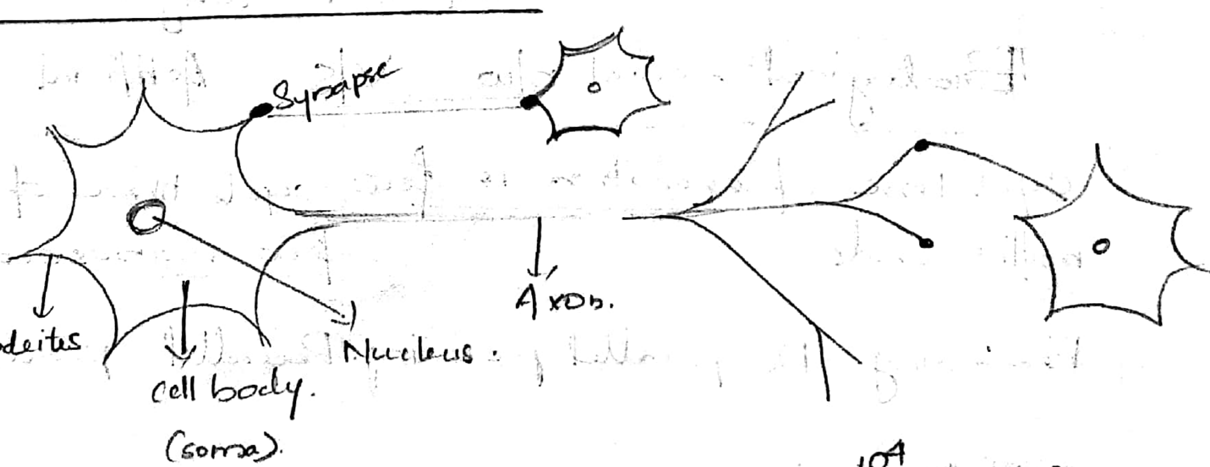
Module 1
Introduction to Soft Computing.

7/8/17

> **Hard Computing v/s Soft computing.**

Hard Computing	Soft computing
Precise systems. precise calculations	Real world problems. like handwriting recognition, speaker recognition, weather forecasting.
Binary logic	Fuzzy logic.
Two states low, high 0, 1	Multiple values 0–1
Precision	Approximation.
Requires program	Program evolve
Maybe sequential	Parallel processing.
Deterministic	Randomness.
Can't tolerate errors.	Tolerate errors.

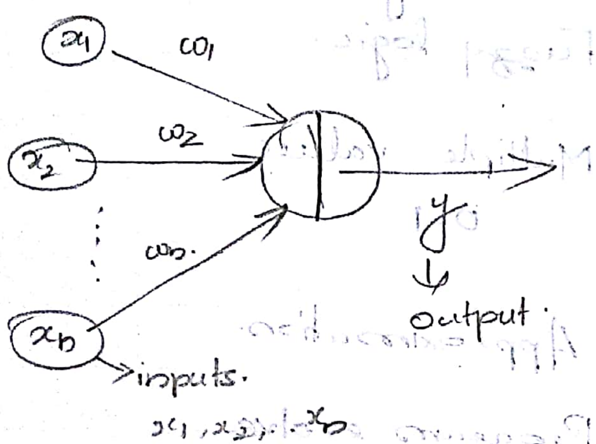
> **Biological Neural Network:**



10^4 synapses.

Biological neuron v/s Artificial neuron

Biological neuron	Artificial neuron
Cell	neuron
Dendrites	Weighted links.
Soma	Net input.
Axon	output.



Net input $y_{in} = x_1 w_1 + x_2 w_2 + \ldots + x_n w_n$.

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

where x_i is the i^{th} input

w_i is i^{th} weight.

Biological neural n/w V/S Artificial neural n/w

Biological neural n/w	Artificial neural n/w
Cycle time of execution is few milliseconds.	cycle time of execution is few nanoseconds.
Processing - Do parallel processing.	Parallel processing is faster.
Complexity is more. 10^{11} neurons 10^{15} interconnections	
Storage capacity - stores info in synapse or interconnection.	older memory is lost when overwritten

Biological Neural N/W v/s Artificial Neural N/W.

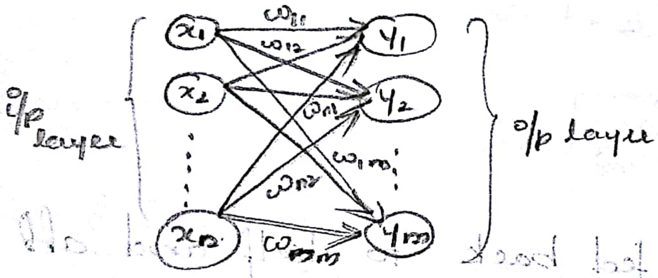
Biological Neural N/W	Artificial Neural N/W
fails to recollect memory	Once stored information is permanently stored.
More fault tolerant	Less fault tolerant
control mechanism done by chemicals.	CPU controls

8/8/17 Tue.

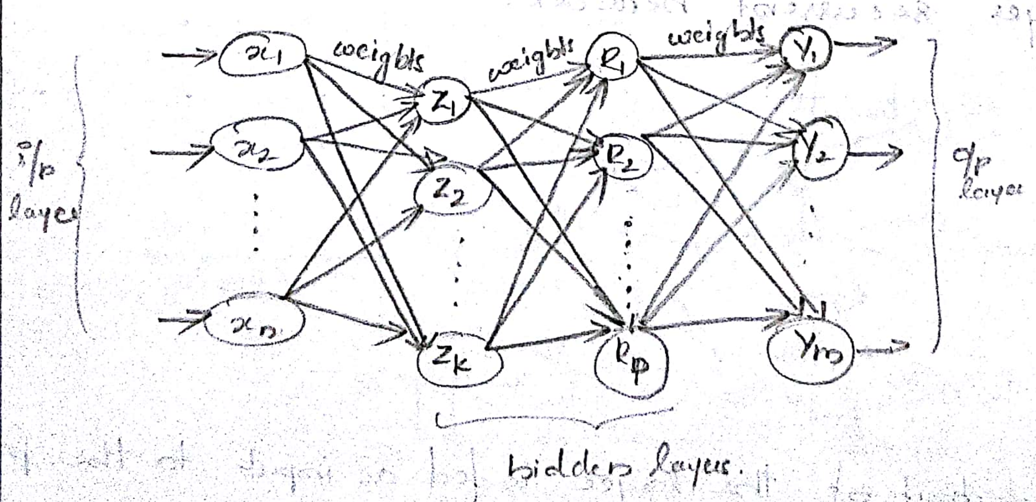
BASIC MODELS OF ARTIFICIAL NEURAL NETWORK

Interconnection:

1. Single layer feed forward network



2. Multi layer feed forward network

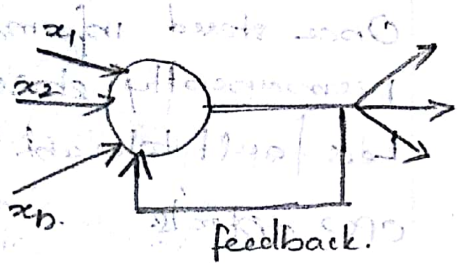


There are hidden layers present

Features are given as i/p

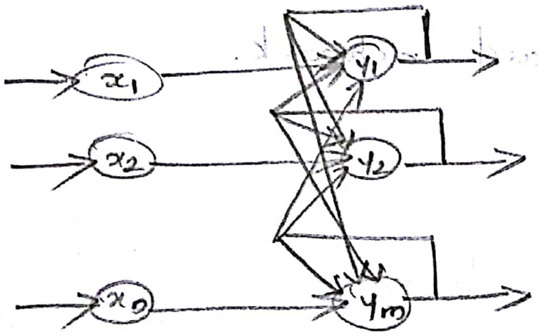
→ N/w becomes more complex when we increase the number of hidden layers.

