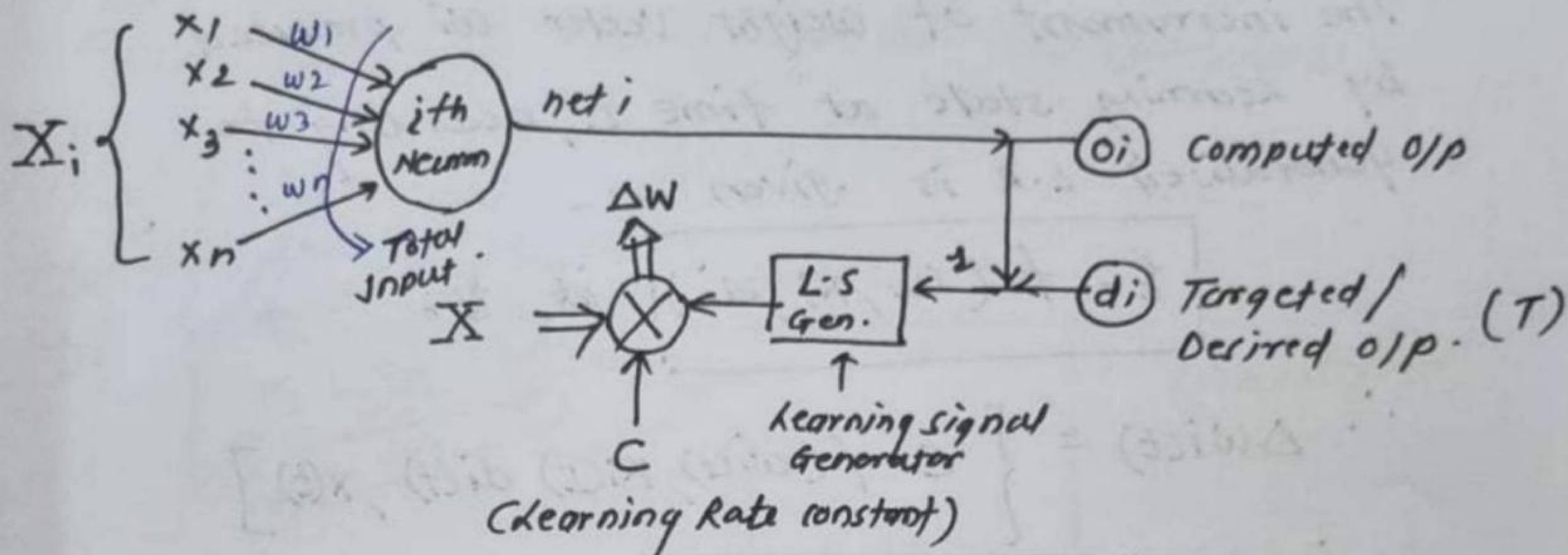
* Generalised Learning Rule :- (AMARI -1990)



consider any ith neuron from NN-Architecture.
having input → X_i , weight vectors - w_i
output → O_i
If we are using supervised Learning then target or Desired o/p d_i (T).

$$x_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad w_i = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{in} \end{bmatrix}, \quad w_i^T = [w_{i1}, w_{i2}, \dots w_{in}]$$

C → Learning Rate constant
r → Learning signal.

Learning ~~rate~~ signal r is general function of w_i (weight vector), X_i, and desired o/p (d_i) → for supervised Learning.

$$\boxed{net\ i = \sum_{i=1}^{n} (x_i, w_i)}$$

Scanned by TapScanner

The increament of weight vector w_i produced by learning state at time t, according to generalised L.R is given as-

$$\boxed{\xi = f(w_i, x_i, d_i)} \quad \text{iA, } \xi A$$

$$\therefore \Delta w_i(t) = \{ c \cdot f(w_i(t), x_i(t), d_i(t), x(t)]$$

ie, $$\boxed{\Delta w_i(t) = c \cdot \xi \cdot x(t)} \quad \text{—— (A)}$$ – Adjustment of wights.

Where, $c \rightarrow$ positive learning rate constant which determines rate of learning.

Thus, The weight vector adopted at time t will becomes at next instant $t+1$ as

$$\boxed{w_i(t+1) = w_i(t) + \Delta w_i(t)}$$

ie, $$w_i(t+1) = w_i(t) + \{ c \cdot f[w_i(t), (x(t), d_i(t)] \cdot x \cdot (t)]$$

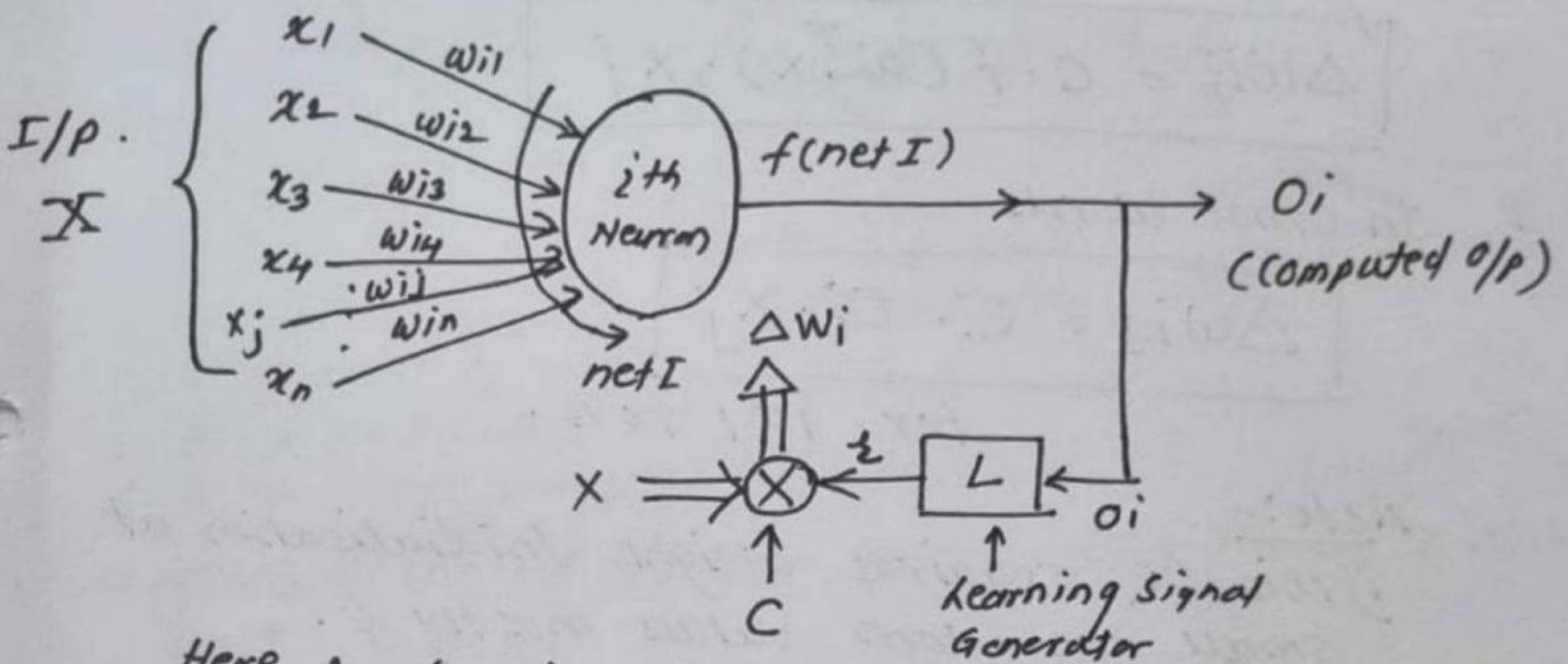
In general form

$$\boxed{w_i^{(k+1)} = w_i^{(k)} + c \cdot \xi \cdot (w_i^k, x^k, d_i^k)}$$

Scanned by TapScanner

Hebbian Learning Rule : (1949 by Donald Hebb)

This rule is applicable for unsupervised Learning.



Here, ε - Learning signal
C - +ve Learning constant

Being unsupervised LR, this Hebbian LR the learning signal ε is computed o/p.

ie, $\boxed{\varepsilon = o_i}$ ie, $(\omega^T . x)$

where $o_i = f(\text{net } I)$

$\boxed{o_i = f(\omega^T . x)}$ $\longrightarrow$ since $\boxed{net = \sum_{i=1}^{n} x_i w_i}$

Thus Adjustment of weights ie, $\Delta\omega_i$ of the weight vector Becomes

$\boxed{\Delta w_i = c . \varepsilon . x}$ $\rightarrow$ using Generalised LR.

$\therefore \boxed{\Delta w_i = c . f(\omega^T . x) . x}$

This is weight adjustment formula for all inputs.

Scanned by TapScanner

Lets consider jth Input to ith Neuron.
with weight ω_{ij}.
Using above formula -

$$\boxed{\Delta W_{ij} = c \cdot f(\omega_i^T x) \cdot x_j}$$

In other words.

$$\boxed{\Delta w_{ij} = C \cdot O_i \cdot X_j}$$

for $i = 1$ to n.

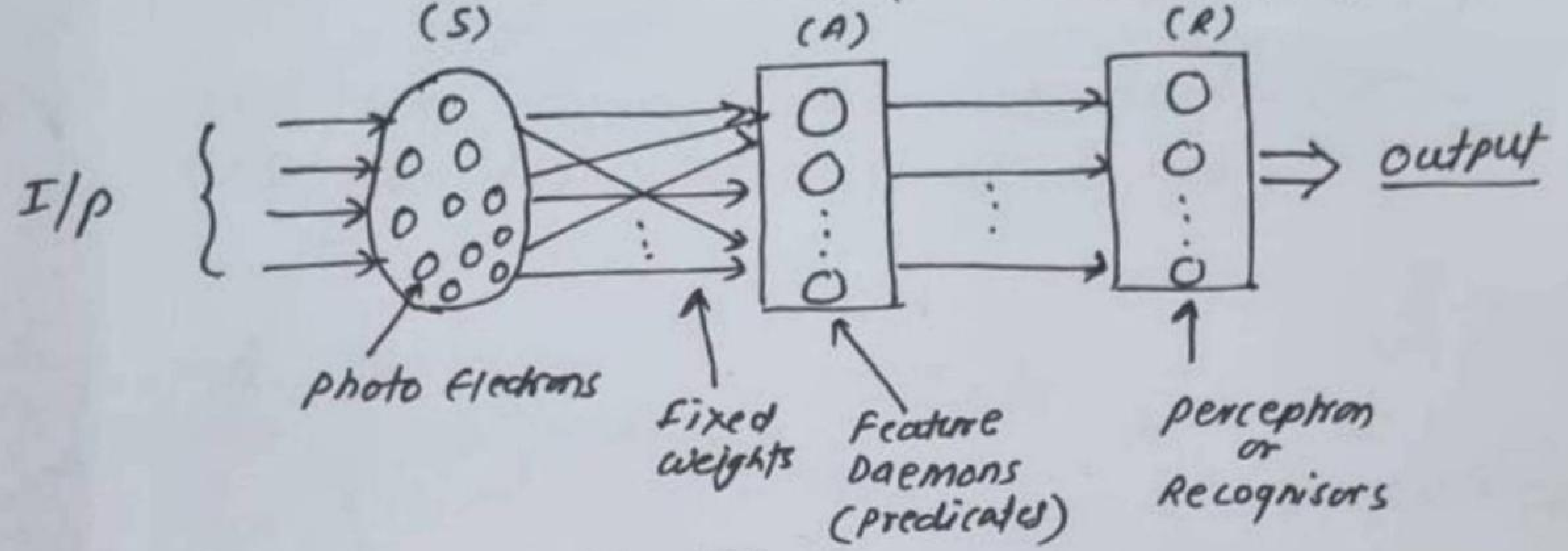
Note:-
1) This LR requires weight Initialisation at small random values mostly 0.
2) This LR represents feed forward, Unsupervised Learning.
3) Thus this rule states that if the cross product of I/P and O/P ie, $O_i \times X_j$ is positive that results in increase of weight ω_{ij} otherwise weight decreases.

—— X ——

Scanned by TapScanner

(Supervised Learning)

2) <u>PERCEPTRON MODEL</u>: (Rosenblatt - 1958)



[S → Sensory unit
A → Association unit
R → Recognation unit]

* Model inspired for the human vision having 3 basic units S, A, R.
* Basic units of S-unit are photo detectors (Aprox. 400)
* Association unit consists of <u>predicates</u> or <u>feature Demons</u>. which store features extracted by S unit.
* R-unit consists of Recognisors or perceptrons which produced o/p.
* weights betn S-A units are fixed.
* weights betn A-R unit are Adjustable.

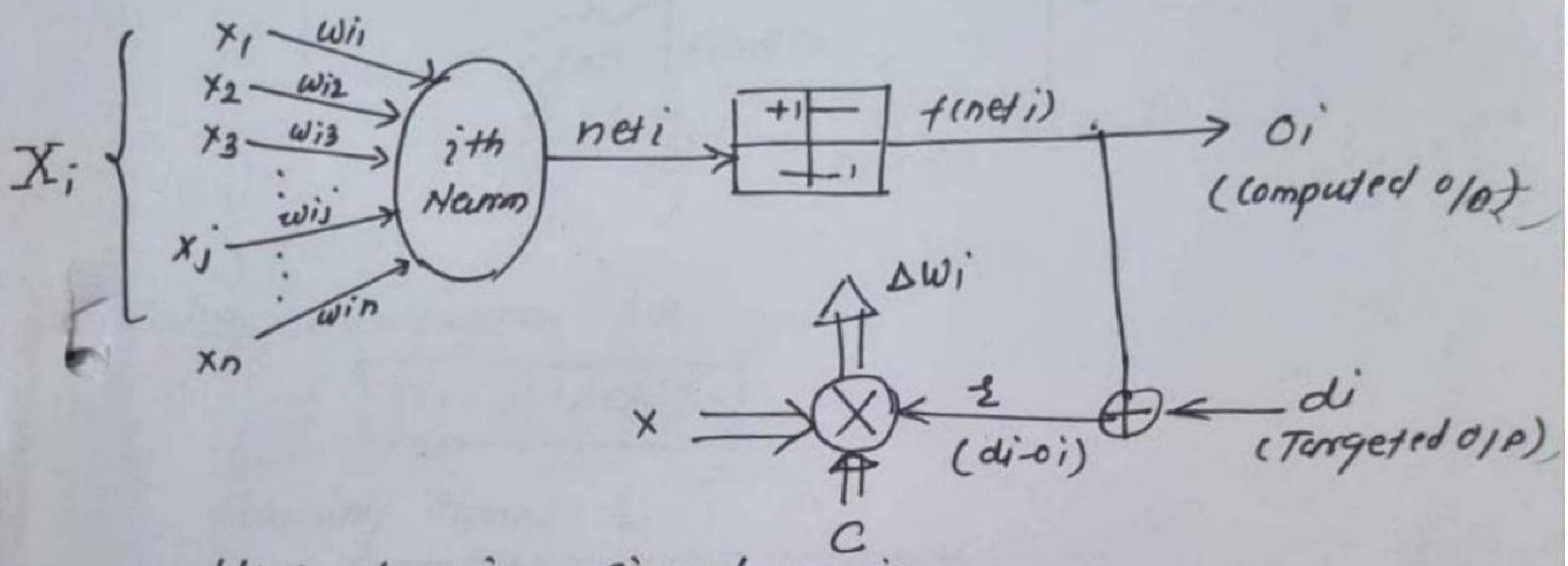
<u>Working</u>:- 1) If the total I/p comming to photo detectors (S-unit) is greater than or equal to threshold o/p is 1 else 0.

2)

Scanned by TapScanner

* Perceptron LR (Rosenblatt - 1958)

This rule is applicable for feed forward networks with supervised Learning.



Here, Learning signal r is

$$\boxed{r = d_i - o_i}$$

$o_i = f(net_i) = sgn(net_i)$

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

$net_i = [w_i]^T \cdot [X]$

$\Delta w_i = C \cdot r \cdot X$

$\therefore \Delta w_i = c\,[d_i - o_i] \cdot X$

$\therefore \Delta w_i = c\,[d_i - sgn(net_i)] \cdot X$

For input Link j

Adjustment of weight Δw_{ij}

$$\boxed{\Delta w_{ij} = c \cdot [d_i - sgn(net_i)] \cdot x_j}$$

If $d_i = -1$, $o_i = 1$

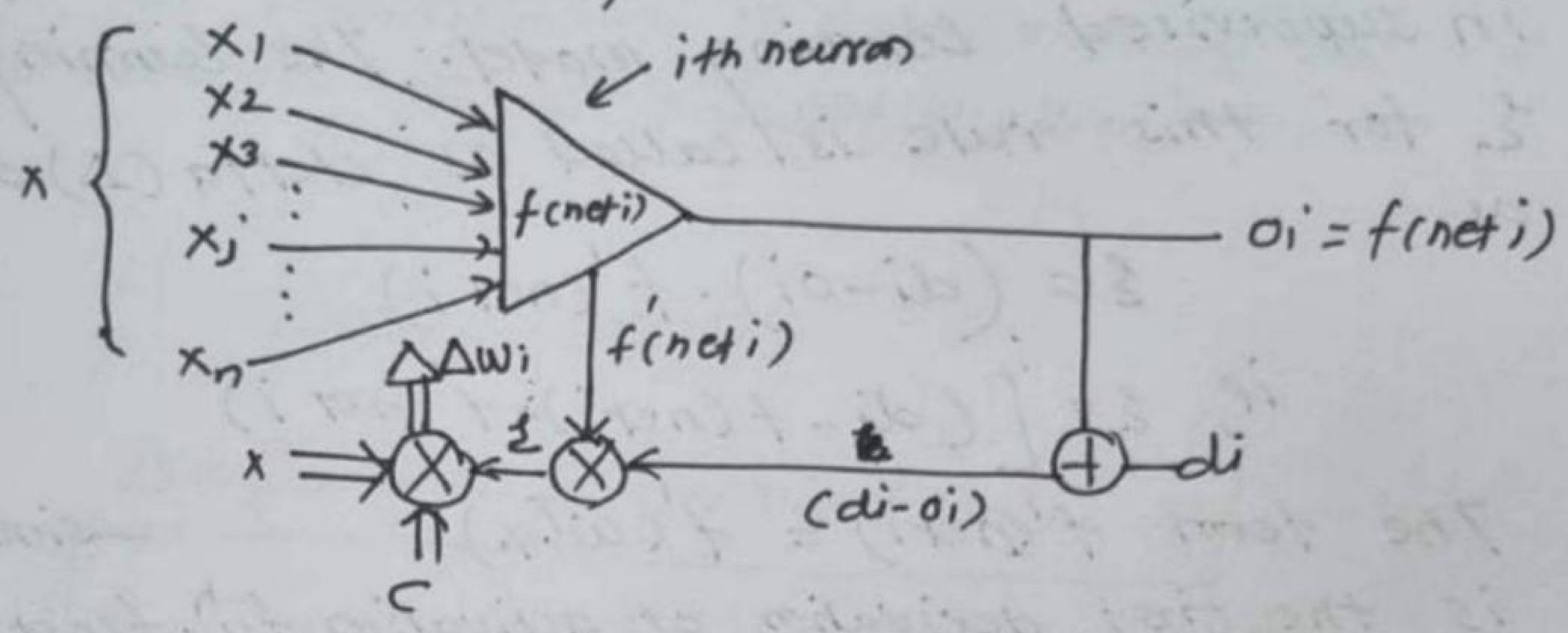
$$\boxed{\Delta w = -2cX}$$

if $d_i = 1$, $o_i = -1$

$$\boxed{\Delta w = +2cX}$$

Scanned by TapScanner

Delta Learning Rule



Using Generalised LR

$$\boxed{o_i = f(net_i)}$$

∴ Learning signal δ is

$$\boxed{\delta = (d_i - o_i) * f'(net_i)}$$

$$\therefore \delta = [(d_i - f(net_i)] \cdot f'(net_i) \quad \text{— since } o_i = f(net_i)$$

Hence

$$\Delta w_i = c \cdot \delta \cdot x$$

$$\therefore \Delta w_i = c[(d_i - o_i) \cdot f'(net_i)] \cdot x$$

$$\Delta w_i = c\{[(d_i - f(net_i)] \cdot f'(net_i)\} \cdot x$$

$$\Delta w_i = c[(d_i - o_i) \cdot f'(net_i)] \cdot x$$

for jth i/p.

$$\boxed{\Delta w_{ij} = c[(d_i - o_i) \cdot f'(net_i)] \cdot x_j} \text{ — (A)}$$

Scanned by TapScanner

This rule is valid for continious activation f^n in supervised Learning model. The learning signal ξ for this rule is called as delta (Δ) and defined as

$$\xi = (d_i - o_i) \cdot f'(net_i)$$

$$\text{ie, } \xi = [(d_i - f(net_i)) * f'(net_i)]$$

The term $f'(net_i) = f'(\omega^T \cdot x)$ — since $net_i = (\omega^T \cdot x)$
is the first derivation of activation f^n. $f(net_i)$

This LR can be also derived from the condition of Least square error b/w o_i and d_i. Calculating the gradient vector wrt ω_i of the square error It can be defined as.

$$\boxed{E = \frac{1}{2}(d_i - o_i)^2}$$ — since error is $E = (d_i - o_i)$

$$E = \frac{1}{2}[d_i - f(\omega^T \cdot x)]^2 \quad \text{— ①}$$

The error gradient vector value is

$$\boxed{\nabla E = -(d_i - o_i) \cdot f'(\omega^T \cdot x) \cdot X} \rightarrow \nabla E = \frac{\partial E}{\partial \omega_i}$$

The components of gradient vectors are

$$\frac{\partial E}{\partial \omega_{ij}} = -(d_i - o_i) \cdot f'(\omega^T \cdot x) \cdot x_j \text{ — for } j^{th} \text{ I/P.}$$

s

Scanned by TapScanner

since the minimisation of error requires the weight change in a negative gradient direction.

ie, $\Delta w_{ij} = -\eta \cdot \nabla E$

$\therefore \Delta w_{ij} = -\eta \cdot [-(d_i - o_i) \cdot f'(w^T \cdot x) \cdot x_j]$

$\therefore \boxed{\Delta w_{ij} = \eta \cdot [(d_i - o_i) \cdot f'(w^T \cdot x) \cdot x_j]}$ — B

Here, η is some +ve constant

Thus from eqn. A & B we can conclude that the weight adjustment term which we get using generalised LR is similar to the value which we compute using error gradient.

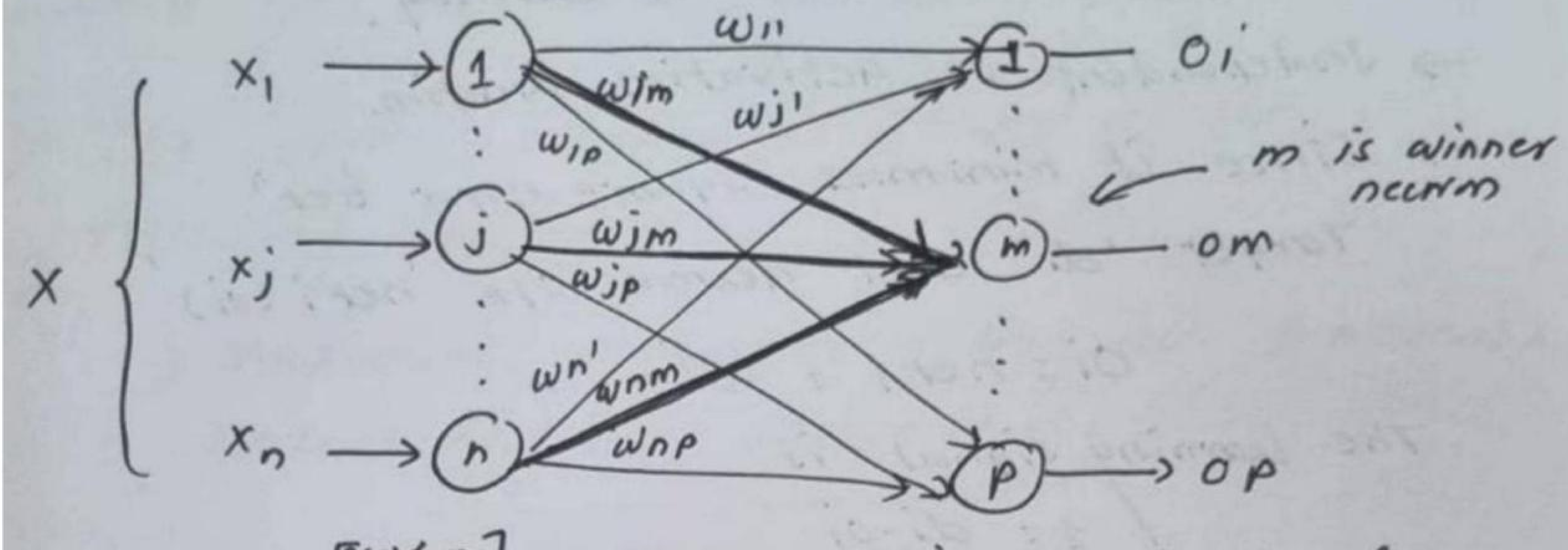
<u>using G.LR</u> - $\Delta w_{ij} = c[(d_i - o_i) \cdot f'(w^T \cdot x) \cdot x_j]$ — A

<u>using Error gradient</u> $\Delta w_{ij} = \eta [(d_i - o_i) \cdot f'(w^T \cdot x) \cdot x_j]$ — B

—x—

Scanned by TapScanner

Competitive LR (Winner take All) Rule



$$w_m = \begin{bmatrix} w_{1m} \\ w_{2m} \\ \vdots \\ w_{jm} \\ \vdots \\ w_{nm} \end{bmatrix}$$

This rule is applicable for Unsupervised learning since there is no desired o/p.

This LR only can be demonstrated or explain for arrange layer of 'p' neurons.
This is example of competative learning in Unsuped Learning.

⇒ This LR based on the assumption that one of the neuron in o/p layer say 'm' neuron has maximum response due to input 'x'. Thus this neuron is decleared as a winning neuron. It has weights - $[w_{1m} \quad w_{2m} \cdots w_{jm} \cdots w_{nm}]$

⇒ For next iteration only these weights undergo updatation using formula

$$\boxed{\Delta w_m = \alpha (X - w_m)}$$

for jth neuron

$$\boxed{\Delta m_j = \alpha (X \, w_{mj})} \quad \text{for } j = 1 \text{ to } n.$$

Scanned by TapScanner

Widrow-Hoff LR (1962)

→ Applicable for supervised Learning.

→ Independent of activation function.

→ Since it minimise square error betⁿ Target 'd_i' and neuron o/p '$net_i (o_i)$'

$$o_i = net_i = \omega^T \cdot x$$

The learning signal is

$$\delta = d_i - o_i$$

i.e.,

$$\boxed{\delta = d_i - \omega^T \cdot x}$$

The weight adjustment for ith neuron is

$$\Delta\omega_i = c \cdot \delta \cdot x$$

$$\therefore \Delta\omega_i = c \cdot [(d_i - \omega^T \cdot x)] \cdot x \quad \text{— (A)}$$

Thus for jth neuron input

$$\Delta\omega_{ij} = c \cdot [(d_i - (\omega^T \cdot x))] \cdot x_j \quad \text{— (B)}$$

for $j = 1$ to n

Its special case of Delta LR.

$$f(\omega_i^T \cdot x) = \omega_i^T \cdot x,$$

Activation funⁿ is $f(\omega^T \cdot x) = f(net_i) = 1$.

The eqⁿ of ΔLR becomes.

$$\boxed{\Delta\omega_{ij} = c \cdot (d_i - o_i) \cdot x_j}$$

Scanned by TapScanner

Where α is a small learning constant and it decreases as learning progreeses.

⇒ The winner selection is based on the following criteria.

1) maximum activation among all p neurons participating in competition.

