P. Rajesh M.Tech.,

I-(1)

Neural Networks:-

Introduction:- By Concept of biological neural networks may hold the key to success of solving intelligent tasks by machines.
The new field is called Artificial Neural networks. which is more apt to describe it as parallel and distributed processing.

*) Brain:- Has A highly Complex, non-linear & Parallel Complex Structural Constituents :⇒ Neuron's.
~~human brain~~ Neurons are interconnected in a very Complex way impossible to simulate.
∴ human brain got typically billions of nerve cells and trillions of interconnections.

*) The n/w of human brain are: Massively Parallel n/w of neurons
✓ 10 - billions of neurons
✓ 60 - trillions of interconnections

→ Artificial neural n/w :- (ANN) :-

NIAEFVN

Usefullness & Capabilities:-
I) It exploits non-linearity :- inter connections of nonlinear neurons.
→ Nonlinearity is distributed throughout.

(ii) The NN are based on Parallel architecture of biological brains of humans ...

2) I/p & o/p Mapping :- (Eg:- Storing info in a Pattern. & o/p Pattern is may be different we have to do adjust.)
↳ Learning ability.

3) Adapting :- Can adapt the face parameters to changes in surrounding environment.

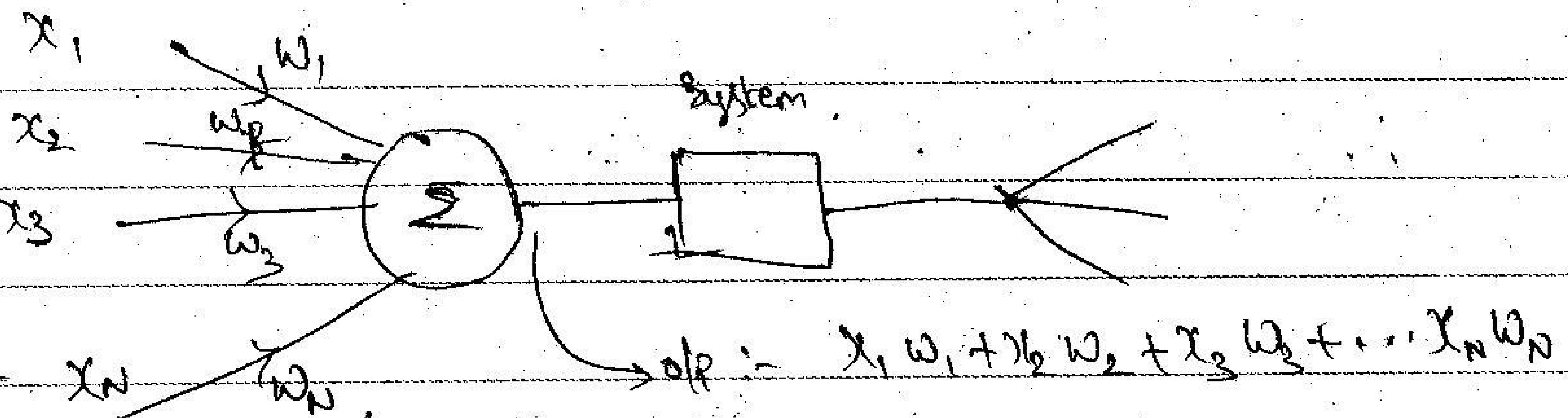
4) Evidential Response :- Decision with a measure of confidence.

5) Fault tolerance :- (If connection is not working ?)
↳ Just degradation happens it is (Graceful degradation)

6) VLSI implementability :- (based on concept of VLSI implementation to be designed neurons for computing).

7) Neuro.biological analogy :- All above should be considered.

Pyramidal Representation :-



(Equivalent Electrical models).

Neural N/w :-

1) **Neural Net :-** is an artificial representation of the human brain that tries to simulate its learning Process.

→ An artificial neural n/w (ANN) is often called a Neural n/w (or) Neural Net (NN).

→ Traditionally (NN) is represented as network of biological neurons in neural system that Process & transmit information.

→ ANN is interconnected group of artificial neurons that uses a mathematical model (or) computational model for information Processing based on Connectionist approach to Computation.

→ ANN is a n/w of simple processing elements (neurons) which can exibit complex global behavior determined ~~by~~ by Connections b/w the Processing elements & Element Parameters. (neurons)

2) **Why Neural N/w :-**

→ Conventional Computers are good for fast arithematic & does what Programmer ~~does~~ Program as to do.

→ The Conventional Computers are not so good for interacting with noisy data (or) data from Environment, massive Parallelism fault tolerance; and adapting to circumstances.

→ NN system help where we can not formulate an algorithm solution. as where we get lot of Examples of behavior

→ NN follows different Paradigm for Computing.

(i) von-Neuman machines based on Processing/memory abstraction of human information Processing

(ii) The NN are based on Parallel architecture of biological brains

→ NN are form of multiprocessors computer system with (I-4)

(i) Simple processing elements

(ii) A high degree of interconnection

(iii) Simple scalar message

(iv) Adaptive interaction b/w elements.

3) History :-

(i) Mcculloch and pitts (1943) :- are generally recognized as the

* designer of first neural n/w. which combine
→ many signal processing unit together which could lead overall increase in computational power.

* Suggested idea :- Neuron has a threshold level and once that level is reached neuron fires.
It is fundamental way which ANN's operate.

→ M&P n/w had a fixed set of weights.

(ii) Hebb (1949) :- developed first learning rule: i.e if

* 2-neurons are active at same time then the strength b/w them should be increased.

→ In 1950 & 1960's. many researches (Block, minsky, Papert, & Rosenblatt) worked on Perceptron.

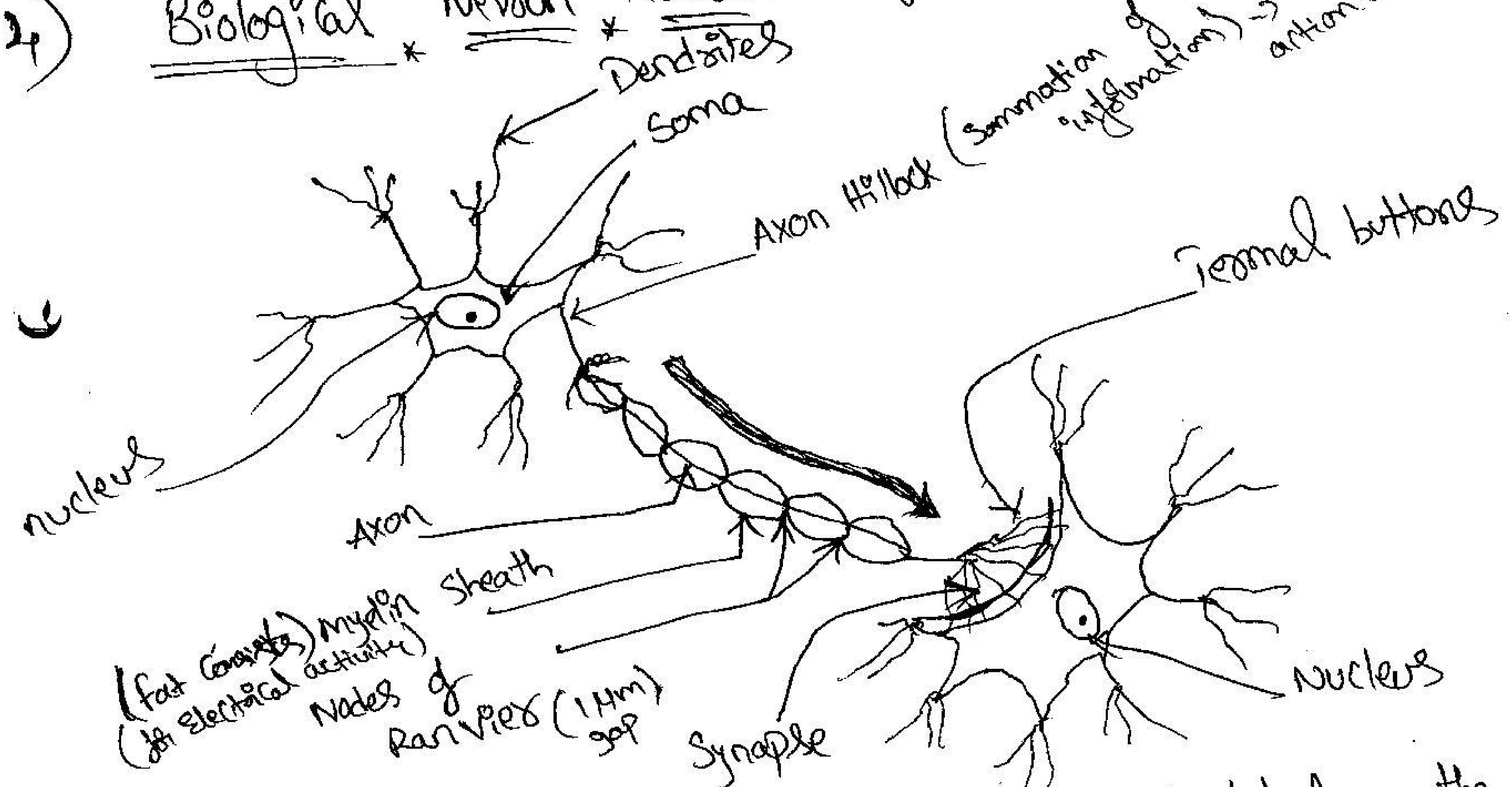
→ The neural n/w model could be proved to converge to correct weights that will solve the problem.

→ The weight adjustments (learning algorithm) used in Perception was found more powerful than learning rule used by Hebb.

(iii) Minsky & Papert (1969) :- showed Perceptron could not learn those functions which are not linearly separable. ③ (I-5)

(iv) In (1985-1986) Parker & LeCun discovered a learning algorithm for multilayer n/w called back propagation that could solve problems that were not linearly seperable.

2) Biological * Neuron * Model :- ~~(Artificial NN)~~ :



(i) Dendrites :- are branching fibers that extended from the cell body (or) soma

(ii) Soma (or) Cell body :- neuron contains the nucleus and other structures, support chemical processing & production of neurotransmitters.

(iii) Axon :- is a singular fiber carries information away from the soma to the synaptic sites of other neurons, muscles, (or) glands

(iv) Axon Hillock :- is the site of summation for incoming information. At any moment, the collective influence of all neurons that conduct impulses to a given neuron will determine whether (or) not an action potential will be intiated at the axon hillock and propagated along the axon.

(v) Myelin Sheath :- Consists of fat containg cells that insolate the axon from electrical activity. (I-6)

-> the insulation acts to increase rate of t_x of signals

-> A gap exists b/w each myelin sheath. cell along the axon.

-> When fat inhibits the ~~elect~~ propagation of electricity the signal jump from one gap to the next.

(vi) Nodes of Ranvier :-

Are gaps about (1 μm) b/w myelin sheath cells along axon are since fat serves as a good insulator,

The myelin sheath speed the rate of t_x of a electrical impulse along the axon.

