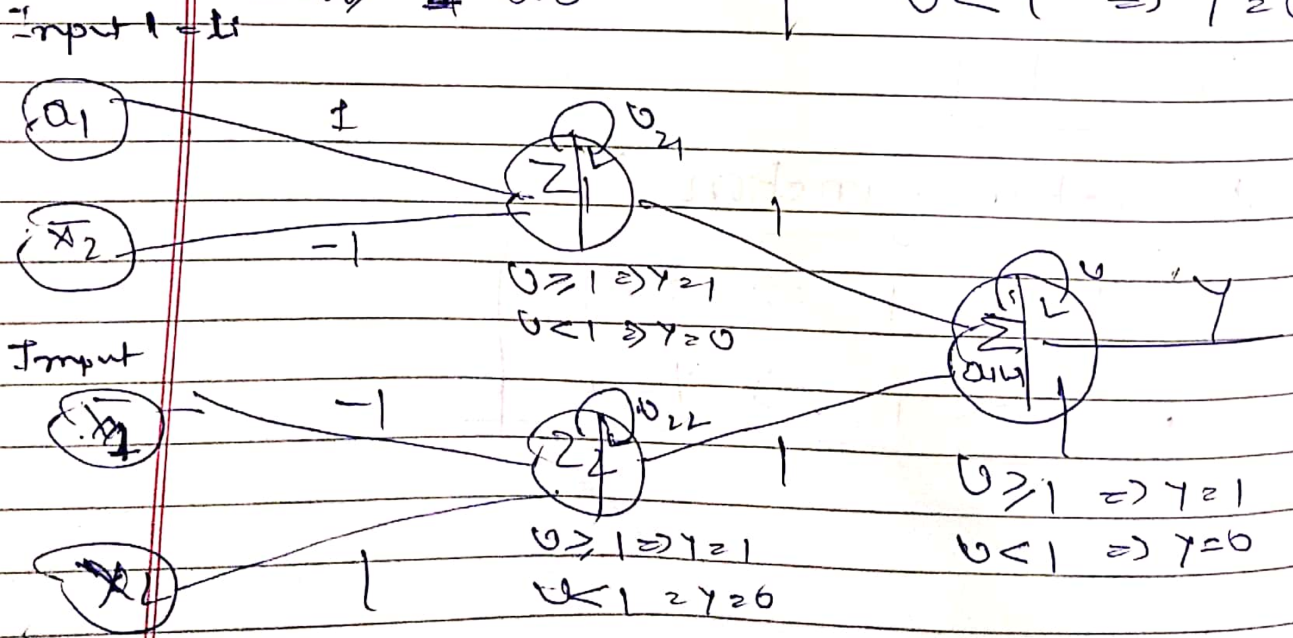
(0,0) $v = 0 \times 1 + 0 \times 1 = 0$

$$\boxed{\begin{array}{ll} (0,1) & v = 0\times 1 + 1 \times 1 = 1 \\ (1,0) & v = 1 \times 1 + 0 \times 1 = 1 \end{array}}$$

$\theta \geq n w - p$ 2(1) − 0

$v \geq 1 \Rightarrow y = 1$

$v < 1 \Rightarrow y = 0$



Linear Separability.

→ ANN does not give exact solution for a non linear problem. However it provides possible approximate solutions to a non linear problem.

→ Linear separability is a concept where in the separation of i/p space into regions is based on whether the n/w response is positive or ~~not~~ negative.

→ A decision line is drawn to separate positive & negative responses. also known as linear separable line or decision support line or decision making line.

→ We need to classify the pattern based on o/p responses.

→ The net i/p for n/w is calculated by

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

→ The decision boundary can be derived by

$$b + \sum_{i=1}^{n} x_i w_i = 0$$

Suppose for two i/ps,

Decision boundary is

$w_1 x_1 + w_2 x_2 + b = 0$

So, $x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2}$

if $w_2 \neq 0$

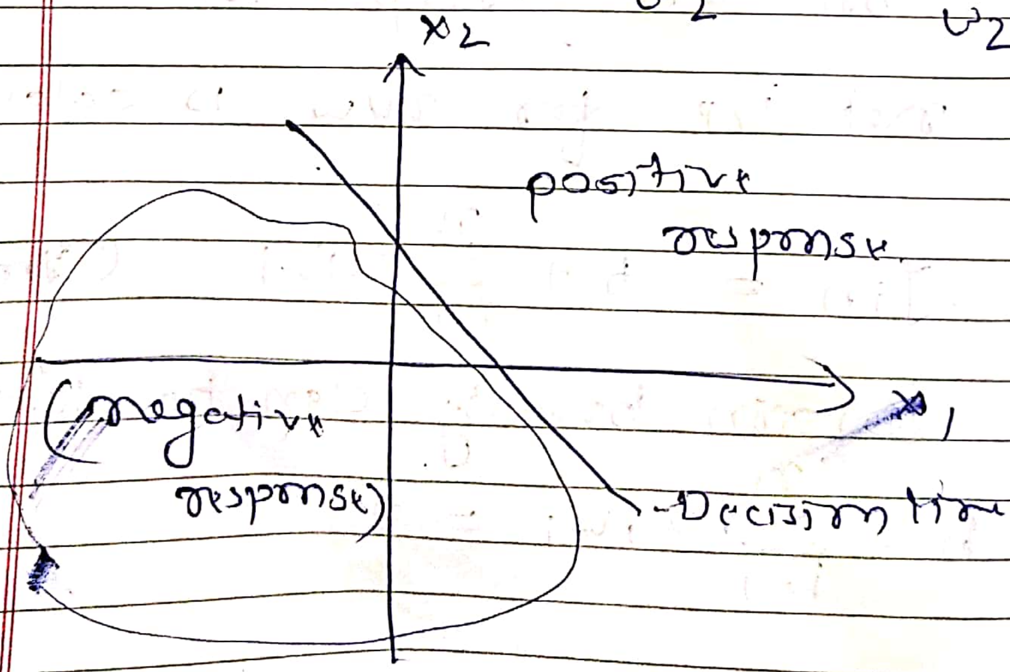
→ If Threshold is used,

net i/p > θ (threshold)