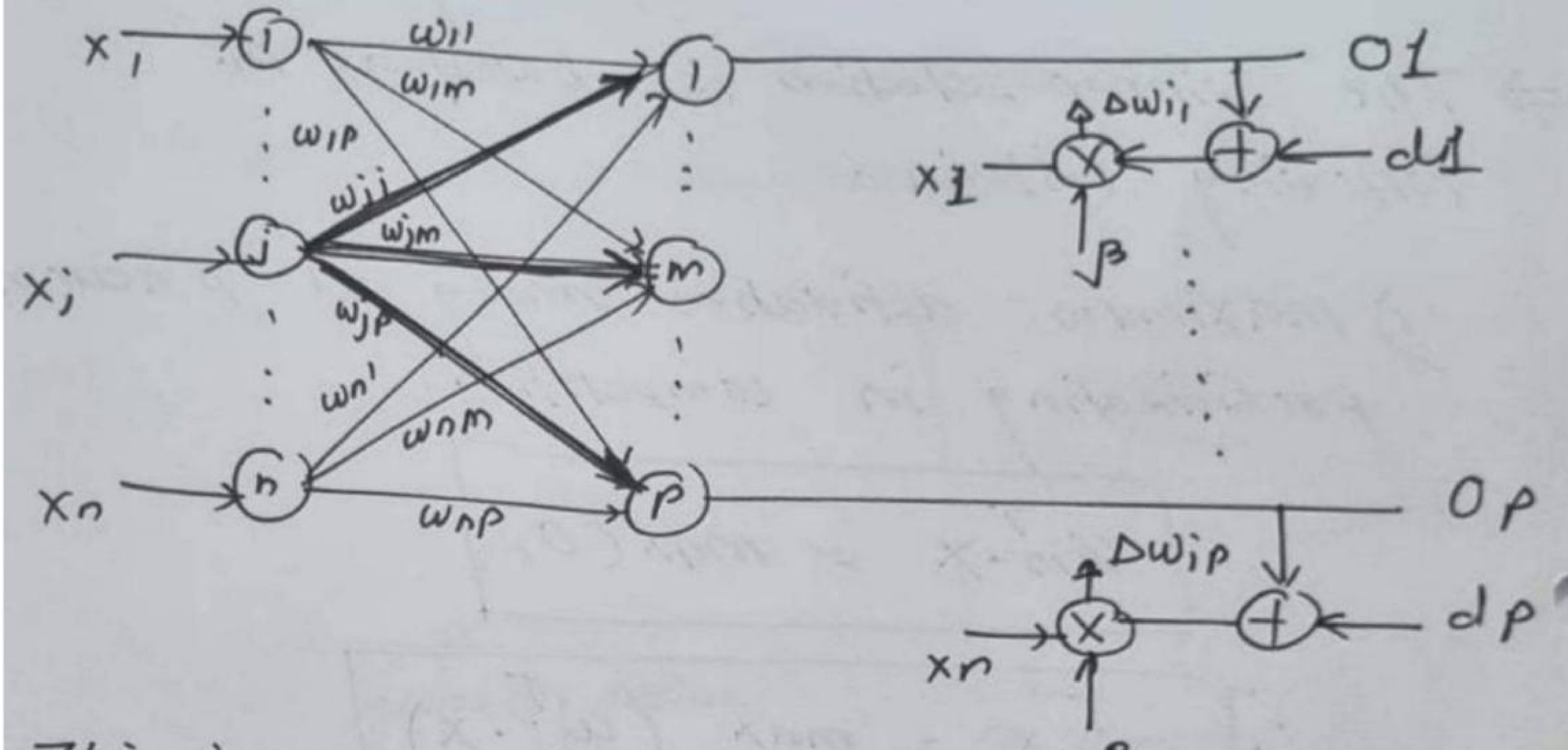
$$\boxed{w_m^T \cdot x = \max(0,}$$

$$\therefore \boxed{w_m^T \cdot x = \max_{i=1 \text{ to } p} (w_i^T \cdot x)}$$

—x—

Scanned by TapScanner

Outstar LR



This is example of supervised LR used to provide learning of repetative and charecteristics properties of I/p - o/p relationship.

It suppose to allow a n/w to extract statistical properties of the I/p & o/p signals.

weight adjustment can be computed as.

$$\Delta w_j = \beta[d_j - w_j]$$

and Individual Weight adjustment

$$\boxed{\Delta w_{mj} = \beta[d - w_{mj}]} \quad \text{for } m = 1 \text{ to } P$$

β is small +ve learning constant and it decreses as learning progreess.

This rule typically ensures that the o/p pattern becomes simillar to undistorted desired o/p after repeteally applying the weight adjustment formula on distorted o/p.

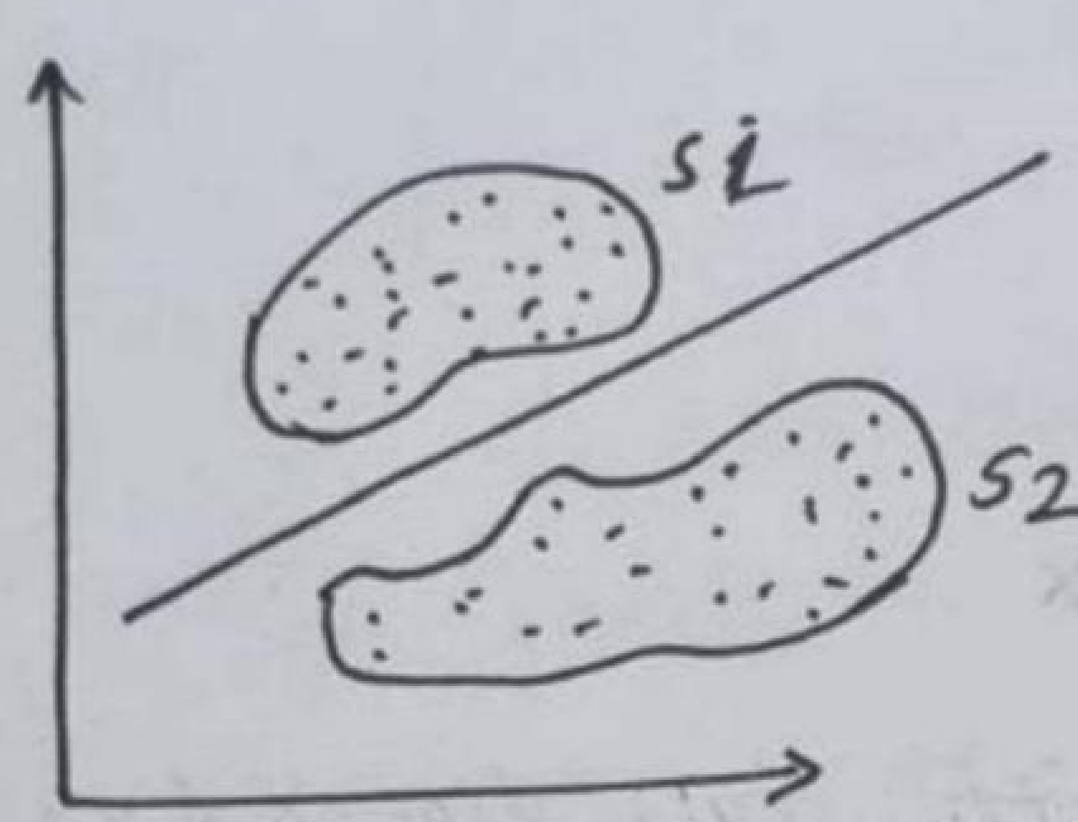
(4) (13)

Scanned by TapScanner

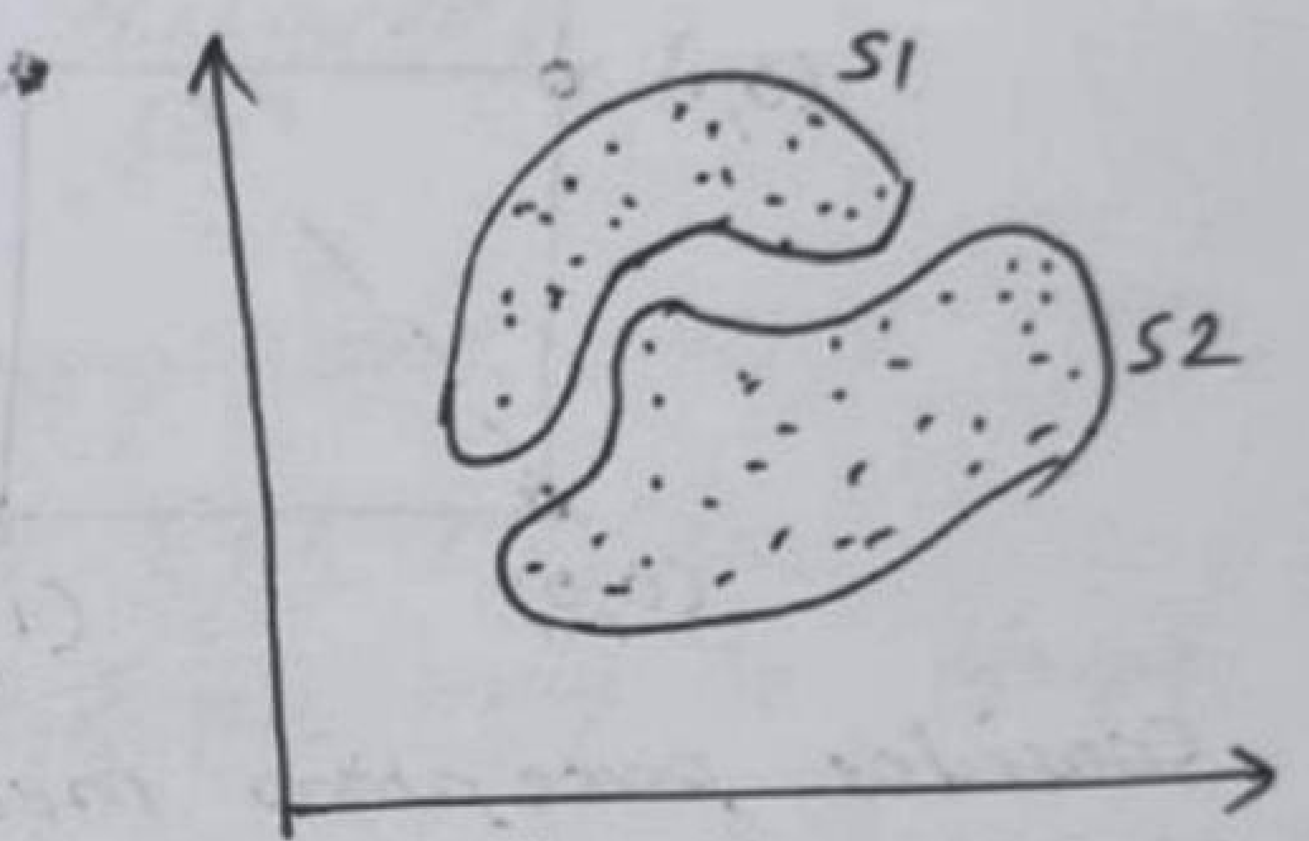
* Linear separability

perceptron model can not handle the tasks which are linearly separables.

Sets of points in 2-D spaces are linearly separable if the [redacted] sets can be separated by a stright line.



① Linearly separable pattens ⓐ



(b) Non-Linearly separable pattens.

⇒ A set of points in n-dimensional space are linearly separable if there is a hyperplane of (n-1) dimension that separates the sets.

The perceptron cannot find weights for classification type of problem that are not linearly separables. It can be illustrated by X-OR problem.

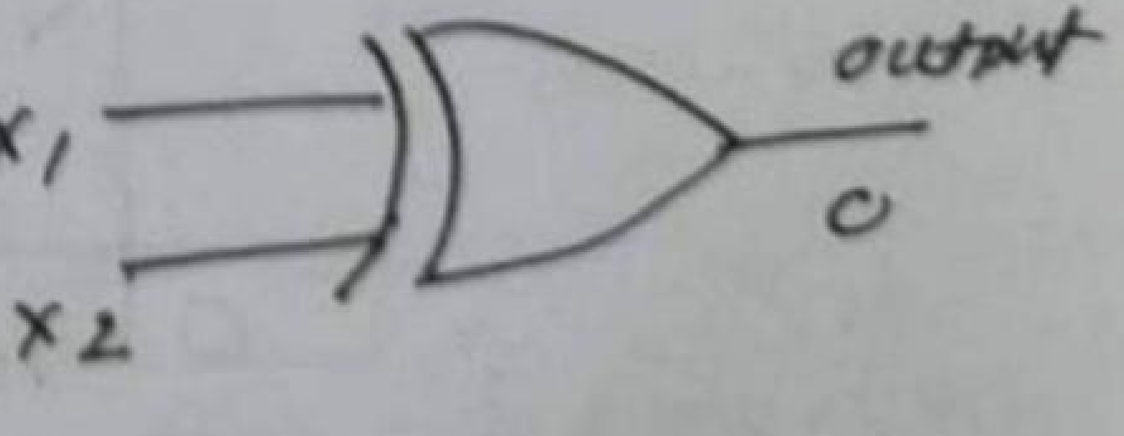
* X-OR problem :-

X-OR is logical operation given as.

x_1	x_2	output
0	0	0
1	1	0
0	1	1
1	0	1

→ Even parity

→ odd parity

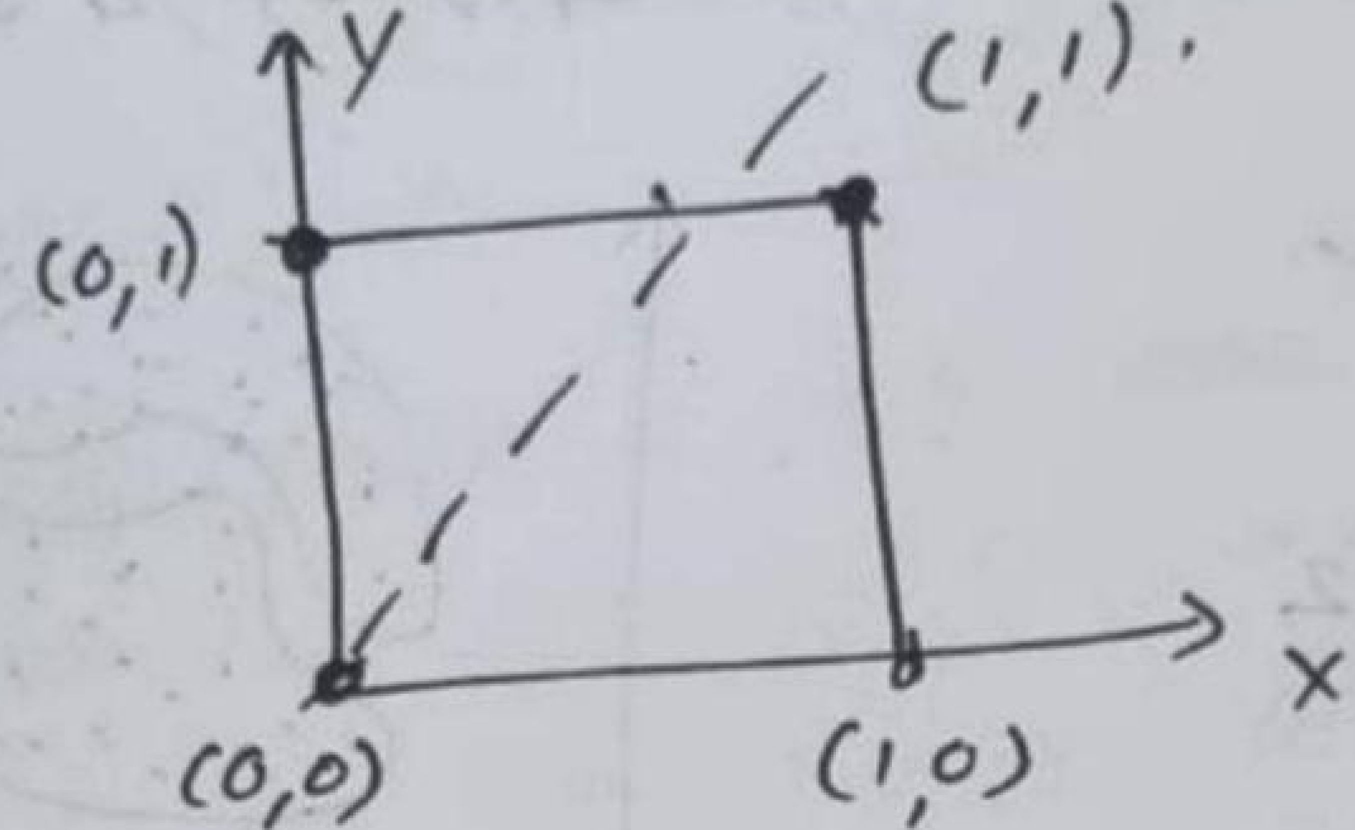


Ⓐ

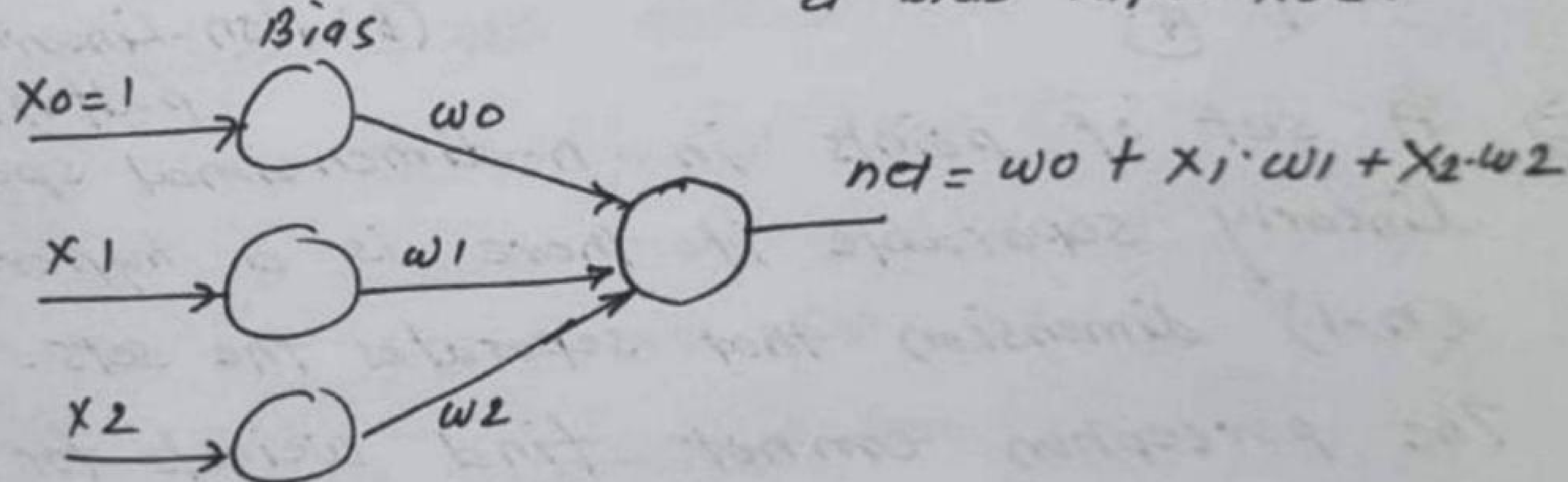
The problem/Task for ANN is to clasify the Inputs as odd parity or Even parity.

[odd parity → odd no. of 1, Even parity → Even no. of 1]

This is impossible to find a line separating even parity input pattens from odd parity i/p pattens.



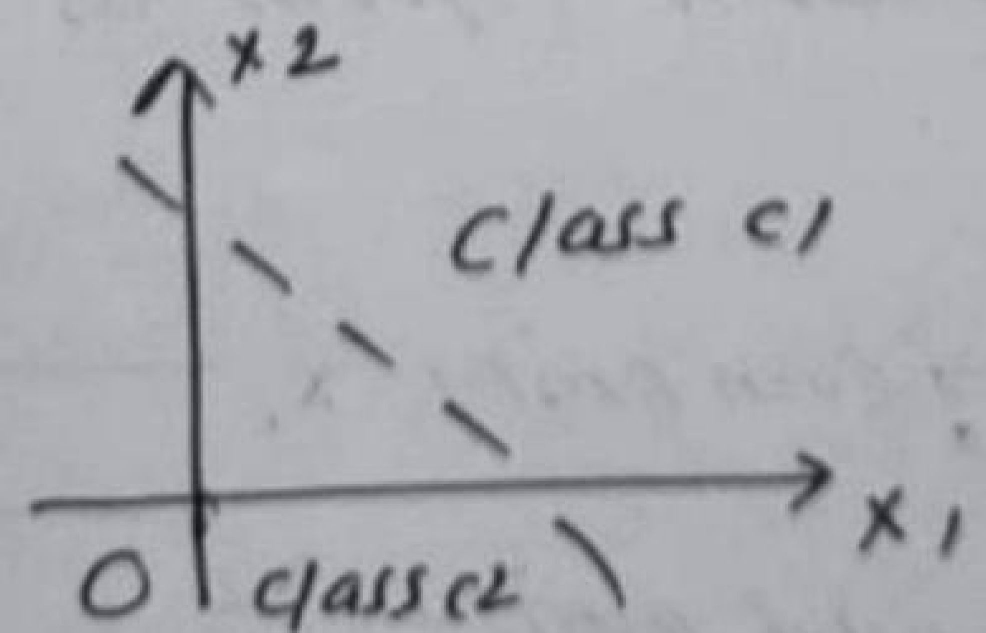
Consider pereptron model with two I/ps. X_1 & X_2 & a bias input $X_0 = 1$



$$net = w_0 + x_1 \cdot w_1 + x_2 \cdot w_2$$

This is eqn. of straight line $(y = mx + c)$

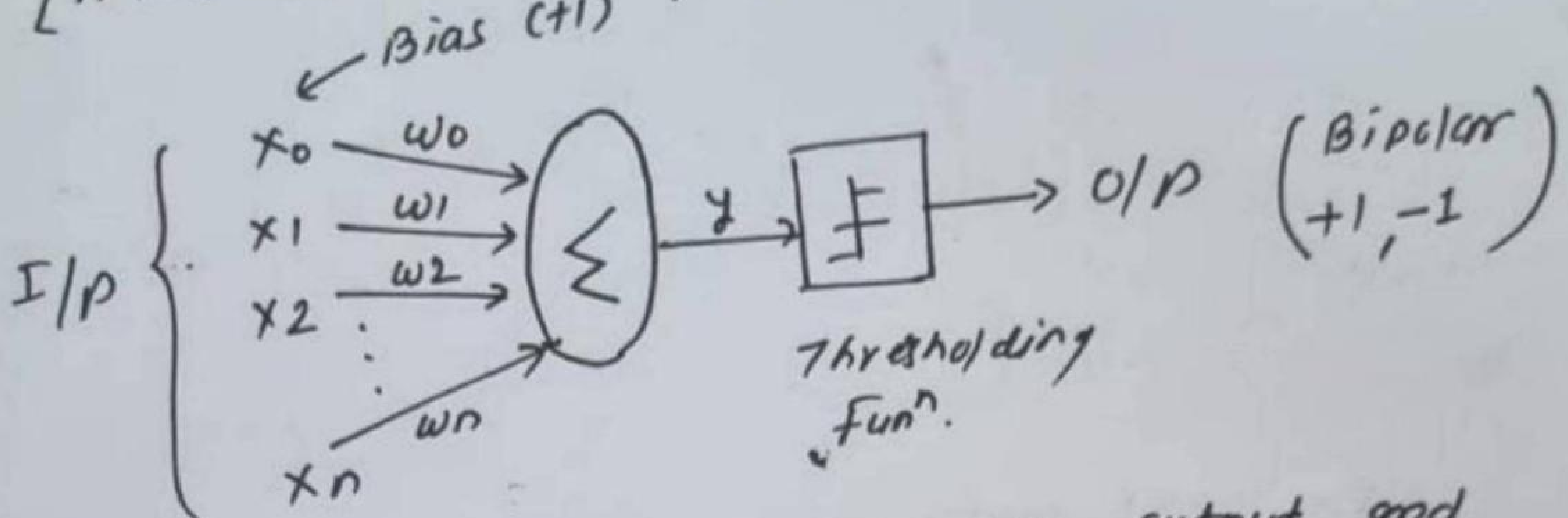
This straight line acts as decision boundary separating the points into class c1 & c2.



15

ADALINE n/w :- [Bernard widrow - Hoff] - 1960

[Adaptive linear n/w]



In this n/w there is only one output and o/p values are bipolar (+1 or -1). However the I/p x_1 & x_2 may be binary, bipolar are real values.

The bias weight is ~~w~~0 and I/p is $x_0 = +1$

Firing Rule :- If the weighted sum of I/p Input. is greater than or equal to 0 then o/p is 1 other wise -1.

This n/w adopts supervised Learning algo. which is simillar to perceptron.

This Leaning algo. is known as Least mean square [LMS] or Delta Rule. and is given as

$$\boxed{\omega_i^{new} = \omega_i^{old} + \alpha(d_i - o_i) \cdot x_i}$$

α - Learning coefficent

d_i - Target

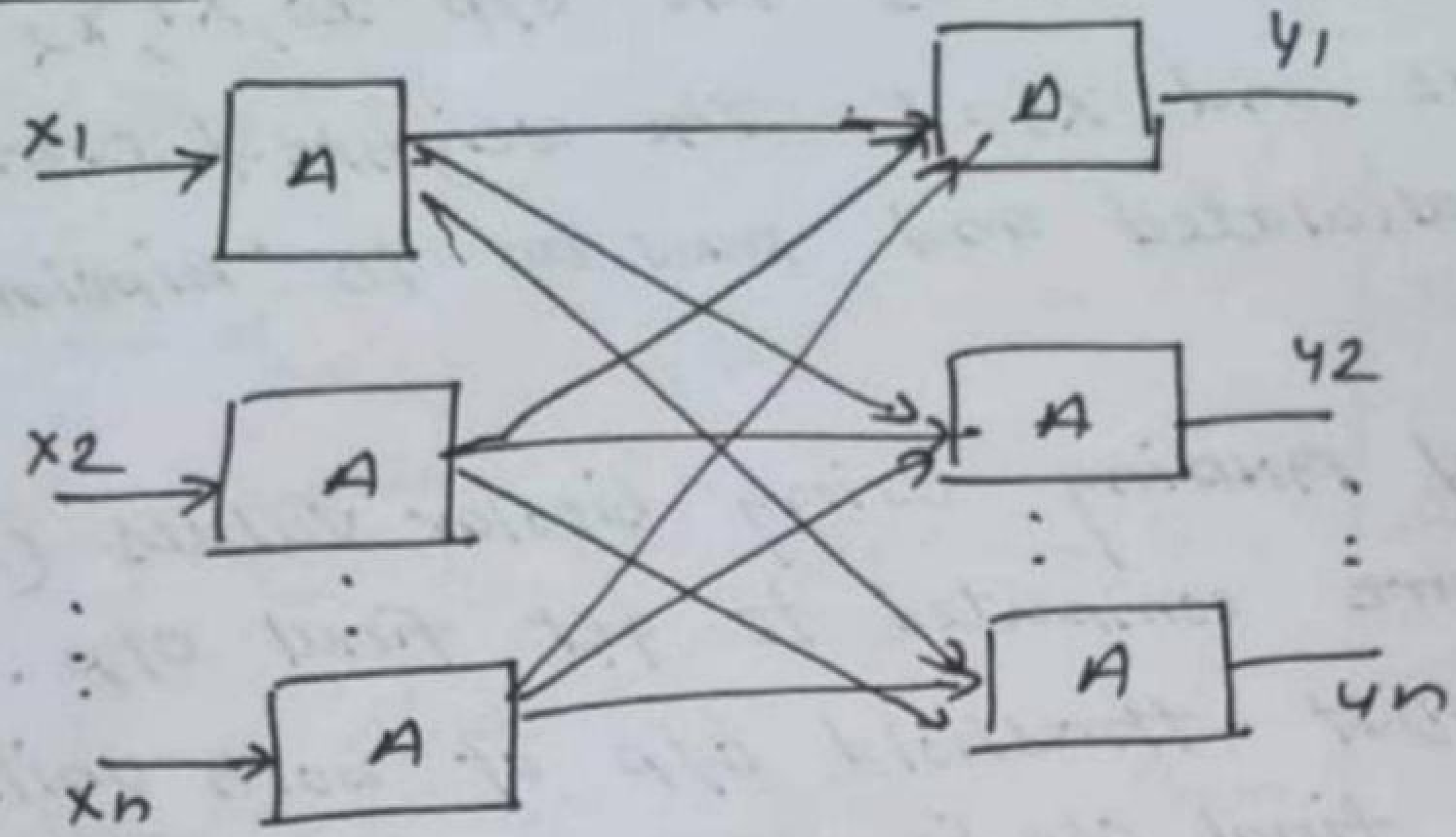
for any jth I/P.

$$\boxed{\omega_{ij}^{new} = \omega_{ij}^{old} + \alpha(d_i - o_i) \cdot x_{ij}}$$

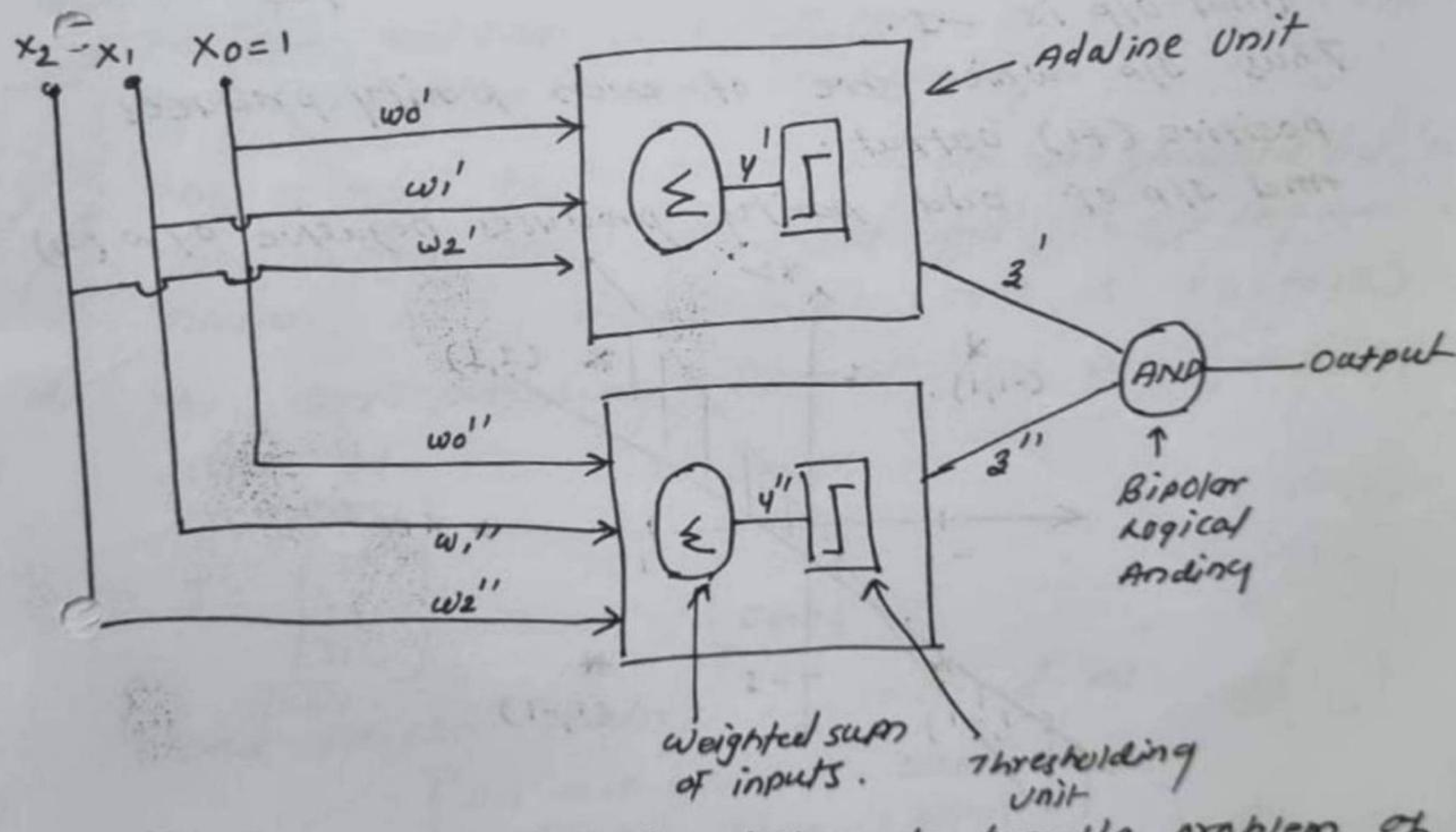
These networks is used in all high speed modems, Telephone switching systems to cancel the echo in long distance communication ckts.