3. Single node with own feedback



When the output is directed back as input to the same or preceding layer it will result in feedback networks.

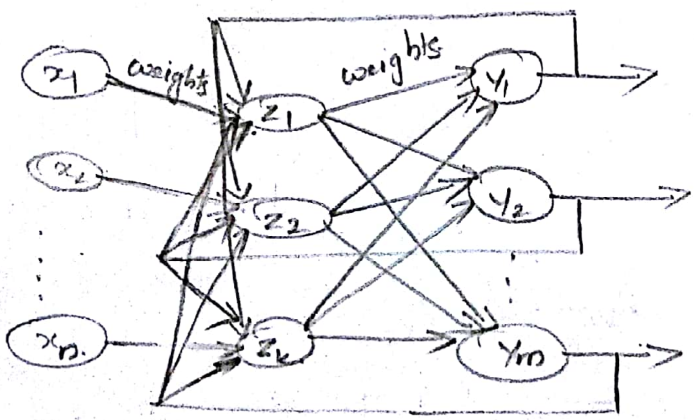
4. Single layer recurrent network.



single layer of i/p & o/p neurons

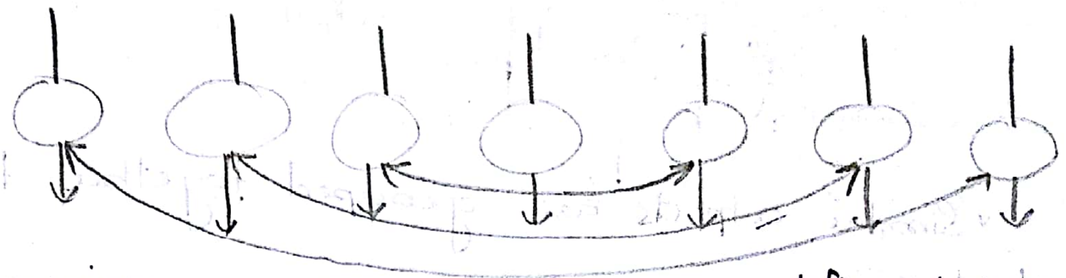
The output is fed back to itself and all other neurons.

5. Multi layer recurrent network.



The output of the layer is fed as input to the previous layer or to the same layer. There will be hidden layers in this network.

6. Lateral inhibition network



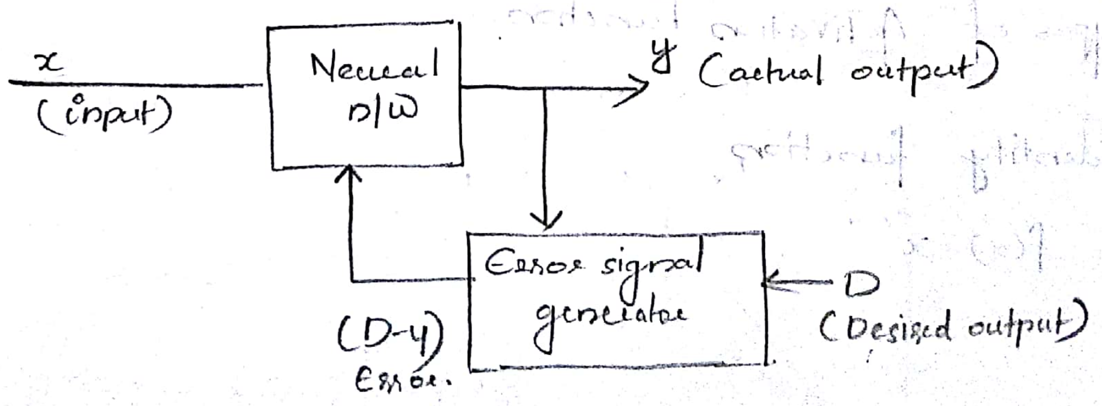
Each processing neuron receives two different inputs: excitory inputs from nearby processing neuron and inhibitory input from more distantly located processing element.

10/8/17 Thu. Learning. - 3 types.

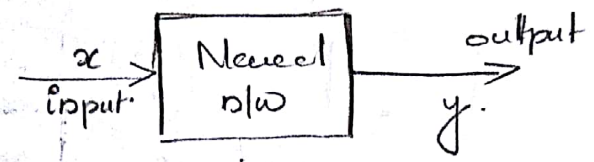
1. Supervised learning.
2. Un-supervised learning.
3. Reinforcement learning

1. Supervised learning:

- under the guidance of a supervisor
- Eg: student learns under the guidance of a teacher.
- correction is made

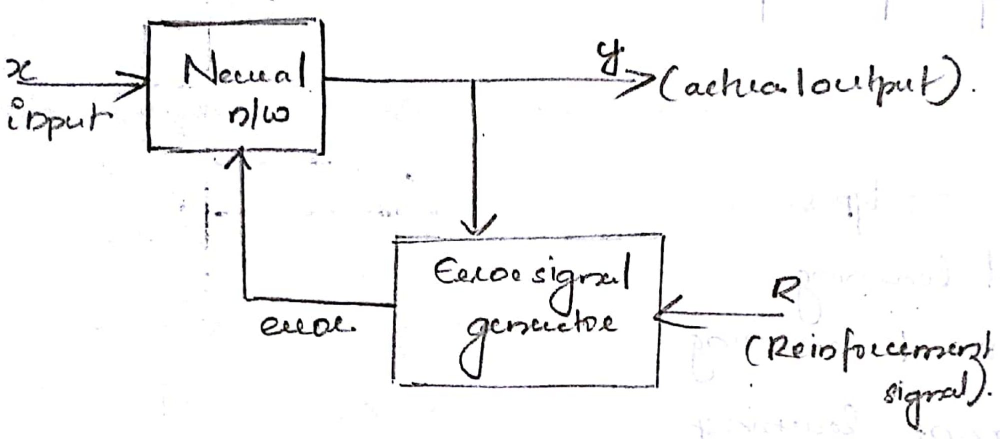


› Un-supervised learning:

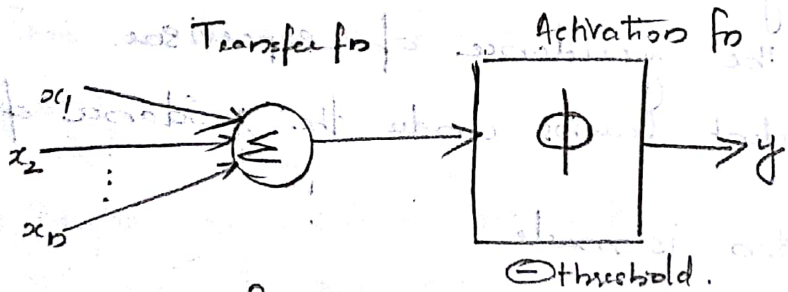


× Similar outputs are grouped together to form clusters.