then the separating line eqn can be

$w_1 x_1 + w_2 x_2 = \theta$

$x_2 = -\frac{w_1}{w_2} x_1 + \frac{\theta}{w_2}$



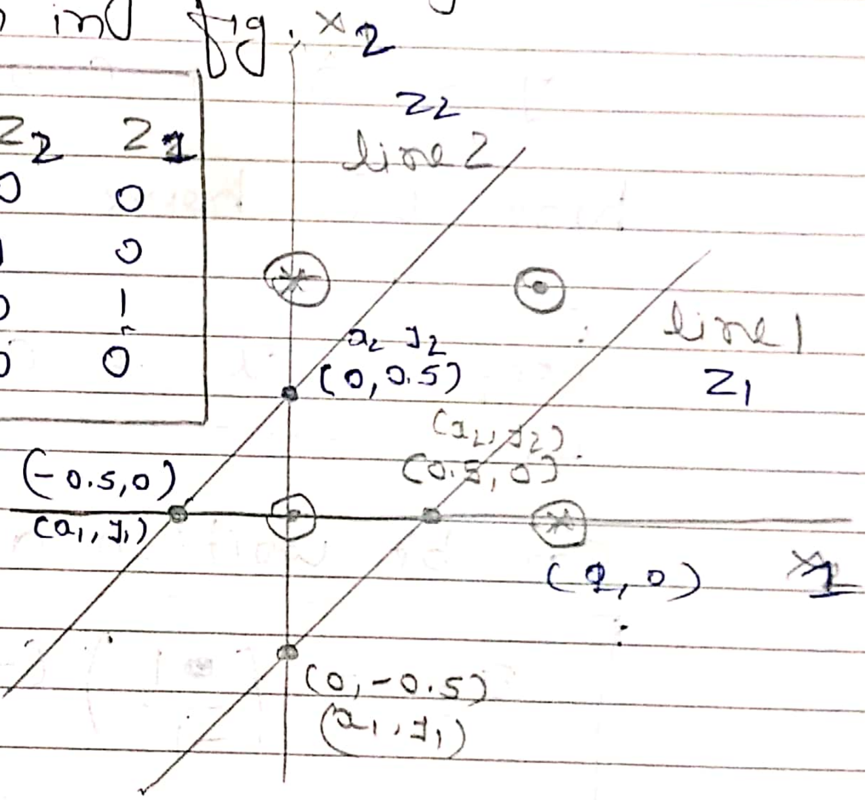
Scanned with CamScanner

$x_1 \bar{x_2} + \bar{x_1} x_2$

Ex:- Using the linear separability concept obtain the response for XOR function.

Note:- XOR is non linear or linearly non separable problem as it's patterns can not be separated by a single line as shown in fig.

x_1	x_2	y	z_2	z_1
0	0	0	0	0
0	1	1	1	0
1	0	1	0	1
1	1	0	0	0



⊙ ⇒ $y = 0$

⊛ ⇒ $y = 1$

⇒ for line 1, let's say o/p is z_1

for inputs x_1 & x_2

we have $y = mx + c$. —— ①

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{0 - (-0.5)}{0.5 - 0} = 1$$

$\boxed{m = 1}$

$C = y_1 - m x_1$

$= -0.5 - ((1) \times 0)$

$$\boxed{C = -0.5}$$

→ put values of m & c in eqn ①

$y = x + (-0.5)$

here we have $y = x_2$ & $x = x_1$

$$\boxed{x_2 = x_1 + (-0.5)}$$

can be written as,

$$\boxed{x_2 = \left(\frac{1}{-1}\right)(-x_1) - \frac{(-0.5)}{(-1)}}$$

└── ②

→ equation for decision boundary is.

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

└── ③

Compare (2) & (3)

$\boxed{w_1 = 1}$ $\boxed{w_2 = -1}$ $\boxed{b = -0.5}$

⇒ for i/p x_1 & x_2, net input v_{z_1} & output z_1 can be calculated as follows.

here we have,

Activation function

$$v_{z_1} > 0 \Rightarrow z_1 = 1$$

$$v_{z_1} \le 0 \Rightarrow z_1 = 0$$

$(0,0)$, $v_{z_1} = 0 \times 1 + 0 \times (-1) - 0.5 = -0.5$

$\Rightarrow z_1 = 0$

$(0,1)$, $v_{z_1} = 0 \times 1 + 1 \times (-1) - 0.5 = -1.5$

$\Rightarrow z_1 = 0$

$(1,0)$, $v_{z_1} = 1 \times 1 + 0 \times (-1) - 0.5 = 0.5$

$\Rightarrow z_1 = 1$

$(1,1)$, $v_{z_1} = 1 \times 1 + 1 \times (-1) - 0.5 = -0.5$

$\Rightarrow z_1 = 0$

So,

x_1	x_2	z_1
0	0	0
0	1	0
1	0	1
1	1	0

Date
Page

$\Rightarrow$ for line 2 let's say o/p is Z_2

$$m = \frac{0.5 - 0}{0 - (-0.5)} = 1$$

$\boxed{m = 1}$

$\boxed{c = 0.5}$

i.e. $y = x + 0.5$ — ①

i.e. $x_2 = x_1 + 0.5$
can be written as.

$$x_2 = \frac{(-1)}{1}(-x_1) - \frac{(-0.5)}{1}$$ — ②

Decision boundary eqn is.

$$x_2 = \frac{w_1}{w_2}(-x_1) - \frac{b}{w_2}$$

$\boxed{w_1 = -1}$, $\boxed{w_2 = 1}$, $\boxed{b = -0.5}$

Activation funn.