Scanned by TapScanner

MADALINE (many Adaline) N/W.



Solving X-OR problem using MADALINE N/W



The use of many Adaline helps to handle problem of non-linear seperability.
Using 2-Adaline we can solve X-OR problem.

Scanned by TapScanner

Every Adaline recives the I/p x_0, x_1, x_2 where x_0 is bias and $x_0 = 1$. The weighted sum of I/p is calculated and pass on to 'Bipolar thresholding unit.'

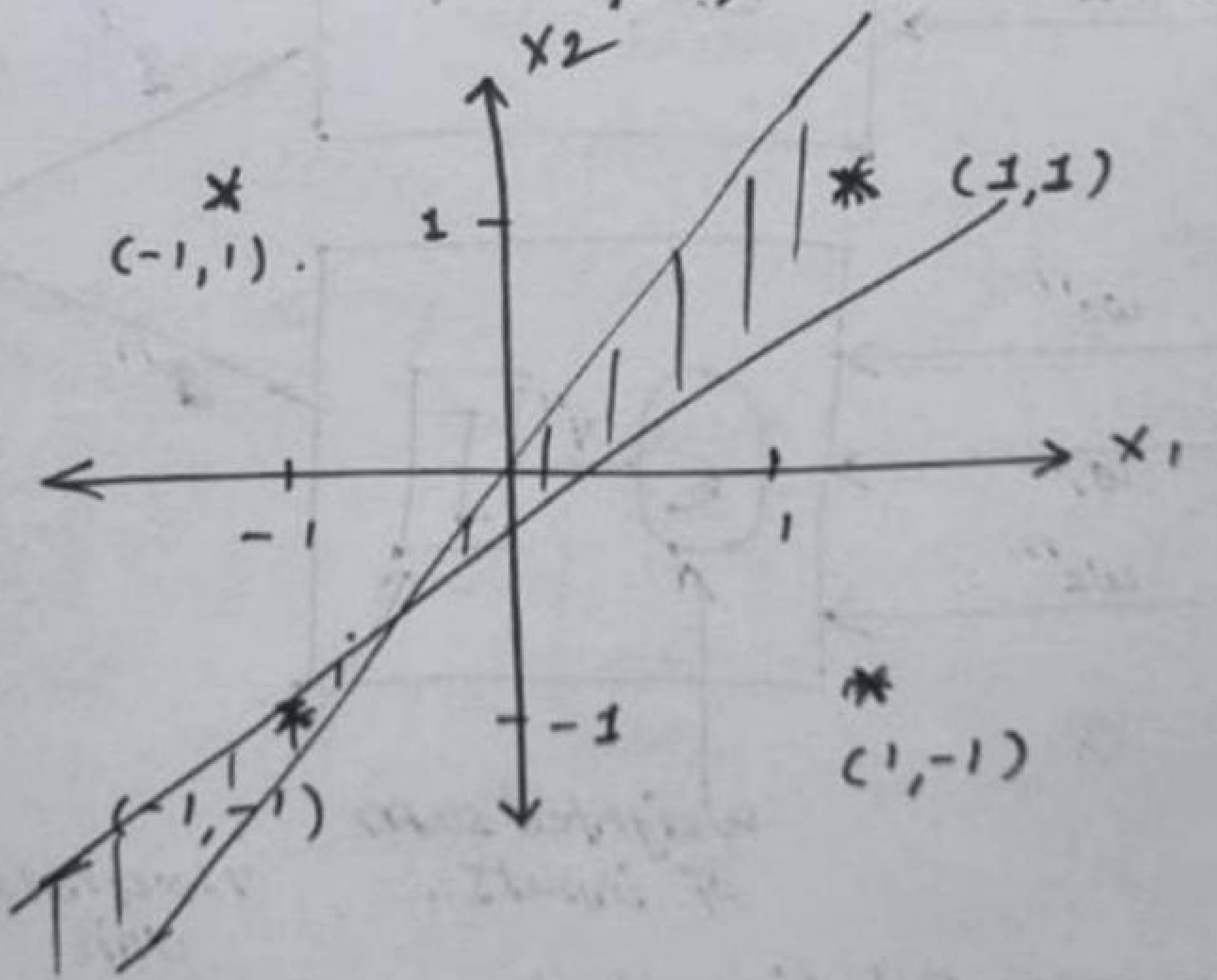
The logical ANDing using bipolar values (1, −1) of 2 units are computed to get final o/p.

<u>condition</u>:- 1) If threshold o/p of both units is +1 or −1 then final o/p is +1.

2) If threshold o/p are different (+1, −1) then the final o/p is −1.

Thus I/p which are of even parity produces positive (+1) output.

And I/p of odd parity produces negative o/p (−1)



T.T

x_0	x_1	x_2	o/p
1	0	0	+1
1	1	1	+1
1	0	1	−1
1	1	0	−1

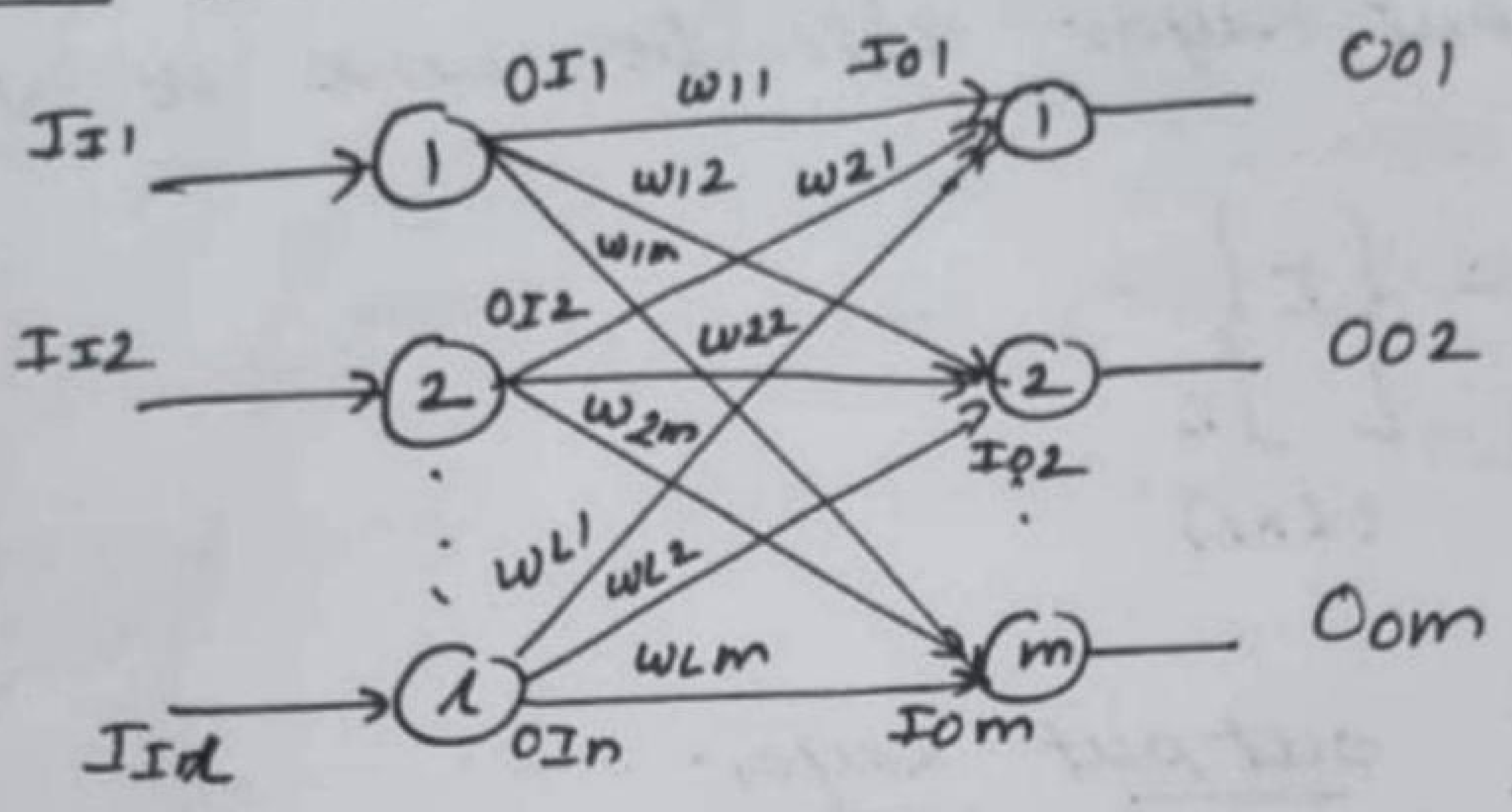
⇒ Binary

1
1
0
0

Scanned by TapScanner

Back propogation Network :- (Supervised Learning)

* **SLFFN Analysis:-**



II → Input to Input Layer

OI → O/P of Input Layer

IO → Input to output Layer

OO → O/P of O/P Layer

⇒ Consider SLFFN with total 'L' neurons in I/P Layer and 'm' neurons in O/P Layer.

→ The weights betⁿ these Layers are denoted by 'w'. Thus weight betⁿ i^{th} input and j^{th} O/P Layer neuron is w_{ij} (for $i = 1$ to n & $j = 1$ to m)

→ The corrosponding I/P and O/P are denoted as vector as.

$$I = \begin{Bmatrix} I_{I1} \\ I_{I2} \\ \vdots \\ I_{IL} \end{Bmatrix}_{(L\times 1)} \qquad O = \begin{Bmatrix} O_{O1} \\ O_{O2} \\ \vdots \\ O_{Om} \end{Bmatrix}_{(m\times 1)}$$

and weight matrix is denoted as.

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{1m} \\ w_{21} & w_{22} & w_{23} & \cdots & w_{2m} \\ \vdots & & & & \\ w_{L1} & w_{L2} & w_{L3} & \cdots & w_{Lm} \end{bmatrix}_{(L\times m)}$$

Assuming linear activation fuⁿ for I/P Layer and Unipolar sigmoidal fuⁿ for O/P Layer.

Scanned by TapScanner

STEP : 1) O/P of Input Layer.

since we use linear activation fu^n for I/p Layer o/p of Input Layer will be same as Input.

ie,

$$\{O\}_I = \{I\}_I$$
$$(l \times 1) \quad (l \times 1)$$

step : 2) I/P to output Layer

consider some j^{th} o/p Layer neuron.

so Input to j^{th} neuron will be.

$$I_{Oj} = O_{I1} \cdot w_{1j} + O_{I2} \cdot w_{2j} + \cdots O_{Il} w_{lj}$$
for $j = 1$ to m

Hence in vector form

$$\{I\}_O = [W]^T \cdot \{O\}_I$$
$$(m \times 1) \quad (m \times l) \quad (l \times 1)$$

step : 3) : O/P of output Layer

since we are using unipolar sigmoidal fu^n.

$$\boxed{O = \frac{1}{1+e^{-\lambda \cdot I}}} \rightarrow \lambda \rightarrow \text{sigmodal gain}$$

$$\{O\}_O = f([W] \cdot [I\}_O)$$

$$\text{ie, } \{O\}_O = \begin{bmatrix} \frac{1}{1+e^{-\lambda \cdot I_{O1}}} \\ \vdots \\ \frac{1}{1+e^{-\lambda I_{Oj}}} \\ \vdots \\ \frac{1}{1+e^{-\lambda I_{Om}}} \end{bmatrix}$$

20

Scanned by TapScanner

Back Propogation Algorithm :- (MLFFN)

Algo:-

Step1:- Initialize the weights.

Step2:- Repeat, for each training patten, train the network for that pattern.

Step 3:- Untill the error is acceptably low.

Step:1) Normalised and initialise the I/p and O/p wrt their maximum ~~weights~~.

Thus, for a n/w of 'l' i/p layer neuron. I/p vector will be:

$$\{I\}_I$$
$$l \times 1$$

and for n o/p layer neurons o/p vector will be $\{O\}_O$
$$n \times 1.$$

Step:2) Assume the no. of neurons in hidden layer lies betn $\boxed{l < m < 2l}$

Step:3) Let matrix [V] represents weights betn I/p - hidden layer and [W] represents hidden-o/p layer.

Initialise weights to small random valuey usually from -1 to 1.

Scanned by TapScanner

for general problem λ can be assumed as 1. and the threshold value can be consider as 0. Then,

$$[V]^0 \text{ and } [W]^0 = \text{random weights.}$$

so, $\Delta V = 0$ & $\Delta W = 0$.

Step:4> Input Layer computation:-

For given training data present one set of I/P and target O/P. present the patten to the I/P layer $\{I\}_I$. By using linear activation fun. the O/P of Input layer can be computed as.

$$\underset{(l\times 1)}{\{O\}_I} = \underset{(l\times 1)}{\{I\}_I}$$

Step:5> Hidden Layer computation

The I/P to hidden layer can be computed as

$$\{I\}_H = [V]^T \cdot \{O\}_I$$

$$\text{ie, } \underset{(m\times 1)}{\{I\}_H} = \underset{(m\times l)}{[V]^T} \cdot \underset{(l\times 1)}{\{I\}_I}$$

In vector form;

$$\begin{Bmatrix} I_{H1} \\ I_{H2} \\ I_{H3} \\ \vdots \\ I_{Hm} \end{Bmatrix} = [V]^T \cdot \begin{bmatrix} I_{I1} \\ I_{I2} \\ \vdots \\ I_{Il} \end{bmatrix}$$

Scanned by TapScanner

Step 6> The o/p of hidden Layer for p^{th} hidden Layer neuron. can be represented as

$$\boxed{O_{Hp} = \frac{1}{1+e^{-\lambda \cdot I_{Hp}}}}$$

i.e, $$\boxed{O_{Hp} = \frac{1}{1+e^{-I_{Hp}}}}$$ $\because \lambda = 1$

Hence

$$\{O\}_{H \atop m\times 1} = \begin{Bmatrix} \frac{1}{1+e^{-I_{H1}}} \\ \vdots \\ \frac{1}{1+e^{-I_{Hp}}} \\ \vdots \\ \frac{1}{1+e^{-I_{Hm}}} \end{Bmatrix}$$

Step: 7> O/p Layer computations.

I/p to o/p Layer. computations

$$\{I\}_{O \atop (n\times 1)} = [W]^T_{(n\times m)} \cdot \{O\}_{H \atop (m\times 1)}$$

Step: 8> O/p of O/p Layer.

The o/p of q^{th} o/p Layer neuron is

$$O_{Oq} = \frac{1}{1+e^{-\lambda \cdot I_{Oq}}} = \frac{1}{1+e^{-I_{Oq}}}$$

$$\{O\}_{O \atop n\times =} = \begin{Bmatrix} \frac{1}{1+e^{-I_{O1}}} \\ \vdots \\ \frac{1}{1+e^{-I_{Oq}}} \\ \vdots \\ \frac{1}{1+e^{-I_{On}}} \end{Bmatrix}$$ (3)

Scanned by TapScanner

Step: 9 > Using Root mean square Error formula:

we can write error as.

$$\boxed{E = \frac{1}{2}(T_O - O_O)^2}$$

$T_O = d_O$ = Expected o/p.

~~Thus for o/p layer we will get.~~

Calculate the Error and difference betn computed o/p & desired (Target) o/p for ith training set.

$$\boxed{E_p = \frac{\sqrt{\Sigma (T_i - O_{Oi})^2}}{n}}$$

for $i = 1$ to n.

Step 10 > find vector $\{d\}$ as.

$$\{d\} = \begin{Bmatrix} (T_1 - O_{Oi}) \cdot O_{O1} \cdot (1 - O_{O1}) \\ \\ (T_K - O_{OK}) \cdot O_{OK} \cdot (1 - O_{OK}) \end{Bmatrix}_{n \times 1}$$

Step 11 > find Y matrix as

$$\underset{m \times n}{[Y]} = \underset{m \times 1}{\{O\}_H} \cdot \underset{1 \times n}{\langle d \rangle}$$

← Transpose of $\{d\}$

Step 12 > Find ΔW as

$$[\Delta W]^{t+1} = \alpha \cdot [\Delta W]^{t} + \eta \cdot [Y]$$

Scanned by TapScanner

Step:13) Find vector e ie {e} as

$$\underset{(m\times 1)}{\{e\}} = \underset{(m\times n)}{[w]} \cdot \underset{(n\times 1)}{\{d\}}$$

Find {d*} as

$$\underset{m\times 1}{\{d^*\}} = \left\{ \begin{array}{c} e_1 \cdot (O_{H1}) \cdot (1 - O_{H1}) \\ \vdots \\ e_i\,(O_{Hi}) \cdot (1 - O_{Hi}) \\ \vdots \\ e_m\,(O_{Hm}) \cdot (1 - O_{Hm}) \end{array} \right\}$$

Find [X] as.

$$\underset{l\times m}{[X]} = \{O\}_I \cdot \langle d^* \rangle$$

$$\underset{(l\times m)}{[X]} = \underset{(l\times 1)}{\{I\}_I} \cdot \underset{(1\times m)}{\langle d^* \rangle}$$

Step: 14) Find ΔV as.

$$\underset{(l\times m)}{[\Delta V]}^{t+1} = \alpha \cdot \underset{(l\times m)}{[V]}^{t} + \eta \cdot [X]$$

Step: 15) Find new weights as

$$\underset{(l\times m)}{[V]}^{t+1} = \underset{(l\times m)}{[V]}^{t} + \underset{(l\times m)}{[\Delta V]}^{t+1}$$

$$\underset{m\times n}{[W]}^{t+1} = \underset{m\times n}{[W]}^{t} + \underset{m\times n}{[\Delta W]}^{t+1}$$

Scanned by TapScanner

Step:16> Find error rate as

$$\text{Error Rate} = \frac{\Sigma EP}{n}$$

$n \rightarrow$ total no. of training sets.

Step:17> Repeat from step (4) to (16) until the convertion of error rate is less than the tollerance value

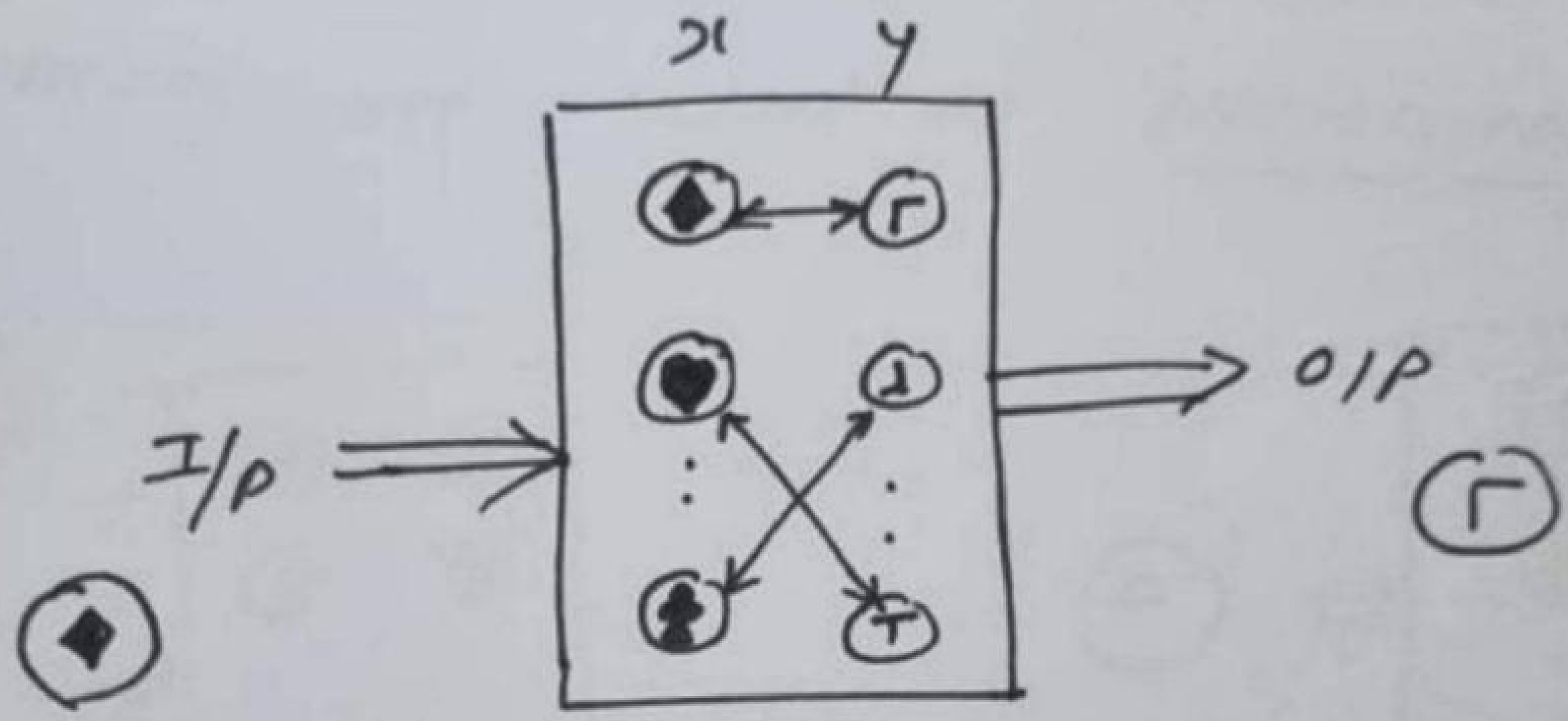
———— x ————

Scanned by TapScanner

Associative memory

It is a store house of associated patterns which are encoded in some form, when a storehouse triggered with any pattern, the associated pattern pair is recalled or output.

The I/p patten will be an exact replica of stored patten or distorted or partial patten.



Here there are associated patten pairs.

$$\begin{array}{ll} & x \quad y \\ \rightarrow & (\blacklozenge, \Gamma) \\ \rightarrow & (\heartsuit, \top) \\ \rightarrow & (\clubsuit, \bot) \end{array}$$

The association is alway represented by directional arrow.

⇒ If the associated patten pair (x,y) are different and the model recalls Y, if x is inputed, then it is reffered as <u>Hetero associative memory</u>.

⇒ If the associated patten pair (x,y) are same then it is refered as ~~Hetto~~ <u>Auto associative memory</u>.

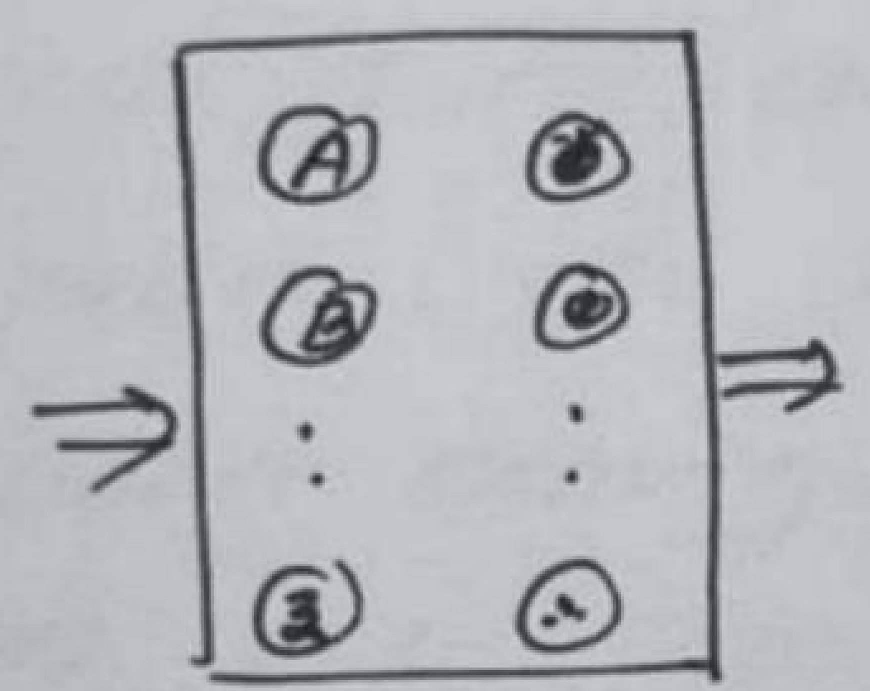
20

Scanned by TapScanner

Heteroassociative memory are used for association. and Autoassociative memory are used for image refinement. ie, if distorted or partial image is given it will recall complete image.