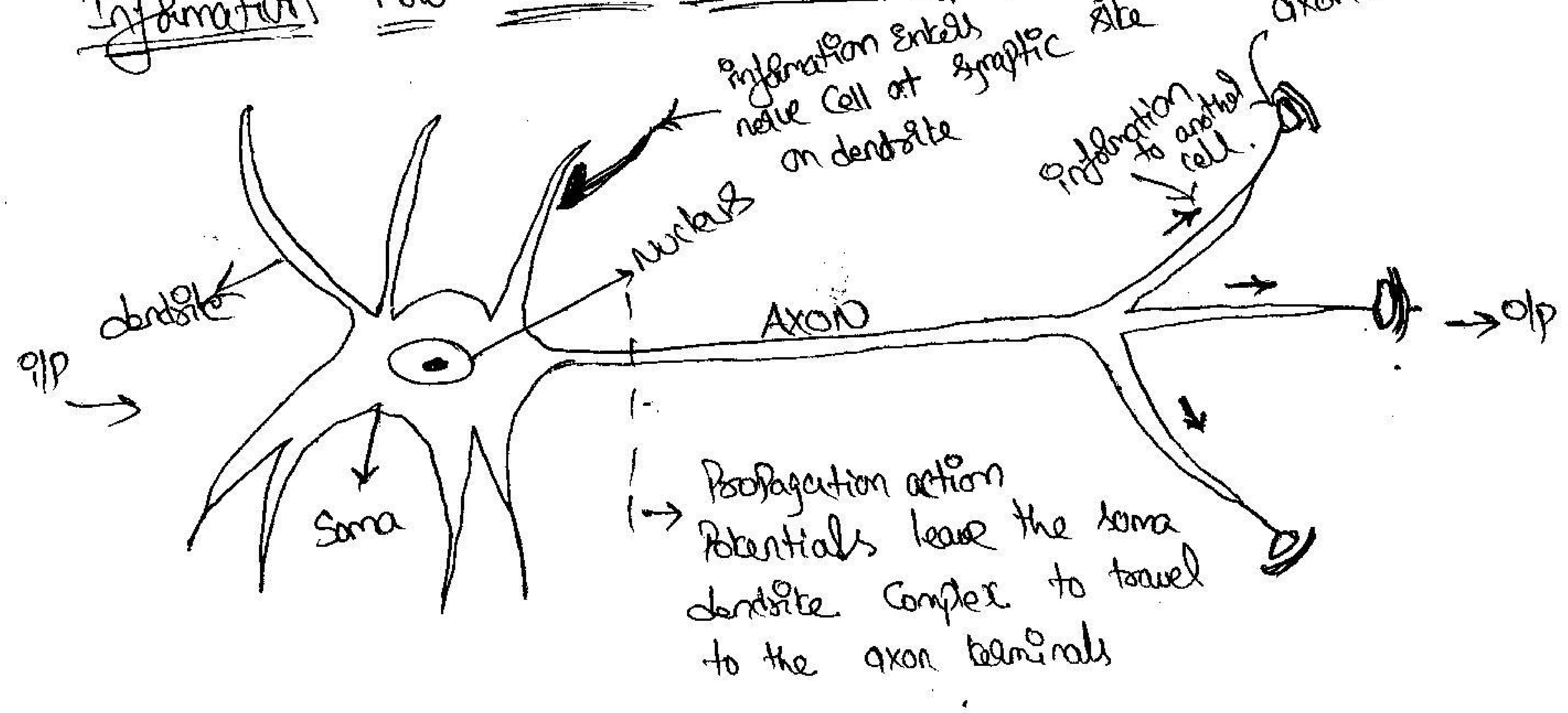
(v). Synapse :- is the point connection b/w two neurons (or) a neuron and muscle (or) a gland.

-> Electro chemical communication b/w neurons takes place at these junctions

(vi) Thermal Buttons :-

A small knobs at the end of an axon that release chemicals called neurotransmitters.

5) Information flow in a Neural Cells :- (Structure of a neural cell in brain)



→ Dendrites receive action from other neurons
→ Soma Process the incoming activation & converts them into o/p activation
→ Axons acts as transmission lines to send activation to other neurons
→ Synapses the junctions allows signal tx b/w the axons and dendrites.
→ The Process of tx is by diffusion of chemical called neuro transmitters

6) Applications of Neural n/w :-

(i) Character Recognition :- This has become very important as handled devices like the Palm Pilot are becoming increasingly popular. Neural n/w can be used to recognize handwritten characters.

(ii) Image Compression :- NN can receive and process vast amount of information at once making them useful in image compression

(iii) Stock market Prediction :- Many factors weigh in whether a given stock will go up on any given day since neural n/w can examine a lot of information quickly and sort it all out, they used to Predict stock Prizes.

(iv) Travelling Salesman Problem :- This NN can solve the traveling salesman problem but only to a certain degree of approximation

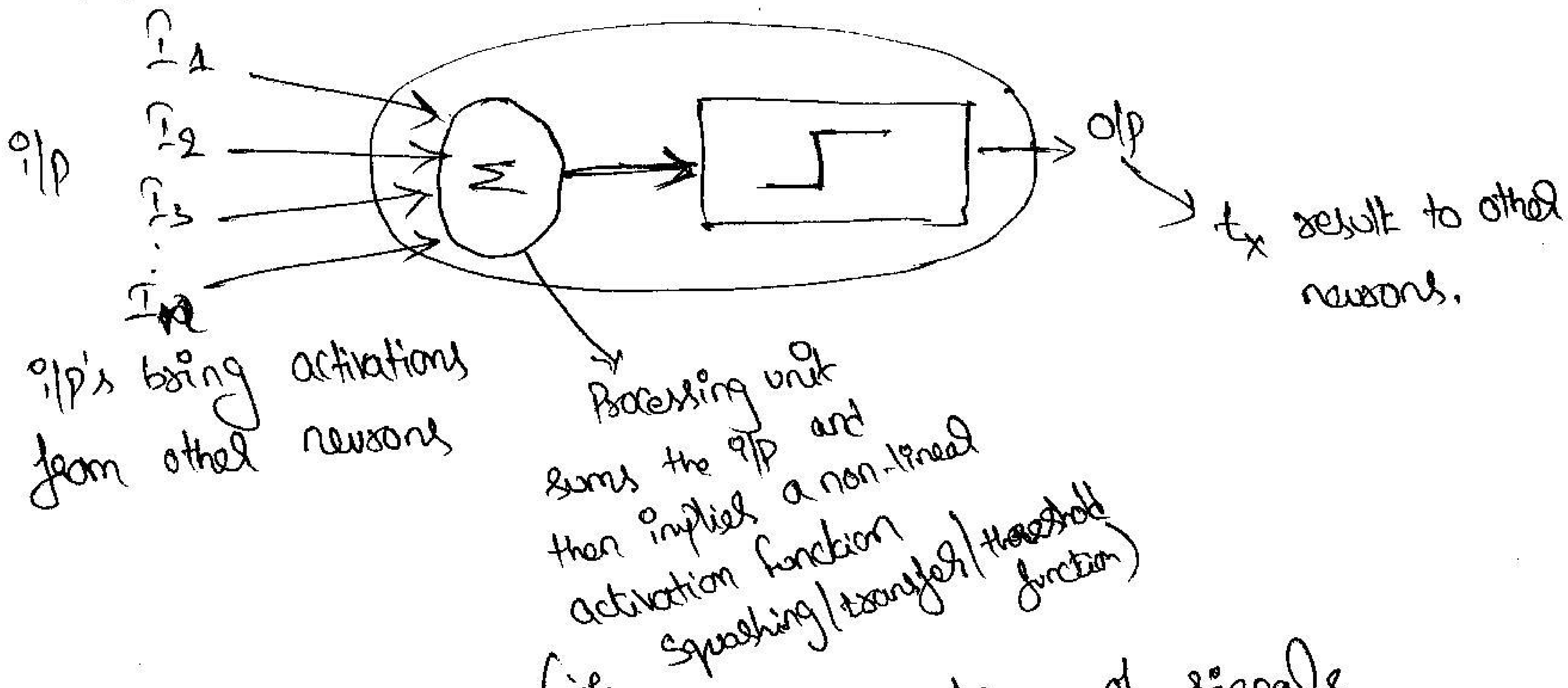
(v) Medicine, Electronic Nose, Security, & Loan applications :- Some applications that their Proof of concept stage with acceptation of NN that will decide wether grant loan (or) not.

(vi) Miscellaneous Applications :- Some interesting (albeit at times a little absurd) applications of NN.

7) Artificial Neuron Model:-

(i) The Mcculloch-Pitts Neuron:-

* This is simplified model of real neuron, known as Threshold neuron.



→ The i/p to neuron arrives in form of signals
→ The signal build up in the cell
→ finally the cell discharges (cell fires) through the o/p
→ The cell can start building up signal again.

8) Notations:-

(i) Scalar:- The no:- of X_i can be added up to give scalar no:

$$S = X_1 + X_2 + X_3 + \cdots X_n = \sum_{i=1}^{n} X_i$$

(ii) vector:- An ordered set of related no:-. Row vectors $(1 \times n)$

$$X = (X_1, X_2 \ldots, X_n) \; ; \; Y = (Y_1, Y_2, Y_3 \ldots Y_n)$$

(iii) Add:- Two vectors of same length added give another vector

$$Z = X + Y = (X_1 + Y_1 \, ; X_2 + Y_2 \, ; X_3 + Y_3 \ldots X_n + Y_n)$$

(iv) Multiply:- Two vector of same length multiplied give scalar

$$P = X \cdot Y = X_1 Y_1 + X_2 Y_2 + X_3 \cdot Y_3 \ldots X_n Y_n = \sum_{i=1}^{n} X_i Y_i$$

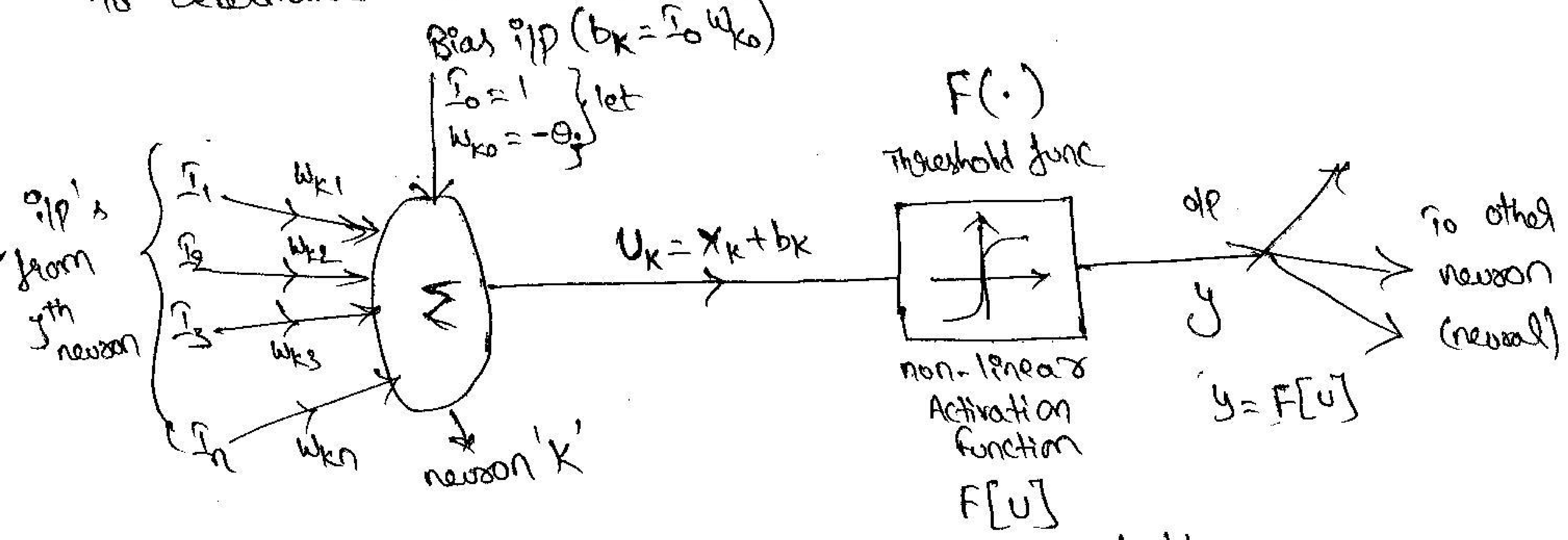
7) **Artificial Neuron & its operation:-**

→ Similar to biological neuron.