$v_{Z_2} > 0 \Rightarrow Z_2 = 1$

$v_{Z_2} \le 0 \Rightarrow Z_2 = 0$

for i/ps

$(0,0)$, $v_{z_2} = 0 \times (-1) + 0 \times 1 - 0.5 = -0.5$.
$\Rightarrow z_2 = 0$

$(0,1)$, $v_{z_2} = 0 \times (-1) + 1 \times 1 - 0.5 = 0.5$
$\Rightarrow z_2 = 1$

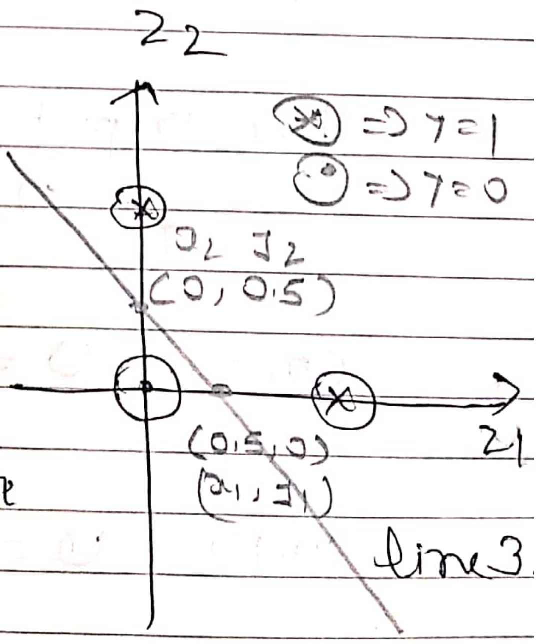
$(1,0)$, $v_{z_2} = 1 \times (-1) + 1 \times 0 - 0.5 = -1.5$
$\Rightarrow z_2 = 0$

$(1,1)$, $v_{z_2} = 1 \times (-1) + 1 \times 1 - 0.5 = -0.5$
$\Rightarrow z_2 = 0$

x_1	x_2	z_2
0	0	0
0	1	1
1	0	0
1	1	0

$\Rightarrow$

x_1	x_2	z_1	z_2	y
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0



for line 3, inputs are z_1 & z_2 & o/p is y

$\boxed{m = -1}$

$c = 0 - (-1)(0.5)$
$= 0.5$

$\boxed{c = 0.5}$

$y = -x + 0.5$

i.e. $x_2 = -x_1 - (-0.5)$ — (1)

decision boundry eqn.

$$x_2 = \frac{w_1}{w_2}(-x_1) - \frac{b}{w_2}$$ — (2)

Compare (1) & (2)

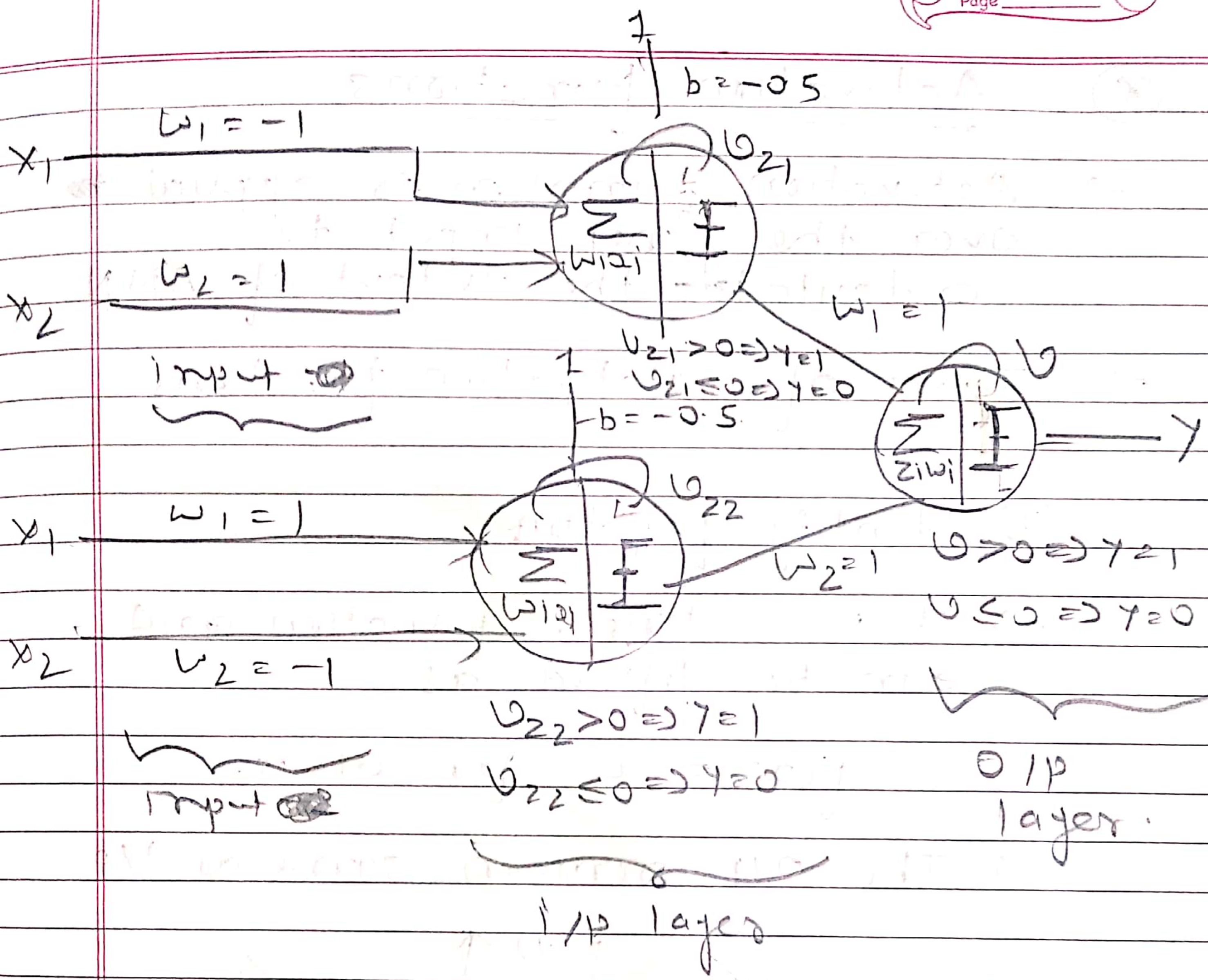
$\boxed{w_1 = 1}$ $\boxed{w_2 = 1}$ $\boxed{b = -0.5}$