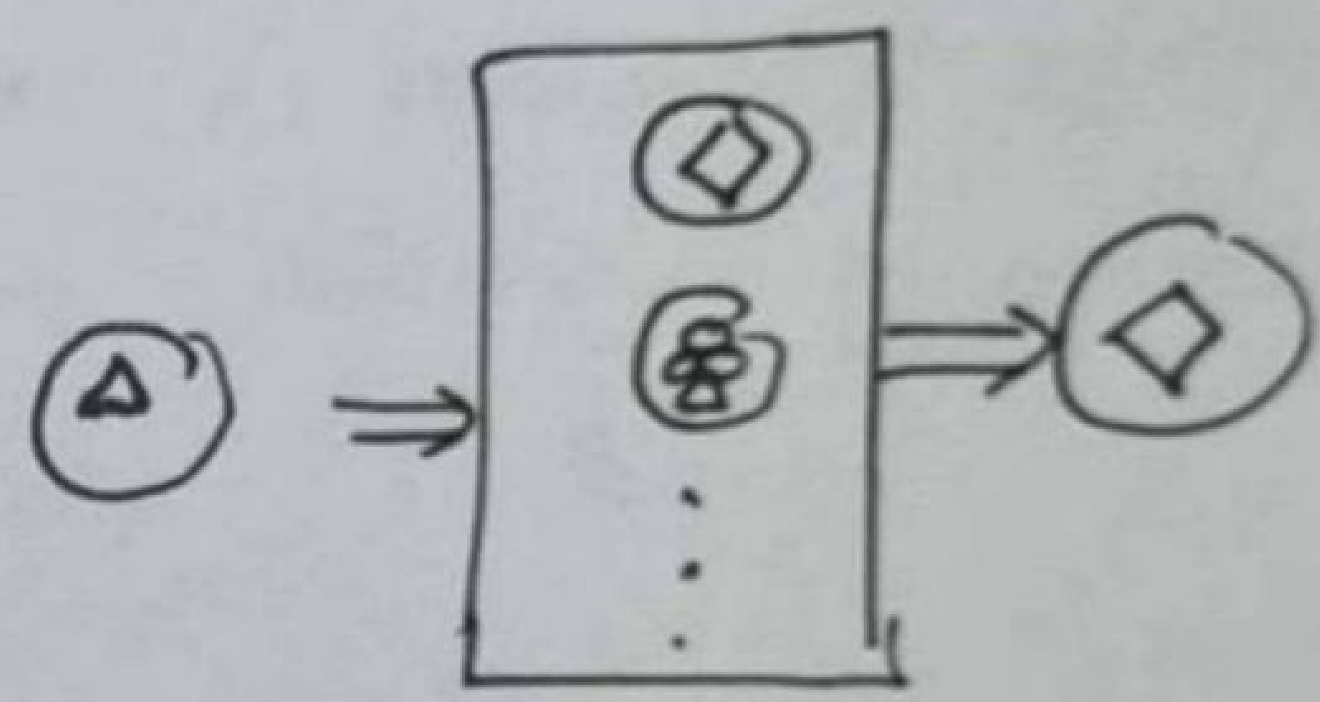
⇒ Associative correlation memory are called as Autocorrelators and heteroassociative correlation memory are called as Heterocorrelations

Ex:-



$x \neq y$

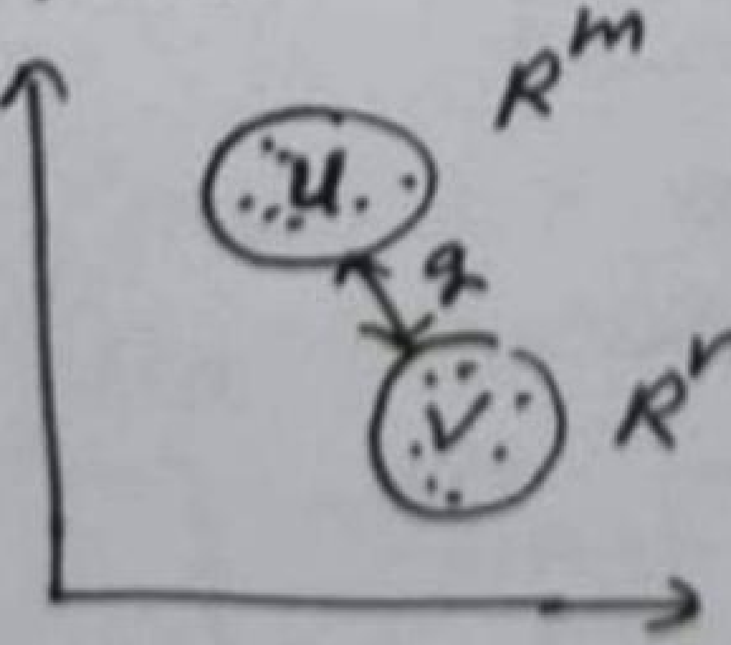
Heteroassociation



$x = y$

Auto association

Thus we can say autoassociative memory is a concept of mapping 'g' between 2 pattern space.



If u belongs to Region M and v belongs to region n. then

$$\boxed{\bar{v} = g(\bar{u})}$$

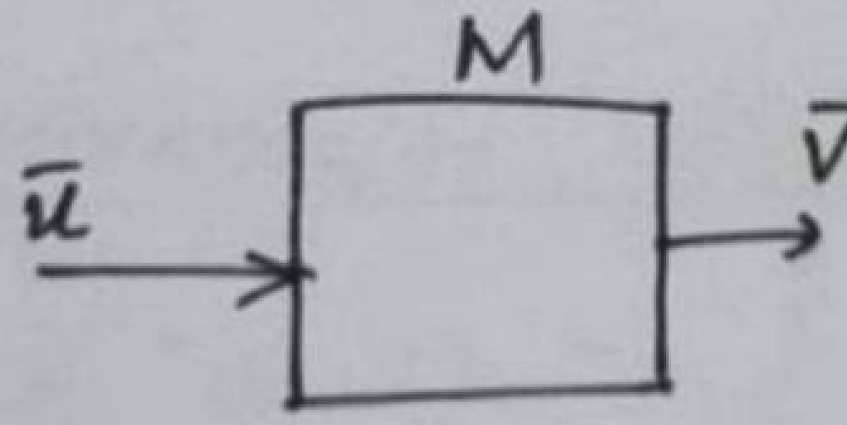
Scanned by TapScanner

Here, g is general non-linear matrix like operator so we can represent.

$$\boxed{\bar{V} = M(\bar{u})}$$

The operator M has different forms for different memory models.
The algorithm which computes 'M' is known as the <u>recording</u> or <u>storage</u> Algo.
In most of the cases 'M' is computed using I/p patten vecter based on principle of recall associative memory.
There are Two models.
① static ② Dynamic.

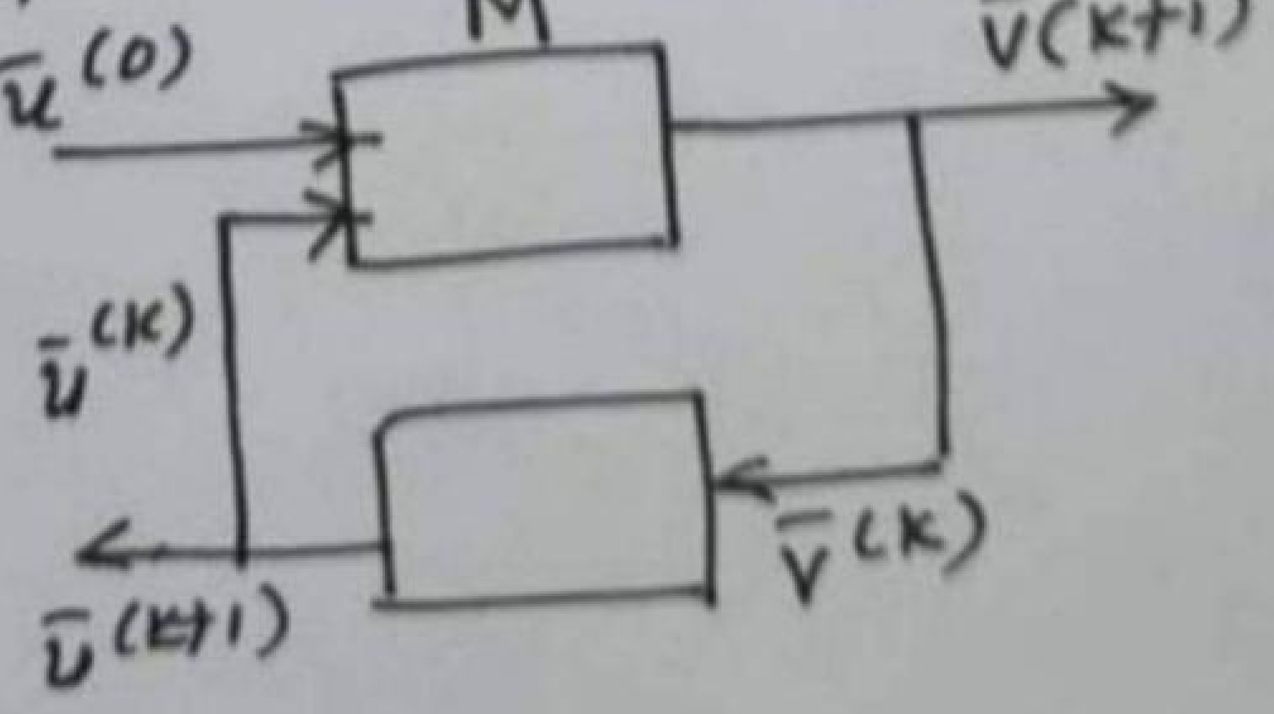
<u>Static</u> :- Static model/N/w recalls the o/p if I/p is given in 1 forward pas.
(Non-reccurrent)



$$\boxed{\bar{V} = M(\bar{u})}$$

<u>Dynamic</u> :- Dynamic n/w recalls the pattern through I/p feed back mechanism which takes time.
(Reccurrent)

$$\boxed{\begin{aligned} \bar{V}^{(k)} &= M(\bar{u}^{(k)}) \\ \bar{V}^{(k+1)} &= M'(\bar{u}^{(k)}, \bar{v}^{(k)}) \end{aligned}}$$



Scanned by TapScanner

Autocorrelators:- (Amari)

The first order Auto-correlator obtain its connection matrix, which is association of patten with itself, by multiplying a patten element with every other patten element.

$$\boxed{T = \sum_{i=1}^{n} [A_i]^T \cdot [A_i]}$$

Here $T = [t_{ij}]$ is $(p \times q)$ connection matrix

$A_i \in \{-1, 1\}^p$.

⇒ The autocorrelation recall equation is a vector matrix multiplication followed by a point wise non-linear threshold operation.

The recall eq^n is given as.

$$\boxed{a_j^{new} = f(\underbrace{a_i \cdot t_{ij}}_{\alpha}, \underbrace{a_j^{old}}_{\beta})}$$

where $A_i = (a_1, a_2, a_3 \cdots a_p)$

The Two parameteter bipolar threshold fu^n is

$$\boxed{f(\alpha, \beta) = \begin{cases} 1, & \text{if } \alpha > 0 \\ \beta, & \text{if } \alpha = 0 \\ -1, & \text{if } \alpha < 0 \end{cases}}$$

Scanned by TapScanner

Ex:- Consider a first order auto correlation having stored pattens -

$$A_1 = (-1, 1, -1, 1)$$
$$A_2 = (1, 1, 1, -1)$$
$$A_3 = (-1, -1, -1, 1)$$

If following pattens

$$A' = (1, 1, 1, -1) \text{ \&}$$
$$A'' = (1, 1, 1, 1)$$ given as input, then check which patten will recalled.

⇒ Find connection matrix as

$$T = T_1 + T_2 + T_3$$

$$T_1 = [A_1]^T \cdot [A_1] = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \cdot [-1, 1, -1, 1] = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

$$T_2 = [A_2]^T \cdot [A_2] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \cdot [1\ 1\ 1\ -1] = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

$$T_3 = [A_3]^T \cdot [A_3] = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \cdot [-1\ -1\ -1\ 1] = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

$$\therefore T = T_1 + T_2 + T_3 = \begin{bmatrix} 3 & 1 & 3 & -3 \\ 1 & 3 & 1 & -1 \\ 3 & 1 & 3 & -3 \\ -3 & -1 & -3 & 3 \end{bmatrix}$$

Scanned by TapScanner

Recognation of stored pattern:-

for i/p pattern $A' = (1, 1, 1, -1)$

$$T = \begin{bmatrix} 3 & 1 & 3 & -3 \\ 1 & 3 & 1 & -1 \\ 3 & 1 & 3 & -3 \\ -3 & -1 & -3 & 3 \end{bmatrix}$$

$$a_j^{new} = f[(A_{j1}, A_{j2}, A_{j3}, A_{j4}), a_j^{old})$$

$$a_{j1}^{new} = f(3+1+3+3, 1)$$

$$= f(10, 1)$$

since $\alpha > 0$

$$\therefore \boxed{a_{j1}^{new} = 1}$$

$$a_{j2}^{new} = f(1+3+1+1, 1)$$

$$= f(6, 1) \quad \because \alpha > 0$$

$$\boxed{a_{j2}^{new} = 1}$$

$$a_{j3}^{new} = f(3+1+3+3, 1)$$

$$= f(10, 1) \quad \because \alpha > 0$$

$$\boxed{a_{j3}^{new} = 1}$$

$$a_{j4}^{new} = f(-3-1-3-3, -1)$$

$$= f(-10, -1)$$

$$= -1 \quad \because \alpha < 0$$

$$a_j^{new} = (1, 1, 1, -1)$$

Since

$a_j^{new} = A2$

A2 will be recaled for A' as input

32

Scanned by TapScanner

Scanned by TapScanner

Recognation of Noisy patten

When $A'' = (1, 1, 1, 1)$ is distorted patten given as input to model.

$$A_j^{new} = (A_{j1}, A_{j2}, A_{j3}, A_{j4})$$

$A_{j1}^{new} = f(3+1+3-3, 1)$
$= f(4, 1)$
$= 1$ (since $\alpha > 0$)

$A_{j2}^{new} = f(1+3+1-1)$
$= f(4, 1)$
$= 1$

$A_{j3}^{new} = f(3+1+3-3, 1)$
$= f(4, 1)$
$= 1$

$A_{j4}^{new} = f(-3-1-3+3, 1)$
$= f(-4, 1)$
$= -1$ since $\alpha < 0$

$\therefore$ $a_j^{new} = A2$ ie, Stored patten A_2 will be recalled if $A'' = (1, 1, 1, 1)$ is given

—— x —— x —— x ——

Another method (Using Hamming distance)

$$HD(X, Y) = \sum_{i=1}^{n} |x_i - y_i|$$

Thus HD of A' from each stored patten is

$HD(A', A1) = [[1-(-1)] + (1-1) + [1-(-1)] + (1-1)] = 4$
$HD(A', A2) = 2$
$HD(A', A3) = 6$

since $HD(A', A2) = 2$ it is very closed hence $A2$ will be recalled

44

Scanned by TapScanner

HW1) $A1 = (1,1,1,-1)$

$A2 = (-1,-1,-1,1)$

$A3 = (1,-1,1,-1)$

$A4 = (-1,1,-1,1)$

$A' = (1,1,1,-1)$

$A'' = (1,-1,1,1)$

HW2) validate your result with HD method.

8

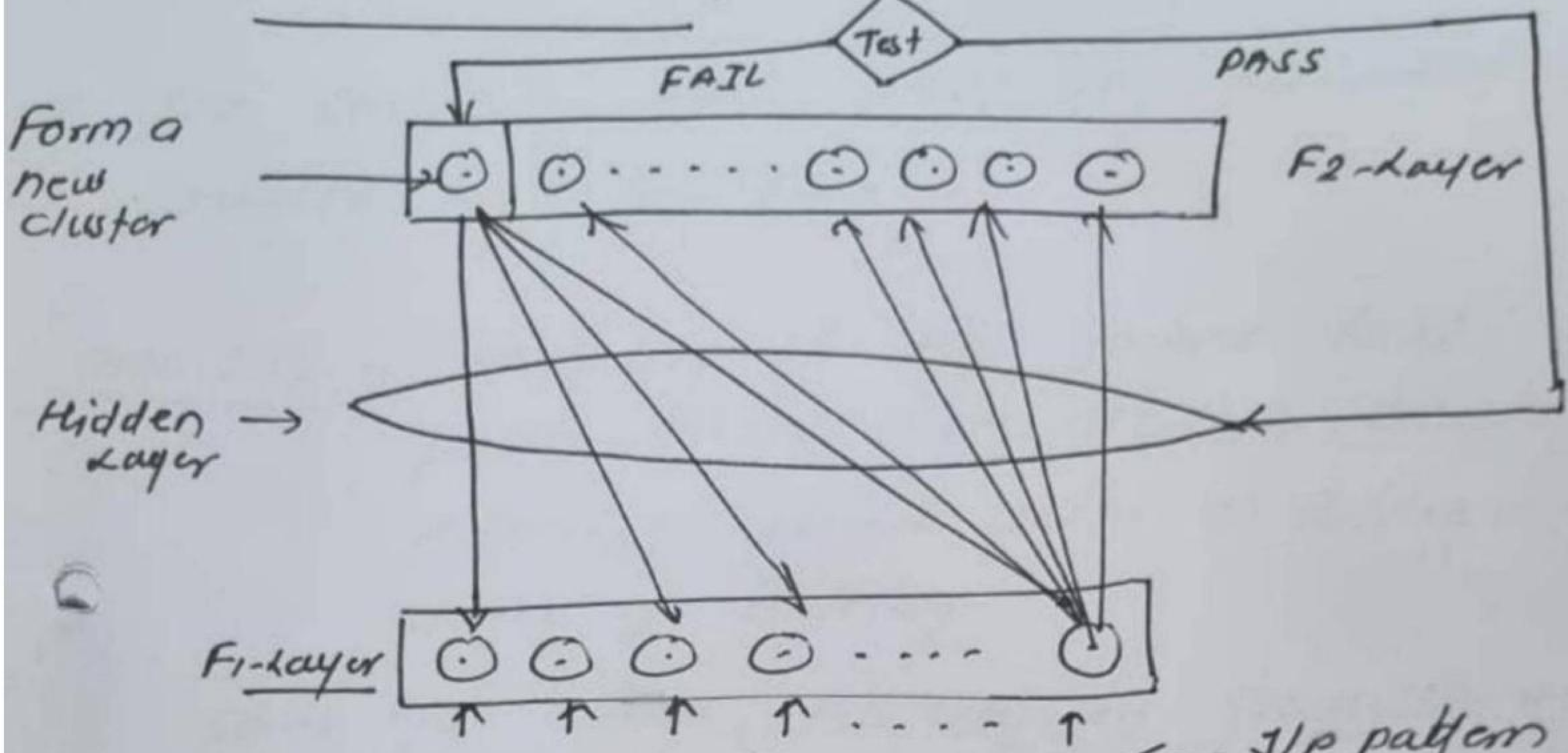
24

Scanned by TapScanner

Scanned by TapScanner

Adaptive Resonance Theory (Grossberg & carpenter)

(ART1 Network)



Simplified ART N/W :-

→ ART1 N/W is used for clustering binary vectors.

→ we can present I/p pattern in any order.
When a pattern is presented an appropriate cluster is choosen and cluster weights are adjusted to the cluster unit to learn the patten.

→ The weights on the cluster unit can be considered as exampler for pattern & placed in cluster.

→ This N/W works in unsupervised mode.
One the N/W is trained we can present training patten several times.

→ The patten may be placed on one cluster unit for the 1st time & then on different cluster unit when it is presented later due to change in weights.

Scanned by TapScanner

→ Thus the patten oscillating among different cluster units at different stages of training.

→ The stability and plasticity dilemma was proposed by Grossberg.

<u>stability</u>:- stability of n/w means that a patten should not oscillate among different cluster units at different stages of training.

some n/w active stability by gradually reducing the learning rate.

<u>plasticity</u>:- It is ability of the n/w to respond to learn new pattern equally at any stage of learning.

—x—

Scanned by TapScanner

1

Fuzzy set:-

Defn:- A fuzzy set can be defined as -

If X is a universal set and x is a particular element of X then fuzzy set $\tilde{A}$ is defined on X must be represented a collection of ordered pairs

$$\boxed{\tilde{A} = \{(x, \mu_{\tilde{A}}(x)),\ x \in X\}}$$

Where each pair $(x, \mu_A(x))$ is a singleton.

In discrit form:-

$$\tilde{A} = \sum_{x_i \in X} \mu_{\tilde{A}}(x_i) \,/\, x_i$$ - using Zadeh notation

In continious form:-

$$\tilde{A} = \int_x \mu_{\tilde{A}}(x) \,/\, x$$

Using Zadeh notation

$$\boxed{\tilde{A} = \sum_{x_i \in X} \frac{\mu_{\tilde{A}}(x_i)}{x_i}}$$

Ex:- Condider a universal set of students X

$$X = \{S1, S2, S3, S4, S5\}$$

Let $\tilde{A}$ is a fuzzy set of 'smart' student. here 'smart' is fuzzy linguistic term.

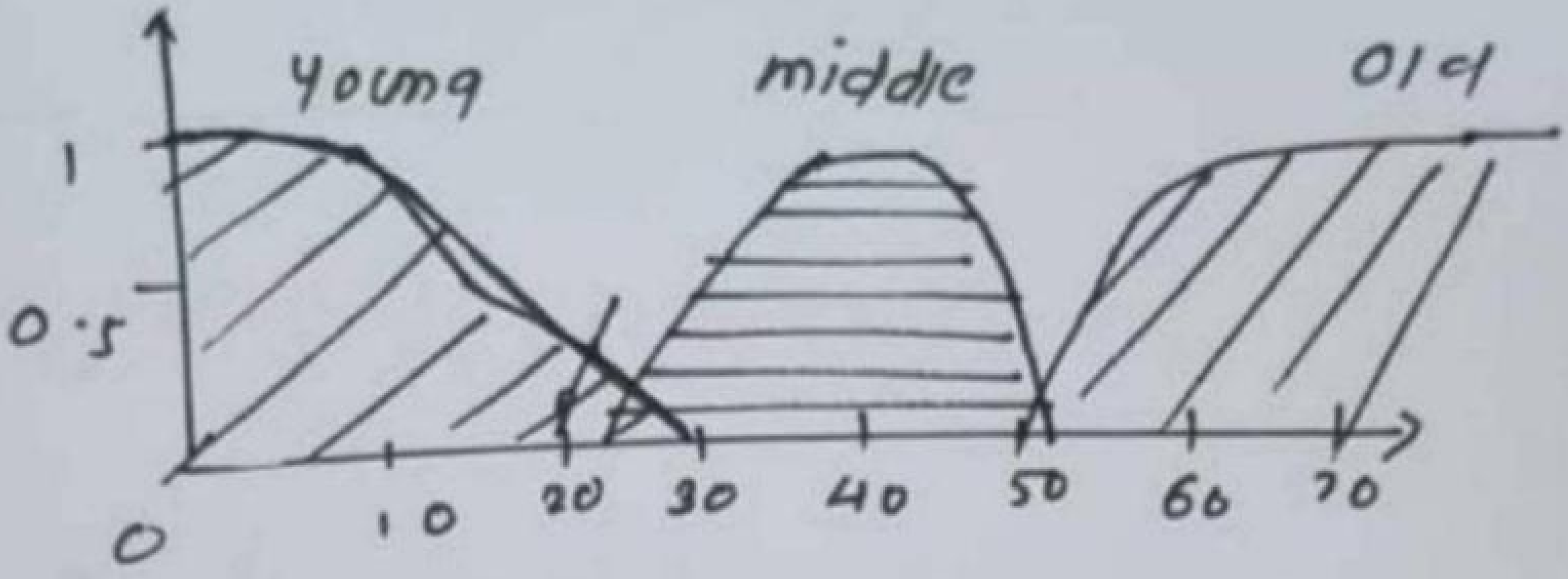
then, $\tilde{A} = \{(S1, 0.5), (S2, 0.7), (S3, 1) (S4, 0), (S5, 0.6)\}$

Here, $\tilde{A}$ indicates that smartness of S1 is 0.5.

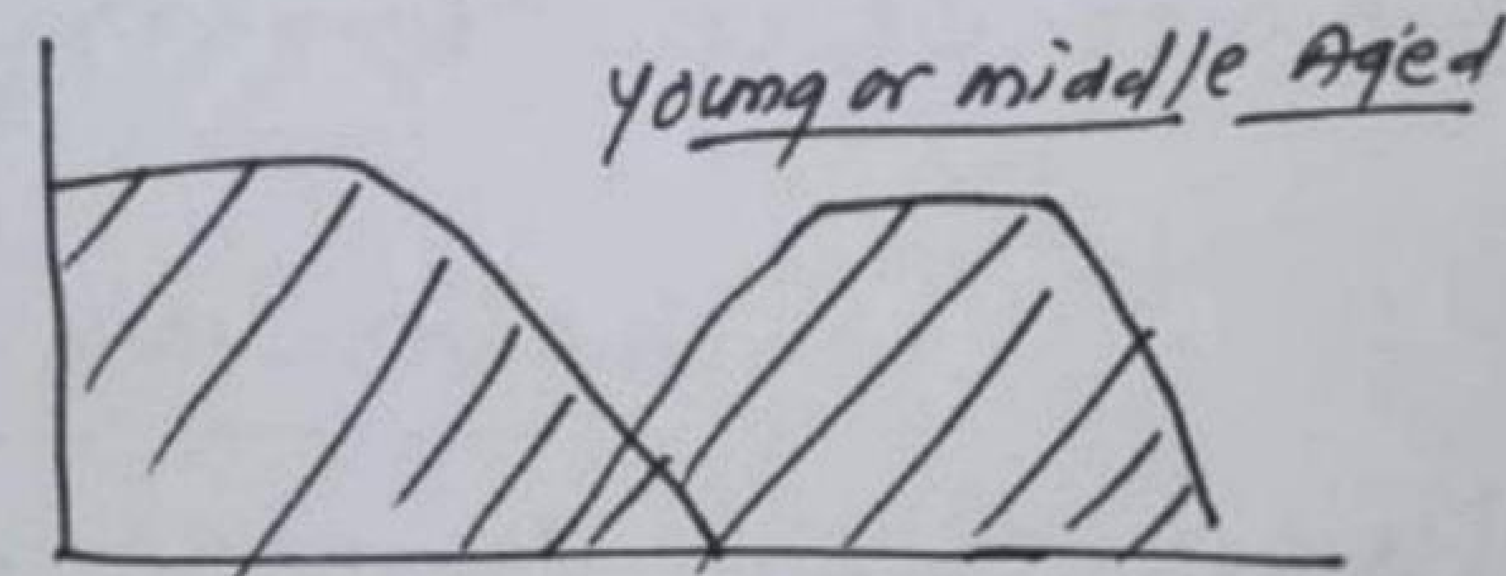
Scanned by TapScanner

Example of operations on FS:-

Consider an example of FS representing young, middle age, old age as.



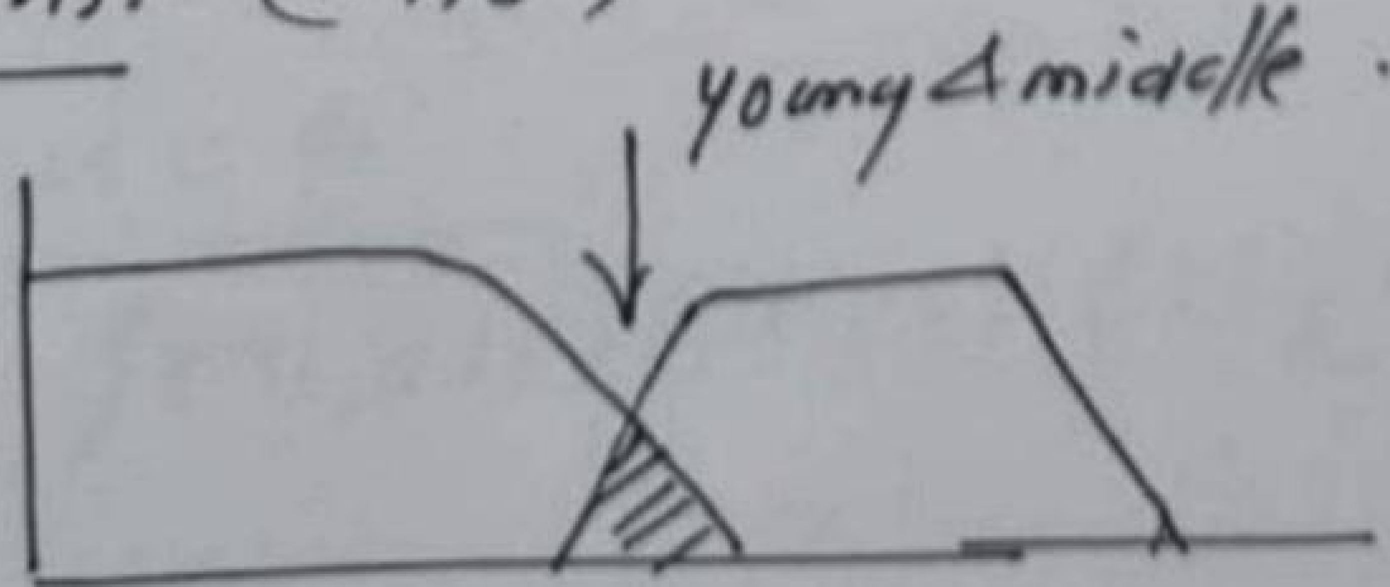
① **Union**:- Let $\tilde{A}$ be the set of young and $\tilde{B}$ is set of middle aged. Thus $\tilde{A} \cup \tilde{B}$ represents the fuzzy set of "young or middle Age"



If $\tilde{A} = \{(x_1, 0.5), (x_2, 0.7), (x_3, 0)\}$
$\tilde{B} = \{(x_1, 0.8), (x_2, 0.2), (x_3, 1)\}$
$\mu_{\tilde{A} \cup \tilde{B}} = \max(\mu_{\tilde{A}}, \mu_{\tilde{B}})$

Then $\tilde{A} \cup \tilde{B} = \{(x_1, 0.8), (x_2, 0.7), (x_3, 1)\}$

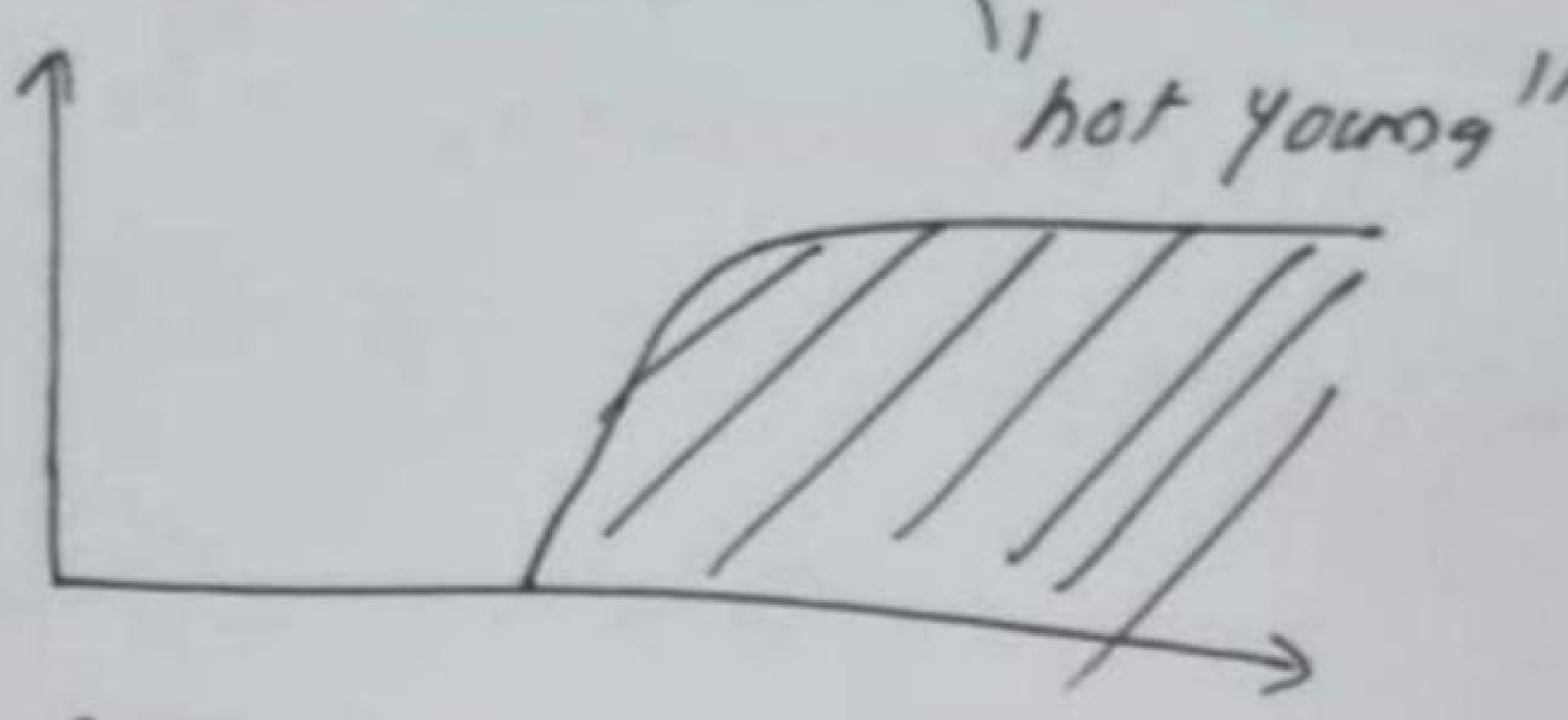
② **Intersection**:- $(\tilde{A} \cap \tilde{B})$



$\tilde{A} \cap \tilde{B} = \{(x_1, 0.5), (x_2, 0.2), (x_3, 0)\}$.