› Reinforcement learning.



→ Activation Function.

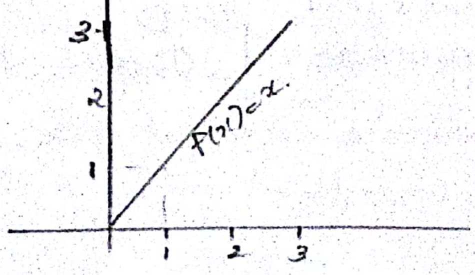


$$y_{in} = \sum_{i=1}^{n} x_i w_i$$

› Types of Activation Function:

1. Identity function.

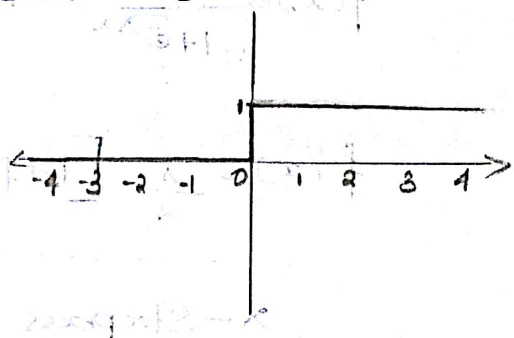
$f(x) = x$.



2. Binary step function.

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

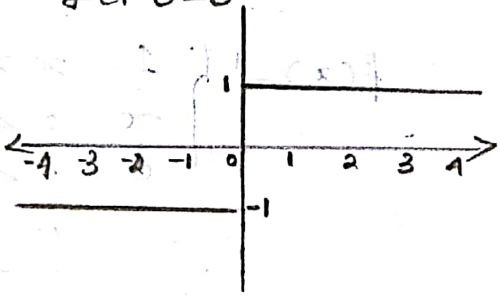
Let $\theta = 0$.



3. Bipolar Step function.

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

Let $\theta = 0$



4. Sigmoidal function.

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

where λ = steepness parameter

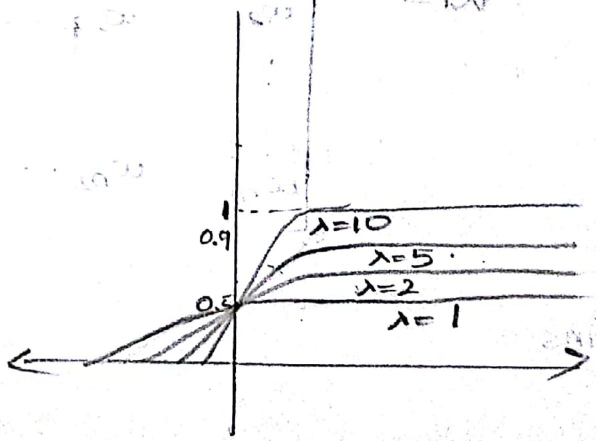
Let $\lambda = 10$.

$$f(0) = \frac{1}{1+e^{0}} = 0.5$$

$\lambda = 5$

$$f(1) = \frac{1}{1+e^{-10}} = 0.9$$

$$f(-1) = \frac{1}{1+e^{10}} =$$



range: 0 to 1.

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

14/8/15 5. Bipolar Sigmoidal function.

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

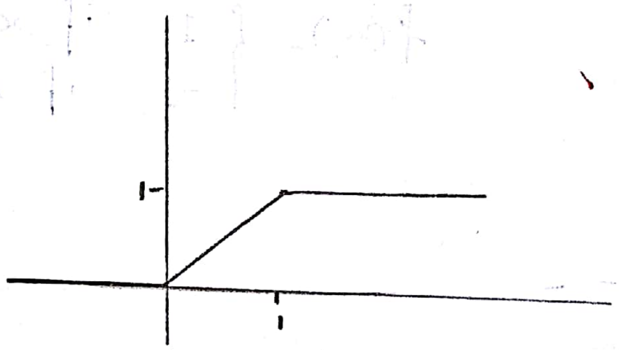
o/p
-1 to $+1$

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

λ - Steepness parameter.

6. Ramp function

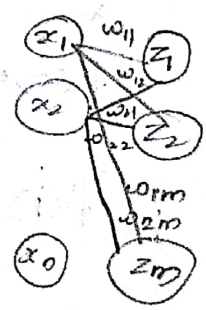
$$f(x) = \begin{cases} 1 & x > 1 \\ x & 0 \le x \le 1 \\ 0 & x < 0. \end{cases}$$



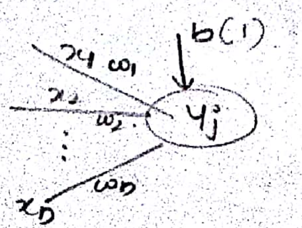
→ Terminologies in ANN:

1. Weight matrix / Connection matrix :

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & & & \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} = \begin{bmatrix} w_1 T \\ w_2 T \\ \vdots \\ w_n T \end{bmatrix}$$



2. BIAS.



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

3. Threshold.

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

$\theta \rightarrow$ Threshold. (output depends on threshold value).

4. Learning rate. (α)

Control on weight adjustment.

The value of α ranges from 0 to 1.

Notations:

→ x_i - Activation of input unit i/ input signal.

→ w_{ij} = Weight in the connection from unit i in one layer to unit j in the next layer.

→ b_j = Bias acting on unit j. [Bias is usually 1]

→ W = Weight matrix.

→ y_{inj} = Net input to unit y_j.

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

→ Norm of x. $\|x\|$

1-norm $\|x\| = |x_1| + |x_2| + \cdots + |x_n|$

2-norm $\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2}$

$\vdots$

p-norm $\|x\|_p = \left(|x_1|^p + |x_2|^p + \cdots + |x_n|^p \right)^{1/p}$.

→ θ_j = Threshold for activation of neuron y_j.

→ S = Training input vector $(s_1\ s_2 \ldots s_n)$

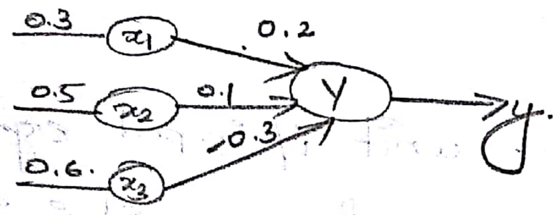
→ T = Training output vector $(t_1\ t_2 \ldots t_n)$.