for input z_1, z_2, net input v & o/p y is calculated as

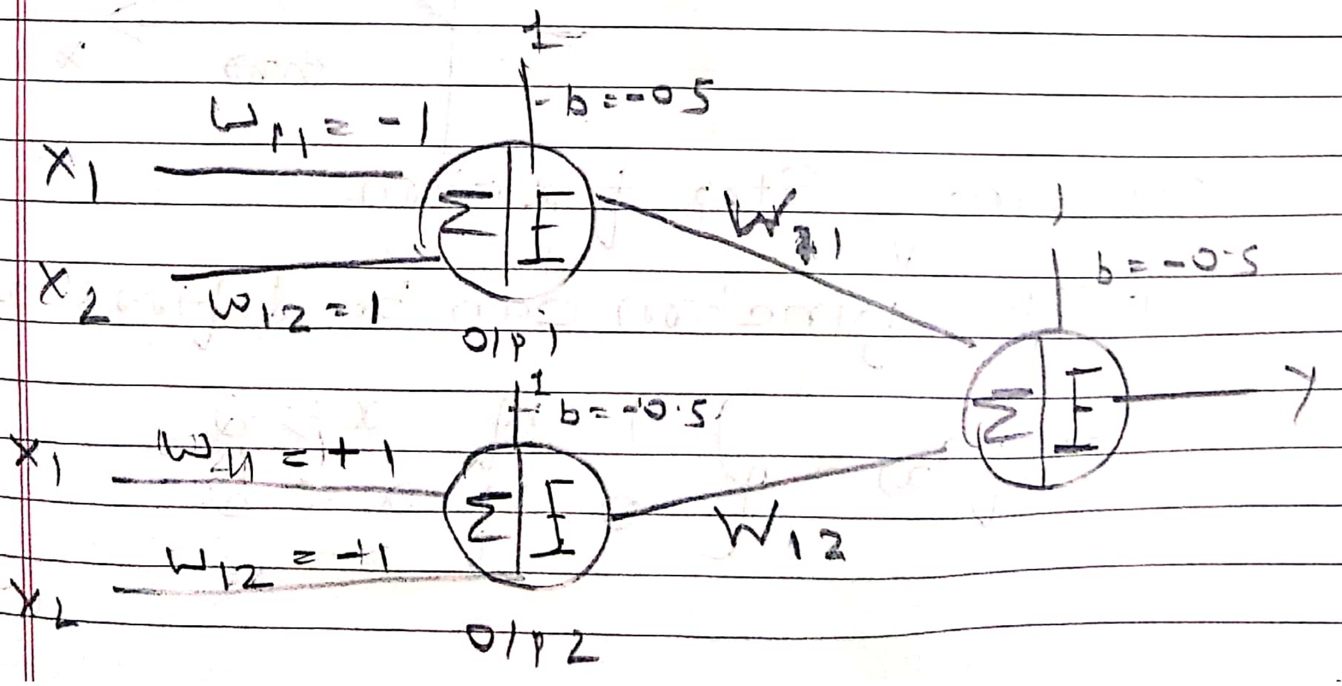
$(0,0)$ $v = 0 \times 1 + 0 \times 1 - 0.5 = -0.5$
$\Rightarrow y = 0$

$(0,1)$ $v = 0 \times 1 + 1 \times (+1) - 0.5 = 0.5$
$\Rightarrow y = 1$

$(1,0)$ $v = 1 \times 1 + 0 \times (1) - 0.5 = 0.5$
$\Rightarrow y = 1$



Weights can be represented as w_{oi}



Activation functions

→ Activation function is applied over the net input to calculate the output of ANN

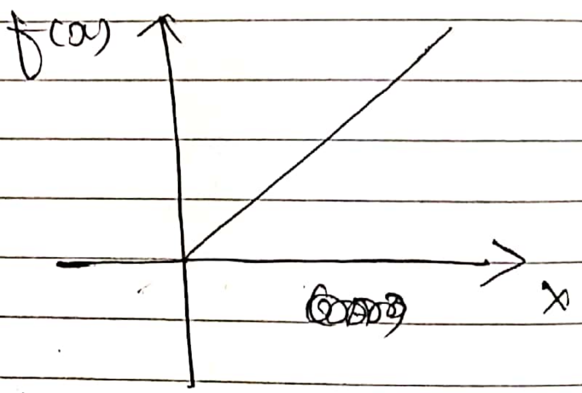
→ Types of Activation functions.

① Identity function

→ It is a linear function and can be defined as

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

→ The O/P remain same as I/P



② Binary Step function

→ This function can be defined as

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

where θ = Threshold

- Widely used in single-layer nets to convert net i/p to o/p. i.e. (1 or 0) binary.

(3) Bipolar Step function

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

where θ = Threshold.

- used in single layer nets to convert net input to output i.e bipolar (+1 or -1)

(4) Sigmoidal functions

- Widely used in back propogation nets because the relation between the value of functions at a point and the value of the derivative at that point

- which reduces computational burden during training.

- Sigmoidal function have 2 types
  (A) Binary Sigmoid
  (B) Bipolar Sigmoid.

(A) Binary Sigmoid function

- known as 'logistic sigmoid or unipolar
- can be defined as

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

λ = constant.