Scanned by TapScanner

③ Compliment :($\tilde{A}^c$)



$$\tilde{A}^c = \{(x_1, 0.5), (x_2, 0.4), (x_3, 1)\}$$

④ Product of FS :- $\tilde{A}\cdot\tilde{B}$

If $\tilde{A} = \{(x_1, 0.2), (x_2, 0.8), (x_3, 0.4)\}$

$\tilde{B} = \{(x_1, 0.4), (x_2, 0), (x_3, 0.1)\}$

$\therefore\ \tilde{A}\cdot\tilde{B} = \{(x_1, 0.08), (x_2, 0), (x_3, 0.04)\}$

⑤ product of FS with crisp no. :-

If $\tilde{A} = \{(x_1, 0.4), (x_2, 0.6) (x_3, 0.8)\}$

and $a = 0.3$

$\therefore\ a\cdot\tilde{A} = \{(x_1, 0.12), (x_2, 0.18), (x_3, 0.24)\}.$

⑥ Power of FS :- $\tilde{A}^{\alpha}$

If $\tilde{A} = \{(x_1, 0.4) (x_2, 0.2) (x_3, 0.7)\}$

& $\alpha = 2$

$\therefore\ \tilde{A}^2 = \{(x_1, 0.16), (x_2, 0.04), (x_3, 0.49)\}.$

Scanned by TapScanner

(7) **Difference** :- $(-)$, $(\,|\,)$

If $\tilde{A} = \{(x_1, 0.2), (x_2, 0.5) (x_3, 0.6)\}$

$\tilde{B} = \{(x_1, 0.1), (x_2, 0.4) (x_3, 0.5)\}$

$\underline{\tilde{A} - \tilde{B} = (\tilde{A} \cap \tilde{B}^c)}$

$\tilde{B}^c = \{(x_1, 0.9), (x_2, 0.6), (x_3, 0.5)$.

$\tilde{A} - \tilde{B} = \{(x_1, 0.2), (x_2, 0.5), (x_3, 0.5)\}$.

(8) **Disjunctive sum** :- $\tilde{A} \oplus \tilde{B}$

$\tilde{A} \oplus \tilde{B} = (\tilde{A}^c \cap \tilde{B}) \cup (\tilde{A} \cap \tilde{B}^c)$

$\tilde{A}^c = \{(x_1, 0.8), (x_2, 0.5), (x_3, 0.4)$

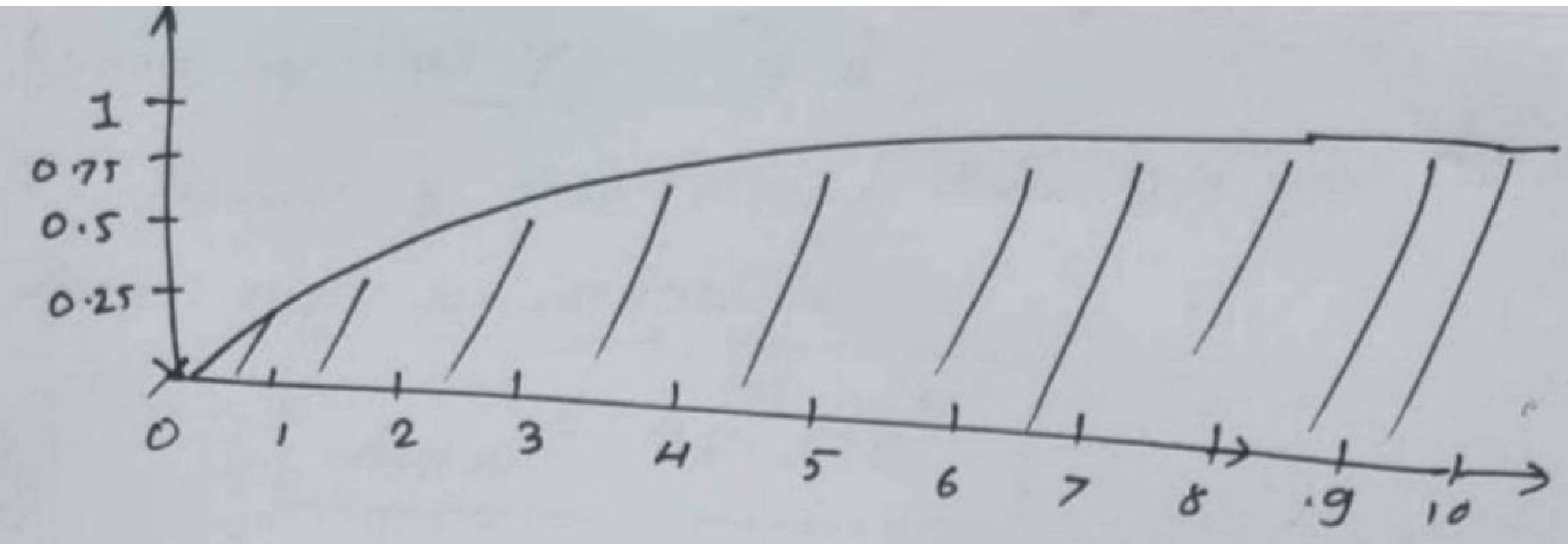
$\tilde{B}^c = \{(x_1, 0.9), (x_2, 0.6) (x_3, 0.5)\}$.

$\therefore (\tilde{A}^c \cap \tilde{B}) = \{(x_1, 0.1), (x_2, 0.4), (x_3, 0.4)\}$

$(\tilde{A} \cap \tilde{B}^c) = \{(x_1, 0.2), (x_2, 0.5) (x_3, 0.5)\}$

$\therefore \underline{\tilde{A} \oplus \tilde{B} = \{(x_1, 0.2), (x_2, 0.5) (x_3, 0.5)}$

Scanned by TapScanner



—o—

Operations on fuzzy set:-

Given X be a universal set and $\tilde{A}$ & $\tilde{B}$ be the two fuzzy sets defined on X with $\mu_{\tilde{A}}(x)$ & $\mu_{\tilde{B}}(x)$ as their membership fun.

① **Union**: $(\tilde{A} \cup \tilde{B})$

The union of two fuzzy sets $\tilde{A}$ & $\tilde{B}$ is a new fuzzy set $\tilde{A} \cup \tilde{B}$ on X with membership fun.

$$\boxed{\mu_{\tilde{A} \cup \tilde{B}}(x) = \max\left(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\right)}$$

② **Intersection**: $(\tilde{A} \cap \tilde{B})$

The intersection of two fuzzy sets $\tilde{A}$ & $\tilde{B}$ is a new fuzzy set $\tilde{A} \cap \tilde{B}$ on X with membership fun.

$$\boxed{\mu_{\tilde{A} \cap \tilde{B}}(x) = \min\left(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\right)}$$

③ **Compliment**: $(\tilde{A}^c)$

The compliment of FS $\tilde{A}$ is a new FS $\tilde{A}^c$ with membership fun.

$$\boxed{\mu_{\tilde{A}^c}(x) = 1 - \left(\mu_{\tilde{A}}(x)\right)}$$

Scanned by TapScanner

4) Product of two FS :- $(\tilde{A}\cdot\tilde{B})$

The product of two FS $\tilde{A}$ & $\tilde{B}$ is a new FS $\tilde{A}\cdot\tilde{B}$ whose mF is defined as.

$$\boxed{\mu_{\tilde{A}\cdot B}^{(x)} = \mu_{\tilde{A}}^{(x)}\cdot\mu_{\tilde{B}}^{(x)}}$$

5) Product of FS with crisp no. $(a.\tilde{A})$

multiplying a FS $\tilde{A}$ by a, a crisp no. results in a new FS $a.\tilde{A}$ with mF.

$$\boxed{\mu_{a.\tilde{A}}^{(x)} = a.\mu_{\tilde{A}}^{(x)}}$$

6) power of FS :- $(\tilde{A}^{\alpha})$

The α power of FS $\tilde{A}$ is a new FS $\tilde{A}^{\alpha}$ whose mF is given as

$$\boxed{\mu_{\tilde{A}^{\alpha}}^{(x)} = \left(\mu_{\tilde{A}}^{(x)}\right)^{\alpha}}$$

7) Difference :- $(-)$ or $(\,|\,)$

The difference of two FS $\tilde{A}$ & $\tilde{B}$ is a new FS $\tilde{A}-\tilde{B}$ whose mF is given as

$$\boxed{\tilde{A}-\tilde{B} = (\tilde{A}\cap\tilde{B}^{c})}$$

8) Disjuction / Disjunctive sum : $\oplus$

The disjuctive sum of two FS $\tilde{A}$ & $\tilde{B}$ is a new FS $\tilde{A}\oplus\tilde{B}$ defined as.

$$\boxed{\tilde{A}\oplus\tilde{B} = (\tilde{A}^{c}\cap\tilde{B})\cup(\tilde{A}\cap\tilde{B}^{c})}$$

Scanned by TapScanner

* α-cuts & strong α-cuts of FS :- ①

α-cut and strong α-cuts are the properties of FS, which can be defined as follows -

i) α-cuts :- α-cut is a new crisp set defined on fuzzy set $\tilde{A}$ over a universe of discourse X as

$$\boxed{A^{\alpha} = \{x \mid A(x) \geq \alpha\}}$$

ii) Strong α-cuts :- strong α-cuts is a new crisp set defined on a FS $\tilde{A}$ over a universal set X as

$$\boxed{A^{\alpha+} = \{x \mid A(x) > \alpha\}}$$

Ex :- Find the α-cuts & strong α-cuts for $\alpha = 0.5$ of set $\tilde{A}$

$$\tilde{A} = \left\{ \frac{0.2}{1} + \frac{0.3}{2} + \frac{0.4}{3} + \frac{0.5}{6} + \frac{0.7}{7} + \frac{0.8}{8} + \frac{0.9}{10} \right\}$$

$$\Rightarrow \tilde{A} = \{(1, 0.2), (2, 0.3), (3, 0.4), (6, 0.5), (7, 0.7), (8, 0.8), (10, 0.9)\}$$

α-cuts :-

i) For $\alpha = 0.2$

$$A^{0.2} = \left\{ \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{10} \right\}$$

$$A^{0.2} = \{1, 2, 3, 6, 7, 8, 10\}$$

ii) For $\alpha = 0.3$

$$A^{0.3} = \{2, 3, 6, 7, 8, 10\}.$$

Scanned by TapScanner

iii) For $\alpha = 0.4$

$A^{0.4} = \{3, 6, 7, 8, 10\}$

iv) For $\alpha = 0.5$

$A^{0.5} = \{6, 7, 8, 10\}$

v) For $\alpha = 0.7$

$A^{0.7} = \{7, 8, 10\}$

vi) For $\alpha = 0.8$

$A^{0.8} = \{8, 10\}$

vii) For $\alpha = 0.9$

$A^{0.9} = \{10\}$

———x———

Strong α-cuts:-

i) For $\alpha = 0.2$

$A^{+0.2} = \{2, 3, 6, 7, 8, 10\}$

ii) For $\alpha = 0.3$

$A^{+0.3} = \{3, 6, 7, 8, 10\}$

iii) For $\alpha = 0.4$ – $A^{+0.4} = \{6, 7, 8, 10\}$

iv) For $\alpha = 0.5$ – $A^{+0.5} = \{7, 8, 10\}$

v) For $\alpha = 0.8$ – $A^{+0.8} = \{10\}$

vi) For $\alpha = 0.9$ – $A^{+0.9} = \{\phi\}$.

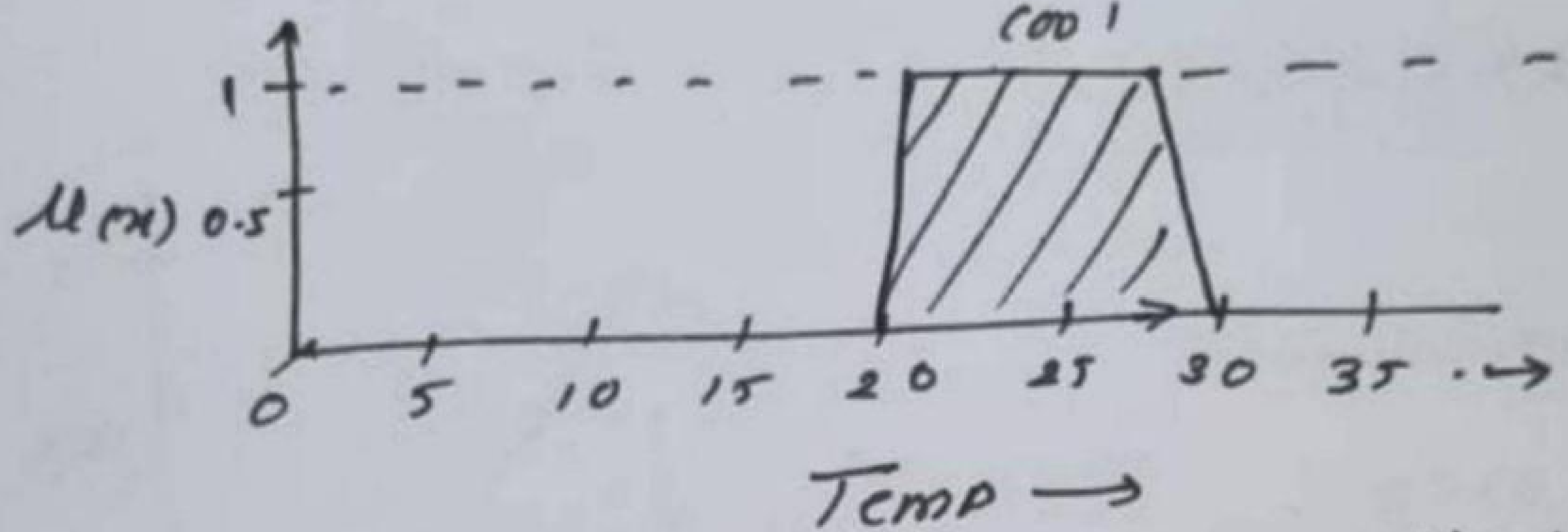
———x———

Scanned by TapScanner

membership function:-

The membership fuⁿ values always described by discrit values. and by continious function.

The fuzzy membership fuⁿ for fuzzy linguistic term "cool" related to temp. may represented as

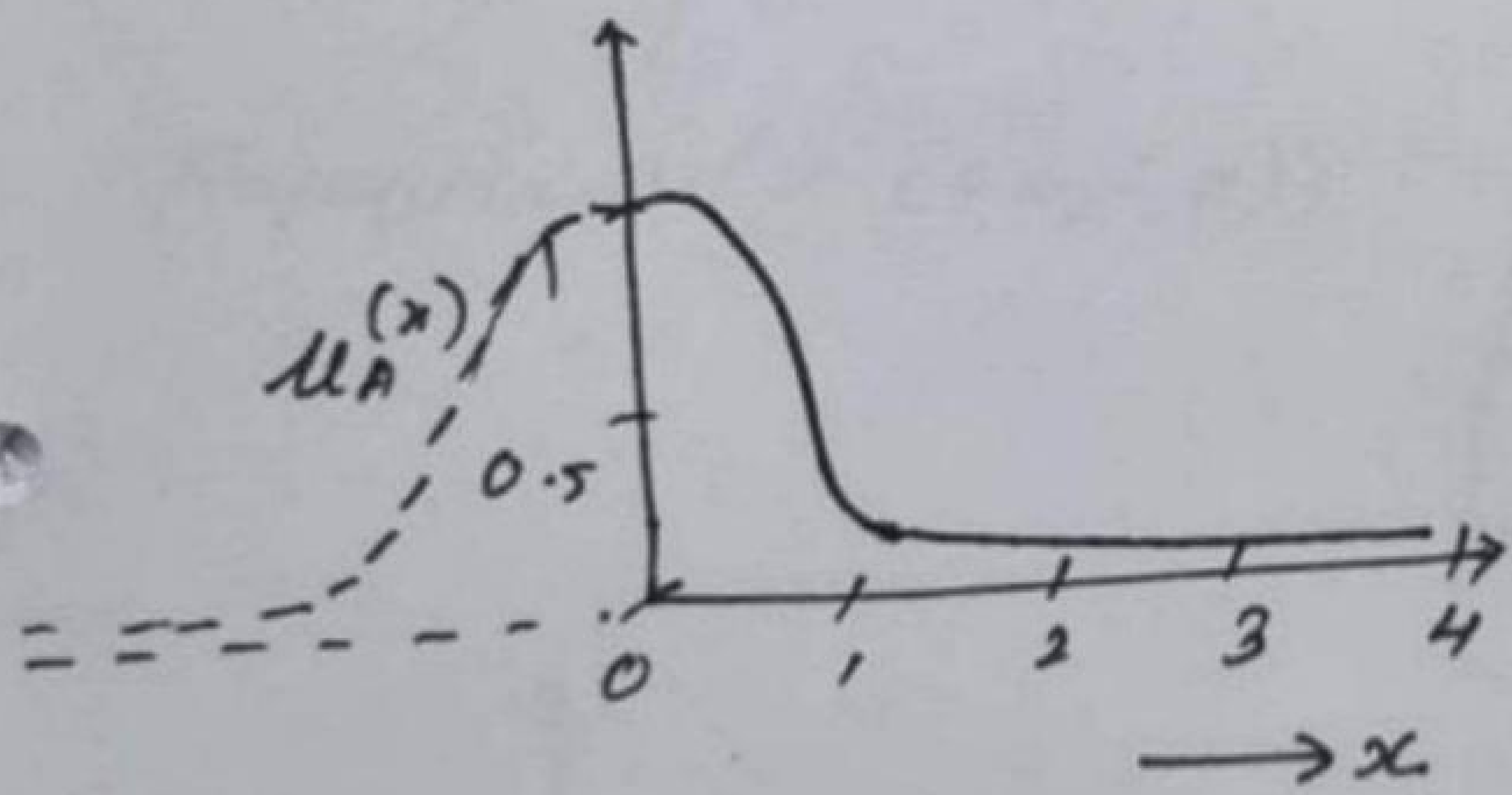


A MF can be also given mathematically.

Ex:- $\boxed{\mu_A^{(x)} = \frac{1}{1+x^2}}$

For $X = \{0, 1, 2, 3, 4\}$

x	$\mu_A^{(x)}$
0	1
1	0.25
2	1/9
3	1/16
4	1/25



Que:- Construct and plot fuzzy set, defined by function $f(x) = x/(x+2)$ on universe of discourse. $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

X	0	1	2	3	4	5	6	7	8	9	10
f(x)	0	1/3 0.33	2/4 0.5	3/5 0.6	4/6 0.66	5/7 0.71	6/8 0.75	7/9 0.77	8/10 0.8	9/11 0.81	10/12 0.82

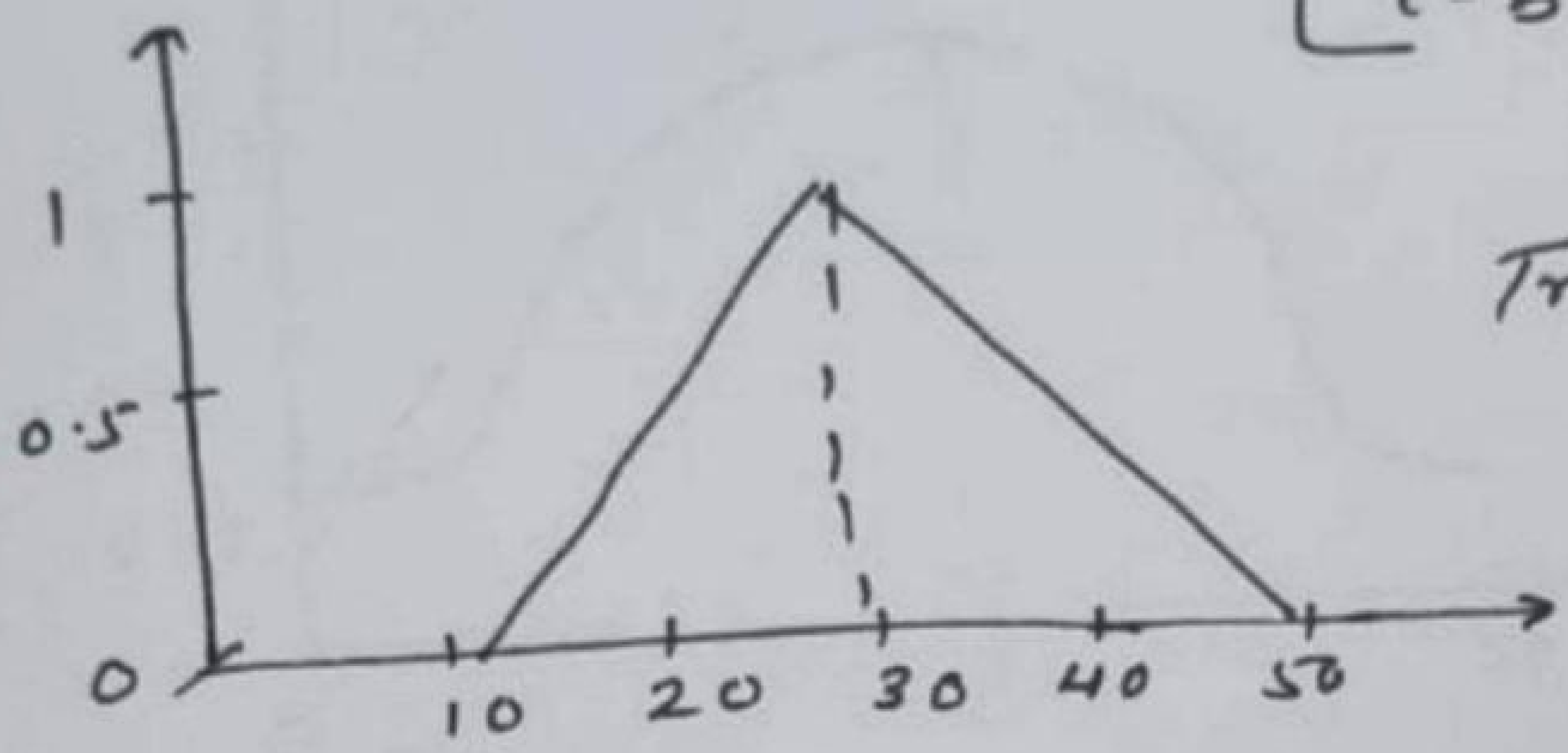
Scanned by TapScanner

Types of membership fu^n:-

① Tringular MF :-

specified by 3 - parameters - $\{a, b, c\}$.

$$\text{tringular}(x, (a,b,c)) = \begin{cases} 0 - x \le a \\ \frac{x-a}{b-a} - a \le x \le b \\ \frac{c-x}{c-b} - b \le x \le c \end{cases}$$



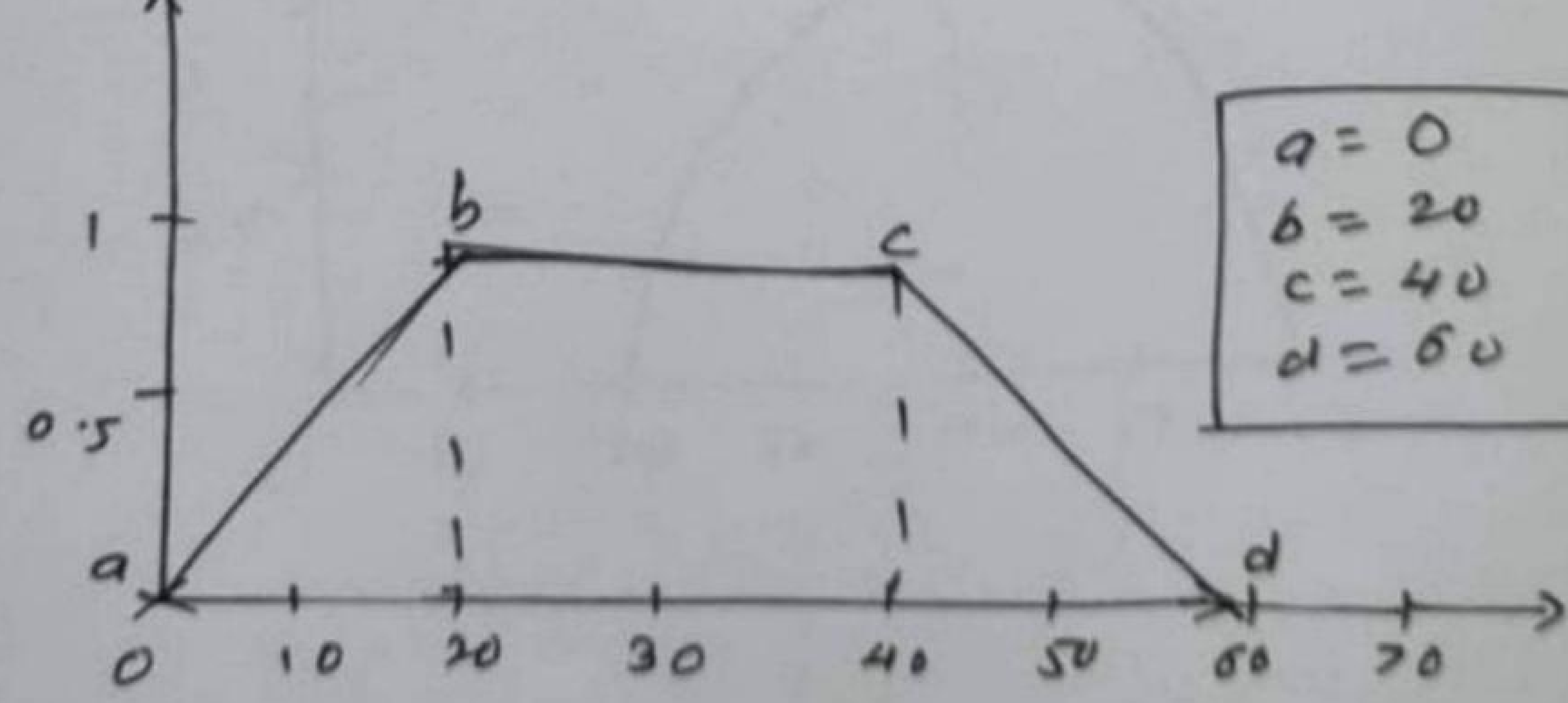
Tringular MF = (x, 10, 30, 50)

a = 10
b = 30
c = 50

(ii) Trapazoidal MF :-

Specified by 4 parameters $\{a, b, c, d\}$.