→ Artificial neuron receives many i/p's representing the o/p of other neurons

→ Each i/p is multiplied by a corresponding weight ; Analogous to synaptic strength.

→ A weight are summed and passed through a Activation function to determine neuron i/p.



where θ_j = j^{th} neuron activation threshold

If $[I_1, I_2, I_3 \ldots\ldots I_n]$ are inputs from j^{th} neuron which having Synaptic weights of K^{th} neurons $[W_{K1}, W_{K2}, W_{K3} \ldots\ldots W_{Kn}]$ and an bias input ($b_K = I_0 . W_{K0}$) as input for K^{th} neuron.

where let ($I_0 = 1$; & $W_{K0} = -\theta_j$) then.

The o/p of K^{th} neuron is

$$X_K = I_1 W_{K1} + I_2 W_{K2} + I_3 W_{K3} + \ldots\ldots I_n . W_{Kn} \rightarrow (a)$$

$$\Rightarrow X_K = \sum_{K=1}^{n} . I_j . W_{Kj} \rightarrow (b) \quad (\because J = 1, 2 \ldots\ldots n)$$

now Complete o/p of K^{th} neuron U_K with summation of bias i/p (b_K) is

$$U_K = X_K + b_K \rightarrow (c)$$

$$\Rightarrow U_k = \sum_{k=1}^{n} I_j W_{kj} + b_k \rightarrow (d)$$

now let us consider from bias i/p b_k that

θ_{k0} $\boxed{b_k = I_0 \cdot W_{k0}}$ with

Case (i) $W_{k0} = -\theta_j$ & ($I_0 = 1$ & $I_0 = 0$) (if) then.

Eq (d) is

$$U_k = \sum_{k=0}^{n} I_j W_{kj} - \theta_j \rightarrow (e) \quad (\because \text{if } b_k = -\theta_j)$$

&

$$U_k = \sum_{k=0}^{n} I_j \cdot W_{kj} \rightarrow (F) \quad (\because \text{if } b_k = 0)$$

$$\therefore \boxed{U_k = \sum_{k=0}^{n} I_j W_{kj}} \rightarrow (g)$$

The vector form Representation is $\boxed{U_k = I \cdot W} \rightarrow (h)$

To determine the complete o/p of neuron n/w 'y' from

Concept of Mcculloch & Pitts model:-

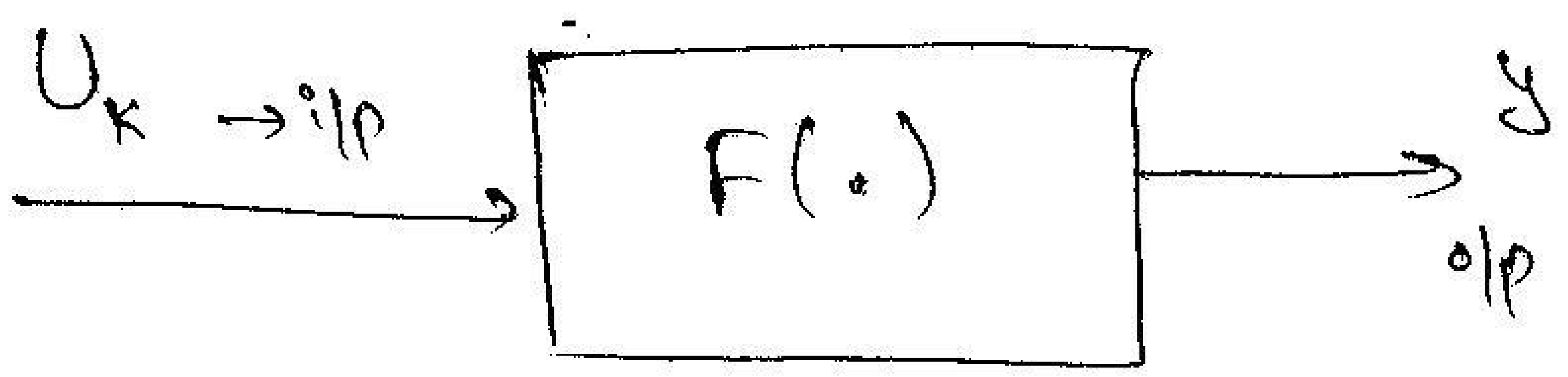
Mcculloch - Pitts Model:- (MP)

Defines neurons has threshold level and once reached that level is fixed.

→ Where MP model has fixed set of weights.

→ To define threshold for neuron the o/p of k^{th} neuron as to be applied for functional representation. In order to defined linear o/p to non linear threshold level representation.

Activation function.

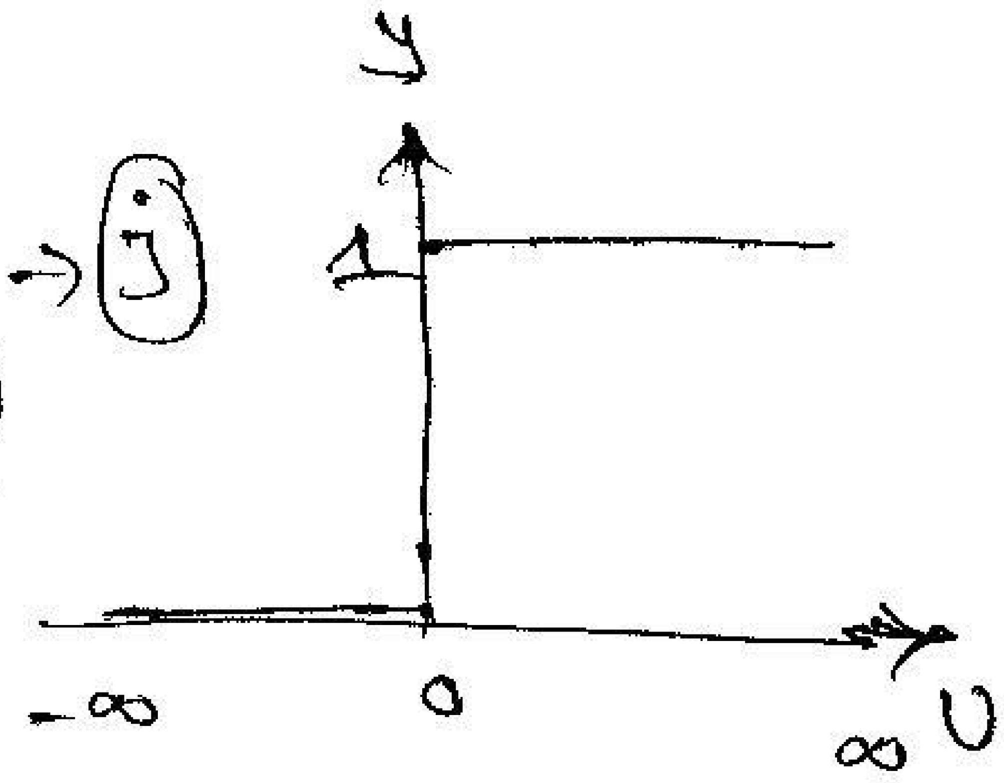


The functional Representation is

$$\boxed{y = F(U)} \rightarrow (i)$$

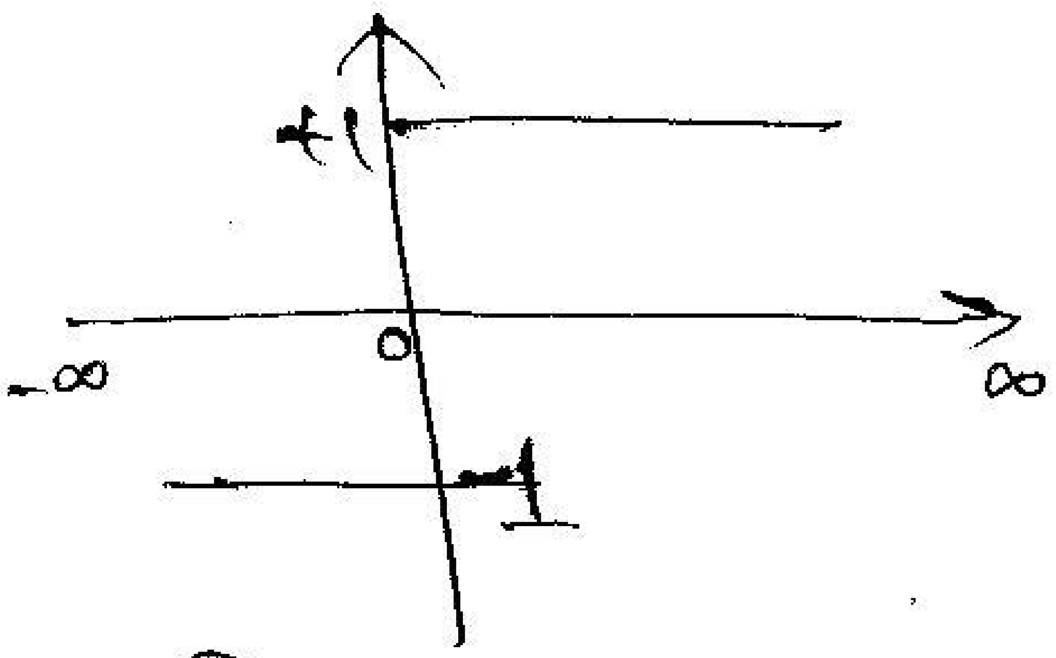
where an i/p & o/p mapping from U to Y.