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

range is 0 to 1

(B) Bipolar sigmoid function

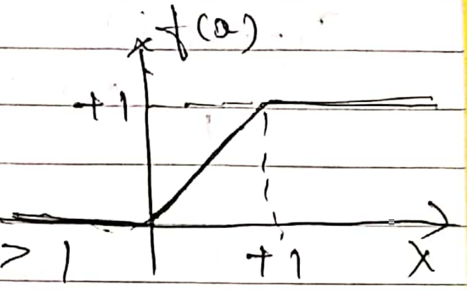
$$f(x) = \frac{2}{1 + e^{-\lambda x}} - 1$$

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

range is -1 to $+1$

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

(5) Ramp function



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

⇒ graphical representation

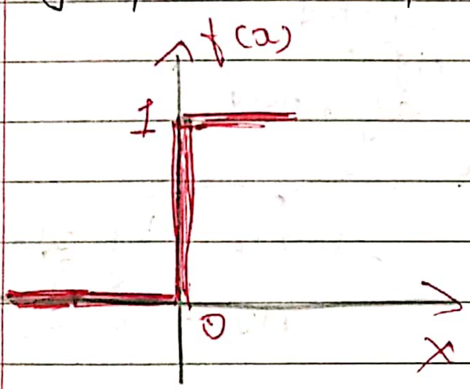


fig binary step funⁿ

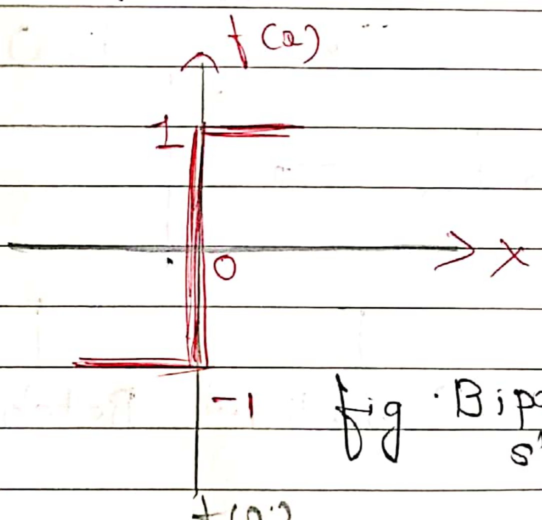


fig Bipolar step fun

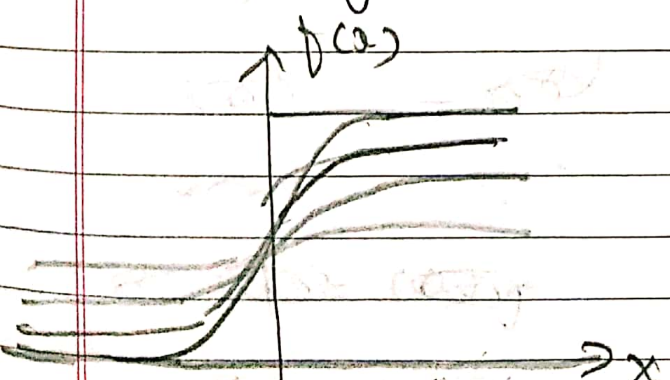


fig. binary sigmoid funⁿ

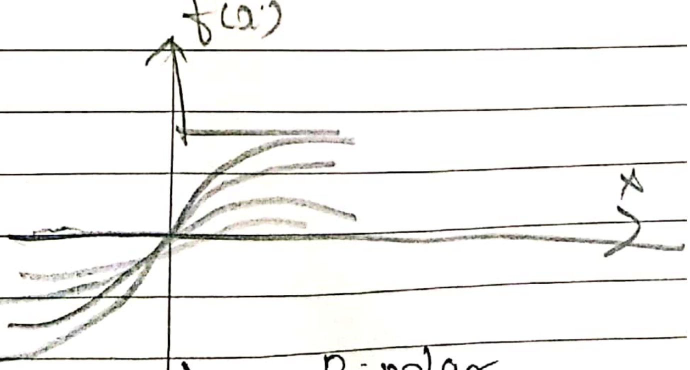


fig Bipolar sigmoid fun

(7) Learning Techniques (Process) in Neural Network

- Supervised Learning
  - Stochastic
  - Gradient Descent (Error correction)
    - Sequential
    - Batch mode

  Teacher + O/P

- Reinforced learning (teacher only checks O/P is right or wrong)

- Unsupervised Learning
  - Hebb rule
    - Based on correlated weight adjustment
  - Competitive Learning (SOM)

  learn by own
  no teacher

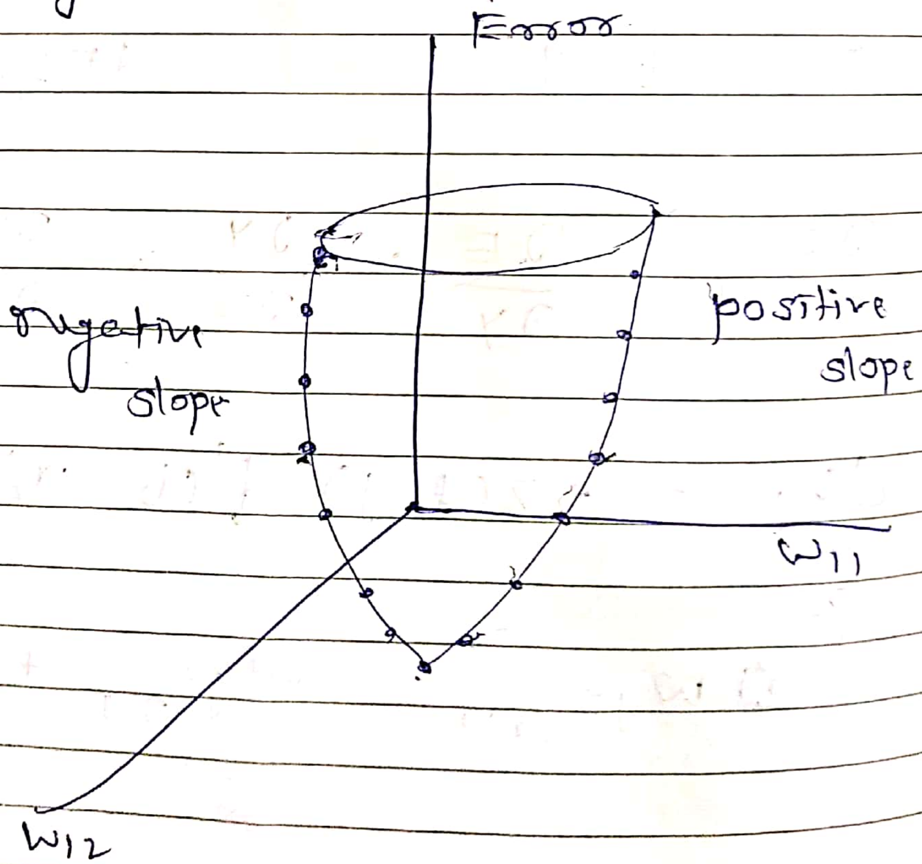
⊛ Gradient Descent Method.
(Steepest Descent)

→ A gradient measures how much an O/p function changes if you change in an I/p

→ Gradient is a partial derivative w.r.t its learning

→ Higher the gradient, steeper the slope & faster the model.