$$\text{Trapazoidal}(x, (a,b,c,d)) = \begin{cases} 0 - x \le 0 \\ \frac{x-a}{b-a} - a \le x \le b \\ 1 - b \le x \le c \\ \frac{d-x}{d-c} - c \le x \le d \\ 0 - d \le 0 \end{cases}$$



a = 0
b = 20
c = 40
d = 60

Scanned by TapScanner

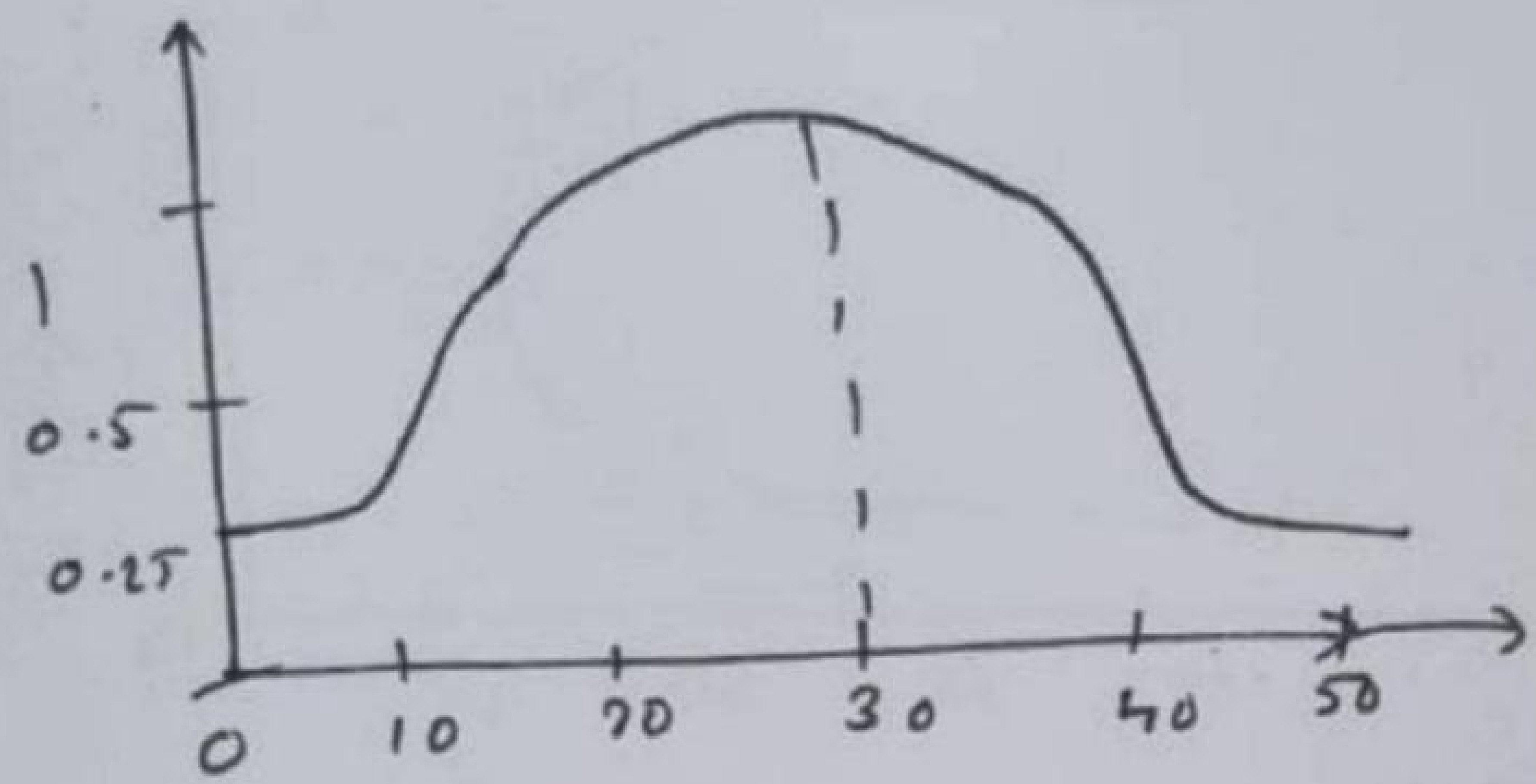
③ Gaussian mf :-

Determined by two parameters (c, a)

$$Gaussian\ (x, c, a) = e^{\frac{1}{2} \cdot \left(\frac{x-c}{a}\right)^2}$$

where c - mf Centre

a - mf width.

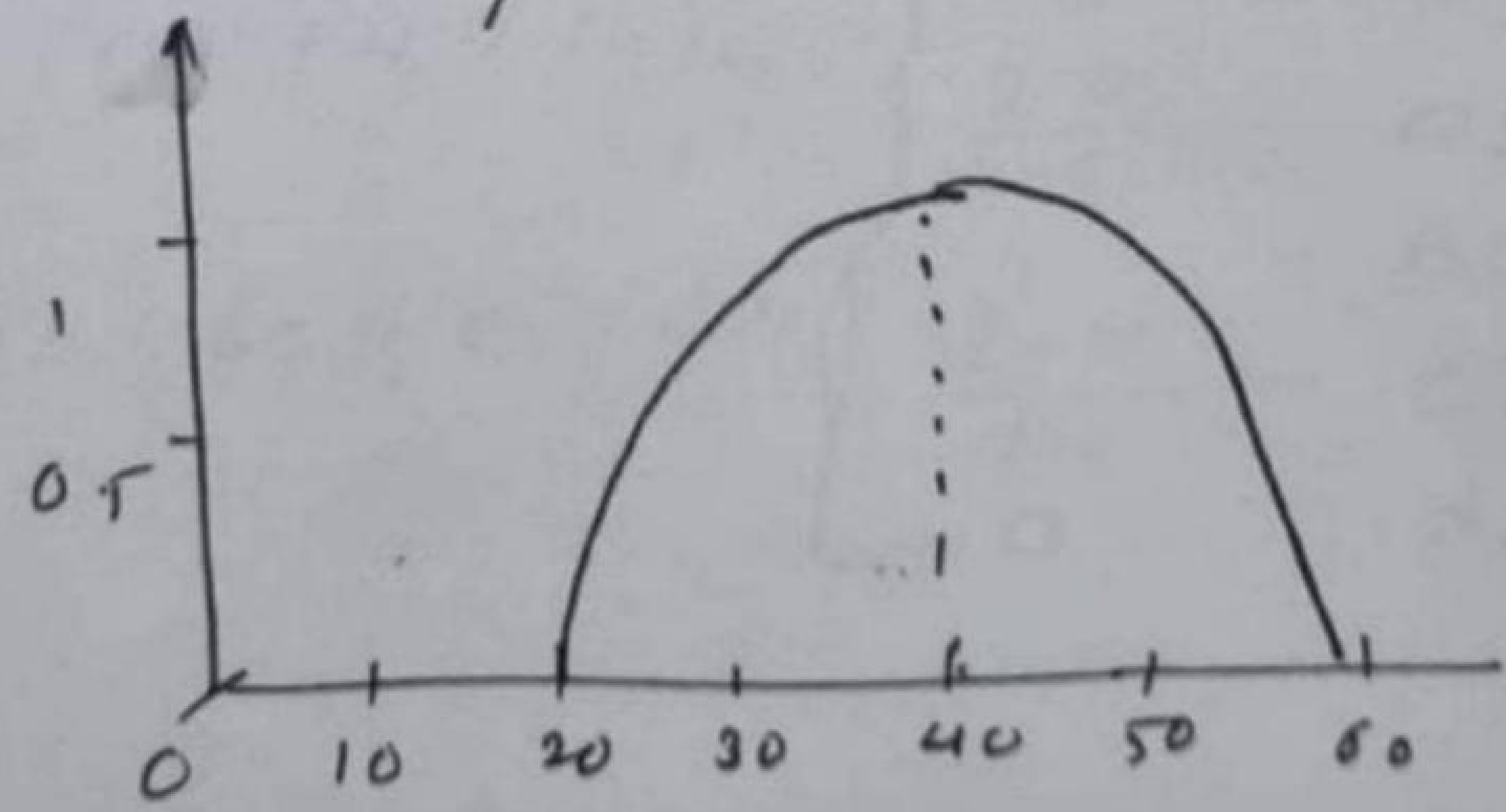


④ Generalised Bell shape mf :-

Defined by 3-parameters. (a, b, c)

$$Bell\ (x, a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right| \cdot 2b}$$

where, b is usually positive

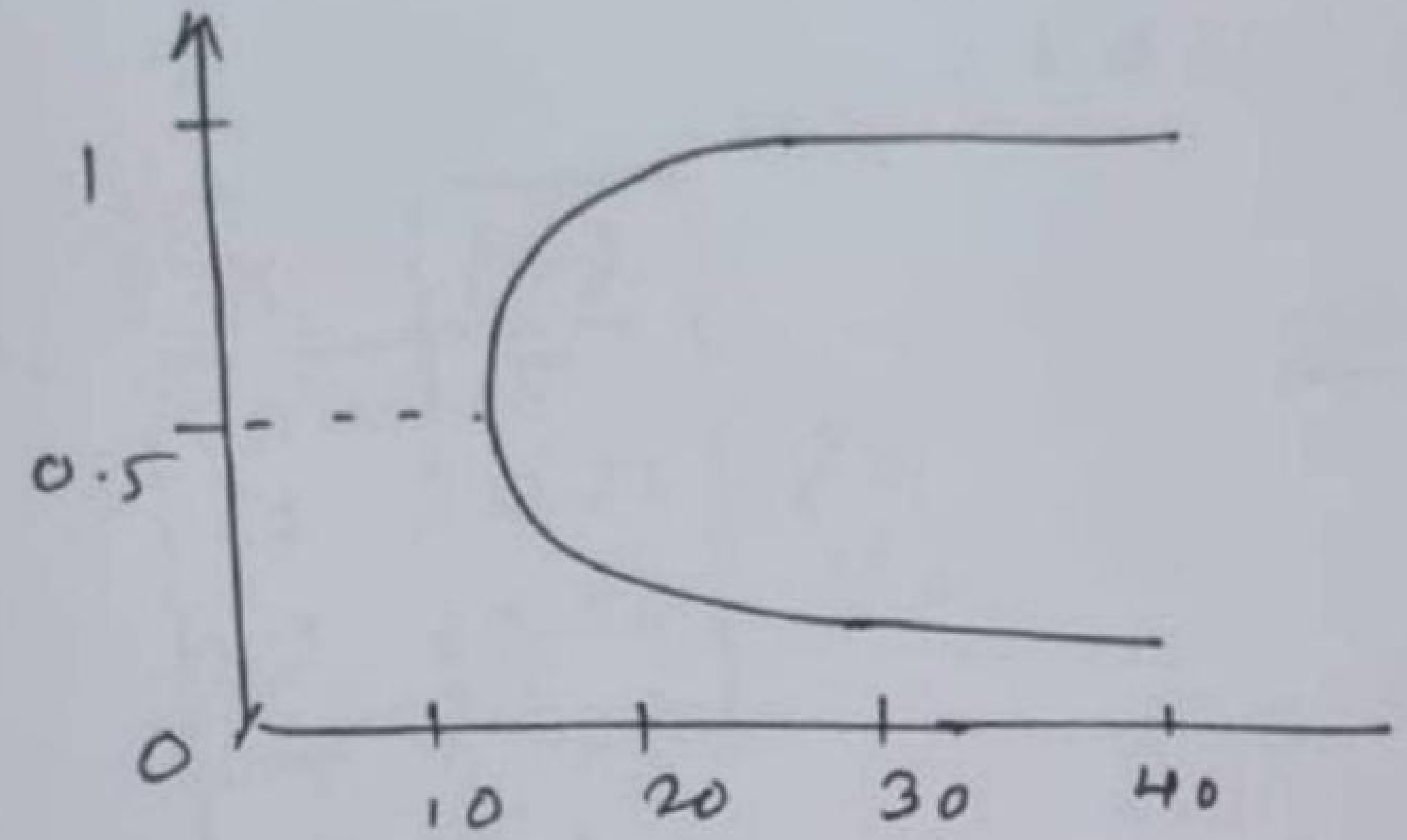


Scanned by TapScanner

5 Sigmoidal MF :-

defined by 3 parameters (a, b, c) as

$$\text{Sigmoidal}(x, a, b, c) = \frac{1}{1 + \exp(-a(x-c))}$$

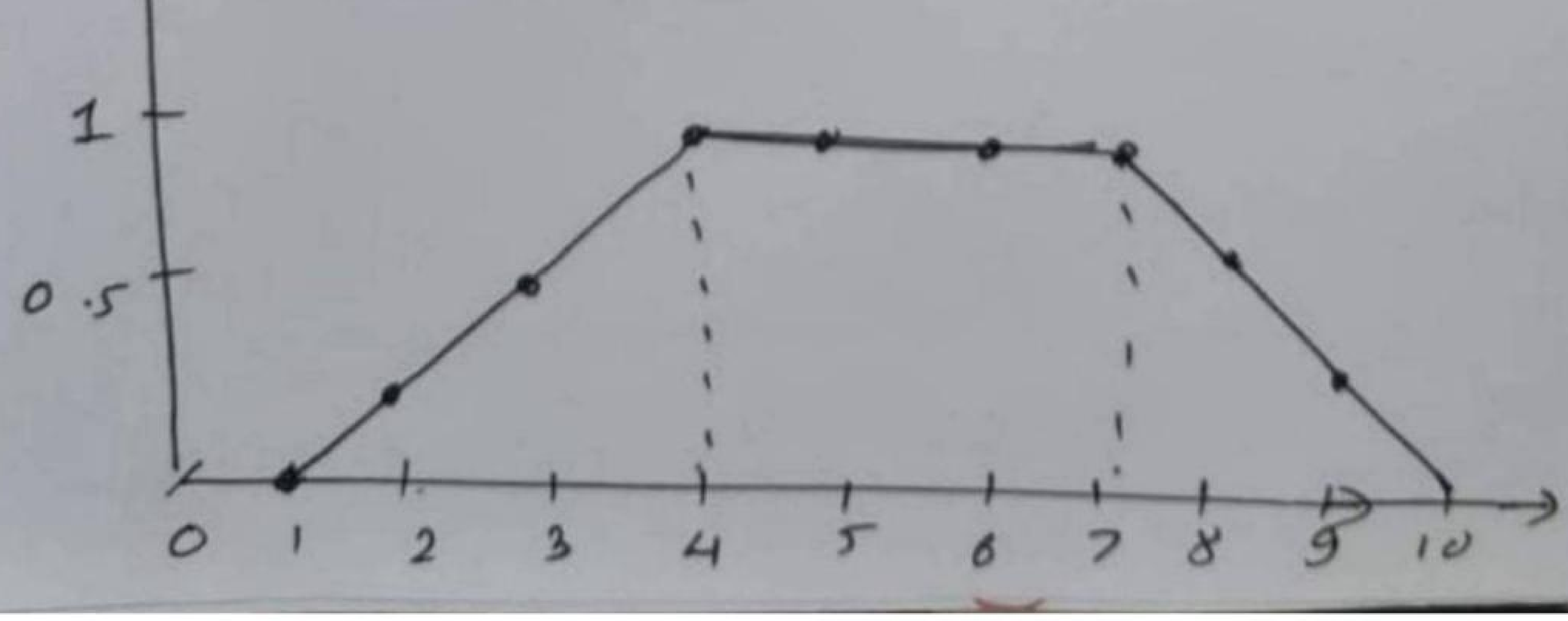


— x —

Numerical :- construct a FS defined by fn $f(x : a, b, c, d)$ given below on universe of discourse $X = \{1, 2, 3, \ldots 10\}$.

$$f(x : 1, 4, 7, 10) = \begin{cases} 0 & - \; x \le a \\ \frac{x-a}{b-a} & - \; a \le x \le b \\ 1 & - \; b \le x \le c \\ \frac{d-x}{d-c} & - \; c \le x \le d \\ 0 & - \; d \le x . \end{cases}$$

$\Rightarrow$ Here $a = 1$, $b = 4$, $c = 7$, $d = 10$



Scanned by TapScanner

Let $\tilde{A}$ is Fs defined by fun.

$f(x : a, b, c, d)$

then

$\tilde{A} = \{(1,0), (2,0.33), (3,0.66), (4,1) (5,1) (6,1) (7,1)$
$(8,0.66), (9,0.33), (10,0)\}.$

x	f(x)	
1	0	
2	0.33	1/3
3	0.66	2/3
4	1	
5	1	
6	1	
7	1	
8	0.66	2/3
9	0.33	1/3
10	0	

using Formula

Scanned by TapScanner

Crisp Relations:-

Cartesian product :- $(A \times B)$

The cartesian product of two crisp sets A & B denoted by $A \times B$ is the set of all ordered pairs such that, the first element in the pair belongs to set A and second element belongs to set B.

ie,
$$\boxed{A \times B = \{(a,b) \mid a \in A \ \& \ b \in B\}}$$

Ex:- $A_1 = \{a,b\}$, $A_2 = \{1,2\}$ & $A_3 = \{\alpha\}$

then,

1) $A_1 \times A_2 = \{(a,1), (a,2), (b,1), (b,2)\}$

$|A_1 \times A_2| = 4$ - $\because$ $|A_1| = 2$, $|A_2| = 2$

2) $A_1 \times A_2 \times A_3 = \{(a,1,\alpha), (a,2,\alpha), (b,1,\alpha) (b,2,\alpha)\}$

Relation:- An n-ary relation denoted as $R(X_1, X_2 \dots X_n)$ among crisp sets $X_1, X_2 \dots$ is a subset of the cartesian product

$\overset{n}{\underset{i=1}{\times}} X_i$.

Scanned by TapScanner

For binary relation $R(X,Y)$, where
$X = \{x_1, x_2, \cdots x_n\}$ &
$Y = \{y_1, y_2, \cdots y_m\}$

The relation matrix R is a 2-D matrix where X - Represents rows and Y represents columns.

and $R(i,j) = 1$, if $(x_i, y_j) \in R$
$R(i,j) = 0$, if $(x_i, y_j) \notin R$

Ex:-
$X = \{1, 2, 3, 4\}$

Here, $|X| = 4$.

then $X \times X = \begin{Bmatrix} (1,1) & (1,2) & (1,3) & (1,4) \\ (2,1) & (2,2) & (2,3) & (2,4) \\ (3,1) & (3,2) & (3,3) & (3,4) \\ (4,1) & (4,2) & (4,3) & (4,4) \end{Bmatrix}$

Let Relation R be defined as.

$R = \{(x,y) \mid y = x+1$ & $x, y \in X$

then, The Relation matrix R can be given as.

$$R = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Scanned by TapScanner

Operations on crisp relations

① Union $[R \cup S]$

$$R \cup S\,(x,y) = \max\left(R(x,y),\, S(x,y)\right)$$

② Intersection $[R \cap S]$

$$R \cap S\,(x,y) = \min\left(R(x,y), S(x,y)\right)$$

③ Compliment : $\bar{R}$ / R^C

$$\bar{R}\,(x,y) = 1 - R(x,y).$$

④ Composition : $R \odot S$

If R is relation on x, y and S is relation on y, z then the composition of relation on x, z is defined as

$$R \odot S = \left\{ (x,z) \;\middle|\; \begin{array}{l} (x,z) \in X \times Z \ \& \\ y \in Y \text{ such that} \\ (x,y) \in R \\ (y,z) \in S \end{array} \right\}.$$

Scanned by TapScanner

⑤ min- max composition:-

Given the relation matrices of Relations R & S.
The min.max composition is defined as

for $T = R \odot S$

Where, $T(x,z) = \max\left[\min\left(R(x,y), (S(y,z)\right)\right]$

Ex:- Let R & S be defined on sets

$X = \{1,3,5\}$ and $Y = \{1,3,5\}$.

$\therefore X \times Y = \{1,3,5\} \times \{1,3,5\}$

$= \{(1,1)(1,3)(1,5), (3,1)(3,3), (3,5)(5,1)(5,3)(5,5)\}$

Let Relation

$R: \{(x,y) \mid y = x+2$

$S: \{(x,y) \mid x < y$

$\therefore R = \{(1,3), (3,5)\}$

$S = \{(1,3), (1,5), (3,5)\}$.

Relation matrix

$$R = \begin{array}{c|c} & \begin{matrix} 1 & 3 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 3 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}, \quad S = \begin{array}{c|c} & \begin{matrix} 1 & 3 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 3 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

Scanned by TapScanner

$$R = \begin{array}{c|ccc} & 1 & 3 & 5 \\ \hline 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \\ 5 & 0 & 0 & 0 \end{array} \qquad S = \begin{array}{c|ccc} & 1 & 3 & 5 \\ \hline 1 & 0 & 1 & 1 \\ 3 & 0 & 0 & 1 \\ 5 & 0 & 0 & 0 \end{array}$$

Using min-max composition

$$T = R \odot S = \begin{array}{c|ccc} & 1 & 3 & 5 \\ \hline 1 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \end{array} \text{ — Result.}$$

$$\underline{R \odot S(1,1)} = \max\left[\min(R(1,1)), (S(1,1))), \min(R(1,3)), (S(1,3))), \min(R(1,5)), (S(1,5))\right]$$

$$= \max\left[\min(0,0), \min(1,0), \min(0,0)\right]$$

$$= \max\left[0, 0, 0\right]$$

$$= 0$$

<u>and so on</u> —

Scanned by TapScanner

Scanned by TapScanner

Fuzzy Relations:-

A fuzzy relation is a fuzzy set defined on a cartesian product of a crisp set $\{x_1, x_2, x_3 \cdots x_n\}$, where n-tuple $(x_1, x_2, x_3 \cdots x_n)$ may have the varing degree of membership within the relation.

The membership value indicates the state of relation between the tuples.

Let R is fuzzy relation betn two set X_1, X_2

Where X_1 is set of dieases.

X_2 is set of symptoms.

$X_1 = \{$Typhoid, viral fever, common cold$\}$

$X_2 = \{$shivering, Running Nose, High fever$\}$

Then

R =

	RN	HF	SH
TY	0.1	0.9	0.8
VF	0.2	0.9	0.7
CC	0.9	0.4	0.6

Scanned by TapScanner

Fuzzy cartesian product :-

Let fuzzy set $\tilde{A}$ is defined on universal set X and fuzzy set $\tilde{B}$ is defined on universal set Y. Let cartesian product betn FS $\tilde{A}$ & $\tilde{B}$ is $\tilde{A} \times \tilde{B}$ and resulting in relation $\tilde{R}$ is given as

$$\tilde{R} = \tilde{A} \times \tilde{B} \subset X \times Y \text{ (subset of X \& Y)}$$

where $\tilde{R}$ has its membership funn given by

$$\boxed{\mu_{\tilde{R}}^{(x,y)} = \mu_{\tilde{A} \times \tilde{B}}^{(x,y)}}$$

$$\boxed{\mu_{\tilde{R}}^{(x,y)} = \min\left[\mu_{\tilde{A}}^{(x,y)}, \mu_{\tilde{B}}^{(x,y)}\right]}$$

Ex:- Let $\tilde{A}$ & $\tilde{B}$ are two FS defined on universal set on x & y. and the fuzzy relation R resulting out of fuzzy cartesian product can be given as

$$X = \{x_1, x_2, x_3\}, \quad Y = \{y_1, y_2\}$$

$$\tilde{A} = \{(x_1, 0.2), (x_2, 0.7), (x_3, 0.4)\}$$

$$\tilde{B} = \{(y_1, 0.5), (y_2, 0.6)\}$$

Then

$$\tilde{R} = \tilde{A} \times \tilde{B} = \begin{array}{c c} & \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} & \begin{bmatrix} 0.2 & 0.2 \\ 0.5 & 0.6 \\ 0.4 & 0.4 \end{bmatrix} \end{array}$$

Scanned by TapScanner

$$\tilde{R}(x_1, y_1) = \min\left[\mu_{\tilde{A}}^{(x_1)}, \mu_{\tilde{B}}^{(y_1)}\right]$$
$$= \min[0.2, 0.5]$$
$$= \underline{0.2}$$

$$\tilde{R}(x_1, y_2) = \min\left[\mu_{\tilde{A}}^{(x_1)}, \mu_{\tilde{B}}^{(y_2)}\right] = \min[0.2, 0.6] = 0.2$$

$$\tilde{R}(x_2, y_1) = \min\left[\mu_{\tilde{A}}^{(x_2)}, \mu_{\tilde{B}}^{(y_1)}\right] = \min[0.7, 0.5] = 0.5$$

$$\tilde{R}(x_2, y_2) = \min\left[\mu_{\tilde{A}}^{(x_2)}, \mu_{\tilde{A}}^{(y_2)}\right] = \min[0.7, 0.6] = 0.6$$

$$\tilde{R}(x_3, y_1) = \min[0.4, 0.5] = 0.4$$

$$\tilde{R}(x_3, y_2) = \min[0.4, 0.6] = 0.4$$

$$\therefore \tilde{R} = \tilde{A} \times \tilde{B} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.2 & 0.2 \\ 0.5 & 0.6 \\ 0.4 & 0.4 \end{bmatrix} \end{array}$$

<u>Operations on fuzzy Relations</u> :-

Let $\tilde{R}$ & $\tilde{S}$ be the two fuzzy relations on $X \times Y$

(1) <u>Union</u> :- $(\tilde{R} \cup \tilde{S})$

$$\mu_{\tilde{R} \cup \tilde{S}}^{(x,y)} = \max\left(\mu_{\tilde{R}}^{(x,y)}, \mu_{\tilde{S}}^{(x,y)}\right)$$

(2) <u>Intersection</u> :- $(\tilde{R} \cap \tilde{B})$

$$\mu_{\tilde{R} \cap \tilde{B}}^{(x,y)} = \min\left(\mu_{\tilde{R}}^{(x,y)}, \mu_{\tilde{R}}^{(x,y)}\right)$$

Scanned by TapScanner

③ **Compliment :** $(\tilde{R}^{c})$

$$\boxed{\mu_{\tilde{R}^{c}}^{(x,y)} = 1 - \mu_{\tilde{R}}^{(x,y)}}$$

④ **Composition:-** $\tilde{R} \odot \tilde{S}$

Suppose $\tilde{R}$ is a fuzzy relation defined on $X \times Y$ and $\tilde{S}$ is fuzzy relation defined on $Y \times Z$ the $\tilde{R} \odot \tilde{S}$ is a fuzzy relation on $X \times Z$ and the fuzzy min-max composition can be defined as -

$$\boxed{\mu_{\tilde{R} \odot \tilde{S}}^{(x,z)} = \max\left[\min\left(\mu_{\tilde{R}}^{(x,y)}, \mu_{\tilde{S}}^{(y,z)}\right)\right]}$$

Scanned by TapScanner

Ex:- Given

$$X = \{x_1, x_2, x_3\}$$
$$Y = \{y_1, y_2\}$$
$$Z = \{z_1, z_2, z_3\}$$

Let $\tilde{R}$ is fuzzy relation as
$$\tilde{R} = \begin{array}{c|cc} & y_1 & y_2 \\ \hline x_1 & 0.5 & 0.1 \\ x_2 & 0.2 & 0.9 \\ x_3 & 0.8 & 0.6 \end{array}$$

and $\tilde{S}$ is fuzzy relation as
$$\tilde{S} = \begin{array}{c|ccc} & z_1 & z_2 & z_3 \\ \hline y_1 & 0.6 & 0.4 & 0.7 \\ y_2 & 0.5 & 0.8 & 0.9 \end{array}$$

Find $\tilde{R} \odot \tilde{S} \rightarrow X \times Z$

$$\Rightarrow \tilde{R} \odot \tilde{S} = \begin{array}{c|ccc} & z_1 & z_2 & z_3 \\ \hline x_1 & 0.5 & 0.4 & 0.5 \\ x_2 & 0.5 & 0.8 & 0.9 \\ x_3 & 0.6 & 0.6 & 0.7 \end{array} \text{ — Result}$$

$$\mu^{(x_1, z_1)}_{\tilde{R} \odot \tilde{S}} = \max[\min(0.5, 0.6), \min(0.1, 0.5)]$$
$$= \max[0.5, 0.1]$$
$$= \underline{0.5}$$

$$\mu^{(x_1, z_2)}_{\tilde{R} \odot \tilde{S}} = \max[\min(0.5, \underline{0.4}), \min(\underline{0.1}, 0.8)]$$
$$= \underline{0.4}$$

$$\mu^{(x_1, z_3)}_{\tilde{R} \odot \tilde{S}} = \max[\min(\underline{0.5}, 0.7), \min(\underline{0.1}, 0.9)]$$
$$= \underline{0.5}$$

and so on ——

Scanned by TapScanner

Ex:- 2) $X = \{x_1, x_2, x_3\}$

$Y = \{y_1, y_2\}$

$Z = \{z_1, z_2, z_3\}$

$$\tilde{R} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.4 & 0.2 \\ 0.2 & 0.7 \\ 0.9 & 0.6 \end{bmatrix} \end{array} \qquad \tilde{S} = \begin{array}{c} \\ y_1 \\ y_2 \end{array} \begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.6 & 0.5 & 0.8 \\ 0.6 & 0.7 & 0.8 \end{bmatrix} \end{array}$$

Find $\tilde{R} \odot \tilde{S}$:-

$\Rightarrow$ Result :

$$\tilde{R} \odot \tilde{S} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.4 & 0.4 & 0.4 \\ 0.6 & 0.7 & 0.7 \\ 0.6 & 0.6 & 0.8 \end{bmatrix} \end{array}$$

Ex: 3) : consider a set $P = \{P_1, P_2, P_3, P_4\}$ of four varieties of paddy plants, ~~set $D = \{D_1, D_2, D_3, D$~~ set $D = \{D_1, D_2, D_3, D_4\}$ of various diseases affecting plants and $S = \{S_1, S_2, S_3, S_4\}$ be the common symptoms of the diseases.