→ X = Input vector $X = \{x_1\ x_2 \ldots x_n\}$ Can be training vector / testing vector

→ $\Delta w_{ij} = w_{ij}(\text{new}) - w_{ij}(\text{old})$.

→ α; learning rate.

1. Calculate the net input ~~neuron~~

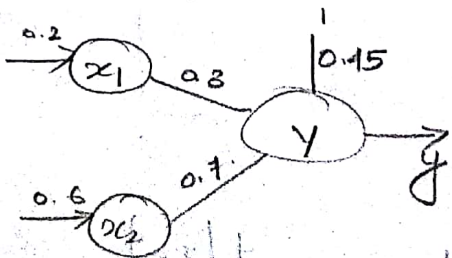


$$y_{in} = 0.3 \times 0.2 + 0.5 \times 0.1 + 0.6 \times -0.3$$
$$= 0.06 + 0.05 - 0.18$$
$$= -0.07$$

0.06
0.05
0.11
$0.18 -$
0.11
0.07

17/8/17 Thu

2. Calculate the net input to the neural n/w given to you



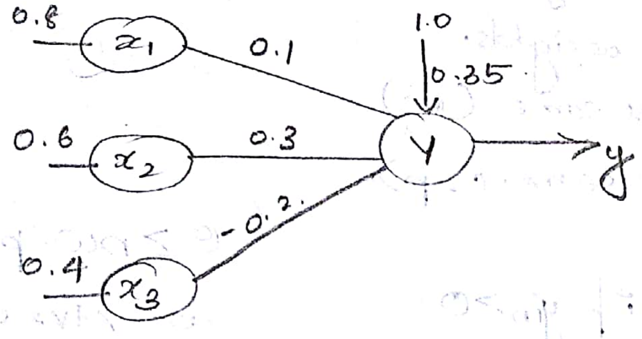
$$y_{in} = 0.2 \times 0.3 + 0.6 \times 0.7 + 0.45 \times 1$$
$$= 0.06 + 0.42 + 0.45$$
$$= 0.93$$

0.42
0.15
0.06

3. Obtain the output of the neuron Y for the nw shown in the figure using activation function as

(i) Binary sigmoidal.

(ii) Bipolar sigmoidal.



Ans: (i)

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

Let $\lambda = 1$

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

$$= 0.8 \times 0.1 + 0.6 \times 0.3 + 0.4 \times -0.2 + 1 \times 0.35$$

$$= 0.08 + 0.18 - 0.08 + 0.35$$

$$= 0.53$$

$$\begin{array}{r} 0.08 \\ 0.18 \\ 0.35 \\ \hline 0.61 - \\ 0.08 \\ \hline 0.53 \end{array}$$

$$f(x) = f(y_{in}) = \frac{1}{1+e^{-0.53}}$$

$$= 0.629$$

Ans (ii) Bipolar sigmoidal

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

$$= \frac{2}{1+e^{-0.53}} - 1$$

$$\cancel{= -0.3705} = 0.2589$$

→ McCulloch-Pitts Neuron. (M-P Neuron).

→ Binary activation [0,1] Either 0 or 1 will be output.

→ Weights can be:

* Excitatory +ve weights
* Inhibitory -ve weights.

All positive weights are same (w)

All negative weights are same. ($-p$).

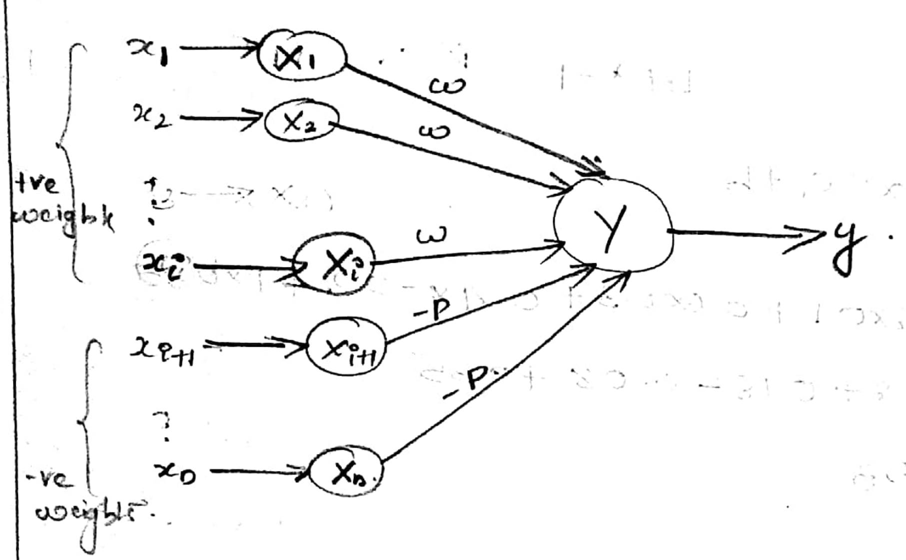
$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq \theta \\ 0 & \text{if } y_{in} < \theta \end{cases}$$

* $\theta \geq nw - p$.

$w \rightarrow$ +ve weight

$p \rightarrow$ -ve weight (only magnitude)

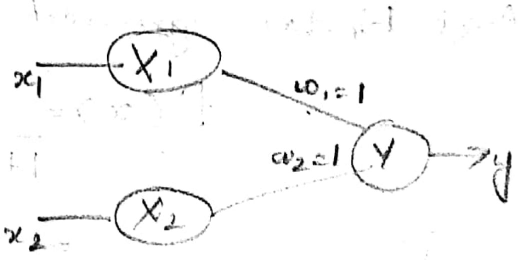
$n \rightarrow$ no of inputs



→ Implement AND function using M-P neuron.

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

2 i/ps.
1 o/p.



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

(0, 0) $y_{in} = 0 \times w_1 + 0 \times w_2$

$= 0 \times 1 + 0 \times 1$

$= \underline{\underline{0}}$

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

$= \underline{\underline{1}}$

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

$= \underline{\underline{1}}$

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

$= \underline{\underline{2}}$

$\theta \geq n w - p.$

$\theta \geq 2 \times 1 - 0$

$\underline{\underline{\theta \geq 2}}.$

$\theta =$ threshold $= 2.$

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} \geq 2 \\ 0 & y_{in} < 2. \end{cases}$$

C PROGRAM.

```
f(x)
 { return x
 }
Calculateyin(x1, w1, x2, w2)
{
  yin = x1 x w1 + x2 x w2
  return yin
}
findyin (yin)
{ if yin >= 2
        return 1
  else
        return 0 }
```

Void main.

{ $w_1 = 1$

$w_2 = 1$

read x_1, x_2

$x_1 = f(x_1)$

$x_2 = f(x_2)$

y_{in} = calculateyin (x_1, w_1, x_2, w_2).

y = findy (y_{in}).