(i) Threshold function :-



if $U_K \geq 0$ then $y = 1$ (+ve)
$U_K < 0$ then $y = 0$ (-ve) others $\Big\} \rightarrow (j)$

(ii) Sigmoid function (or) (s-shape function) :-
to define o/p is at +1 (or) -1



if $Sgn(U_K) \geq 0$ then $y = 1$
$Sgn(U_K) < 0$ then $y = -1$ $\Big\} \rightarrow (K)$

∴ The total o/p mp-model 'y' is represented as

$$\boxed{y = Sgn\left(\sum_{K=1}^{n} i_j W_{kj} - \theta\right)} \rightarrow (l)$$

here $\theta \rightarrow$ the neuron activation threshold of k^{th} neuron.

if $\sum_{K=1}^{n} i_j W_{kj} \geq \theta$ then o/p $y = 1$

$\sum_{K=1}^{n} i_j W_{kj} < \theta$ then o/p $y = 0$ $\Big\} \rightarrow (m)$ //.

To identify the o/p $y = +1$ (or) -1 then. from above (m) Eq & (l) and comparing with Eq (K) we define the o/p of neuron is '+1' (or) '-1'

Note :-

→ As i/p applied from j^{th} neuron are summed with weighted i/p and bias i/p and given an o/p which is said to be function o/p of k^{th} neuron represented 'linear function'

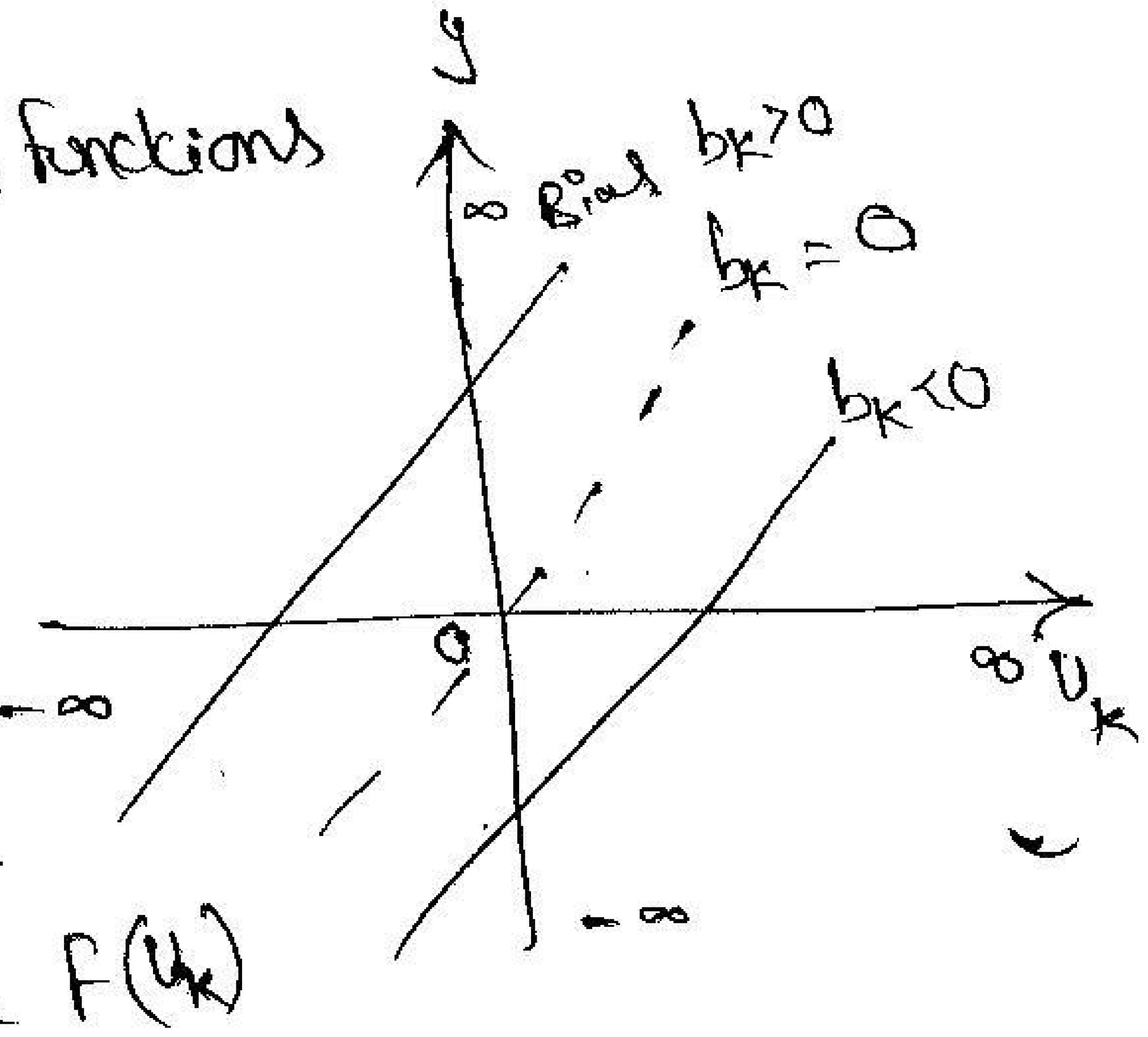
→ the o/p of summation of i/p & synaptic weights are taken as i/p to Activation function.

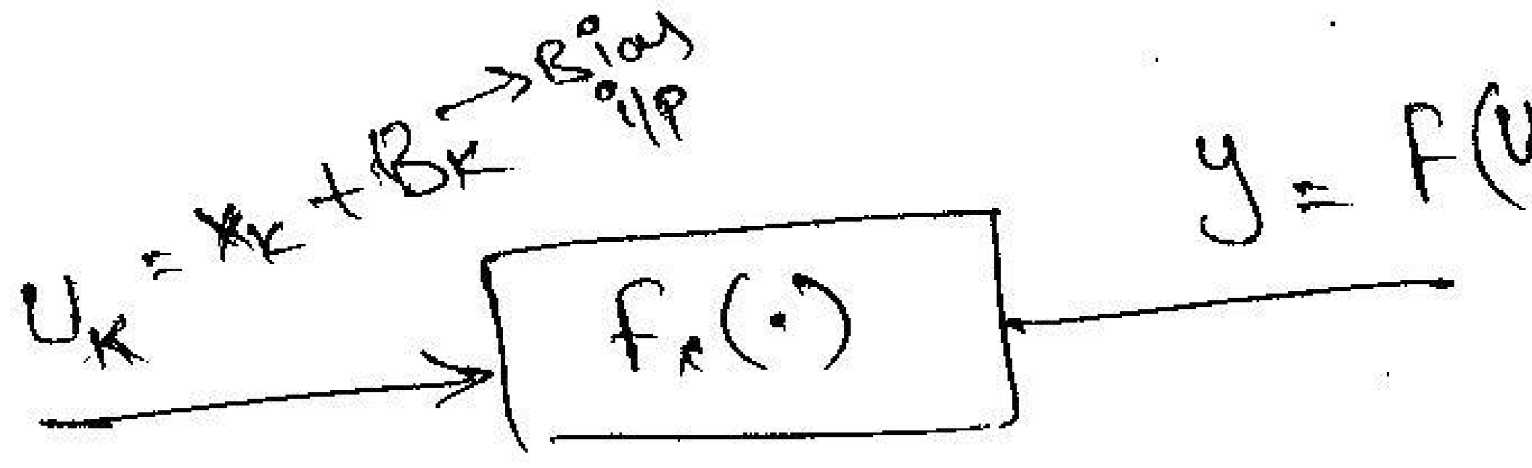
→ in order to filter the non-linear function of total o/p response of Artifical neuron n/w. either (+ve (or) -ve).

8) Activation function :- (F)

Perform a mathematical operations on the signal o/p. the most Common activation function are

→ threshold function } → Pre-Activation functions
→ Linear function
→ Piecewise Linear function.
→ Sigmoidal (S-shape) function → unipolar, bipolar
→ Tangent hyperbolic function.





$$X_k = \sum_{j=1}^{n} x_j W_{kj} - \theta_j$$

θ_j → Threshold internal of j^{th} neuron.

→ The Sigmoid & hyperbolic tangent functions perform well for prediction and Process forecasting types of Problems

→ Radial bias function Proves more effective for n/w & highly recommeded function for those Problems involving faultdiagnosis.

→ Hard limiter for classification Problems.

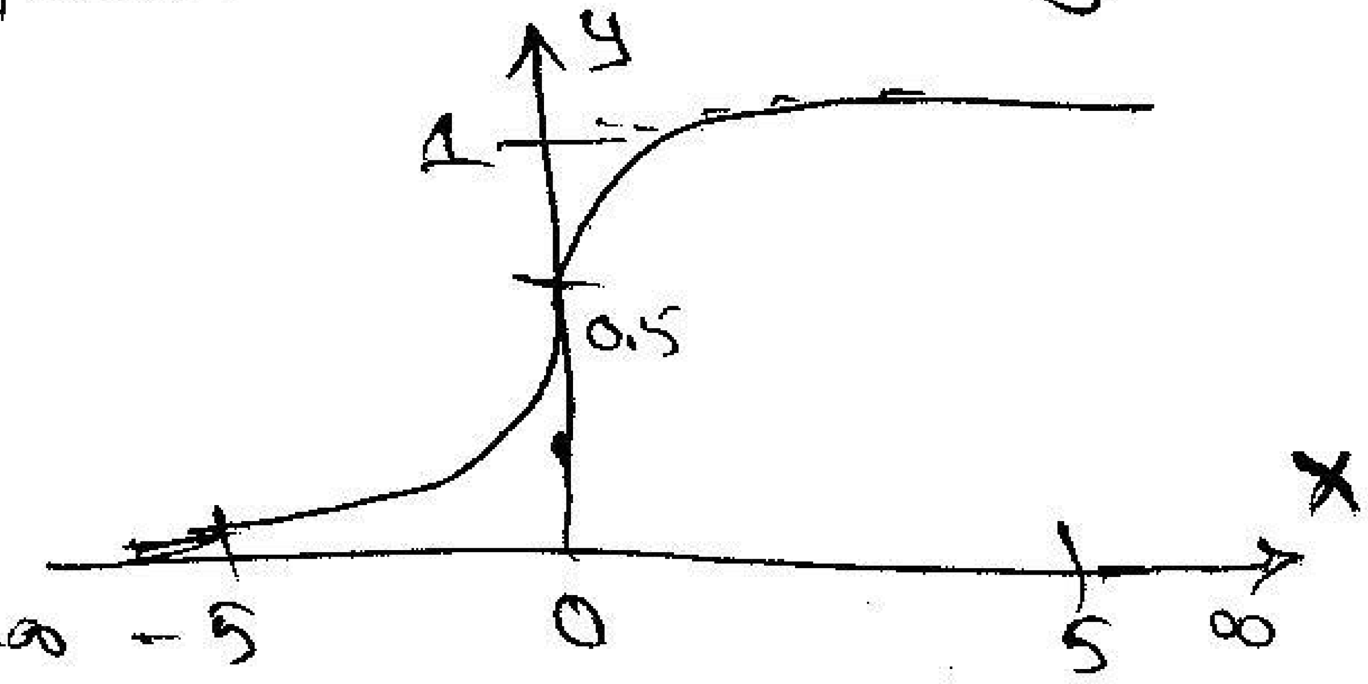
Types of Activation functions:-

→ behavioral of artifical neuron depends on both synaptic weights and activation function can be represented ; log, e^x, x^3 ... etc., mathematically.