→ But if slope is 0, a model stops learning



Widro-hoff rule

(Delta rule)

here

$$\Delta w_{ji} = \alpha (t-y) x_i$$

OR

where α is a learning ~~range~~ rate

$\Rightarrow$ α (learning rate) ranges from $\boxed{0 \text{ to } 1}$

Note:- If initial weights are taken as 0.

that is $y_{in} = 0 \Rightarrow y = 0$

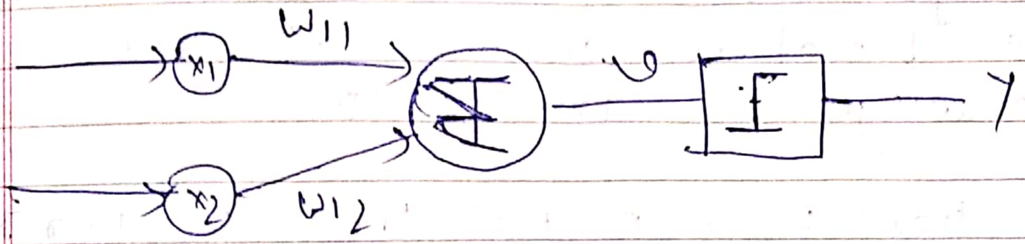
$$\Delta w_{ji} = \alpha (t - y) x_{ji}$$

$$\boxed{\Delta w_{ji} = \alpha t x_{ji}}$$

for single output $j = 1$

$$\boxed{\Delta w_i = \alpha t x_i}$$

(*) Finding the equation of Δw_{ji} (Error) in neural net

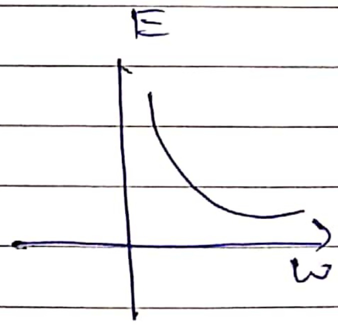


$$u = \sum_{1}^{n} W x_i w_{ji} \qquad \frac{\partial u}{\partial w_{ji}} = x_i$$

$$y = f(u) \qquad \partial y / \partial u = f'(u)$$

$$E = (t - y)^2 \qquad dE / \partial y = -2(t - y)$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial u} \times \frac{\partial u}{\partial w_{ji}}$$



$$\Delta w_{ji} = -2(t - y)\, f'(u)\, x_i$$

So

$$w_{ji\,new} = w_{ji\,OLD} + \Delta w_{ji}$$

Date
Page

(x) Learning Process in Biological Neuron
OR
learning in BNN
OR (Perceptron learning rules) =
Hebb learning / Hebb rule.
(1949)

→ "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 of the cells, such that A's efficiency, as one of the cell firing B, is increased."

→ According to hebb rule

The weight vector is found to increase proportionally to the product of the input & learning signal.

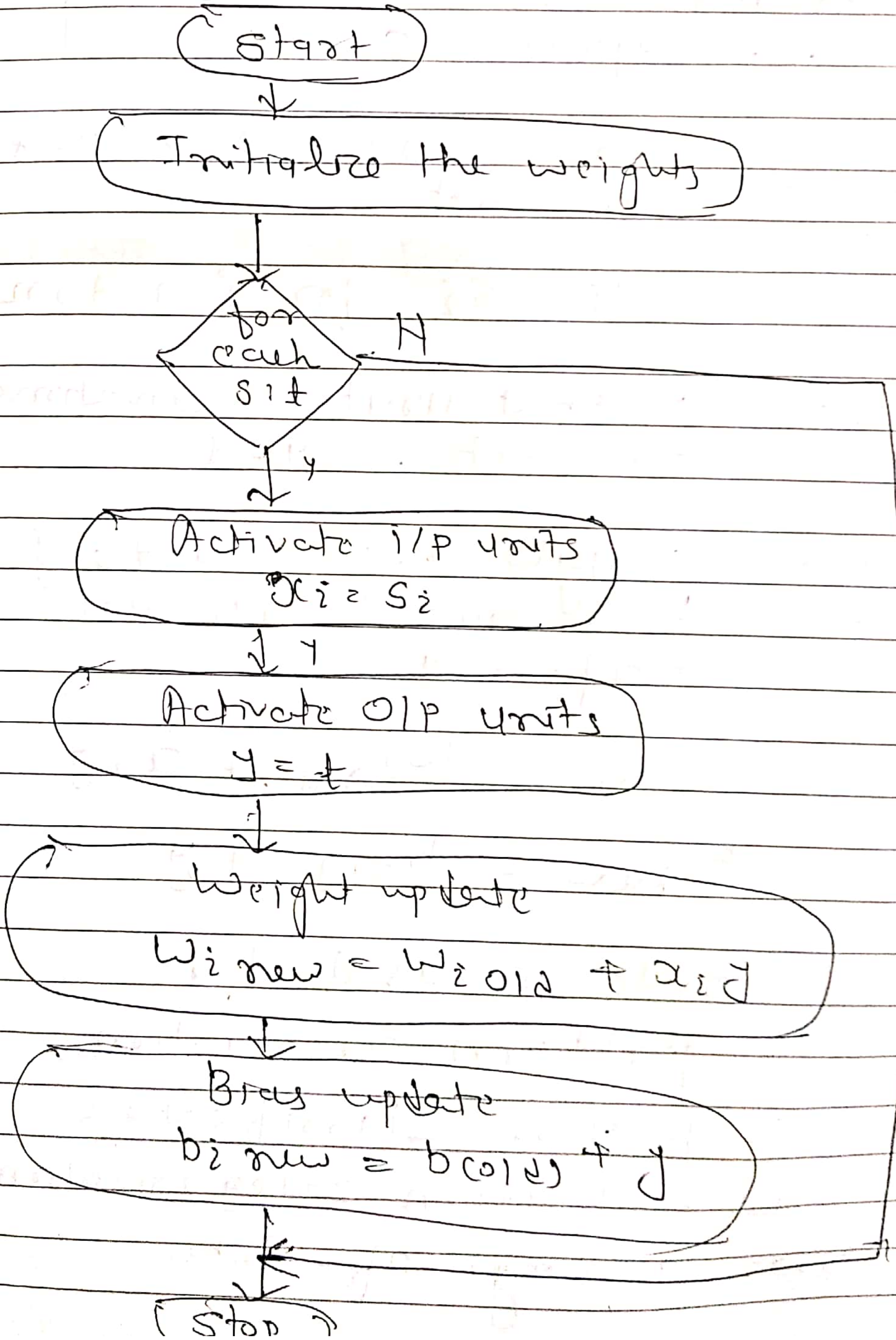
→ Here the learning signal is equals to neuron's o/p