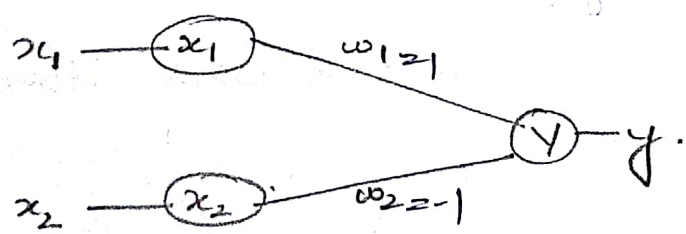
print y

}.

22/08/17 Tue

Implementing AND NOT.

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



$y_{in}(1,1)$ * $w_1 = 1$ $w_2 = 1$

$= 1\times1 + 1\times1$

$= 2$

$y_{in}(1,0)$

$= 1\times1 + 0\times1$

$= 1 + 0 = 1$.

$(0,1)$

$= 0\times1 + 1\times1$

$= 1$

(0,0)

$= 0\times1+0\times1$

$= 0.$

$w_1 = 1 \quad w_2 = -1$

(1,1)

$= 1\times1 + 1\times-1$

$= 1-1=0.$

(1,0) $= 1\times1+0\times-1$

$= 1$

(0,1) $= 0\times1 + 1\times-1$

$= -1$

(0,0) $= 0\times1 + 0\times-1$

$= 0.$

$\theta \geq nw - p.$

$\theta \geq 2\times1-1$

$\theta \geq 1$

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

$ XOR using M-P neuron

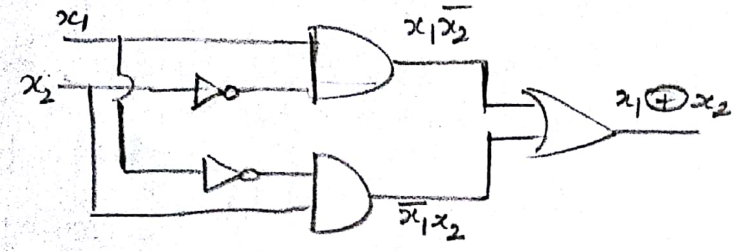
x_1	x_2	y.
1	1	0
1	0	1
0	1	1
0	0	0

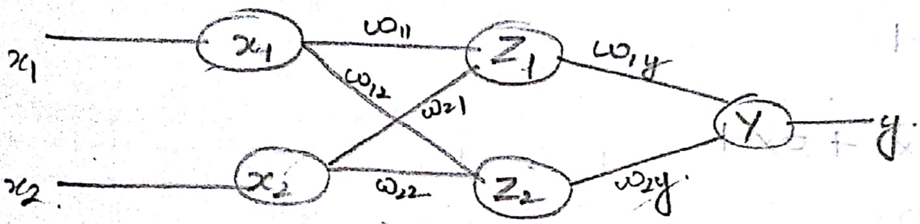
$x_1 \oplus x_2 = x_1\overline{x_2} + \overline{x_1}x_2.$

$Z_1 = x_1\overline{x_2}$

$Z_2 = \overline{x_1}x_2$

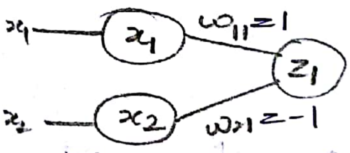
$Z_1 + Z_2$





→ $Z_1 = x_1\overline{x_2}$

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



Assume $\omega_{11} = \omega_{21} = 1.$

Net input.

$z_{1in}(0,0)$

$= 0\times1 + 0\times1$

$= 0$

$z_{1in}(0,1)$

$= 0\times1 + 1\times1$

$= 1$

$(1,0)$

$= 1\times1 + 0\times1 = 1$

$(1,1)$

$= 1\times1+1\times1$

$= 2.$

Assume $\omega_{11}=1, \omega_{21}=-1$

$z_{1in}(0,0)$

$= 0\times1+0\times-1$

$=0$

$z_{1in}(0,1)$

$=0\times1+1\times-1$

$=-1$

$z_{1in}(1,0)$

$=1\times1+0\times-1$

$=1$

$z_{1in}(1,1)$

$=1\times1+1\times-1$

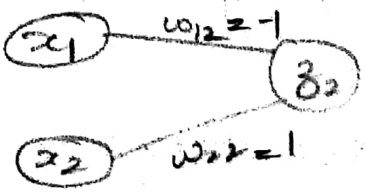
$=0.$

$\theta \geq nw-p$

$\geq 2\times1-1$

$\theta \geq 1$

→ $Z_2 = \bar{x}_1 x_2$



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

Assume ~~(1,1)~~ $\omega_{12}=1, \omega_{22}=1$

$z_{2in}(0,0)$

$= 0\times-1+0\times1$

$=0$

$z_{2in}(0,1)$

$=0\times-1+1\times1$

$=1$

$z_{2in}(1,0) = 1\times-1+0\times1$

$=-1$

$z_{2in}(1,1) = 1\times-1+1\times1$

$=0$

$\theta \geq nw - p$

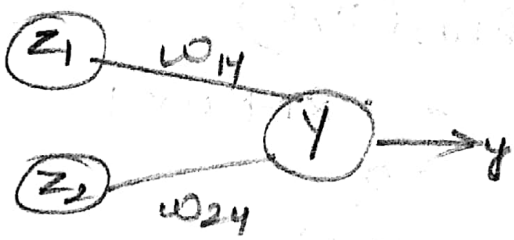
$\geq 2 \times 1 - 1$

$\theta \geq 1$

29/8/17 Tue.

$Z_1 + Z_2$.

net $Y_{in} = w_{1y} Z_1 + w_{2y} Z_2$



Z_1 Z_2 y

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

Calculate net inputs

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

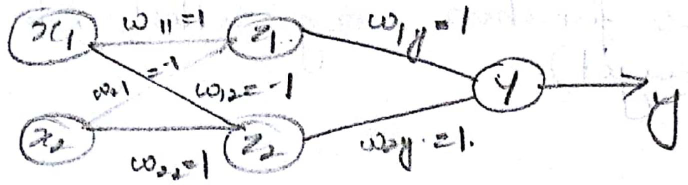
$y_{in(0,1)} = 1 \times 0 + 1 \times 1 = 1$

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

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

$\theta \geq nw - p$.

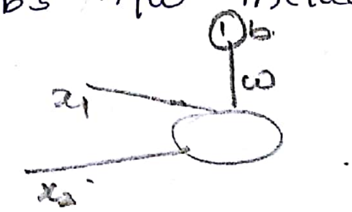
$\geq$



M-P Neuron for XOR

→ HEBB'S NETWORKS

Hebb's n/w includes bias.



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

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

~~Algorithm:~~

Training Algorithm

Step 0: Initialise base $w_i = 0$ for all $i = 1$ to n. where n is the total number of input neurons.

Step 1: Steps 2 to 4 is performed for each input training vector and target output pair (S:t)

Step 2: The input units activations are set using identity function.

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

Step 3: Output units activations are set.

ie $y = t$.

Step 4: Adjust weights

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

2) Implement AND function using a Hebb n/w (use bipolar input & target).

AND.

x_1	x_2	b	Target.
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

Training Algo:

$w_1 = w_2 = b = 0$

Take first i/p.