→ Sigmoid functions are commonly used activiation functions in multilayered feed forward neural n/w.

(i) Sigmoid function (unipolar sigmoid) :- used in (Prediction & Process forecasting).

$$\boxed{y(x) = F(x) = \frac{1}{1+e^{-x}}} \rightarrow (a)$$



range is $0 < y < 1$

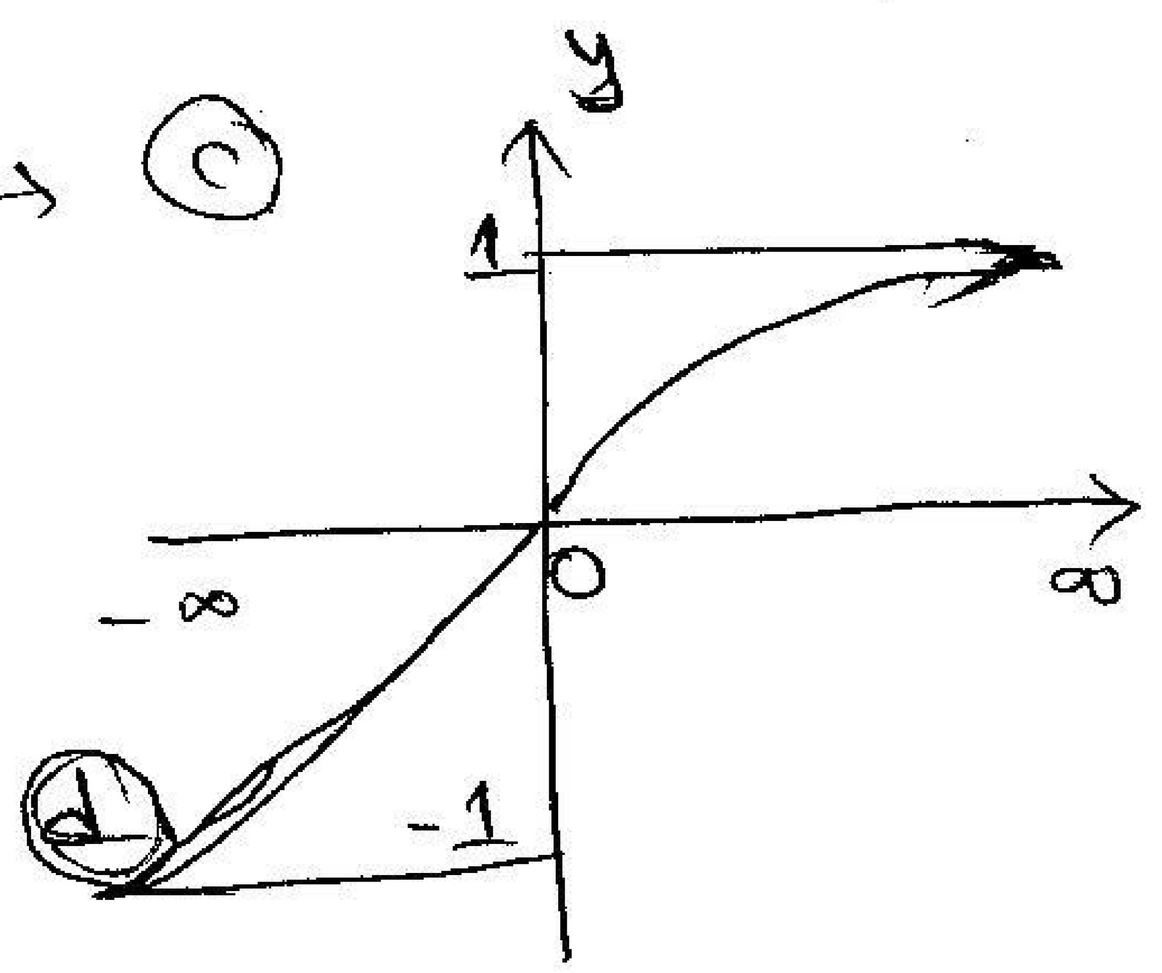
The derivative of function is

$$\boxed{y'(x) = F'(x) = F(x)\,(1 - F(x))} \rightarrow (b)$$

$$\because x = \sum_{j=1}^{n} x_j w_{kj} - \theta_j$$

(ii) Hyperbolic tangent (bipolar sigmoid) function:- (Prediction & Process forecasting)

$$\boxed{y(x) = F(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}} \rightarrow (c)$$



range of signal is $-1 < y < 1$

derivative as

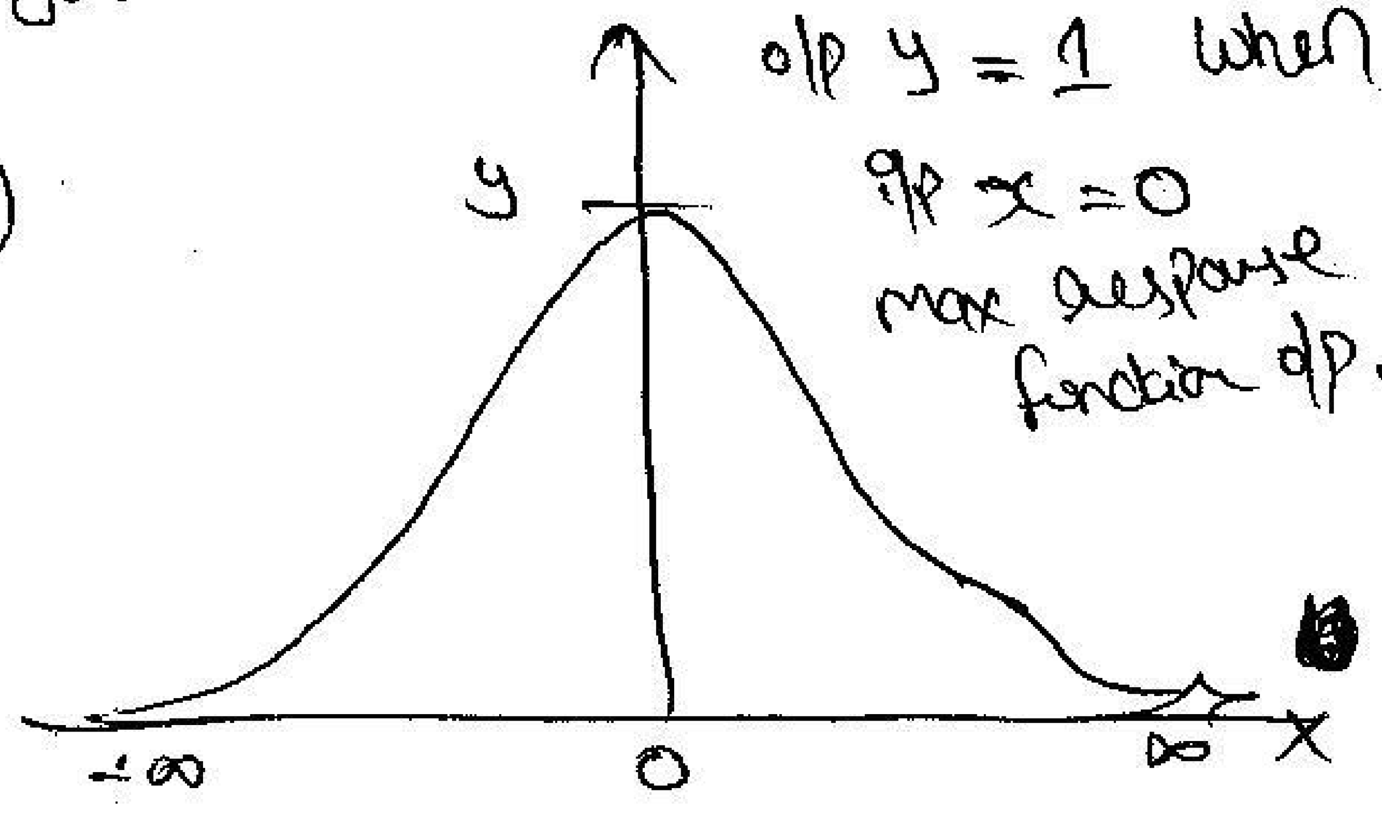
$$\boxed{y'(x) = F'(x) = 1 - [F(x)]^2} \rightarrow (d)$$

(iii) Radial basis function :- (used for fault diagnosis problems)

The Gaussian function is mostly commonly used in radially Symmetric function, the characteristics function is in mathematic description.

$$\boxed{y(x) = F(x) = e^{(-x^2/2)}} \rightarrow (e)$$

$0 < y < 1$



o/p $y = 1$ when i/p $x = 0$ max response function o/p.

derivative

$$y'(x) = F'(x) = -x \cdot e^{-x^2/2} \rightarrow (f)$$

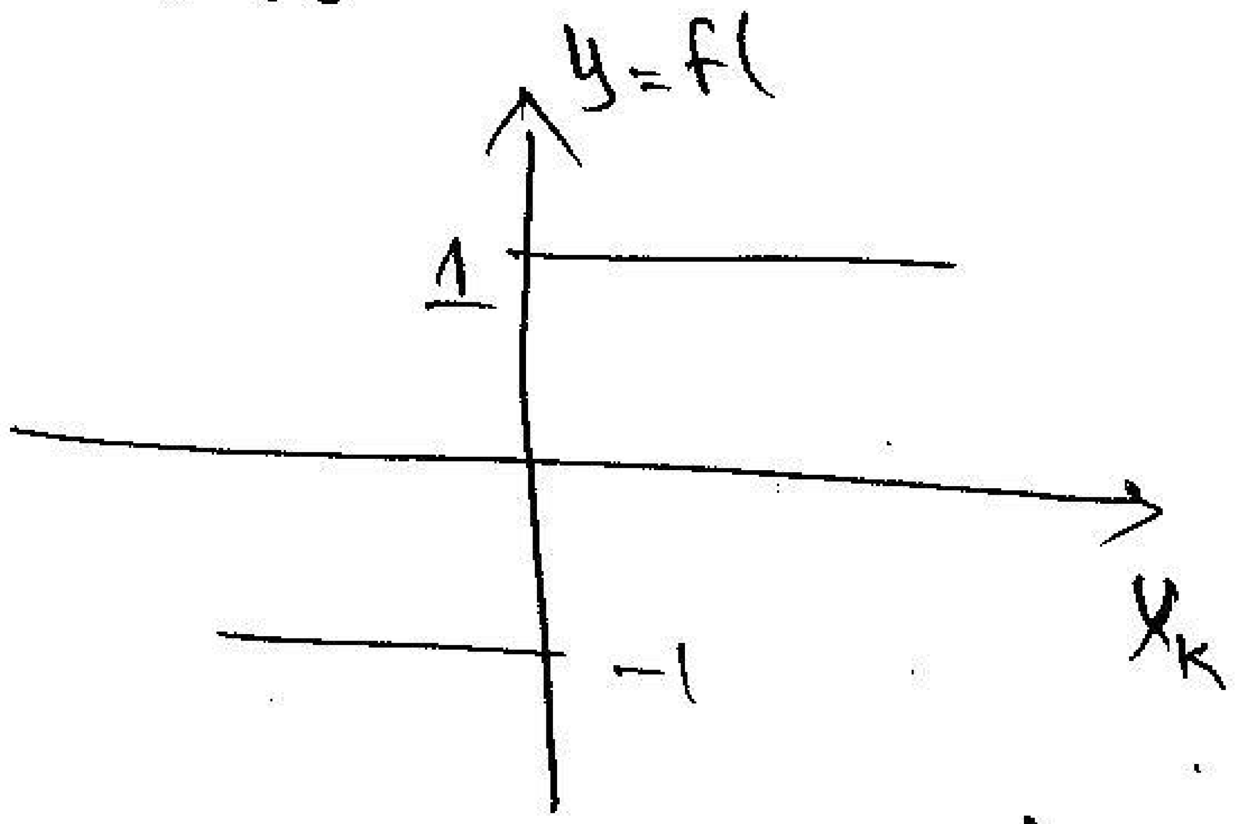
(iv) Hard limiter:- (classification of problems):-