→ here the weights can be increased by modification in the synaptic gap.

→ The weight update in hebb rule is given by

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

⇒ <u>Training algorithm</u> (Hebb training)



⇒ Algorithm for hebb training.

Step 0 :- Initialize the weights

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

Step 2 : Input units activations fun are set

$x_i = s_i$ for $i = 1$ to n

Step 3 Output units activations are set, $y = t$

Step 4 Weight adjustments & bias adjustments are performed.

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

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

→ Hebb rule is used for
- pattern association
- pattern classification
- & pattern categorization

over a range of data.

	I				O	
Ex:	*	*	*	*	*	*
	O	*	O	*	O	*
	*	*	*	*	*	*

* = 1
O = 0

	I/P									O/P
Pattern	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y
I	1	1	1	-1	1	-1	1	1	1	1
O	1	1	1	1	-1	1	1	1	1	-1

→ Initialize all weights w_1 to $w_9 = 0$ & $b = 0$

① for input pattern I change in w

$$\Delta w_i = x_i y$$

So,

$\Delta w_1 = 1 \times 1 = 1$

$\Delta w_2 = 1 \times 1 = 1$

$\Delta w_3 = 1 \times 1 = 1$

$\Delta w_4 = -1 \times 1 = (-1)$

$\Delta w_5 = 1 \times 1 = 1$

$\Delta w_6 = -1 \times 1 = (-1)$

$\Delta w_7 = 1 \times 1 = 1$

$\Delta w_8 = 1 \times 1 = 1$

$\Delta w_9 = 1 \times 1 = 1$

$\Delta b = y = 1$

→ Calculate new weights.