$[x_1 \; x_2 \; b] = [1 \; 1 \; 1]$ $\quad y=1$

Apply hebb rule.

$w_i(new) = w_i(old) + x_i y$

$w_1(new) = w_1(old) + x_1 y$

$= 0 + 1 \times 1 = \underline{\underline{1}}$

$w_2(new) = w_2(old) + x_2 y$

$= 0 + 1 \times 1 = \underline{\underline{1}}$

$b(new) = b(old) + y$

$= 0 + 1 = \underline{\underline{1}}$

Take 2^{nd} input

$[x_1 \quad x_2 \quad b] = [1 \quad -1 \quad 1]$

$y = -1$

Apply Hebb rule.

$w_1(new) = w_1(old) + x_1 y$

$= 1 + 1 \times -1 = \underline{\underline{0}}$

$w_2(new) = w_2(old) + x_2 y$.

$= 1 + -1 \times -1 = \underline{\underline{2}}$

$b(new) = b(old) + y$

$= 1 + -1 = \underline{\underline{0}}$.

Take 3^{rd} input

$[x_1 \quad x_2 \quad b] = [-1 \quad 1 \quad 1]$

$y = -1$

Apply Hebb Rule:

$w_1(new) = w_1(old) + x_1 y$

$= 0 + -1 \times -1 = \underline{\underline{1}}$

$w_2(new) = 2 + 1 \times -1 = \underline{\underline{1}}$

$b(new) = 0 + -1 = \underline{\underline{-1}}$

Take 4^{th} input

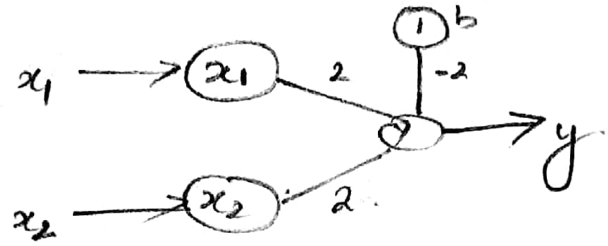
$[x_1 \quad x_2 \quad b] = [-1 \quad -1 \quad 1]$

$y = -1$

$w_1(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

$w_2(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

$b(new) = -1 + -1 = \underline{\underline{-2}}$

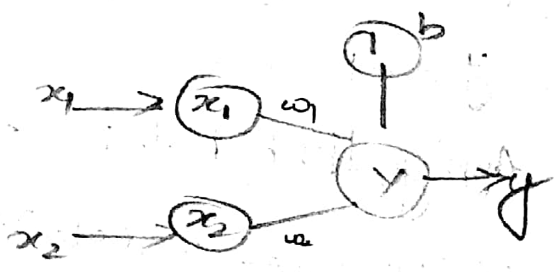


x_1	x_2	y
1	1	2 → +ve excitation
1	-1	-2
-1	1	-2
-1	-1	-6

(rows 2–4: inhibitions)

b.w 2) Design a hebb n/w to implement OR function (Use bipolar input & target).

x_1	x_2	b	Target
1	1	1	1
1	-1	1	1
-1	1	1	1
-1	-1	1	-1



$w_1 = w_2 = b = 0$

* First i/p

$[x_1 \ x_2 \ b] = [1 \ 1 \ 1]$ $\quad y = 1$

Apply hebb rule.

$w_1(\text{new}) = 0 + 1 \times 1 = \underline{1}$

$w_2(\text{new}) = 0 + 1 \times 1 = \underline{1}$

$b(\text{new}) = 0 + 1 = \underline{1}$

Scanned by CamScanner

* Second input

$[x_1 \quad x_2 \quad b] = [1 \quad -1 \quad 1] \quad y = 1$

$w_1(new) = 1 + 1 \times 1 = \underline{\underline{2}}$

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

$b(new) = 1 + 1 = \underline{\underline{2}}$

* Third input

$[x_1 \quad x_2 \quad b] = [-1 \quad 1 \quad 1] \quad y = 1$

$w_1(new) = 2 + -1 \times 1 = \underline{\underline{1}}$

$w_2(new) = 0 + 1 \times 1 = \underline{\underline{1}}$

$b(new) = 2 + 1 \times 1 = \underline{\underline{3}}$

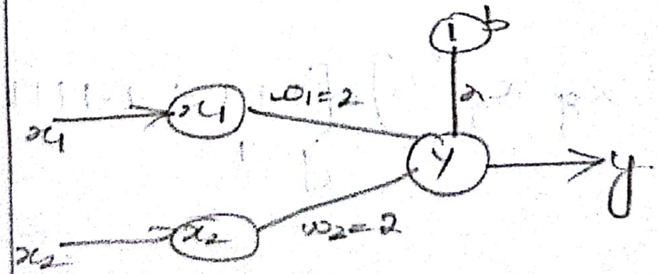
* Fourth i/p

$[x_1 \quad x_2 \quad b] = [-1 \quad -1 \quad 1] \quad y = -1$

$w_1(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

$w_2(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

$b(new) = 3 - 1 = \underline{\underline{2}}$



x_1	x_2	y
1	1	6
1	-1	2
-1	1	2
-1	-1	-2 → inhibition

+ve excitation (for rows 1–3)

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} \geq 2 \\ -1 & y_{in} < 2 \end{cases}$$

11/9/23 Mon

Using hebb rule find the weights required to perform following classification for the given input pattern.

The '+' symbol represents value 1 and the empty squares represent '-1'. Consider I belongs to the target class 1 and 'O' belongs to target class -1

x_1	x_2	x_3
+	+	+
x_4	x_5 +	x_6
x_7 +	x_8 +	x_9 +

'I'

1

+	+	+
+		+
+	+	+

'O'

-1

Ans:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	b	y
1	1	1	-1	1	-1	1	1	1	1	1
1	1	1	1	-1	1	1	1	1	1	-1

$w_1 = w_2 = \dots w_9 = b = 0.$

$[x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ b] = [1\ 1\ 1\ -1\ 1\ -1\ 1\ 1\ 1\ 1]$

$y = 1$

$w_1 new = w_1(old) + x_1 y$

$= 0 + 1 \times 1 = \underline{\underline{1}}$

$w_2 new = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_3 new = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_4 new = 0 + -1 \times 1 = \underline{\underline{-1}}$

$w_5 new = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_6 new = 0 + -1 \times 1 = \underline{\underline{-1}}$

Scanned by CamScanner

$w_7(new) = 0 + 1 \times 1 = \underline{1}$

$w_8(new) = 0 + 1 \times 1 = \underline{1}$

$w_9(new) = 0 + 1 \times 1 = \underline{1}$

$b(new) = 0 + 1 = \underline{1}$

x. $[x_1 \; x_2 \; x_3 \; x_4 \; x_5 \; x_6 \; x_7 \; x_8 \; x_9 \; b] = [1 \; 1 \; 1 \; 1 \; -1 \; 1 \; 1 \; 1 \; 1 \; 1] \; y = -1$