It is mostly used in classification of patterns

$$F(X_k) = \text{Sign}(X_k) = \begin{cases} +1 & v \geq 0 \\ -1 & v < 0 \end{cases}$$

This function is not differentiable

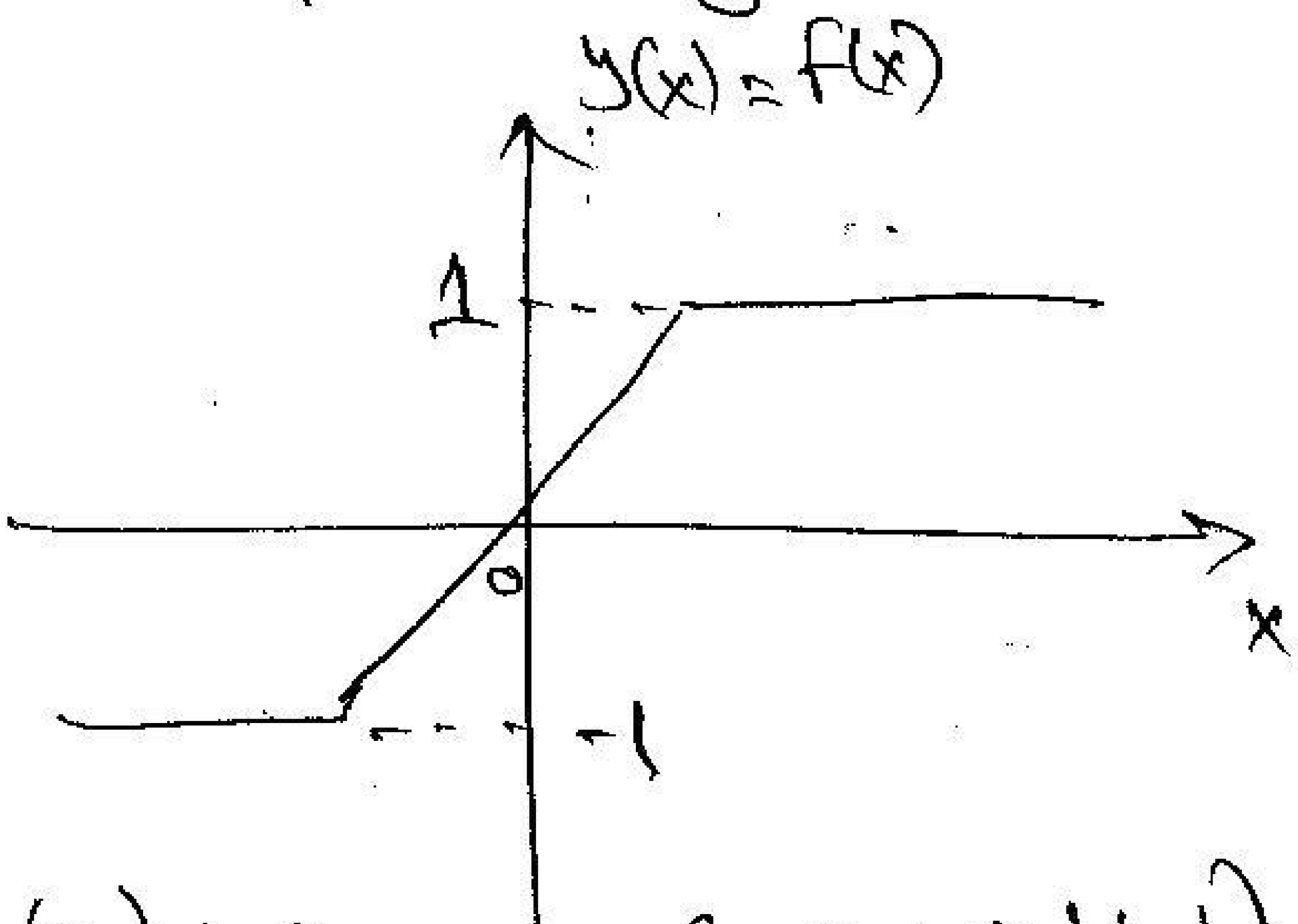
∴ It cannot be used for continuation type of applications.



(v) Piecewise linear function:- (called saturating linear func)

This activation function is called saturating linear function and can have either a binary (or) bipolar range for saturation limits of o/p.

→ The mathematical model is



$$y(x) = F(x) = \begin{cases} -1 & \text{if } \text{Sgn}(x) < 0 \rightarrow (\text{-ve weighted}) \\ \text{Sgn}(x) & \text{if } -1 \geq x \geq 1 \rightarrow (\text{proportional value to } -1 \text{ to } 1) \\ 1 & \text{if } \text{Sgn}(x) \geq 0 \ (\text{+ve weighted sum}) \end{cases}$$

Sigmoidal function (S-shape function) :-

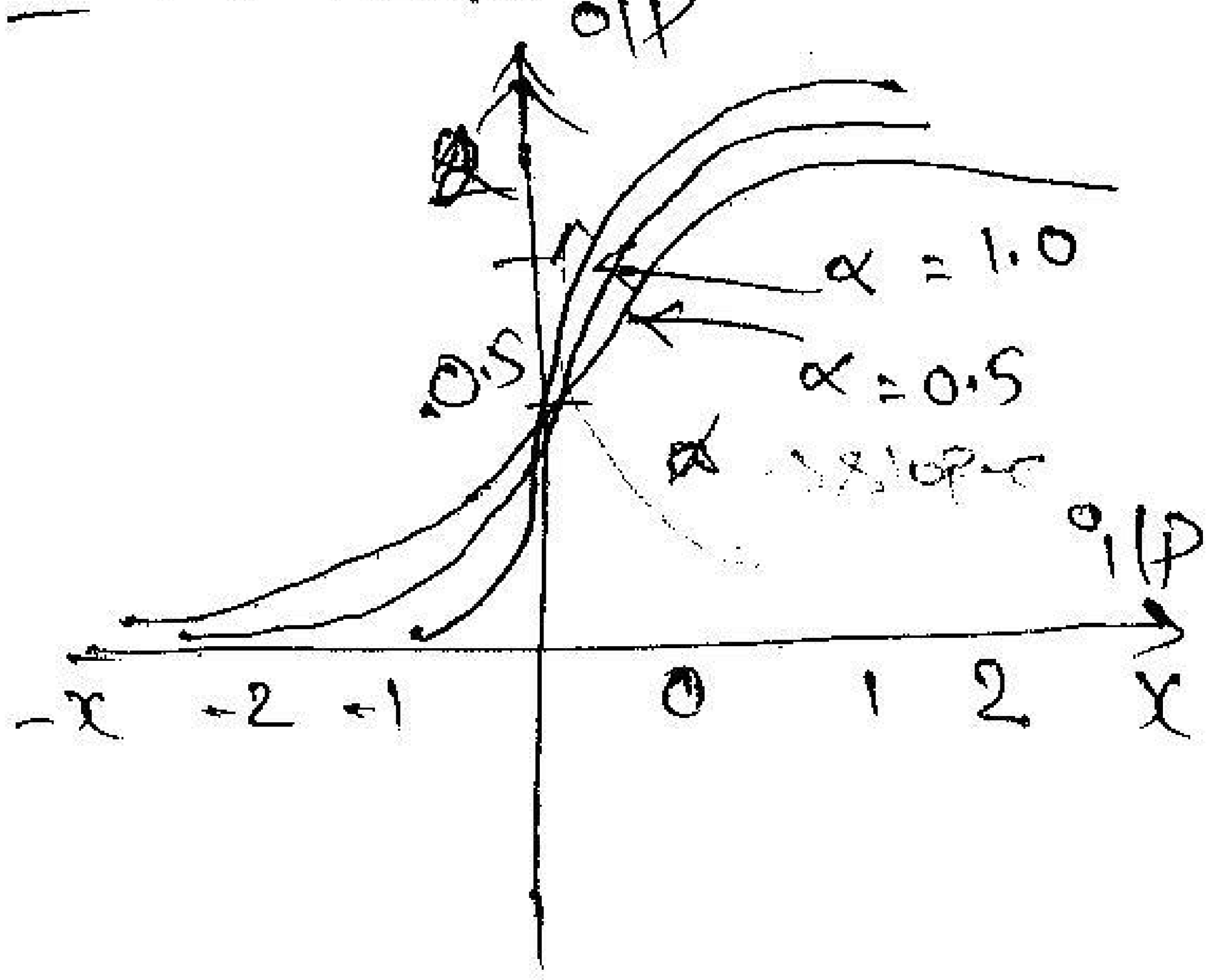
The non-linear curved S-shape function is called the sigmoid function.

→ It is most common type of activation used to construct neural n/w.

→ The mathematically well behaved differentiable and strictly increasing function.

$$y = f(x) = \frac{1}{1+e^{-\alpha x}} \; ; \; 0 \le f(x) \le 1$$

$$y = f(x) = \frac{1}{1+e^{(-\alpha x)}} \; ; \; 0 \le f(x) \le 1$$



This explained as

$y = 0$ for large -ve i/p values

$y = 1$ for large +ve value with.

α → Slope Parameter.

λ → also be represented as Parameter.

By varying α - different shapes of function can be obtained which adjusts the abruptness of the function as it changes b/w the two asymptotic values.

9) **Neural N/w Viewed as Directed graphs:-**

Signal flow graph with ~~loops~~ a well-defined set of rules were originally developed for linear n/w.

The presence of non-linearity in model of a neuron limits the scope of their application to neural n/w.