Let $\tilde{R}$ be relation on $P \times D$ and $\tilde{S}$ be the relation on $D \times S$.

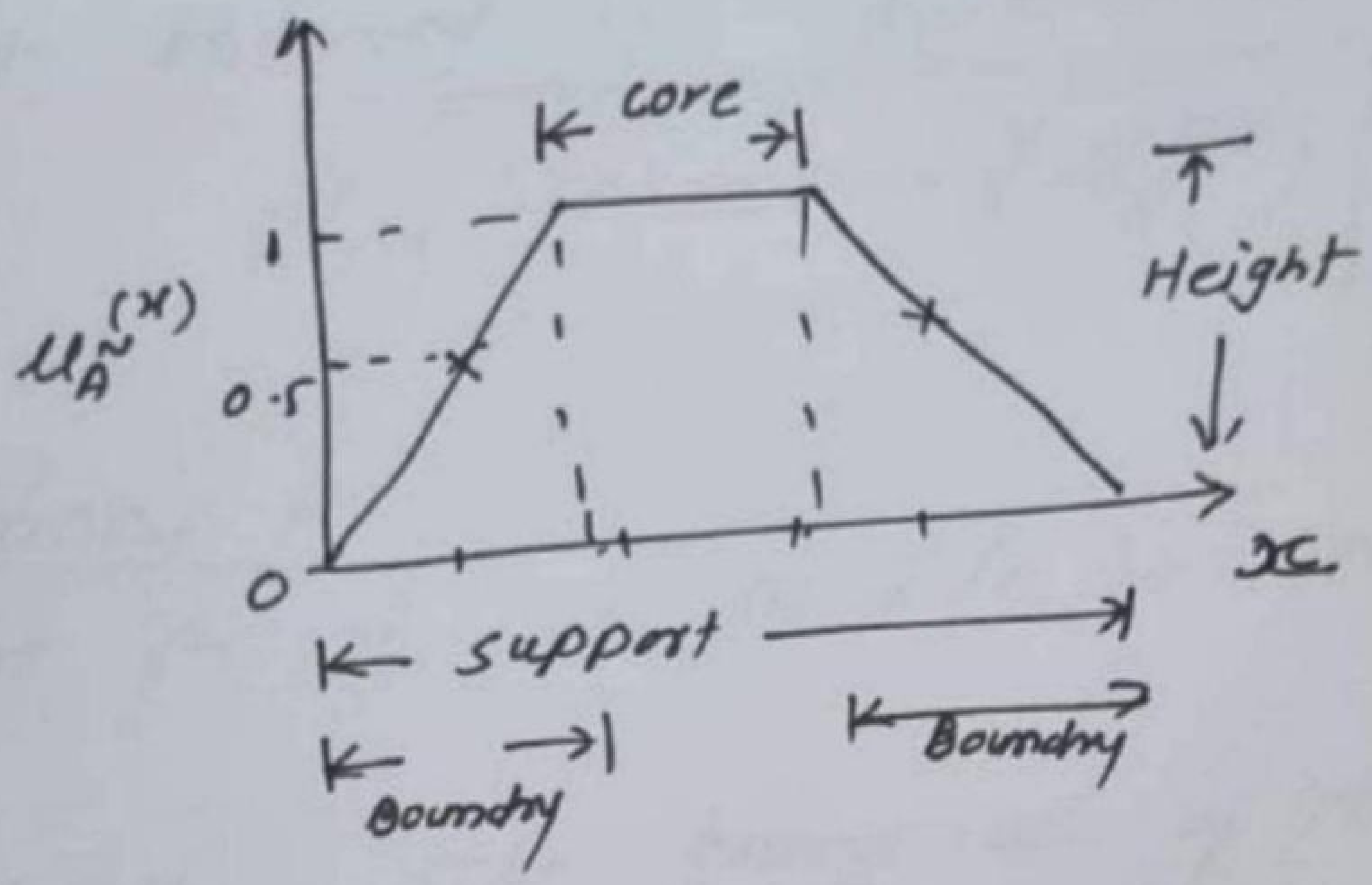
for $$\tilde{R} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & D_3 & D_4 \end{array} \\ \begin{bmatrix} 0.6 & 0.6 & 0.9 & 0.8 \\ 0.1 & 0.2 & 0.9 & 0.8 \\ 0.9 & 0.3 & 0.4 & 0.8 \\ 0.9 & 0.8 & 0.1 & 0.2 \end{bmatrix} \end{array}, \qquad \tilde{S} = \begin{array}{c} \\ D_1 \\ D_2 \\ D_3 \\ D_4 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & S_4 \end{array} \\ \begin{bmatrix} 0.1 & 0.2 & 0.7 & 0.9 \\ 1 & 1 & 0.4 & 0.6 \\ 0 & 0 & 0.5 & 0.9 \\ 0.9 & 1 & 0.8 & 0.2 \end{bmatrix} \end{array}$$

obtain the association of plants with the different symptoms of dieases using min-max composition.

Scanned by TapScanner

Some concepts in fs Theory

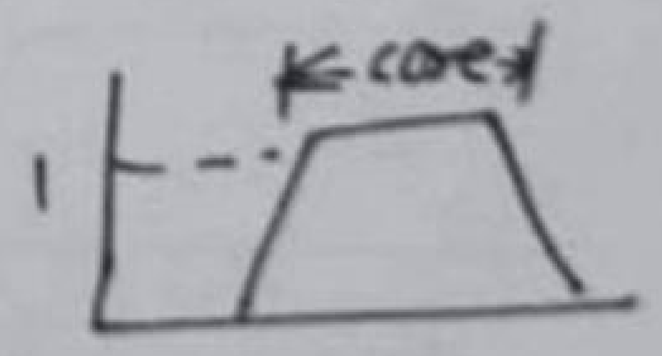
consider a mf as



1) <u>Core</u> $[C(\tilde{A})]$:-

The core of membership function for some fuzzy set $\tilde{A}$ is defined as the regim of universe of discourse that is charectoranied by complete or full membership in set $\tilde{A}$.

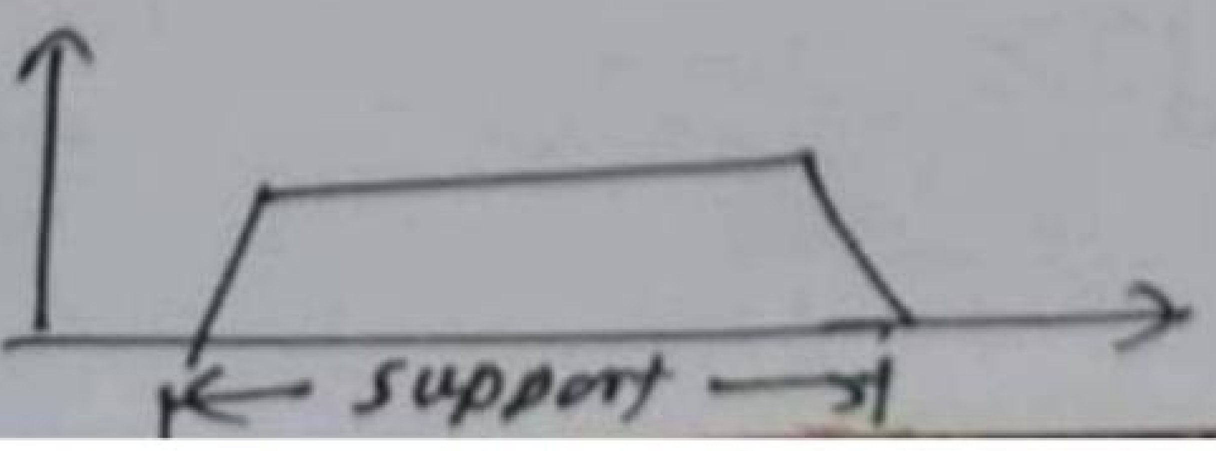
Thus core contains those elements of universal set such that $\mu_{\tilde{A}}^{(x)} = 1$.



2) <u>Support</u> $[S(\tilde{A})]$

The support of mf for some fs $\tilde{A}$ is defined as the regim of universe that is charectarised by non-zero membership in set $\tilde{A}$.

Thus support consists of those elements x of the univsal set such that $\mu_{\tilde{A}}^{(x)} > 0$

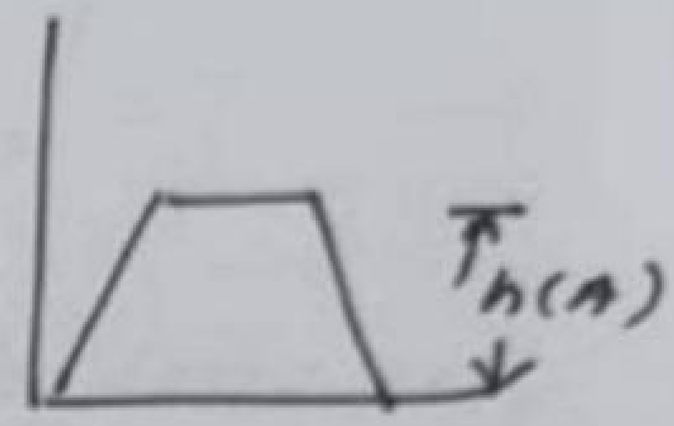


Scanned by TapScanner

③ Height of fs [h(A)]

The height of fs $\tilde{A}$ is the larger membership grade obtained by any element in set $\tilde{A}$.

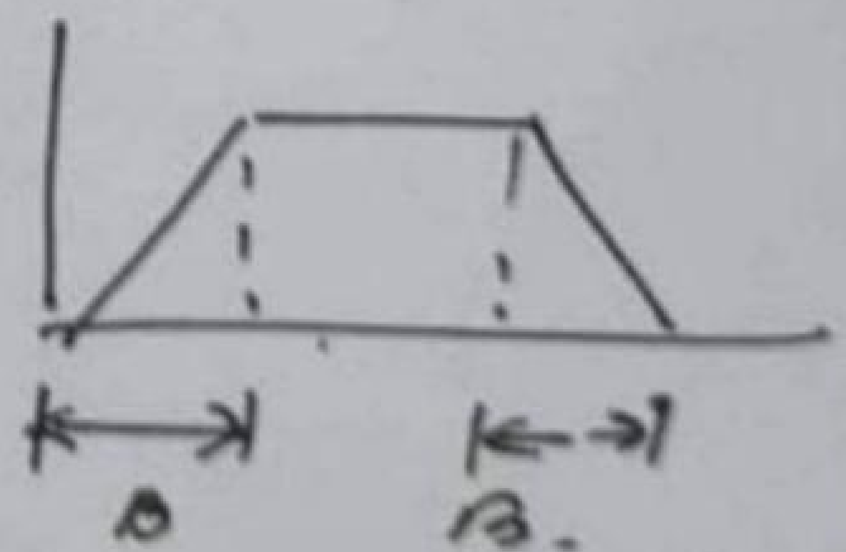
ie, $\boxed{h(A) = \max\left(\mu_{\tilde{A}}^{(x)}\right)}$



④ Normal fs:-

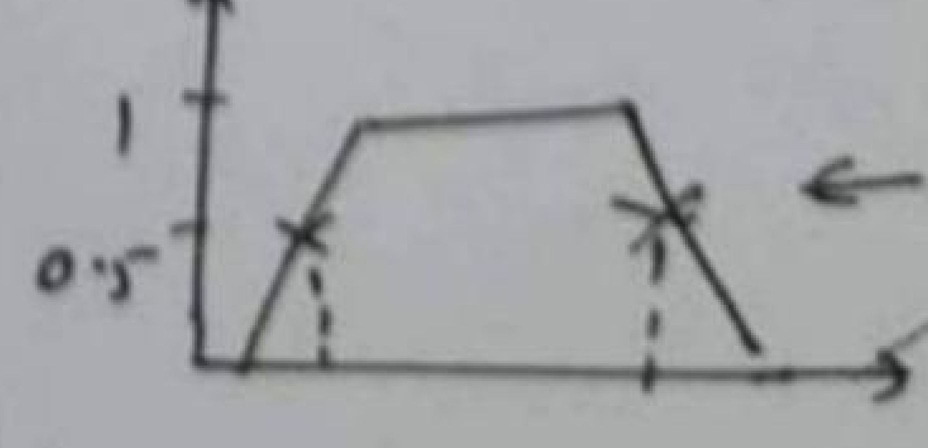
The fs $\tilde{A}$ is said to be normal if $h(A) = 1$

⑤ Boundary:- The boundries of mF for some fs $\tilde{A}$ are defined as the region of universe that have non-zero membership but not complete membersp ie, The boundries consists of those elements x of the universe such that $x < \mu_{\tilde{A}}^{(x)} < 1$.



⑥ Cross-over points:- The cross over points of a mF $\tilde{A}$ is defined as the element in the universe for which a particular fuzzy set $\tilde{A}$ has value equal to 0.5.

ie, $\mu_{\tilde{A}}^{(x)} = 0.5$

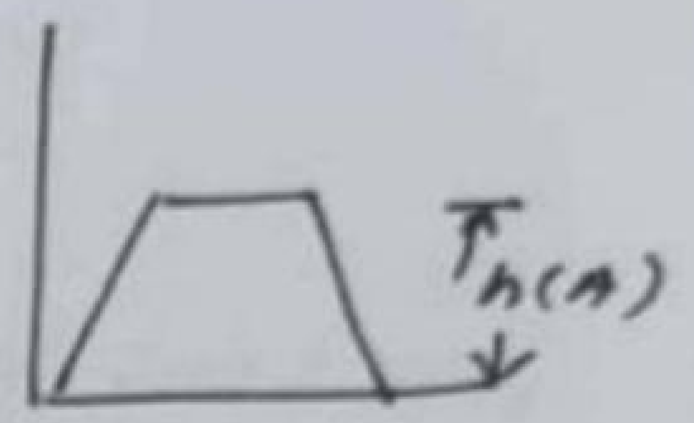


Scanned by TapScanner

③ **Height of fs [h(A)]**

The height of fs $\tilde{A}$ is the larger membership grade obtained by any element in set $\tilde{A}$.

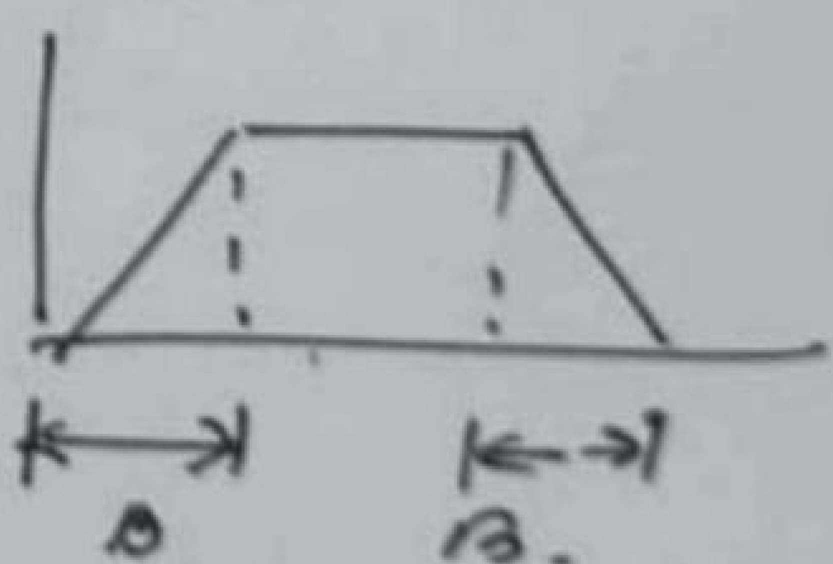
ie. $\boxed{h(A) = \max\left(\mu_{\tilde{A}}^{(x)}\right)}$



④ **Normal fs:-**

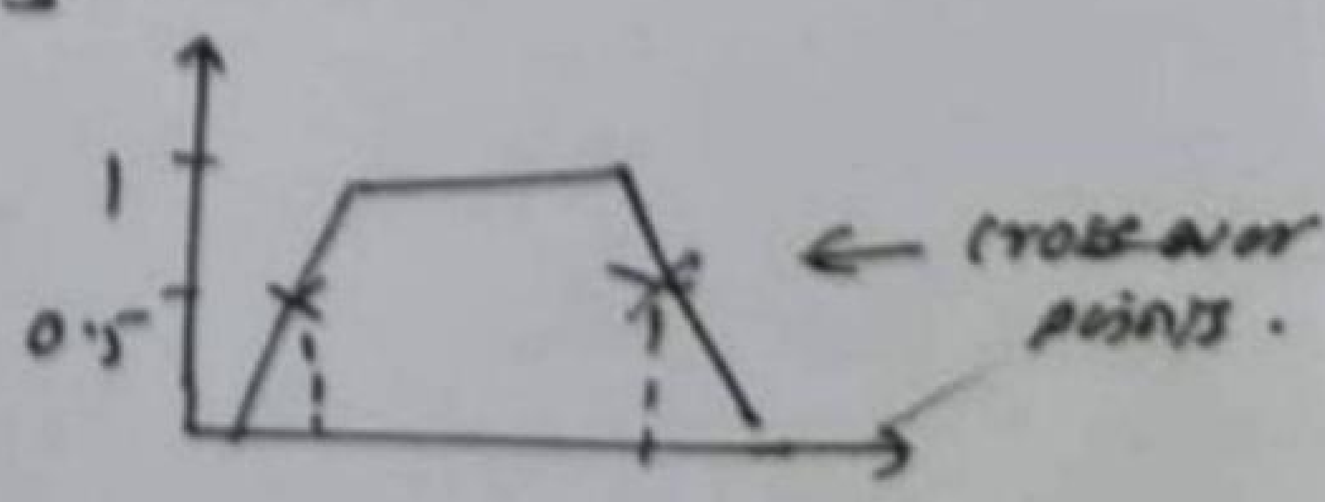
The fs $\tilde{A}$ is said to be normal if $h(A) = 1$

⑤ **Boundary:-** The boundries of MF for some fs $\tilde{A}$ are defined as the region of universe that have non-zero membership but not complete membership ie, The boundries consists of those elements x of the universe such that $x < \mu_{\tilde{A}}^{(x)} < 1$.



⑥ **cross-over points:-** The cross over points of a MF $\tilde{A}$ is defined as the element in the universe for which a particular fuzzy set $\tilde{A}$ has value equal to 0.5.

ie. $\mu_{\tilde{A}}^{(x)} = 0.5$



Scanned by TapScanner

Ex:-

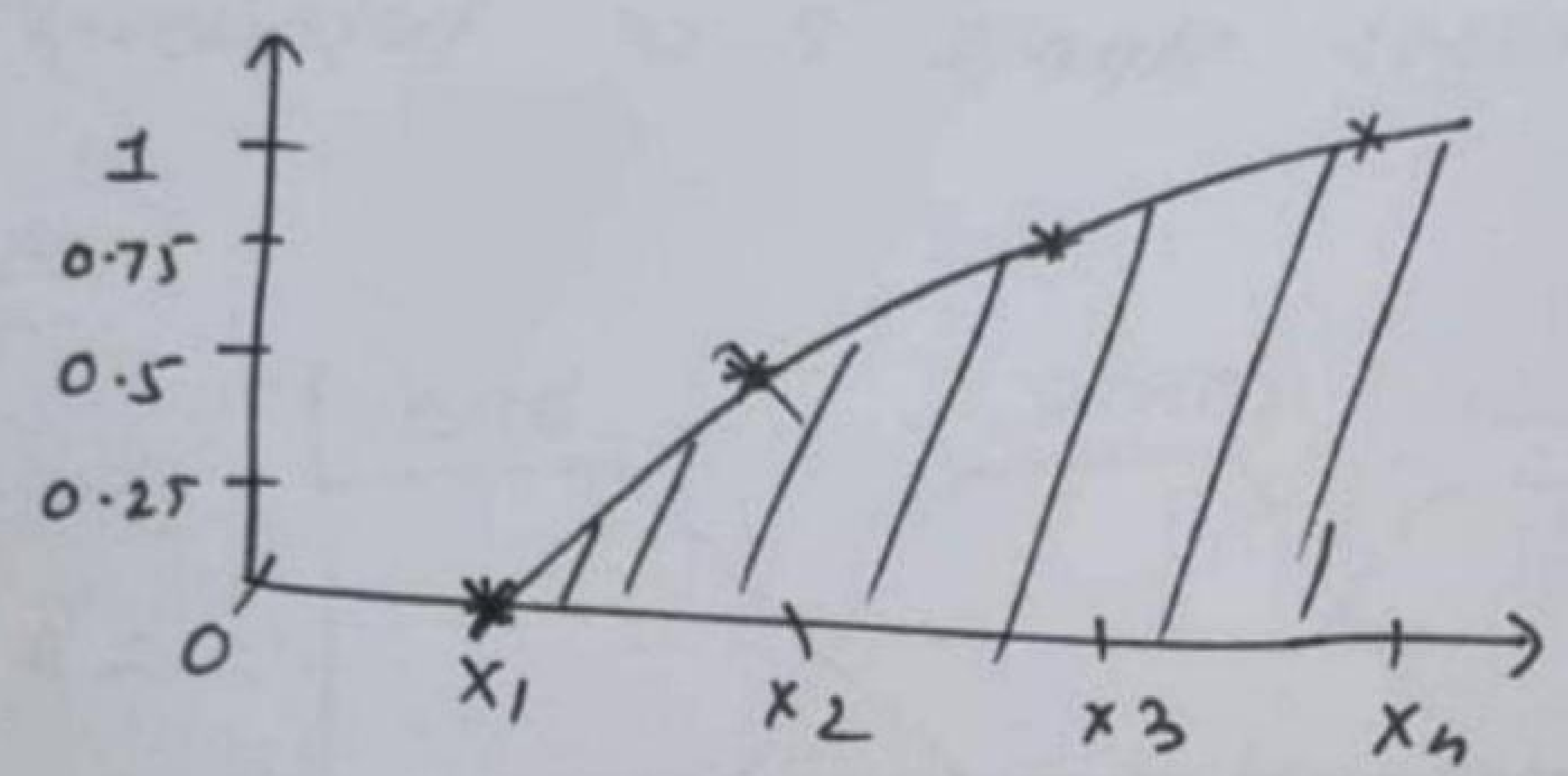
Consider the univrsarsal set $X = \{x_1, x_2, x_3, x_4\}$

Let $\tilde{A}$ is FS defined on univ. set x as

$$\tilde{A} = [(x_1, 0), (x_2, 0.5), (x_3, 0.75), (x_4, 1)]$$

find - 1) support 2) Core 3) height 4) Cross over points of FS. , 5) Is this set normal?

$\Rightarrow$



1) Support of FS $\tilde{A}$ -

$$S(A) = \{x_1, x_2, x_3, x_4\}$$

2) Core of FS $\tilde{A}$ -

$$C(A) = \{x_4\}$$

3) height of FS $\tilde{A}$ -

$$h(A) = 1$$

4) Corss over point of FS $\tilde{A}$ is x_2

5) yes this FS is normal since $h(A) = 1$.

Scanned by TapScanner

Defuzzification

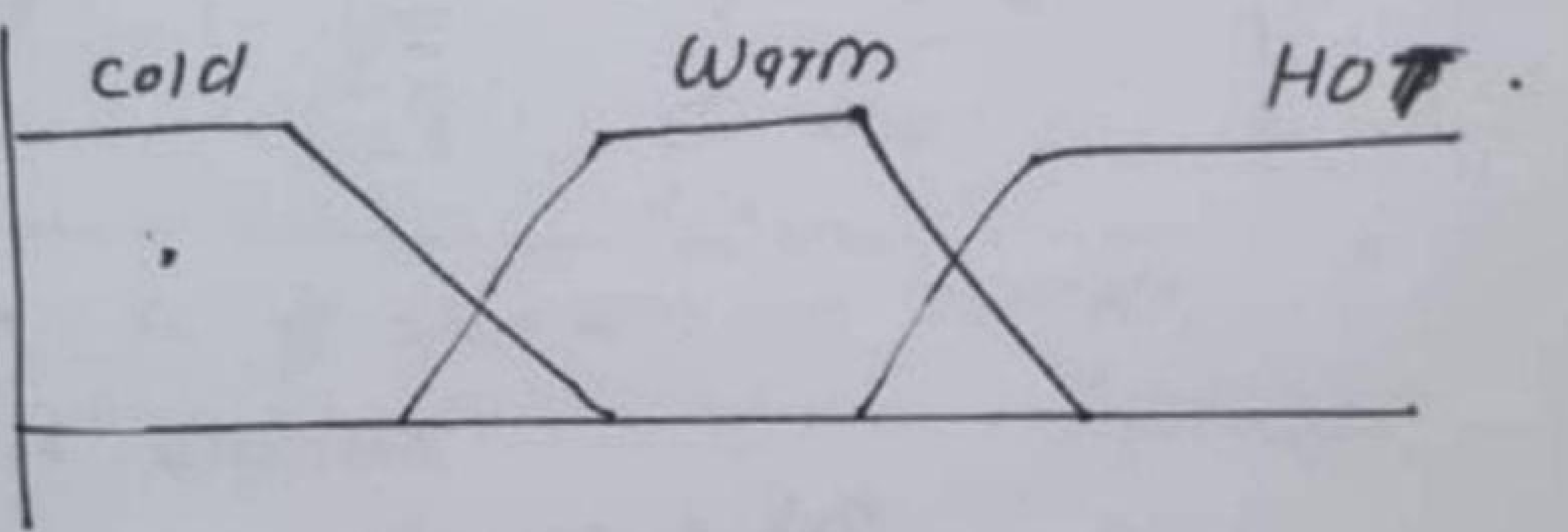
The conversion of FS to a single crisp value is called as Defuzzification.

It is reverse of fuzzification.

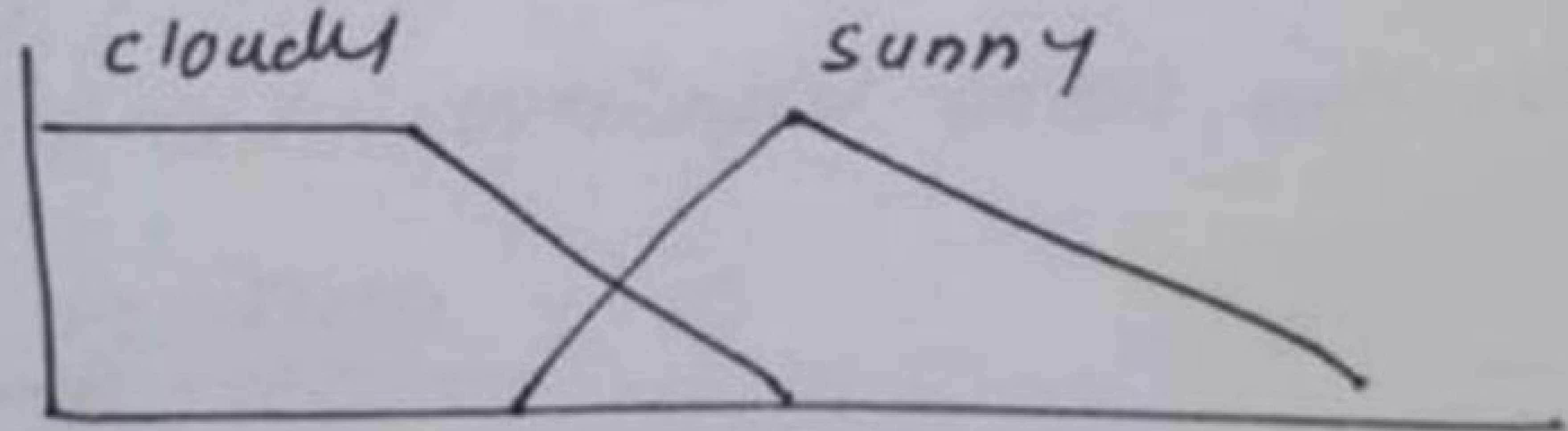
For a system whose o/p is fuzzy, it is easier to take a crisp decision, if the output is represented as a single scaler quantity.