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

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

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

$w_4(new) = -1 + 1 \times -1 = \underline{-2}$

$w_5(new) = 1 + -1 \times -1 = \underline{2}$

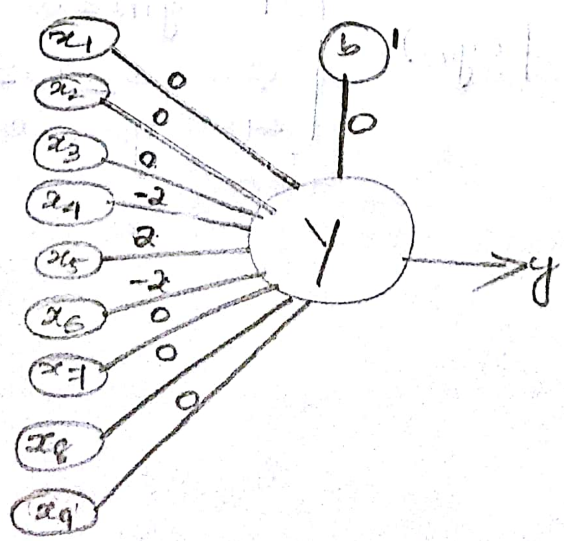
$w_6(new) = -1 + 1 \times -1 = \underline{-2}$

$w_7(new) = 1 + 1 \times -1 = \underline{0}$

$w_8(new) = 1 + 1 \times -1 = \underline{0}$

$w_9(new) = 1 + 1 \times -1 = \underline{0}$

$b(new) = 1 + -1 = \underline{0}$.



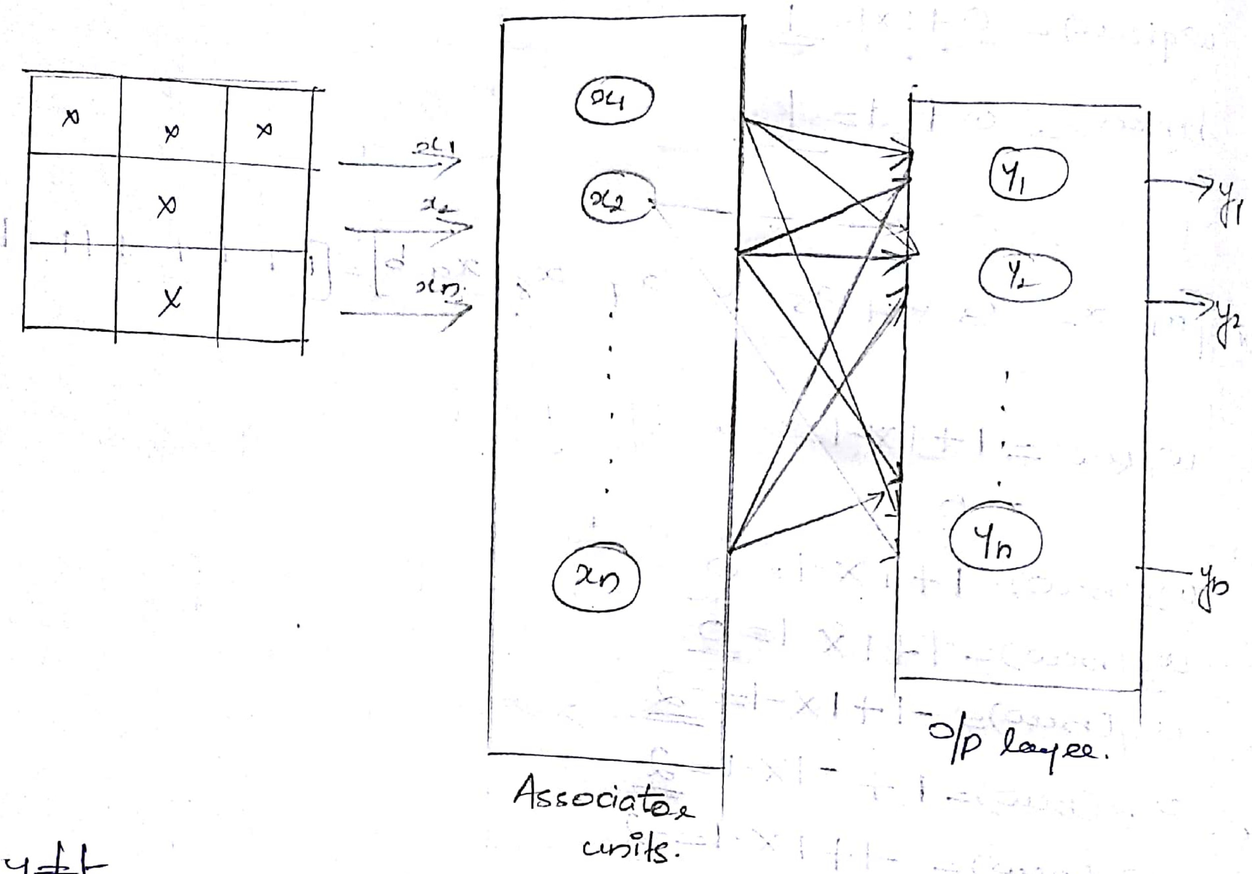
First input $y_{in} = 6$ (0+0+0+2+2+2+0+0+0+0) Excitation

Second input $y_{in} = -6$ (0+0+0-2-2-2+0+0+0+0) Inhibition

14/9/17 Thu

MODULE 2

PERCEPTRON NETWORK.