→ A signal flowgraph is a n/w of directed links (branches) that are interconnected at certain points called nodes. A typical node j has associated node signal x. A typical direct link originates at node j and terminates on node 'k'. It has associated transfer function (or) transmittance that specifies manner of signal.

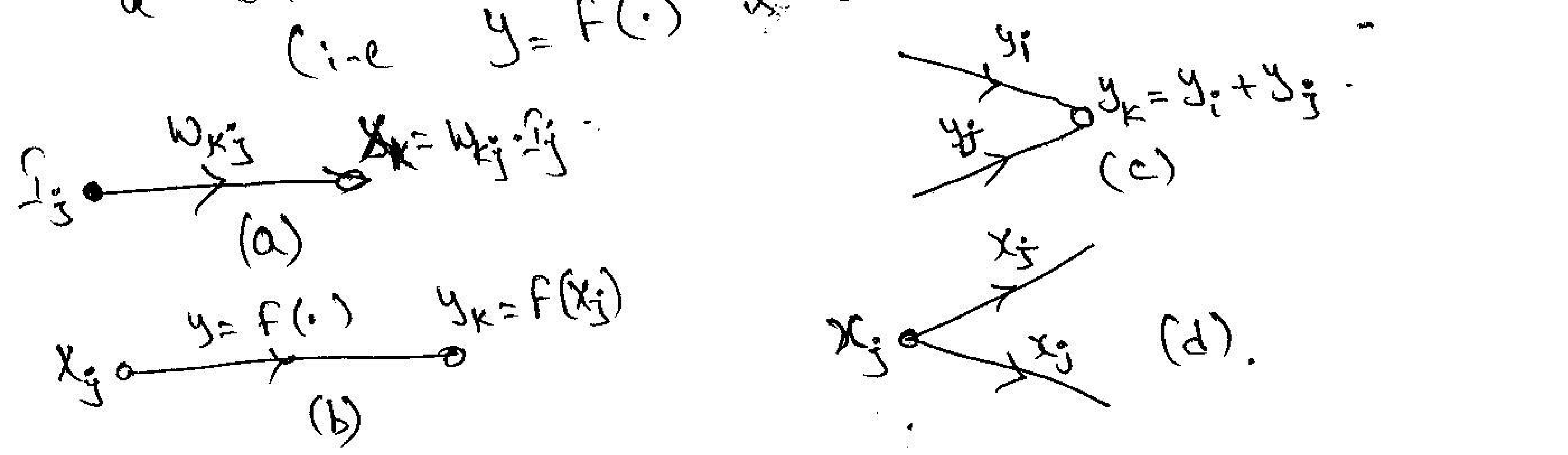
→ The flow of signals in various parts of the graph is dictated by three rules.

Rule : 1 :- A signal flows along a link only in direction defined by arrow on the link.

2 - different link types → Synaptic link ✓ → I/P & o/p ; → Activation link. ✓ o/p

i) Synaptic link :- behavior is governed by linear i/p - o/p relation (i.e x_j multiplied with w_{kj} gives y_k o/p).

ii) Activation link :- behavior is governed in general by a non-linear i/p - o/p relation (i.e $y = F(\cdot)$ is non linear activation function).



Rule - 2:-

A node signal Equals the algebraic sum of all signals Entering Pertient node via incoming links (fig-c)

Rule :- 3:-

The signal at a node is transmitted to Each outgoing link orginating from that node with transmission being entirely independent of transfer functions. of outgoing links. (fig-d)

is Case of synaptic divergence (or) fan-out.

→ A neural n/w is a directed graph consisting of nodes with interconnecting synaptic and Activation link are characterized by four properties:

i) Each neuron is represented by set of linear synaptic links an Externally applied bias and possibly non-linear activation link. The bias is represented by synaptic link connected to i/p fixed at +1.

ii) The synaptic link of neuron weight their respective i/p signal.

iii) The weighted sum of i/p signals defines the induced local field of neuron in question.

iv) The activation link squashes the induced local field of neuron to produce o/p.

3 - Graphical Representation of neuron:-

Providing functional description of n/w

✓ i) block diagram - Providing functional description of n/w

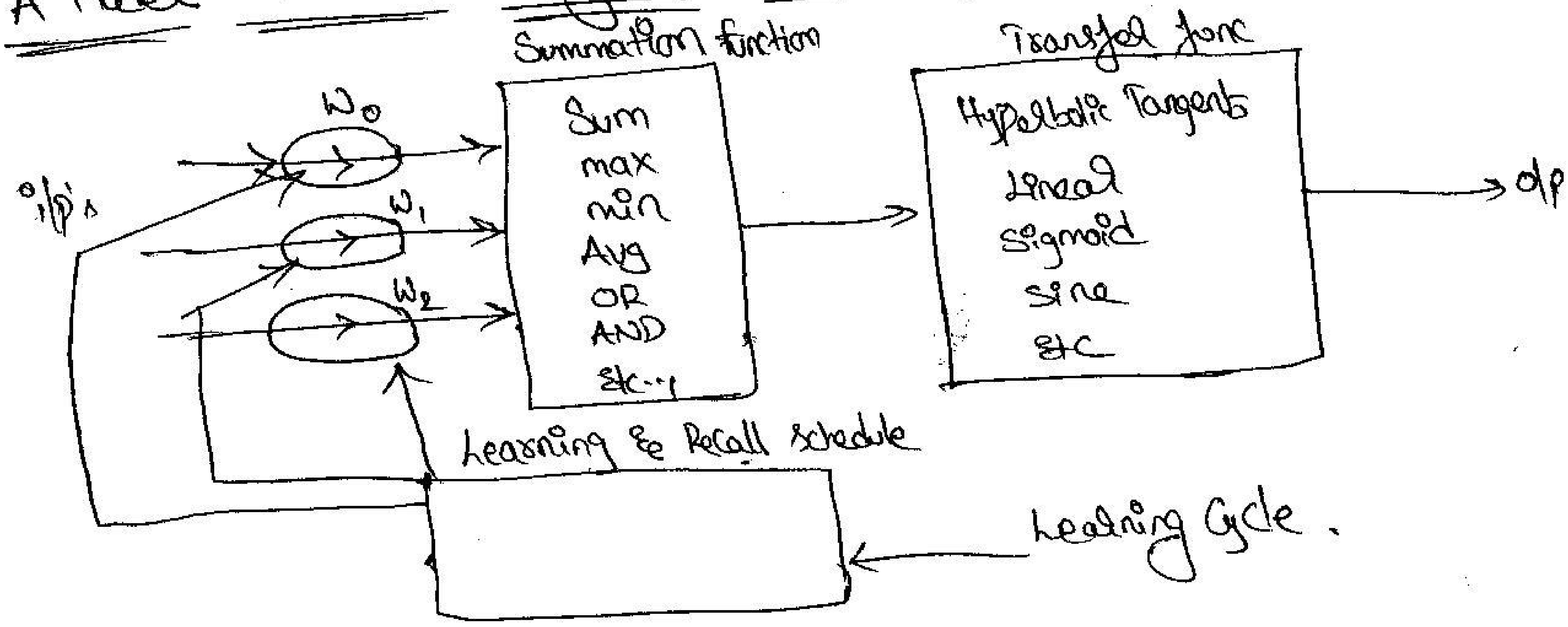
✓ ii) Architectural graph, describing the n/w Layout.

✓ iii) Signal flow graph; Providing a complete description of signal flow in the n/w.

10) History:-

Author	Year	Idea	Implementation.
1) Mcculloch - Pitts	1943	neuron has threshold level once reached neuron fires	Ist Artificial neuron model.
2) Hebbs	1949	If 2-neurons are active same time then strength b/w them should be increased	Ist learning rule
3) Rosen blatt	1950 & 1960	Converge to Correct weights & solve Problems	Introduced Perceptron (learning algorithm)
4) Minsky & Papert	1969	Perceptron could not learn function which are not linear	—
5) Parker & LeCun	1985-1986	Learning algorithm for multilayer called Back Propagation	

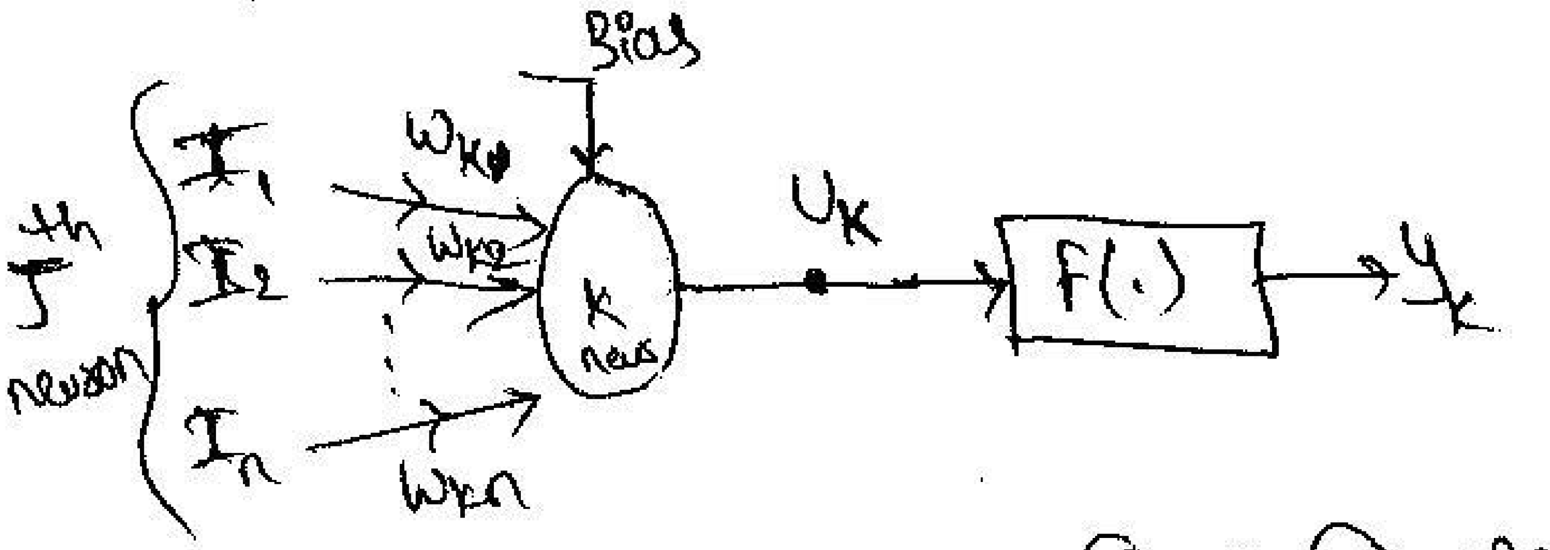
11) A Model of a Processing Element (Electronic implementation):-



Why to use Linear Model and Non-linear Model in neural n/w:-

for the Concept of single layer (or) Multilayer neural n/w to identify the Active neurons in the network based on Activation functional process it is identified.