Ex:-

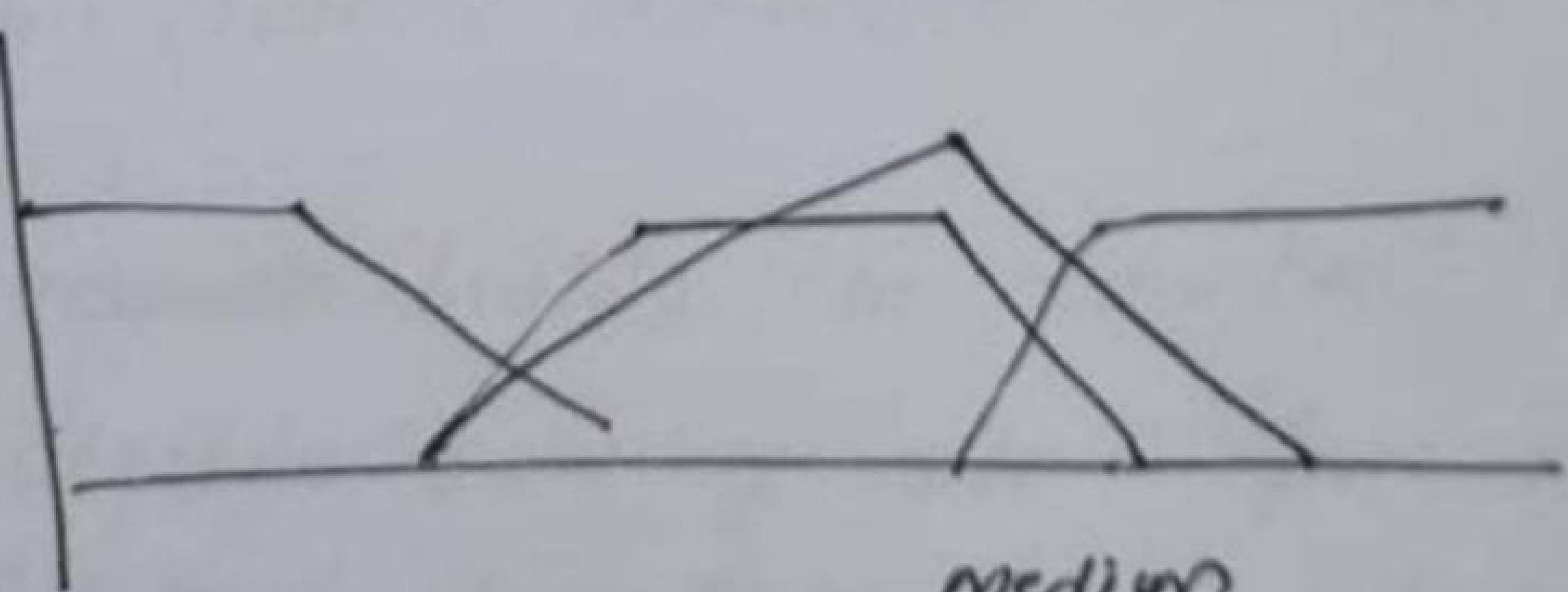
$\tilde{A}$ =

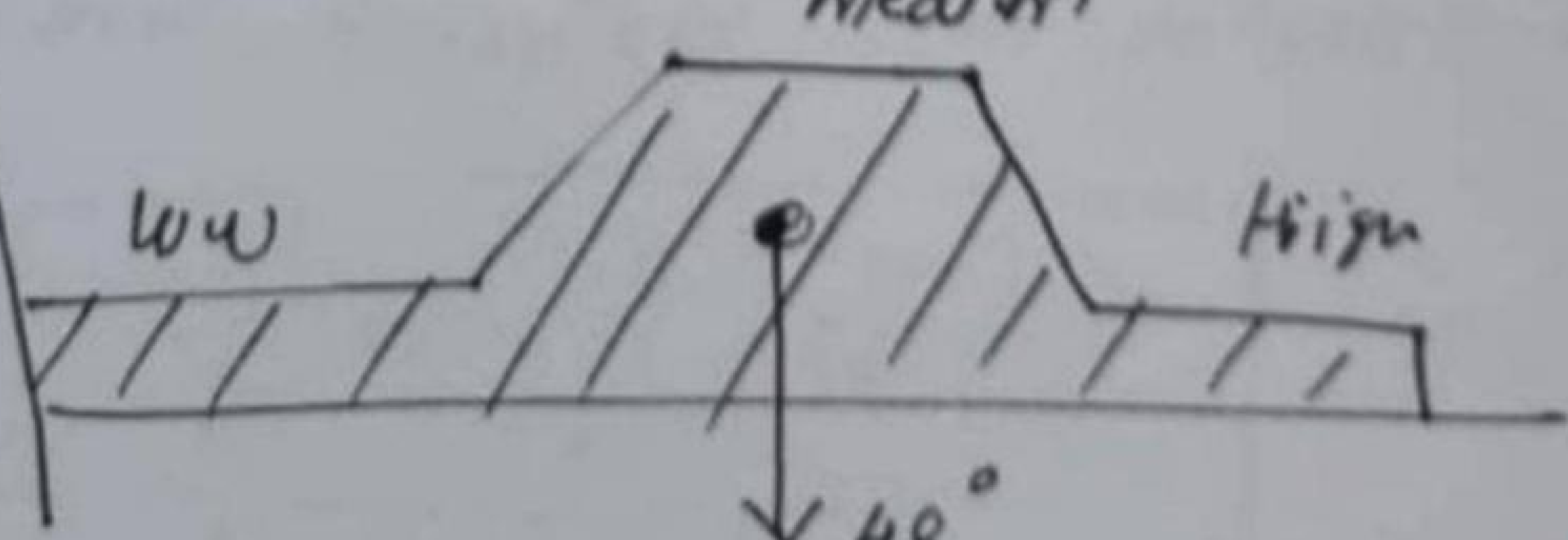


$\tilde{B}$ =



$\tilde{A} \cup B$ =





Defuzzification

There are 3 methods used for Defuzzifications-

1. Centroid method (centre of area / centre of gravity)
2. Centre of sum.
3. Mean of maxima.

1. Centroid method :-

In this method the centre of area is calculated which is occupied by fuzzy set. It is represented as.

$$\boxed{x^* = \sum_{i=1}^{n} \frac{x_i \cdot \mu x_i}{\sum_{i=1}^{n} \mu x_i}}$$

$n \rightarrow$ No. of elements in sample.

$x_i \rightarrow$ elements

$\mu(x_i) \rightarrow$ membership f^n.

- In this method overlaped areas are considered once.

2. Centre of sum :-

In this method overlaped area are considered twice.

COS ~~results~~ builds the resultant mf by taking the algebric sum of outputs ~~of~~ from each of the contributed fuzzy set. The defuzzified value x^* is.

$$\boxed{x^* = \frac{\sum_{i=1}^{N} x_i \cdot \sum_{K=1}^{n} \mu_{\tilde{A}_K}(x_i)}{\sum_{i=1}^{N} \sum_{K=1}^{n} \mu_{\tilde{A}_K}(x_i)}}$$

(68)

Scanned by TapScanner

FUZZY IF-THEN-ELSE

Fuzzy propositions:-

A fuzzy proposition is a statment which aquires a fuzzy truth value.

Thus, given $\tilde{P}$ to to a fuzzy proposition, $T(\tilde{P})$ represents a truth value (0-1). attached to $\tilde{P}$.

Fuzzy propositions are associated with fuzzy sets. The fuzzy membership value associated with the fuzzy set $\tilde{A}$ for $\tilde{P}$ is treated as the fuzzy truth value $T(\tilde{P})$.

ie, $T(\tilde{P}) = \mu_{\tilde{A}}^{(x)}$ where $0 \leq \mu_{\tilde{A}}^{(x)} \leq 1$.

Example:-

$\tilde{P}$: Ram is honest

$T(\tilde{P}) = 0.8$, if $\tilde{P}$ is partially true.

$T(\tilde{P}) = 1$, if $\tilde{P}$ is absolutly true.

Fuzzy Connectives:-

(i) Negation : $-$

(ii) Disjunction : $\vee$

(iii) conjuction : $\wedge$

(iv) Implication : $\Rightarrow$

Scanned by TapScanner

Ex:- Let $\tilde{P}$, $\tilde{Q}$ are fuzzy propositions and $T(\tilde{P})$, $T(\tilde{Q})$ are their truth values.

symbol	connective	usage	Definition
$-$	Negation	$\tilde{P}^c$	$1-T(\tilde{P})$
$\vee$	Disjunction	$\tilde{P} \vee \tilde{Q}$	$\max[T(\tilde{P}), T(\tilde{Q})]$
$\wedge$	Conjunction	$\tilde{P} \wedge \tilde{Q}$	$\min[T(\tilde{P}), T(\tilde{Q})]$
$\Rightarrow$	Implication	$\tilde{P} \Rightarrow \tilde{Q}$	$\tilde{P}^c \vee \tilde{Q} = \max(1-T(\tilde{P}), T(\tilde{Q}))$

$\tilde{P}$ & $\tilde{Q}$ related by the $\Rightarrow$ operator are known as antecedent and consequent respectively.

$\Rightarrow$ represents the If-THEN statment as

If x is $\tilde{A}$ THEN y is $\tilde{B}$, is equivalent to

$$\tilde{R} = (\tilde{A} \times \tilde{B}) \cup (\tilde{A}^c \times y)$$

The membership fun. of $\tilde{R}$ is given by

$$\mu_{\tilde{R}}(x,y) = \max[\min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)), 1-\mu_{\tilde{A}}(x)$$

Scanned by TapScanner

Ex:- 1

$\tilde{P}$: Mary is efficient. $T(\tilde{P}) = 0.8$

$\tilde{Q}$: Ram is efficient, $T(\tilde{Q}) = 0.65$

(i) $\tilde{P}^c$: Mary is not efficient

$$T(\tilde{P})^c = 1 - T(\tilde{P}) = 1 - 0.8 = 0.2$$

(ii) $\tilde{P} \wedge \tilde{Q}$: Mary is efficient and so is Ram.

$$T(\tilde{P} \wedge \tilde{Q}) = \min(T(\tilde{P}), T(\tilde{Q}))$$
$$= \min(0.8, 0.65)$$
$$= 0.65$$

(iii) $T(\tilde{P} \vee \tilde{Q})$: Either Mary or RAM is efficient

$$T(\tilde{P} \vee \tilde{Q}) = \max(T(\tilde{P}), T(\tilde{Q}))$$
$$= \max(0.8, 0.65)$$
$$= 0.8.$$

(iv) $\tilde{P} \Rightarrow \tilde{Q}$: If Mary is efficient then so is RAM

$$T(\tilde{P} \Rightarrow \tilde{Q}) = \max(1 - T(\tilde{P}), T(\tilde{Q}))$$
$$= \max((1 - 0.8), 0.65)$$
$$= \max(0.2, 0.65)$$
$$= 0.65$$

Scanned by TapScanner

Ex:2 :- Let $X = [a, b, c, d]$

$Y = [1, 2, 3, 4]$

and $\tilde{A} = \{(a,0), (b,0.8), (c,0.6), (d,1)\}$

$\tilde{B} = \{(1,0.2), (2,1), (3,0.8), (4,0)\}$

$\tilde{C} = \{(1,0), (2,0.4), (3,1), (4,0.8)\}$.

- Determine the implication relations.

(i) If x is $\tilde{A}$ THEN y is $\tilde{B}$.

(ii) If x is $\tilde{A}$ THEN y is $\tilde{B}$ ELSE y is $\tilde{C}$

$\Rightarrow$

(i) compute

$\tilde{R} = (\tilde{A} \times \tilde{B}) \cup (\tilde{A}^c \times Y)$ where.

$$\mu_{\tilde{R}}^{(x,y)} = \max\left(\min\left(\mu_{\tilde{A}}^{(x)}, \mu_{\tilde{B}}^{(y)}\right), 1 - \mu_{\tilde{A}}^{(x)}\right)$$

- $$\tilde{A} \times \tilde{B} = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2 & 0.8 & 0.8 & 0 \\ 0.2 & 0.6 & 0.6 & 0 \\ 0.2 & 1 & 0.8 & 0 \end{bmatrix} \end{array}, \quad \tilde{A}^c \times Y = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 0.4 & 0.4 & 0.4 & 0.4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Here Y could be viewed as $\{(1,1) (2,1) (3,1) (4,1)\}$ a fuzzy set all of whose elements x have $\mu(x) = 1$

$$\therefore \tilde{R} = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0.2 & 0.8 & 0.8 & 0.2 \\ 0.4 & 0.6 & 0.6 & 0.4 \\ 0.2 & 1 & 0.8 & 0 \end{bmatrix} \end{array}$$

$\rightarrow$ This Represents If x is $\tilde{A}$ THEN y is $\tilde{B}$.

Scanned by TapScanner

(ii) <u>Compute</u>

$$\tilde{R} = (\tilde{A} \times \tilde{B}) \cup (\tilde{A}^c \times \tilde{C}) \text{ where.}$$

$$\mu_{\tilde{R}}(x, y) = \max\left[\min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)), \min(1 - \mu_{\tilde{A}}(x), \mu_{\tilde{C}}(y))\right]$$

$$\tilde{A} \times \tilde{B} = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2 & 0.8 & 0.8 & 0 \\ 0.2 & 0.6 & 0.6 & 0 \\ 0.2 & 1 & 0.8 & 0 \end{bmatrix} \end{array}$$

$$\tilde{A}^c \times \tilde{C} = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & 0.4 & 1 & 0.8 \\ 0 & 0.2 & 0.2 & 0.2 \\ 0 & 0.4 & 0.4 & 0.4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

<u>Therefore</u>

$$\tilde{R} = \max\left((\tilde{A} \times \tilde{B}), (\tilde{A}^c \times \tilde{C})\right)$$

$$\tilde{R} = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & 0.4 & 1 & 0.8 \\ 0.2 & 0.8 & 0.8 & 0.2 \\ 0.2 & 0.6 & 0.6 & 0.4 \\ 0.2 & 1 & 0.8 & 0 \end{bmatrix} \end{array}$$

The above $\tilde{R}$ represents
If x is $\tilde{A}$ THEN y is $\tilde{B}$ ELSE y is C.

Scanned by TapScanner

Fuzzy Inferences (Approximate reasoning)

Refers to computational procedures used for evaluating linguistic descriptions.

Two important infering procedures are-

(i) Generalised modus Ponens (GMP)

(ii) Generalised modus Tollens (GMT).

i) GMP is stated as -

$$\text{IF } x \text{ is } \tilde{A} \text{ THEN } y \text{ is } \tilde{B}$$

$$\rightarrow \frac{x \text{ is } \tilde{A}'}{y \text{ is } \tilde{B}'}$$

Here $\tilde{A}, \tilde{B}, \tilde{A}', \tilde{B}'$ are fuzzy terms.

Every fuzzy linguistic statment above the line is analytically known and what is below is analytically unknown.

To compute the membership function of $\tilde{B}'$ the min-max composition of fuzzy set $\tilde{A}'$ with $\tilde{R}(x,y)$ which is known implication relation (IF-THEN) relation is used. that is

$$\tilde{B}' = \tilde{A}' \circ \tilde{R}(x,y)$$

ie, $\mu_{\tilde{B}'}(y) = \max(\min(\mu_{\tilde{A}'}(x), \mu_{\tilde{R}}(x,y)))$

where, $\mu_{\tilde{A}'}(x)$ is mf of $\tilde{A}'$, $\mu_{\tilde{R}}(x,y)$ is mf of implication relation & $\mu_{\tilde{B}'}(y)$ is mf of $\tilde{B}'$.

Scanned by TapScanner

(ii) GMT has the form.

IF x is $\tilde{A}$ THEN y is $\tilde{B}$.

$$\rightarrow \quad \frac{y \text{ is } \tilde{B}'}{x \text{ is } \tilde{A}'}$$

The membership of $\tilde{A}'$ is computed as

$$\tilde{A}' = \tilde{B}' \odot \tilde{R}(x, y)$$

$$\text{i.e., } \mu_{\tilde{A}'}^{(x)} = \max\left[\min\left(\mu_{\tilde{B}'}^{(y)}, \mu_{\tilde{R}}^{(x,y)}\right)\right]$$

—— o ——

Ex:- Apply the fuzzy modus Ponens rule to deduce Rotation is quit slow. given

(i) If the temp. is high then the rotation is slow.

(ii) The Temp. is very very high.

Let $\tilde{H}$ (High), $V\tilde{H}$ (very high), $\tilde{S}$ (slow) and $Q\tilde{S}$ (Quite slow) indicats the associated fuzzy set as follows.

for $X = \{30, 40, 50, 60, 70, 80, 90, 100\}$ the set of temperatures

and $Y = \{10, 20, 30, 40, 50, 60\}$ the set of rotation/min.

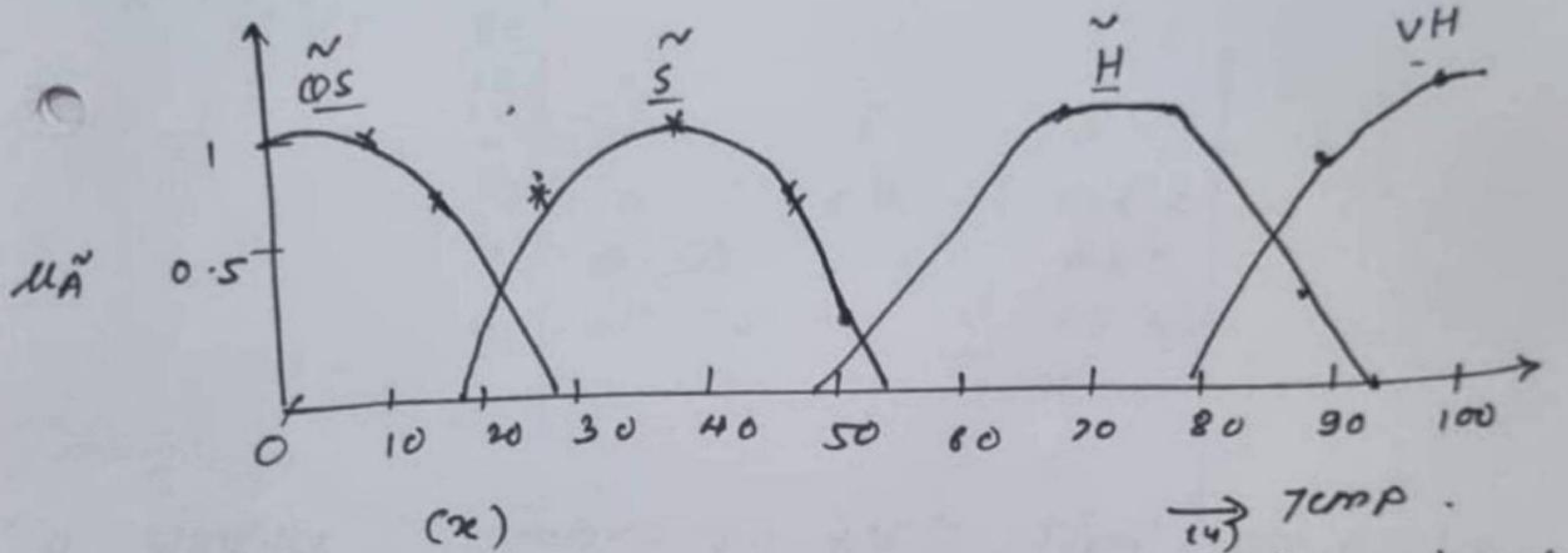
Scanned by TapScanner

$\tilde{H} = \{(70,1)\ (80,1)\ (90,0.3)\}$.

$\tilde{VH} = \{(90,0.9),\ (100,1)\}$

$\tilde{QS} = \{(10,1)\ (20,0.8)\}$

$\tilde{S} = \{(30,0.8)\ (40,1)\ (50,0.6)\}$



(i) If the temp. is high then rotation is slow

To derive $\tilde{R}(x,y)$ representing the above implication relation.

$$\tilde{R}(x,y) = \max(\tilde{H} \times \tilde{S},\ \bar{\tilde{H}} \times y)$$

$$\tilde{H} \times \tilde{S} = \begin{array}{c} \\ 30 \\ 40 \\ 50 \\ 60 \\ 70 \\ 80 \\ 90 \\ 100 \end{array} \begin{array}{c} \begin{array}{cccccc} 10 & 20 & 30 & 40 & 50 & 60 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 1 & 0.6 & 0 \\ 0 & 0 & 0.8 & 1 & 0.6 & 0 \\ 0 & 0 & 0.3 & 0.3 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Scanned by TapScanner

$$\tilde{H}^c \times y = \begin{array}{c|cccccc} & 10 & 20 & 30 & 40 & 50 & 60 \\ \hline 30 & 1 & 1 & 1 & 1 & 1 & 1 \\ 40 & 1 & 1 & 1 & 1 & 1 & 1 \\ 50 & 1 & 1 & 1 & 1 & 1 & 1 \\ 60 & 1 & 1 & 1 & 1 & 1 & 1 \\ 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 80 & 0 & 0 & 0 & 0 & 0 & 0 \\ 90 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 100 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$$\tilde{R}(x,y) = \begin{array}{c|cccccc} & 10 & 20 & 30 & 40 & 50 & 60 \\ \hline 30 & 1 & 1 & 1 & 1 & 1 & 1 \\ 40 & 1 & 1 & 1 & 1 & 1 & 1 \\ 50 & 1 & 1 & 1 & 1 & 1 & 1 \\ 60 & 1 & 1 & 1 & 1 & 1 & 1 \\ 70 & 0 & 0 & 0.8 & 1 & 0.6 & 0 \\ 80 & 0 & 0 & 0.8 & 1 & 0.6 & 0 \\ 90 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 100 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

To deduce Rotation is quite slow we make use of composition rule.

$$\tilde{Q}s = V\tilde{H} \odot R(x,y)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 1 \end{bmatrix} \times \begin{bmatrix} \\ \\ \end{bmatrix}$$

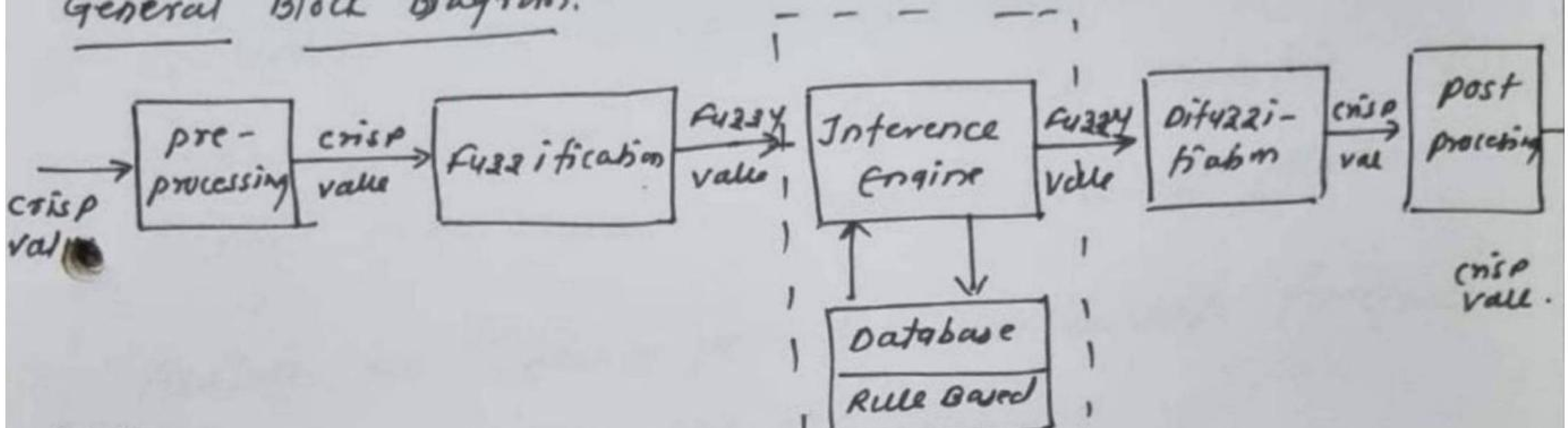
$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Scanned by TapScanner

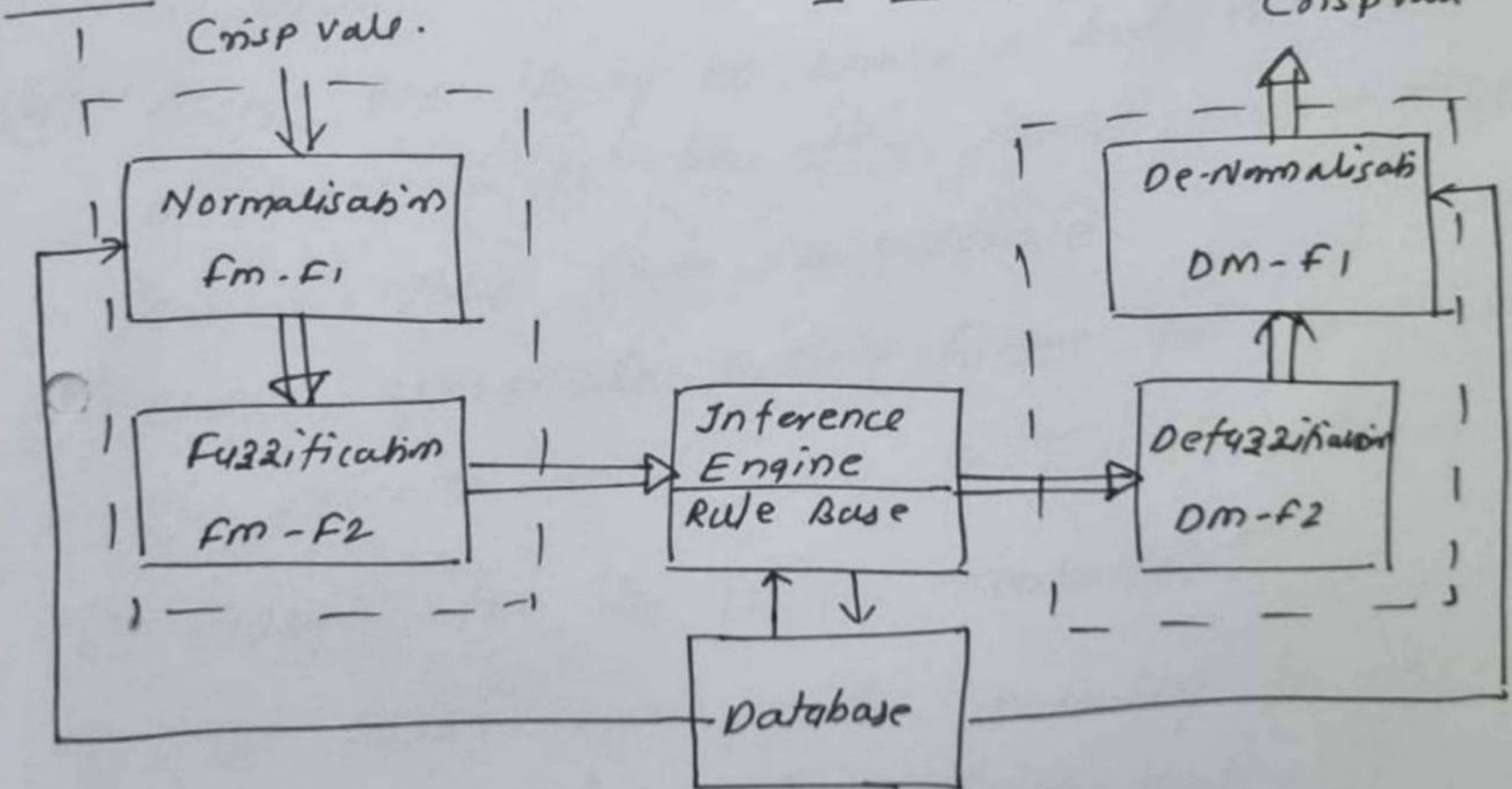
Fuzzy control systems:-

FKBC - fuzzy knowledge based controllers.

General Block Diagram:-



FKBC -



$\Longrightarrow$ computational flow

$\longrightarrow$ Information flow

Fm - F1 — fuzzification Normalisation

Fm - F2 — fuzzification module

DM - F1 — Diffuzification Normalisation

DM - F2 $\rightarrow$ Deffuzification modu.

Scanned by TapScanner

Steps for Designing FLC:-

1. Identify the variables (Input states & output) of the problem.
2. Partition the universe of discourse or interval spanned by each variable into a number of fuzzy sets, assigning each linguistic label.
3. Assign or determine the membership function for each fuzzy subsets.
4. Assign the fuzzy relationship betⁿ the inputs or fuzzy subsets on other hand output fuzzy subsets. Thus form a rulebase.
5. Choose appropriate scaling factor for input and o/p variables to [0 to 1] or [-1, 1] interval.
6. Fuzzyfy the I/p to the controller.
7. Use fuzzy approximate reasoning to infer the o/p contributed from each rule.
8. Aggregate the fuzzy outputs recommended by each rule.
9. Apply defuzzification to form crisp o/p.

— x —

Scanned by TapScanner