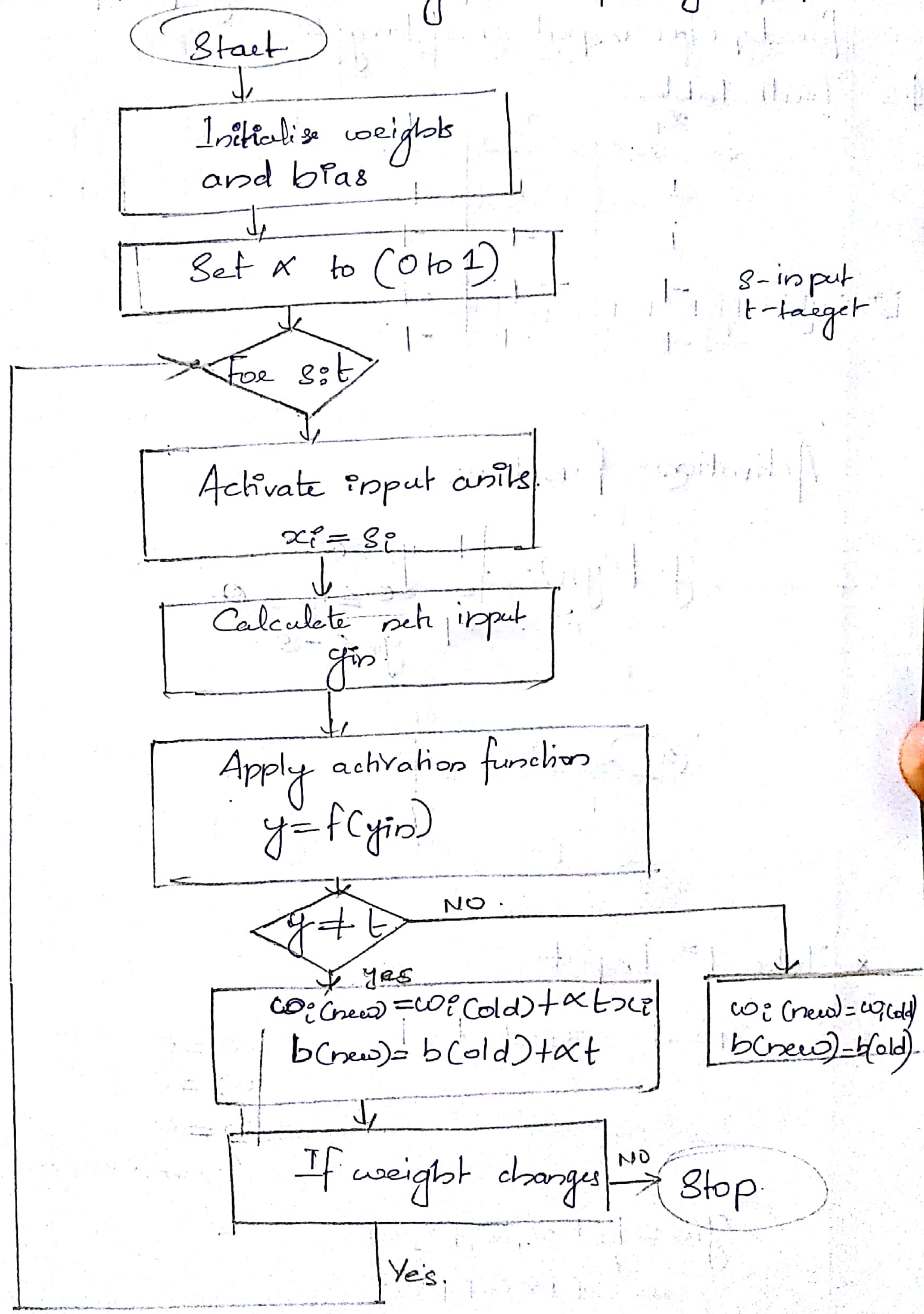
If $y \neq t$

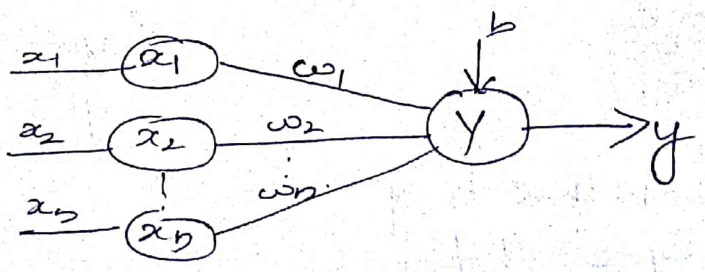
$$w_i(new) = w_i(old) + \alpha t x_i$$

$$b(new) = b(old) + \alpha t$$

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} > \theta \\ 0 & -\theta \leq y_{in} \leq \theta \\ -1 & y_{in} < -\theta \end{cases}$$

Flow Chart. Perceptron Training Method for single output:





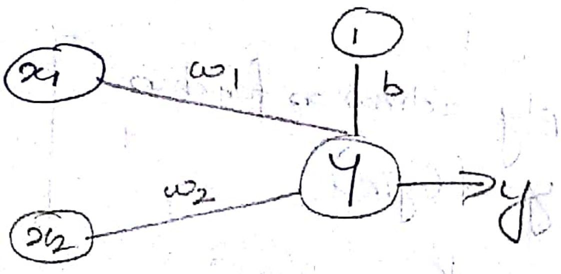
1. Implement AND function using perceptron networks for bipolar input and target

Ans. Truth table

x_1	x_2	t
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Activation function:

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} > 0 \\ 0 & y_{in} \leq 0 \\ -1 & y_{in} < 0 \end{cases}$$



* Take 1st Input

$x_1 = 1 \quad x_2 = 1 \quad t = 1$

$w_1 = 0$
$w_2 = 0$
$b = 0$

Assume $\theta = 0$.

$$y_{in} = b + x_1 w_1 + x_2 w_2$$
$$= 0 + 1 \times 0 + 1 \times 0$$
$$= \underline{\underline{0}}$$

if $y_{in} = 0$
$y = 0$

Check whether $t = y$

$t = 1 \, ; \, y = 0 \qquad t \neq y$

Scanned by CamScanner

$w_1(new) = w_1(old) + \alpha t x_1$ | Assume $\alpha = 1$

$= 0 + 1 \times 1 \times 1$

$= \underline{\underline{1}}$

$w_2(new) = w_2(old) + \alpha t x_2$

$= 0 + 1 \times 1 \times 1$

$= \underline{\underline{1}}$

$b(new) = b(old) + \alpha t$

$= 0 + 1 \times 1$

$= \underline{\underline{1}}$.

* Take $\underline{2^{nd} \text{ input}}$

$x_1 = 1 \quad x_2 = -1 \quad t = -1$ | $w_1 = 1$, $w_2 = 1$, $b = 1$

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

$= 1 + 1 \times 1 + -1 \times 1$

$= \underline{\underline{1}}$

if $y_{in} = 1$

$y = 1 = f(y_{in})$.

$y = 1 ; t = -1 \quad y \neq t$.

$w_1(new) = w_1(old) + \alpha t x_1$ | Assume $\alpha = 1$.

$= 1 + 1 \times -1 \times 1$

$= \underline{\underline{0}}$

$w_2(new) = w_2(old) + \alpha t x_2$

$= 1 + 1 \times -1 \times -1$

$= \underline{\underline{2}}$.

$b(new) = b(old) + \alpha t$

$= 1 + 1 \times -1$

$= 0$ //

* Take 3rd input

$x_1 = -1 \quad x_2 = 1 \quad t = -1$ | $w_1 = 0$, $w_2 = 2$, $b = 0$

$y_{in} = b + x_1 w_1 + x_2 w_2$
$= 0 + -1 \times 0 + 1 \times 2$
$= 2$

If $y_{in} = 2$.

$y_{in} > 0$.
$y = 1$
$y \neq t$.

$w_1(new) = w_1(old) + \alpha t x_1$
$= 0 + 1 \times -1 \times -1$
$= 1$

$w_2(new) = w_2(old) + \alpha t x_2$
$= 2 + 1 \times -1 \times 1$
$= 1$

$b(new) = b(old) + \alpha t$
$= 0 + 1 \times -1$
$= -1$

* Take 4th input

$x_1 = -1 \quad x_2 = -1 \quad t = -1$ | $w_1 = 1$, $w_2 = 1$, $b = -1$

$y_{in} = b + x_1 w_1 + x_2 w_2$
$= -1 + -1 \times 1 + -1 \times 1$
$= -3$

If $y_{in} = -3$ $y_{in} < 0$ $y = -1$

$y = t$. No updation

$w_1 = 1$

$w_2 = 1$

$b = -1$