$$w_{i\,new} = w_{i\,old} + \Delta w_i$$

w_1 new $= 0 + 1 = 1$

w_2 new $= 1$

w_3 new $= 1$

w_4 new $= (-1)$

w_5 new $= +1$

w_6 new $= (-1)$

w_7 new $= 1$

w_8 new $= 1$

w_9 new $= 1$

B (new) $= 1$

So
$W_{new} = [1\ 1\ 1\ -1\ 1\ -1\ 1\ 1\ 1\ 1]$

⇒ (2) for Second pattern O, weight change is

$$\Delta w_i = x_i y$$

$y = -1$ $\quad W_{new} = [1\ 1\ 1\ -1\ 1\ -1\ 1\ 1\ 1\ 1]$

w_i new $= w_{i\,old} + \Delta w_i$

w_1 new $= 1 + 1 \times (-1) = 0$

w_2 new $= 1 + 1 \times (-1) = 0$

w_3 new $= 1 + 1 \times (-1) = 0$

$w_{4\,new} = -1 + 1 \times (-1) = -2$

$w_{5\,new} = 1 + (-1) \times (-1) = 2$

$w_{6\,new} = -1 + 1 \times (-1) = -2$

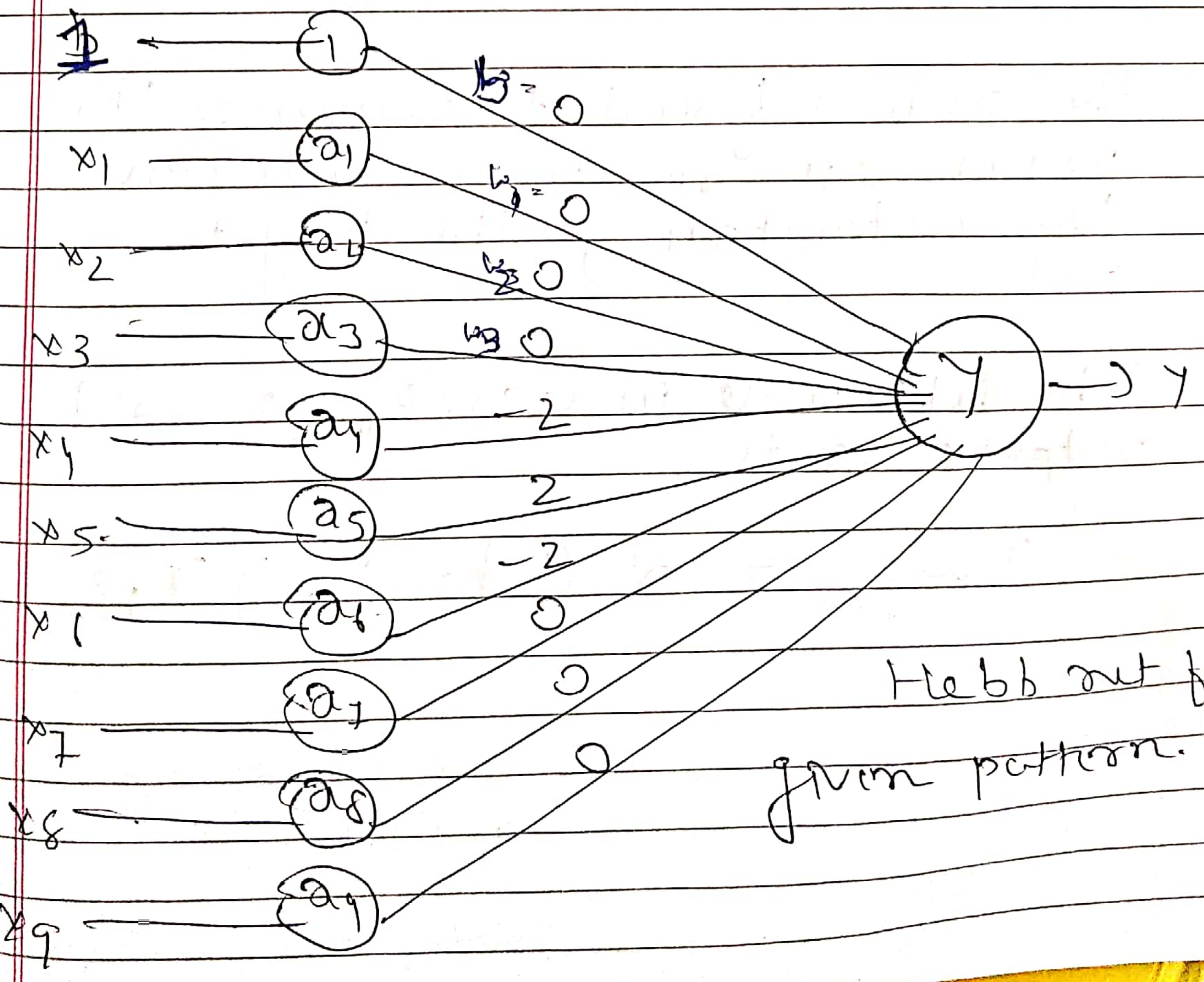
$w_{7\,new} = 1 + 1(-1) = 0$

$w_{8\,new} = 1 + 1(-1) = 0$

$w_{9\,new} = 1 + 1(-1) = 0$

$b_{new} = 1 + 1 \times (-1) = 0$

$W_{new} = [0\ 0\ 0\ -2\ 2\ -2\ 0\ 0\ 0\ 0]$



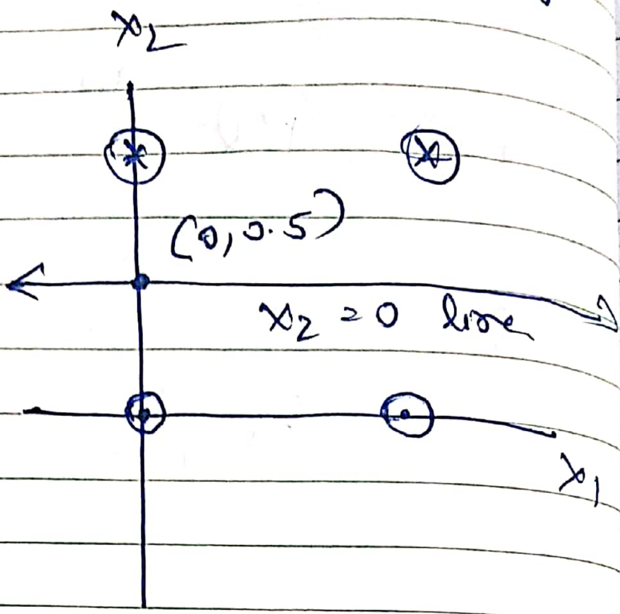
Hebb net for given pattern.

Ex:- Find weight & bias for given function using Linear Separability concept.

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

$y = 0 \Rightarrow \odot$

$y = 1 \Rightarrow \otimes$



$y \times 1 + a \times 0 - 0.5 = 0$

x_1 x_2

here $m = 0$

$c = y_1 - m x_1$

$= 0.5 - 0$

$= 0.5$

Decision boundary Line is

$w_1 x_1 + w_2 x_2 + b = 0$

Here line is $x_2 = 0$

So, Decision boundary Line is.

$w_1 x_1 + b = 0$

$x_1 = -b/w_1$

$b = 0.5$ & $w_1 = 1$, $w_2 = 0$

Test:-

$y(0,0) = 0 \times 1 + 0 \times 0 - 0.5 < 0$

$\Rightarrow y = 0$

Scanned with CamScanner

$v(0,1) = 0 \times 1 + 1 \times 0 - 0.5 < 0$

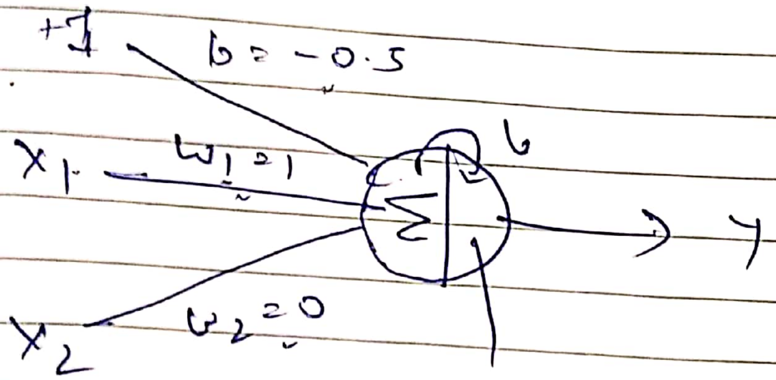
$\Rightarrow y = 0$

$v(1,0) = 1 \times 1 + 0 \times 0 - 0.5 > 0$

$\Rightarrow y = 1$

$v(1,1) = 1 \times 1 + 0 \times 1 - 0.5 > 0$

$\Rightarrow y = 1$



$v \le 0 \Rightarrow y = 0$

$v > 0 \Rightarrow y = 1$