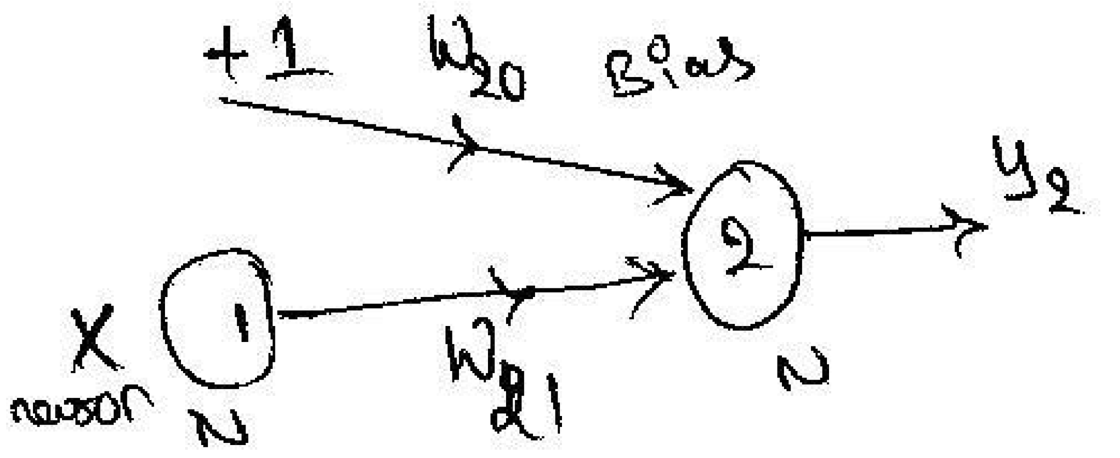
Linear model:- $(y_k = U_k)$ → used in data fit model.



$$U_k = \sum_{I=1}^{n} I_j \cdot W_{kj}$$

$$U_K = I_1 W_{k1} + I_2 W_{k2} + I_3 W_{k3} \cdots \cdots I_j W_{kj}$$

if 2-node's of ① & ② are Considered.



$$y_2 = W_{21} X + W_{20}$$

X → i/p from neuron 'x'

W_{21} → slope

W_{20} → bias is intercept

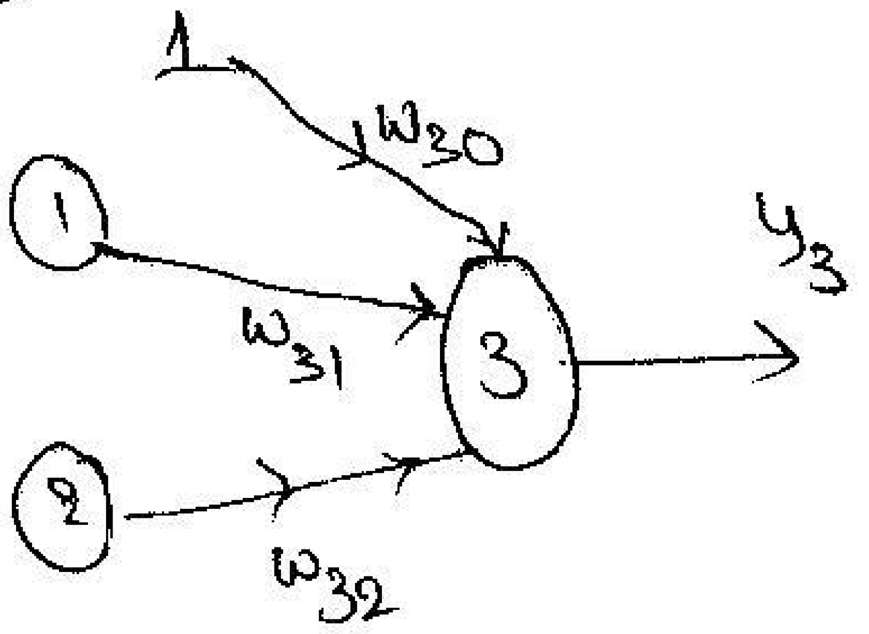
⇓

$y = mx + c$

m → slope / Gradiant (direction)

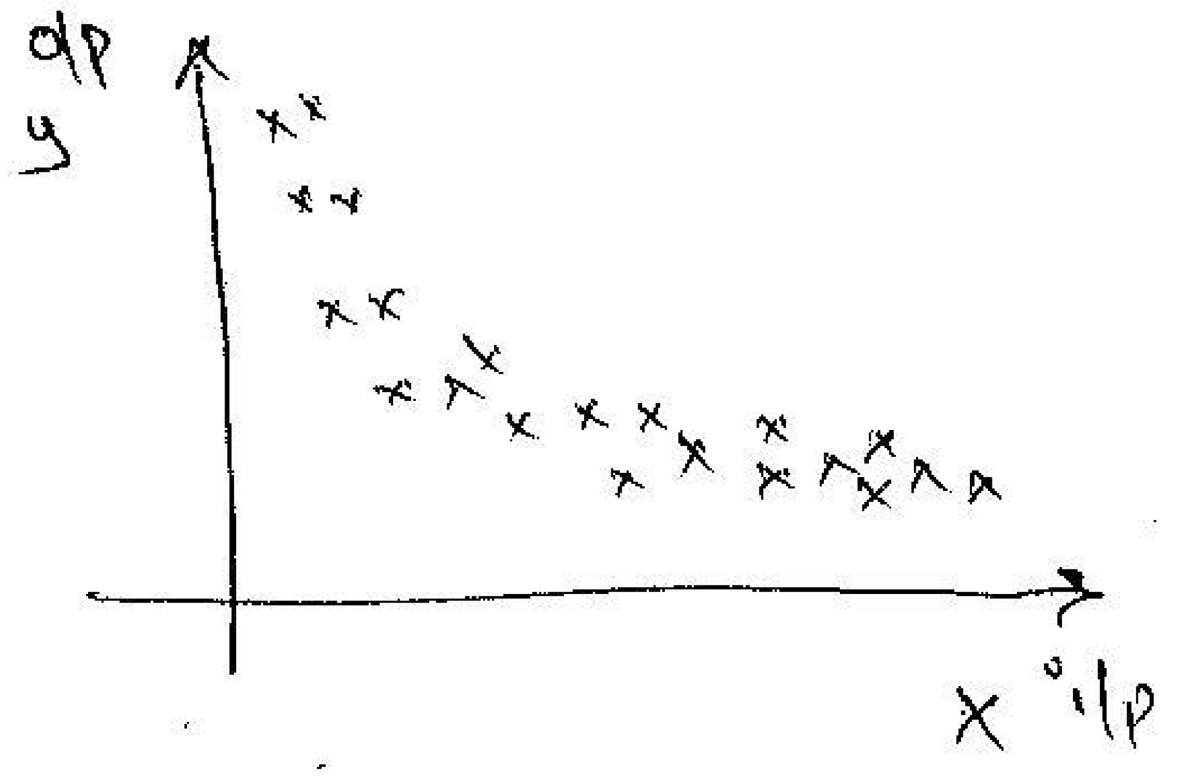
c → intercept

if 3-nodes N_1 & N_2 & N_3 are Considered

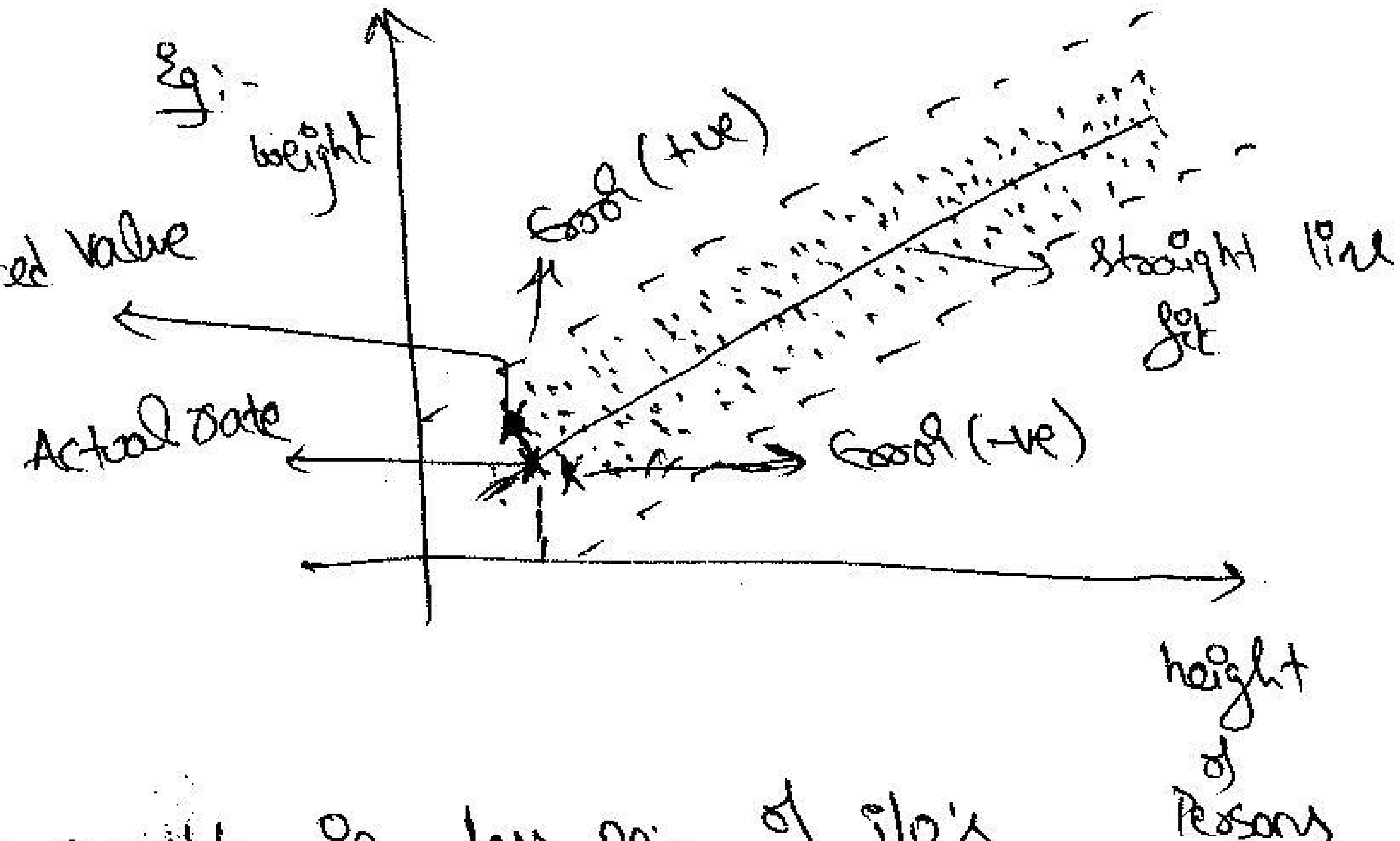


$$y_3 = W_{30} + W_{31} \cdot x_1 + W_{32} \cdot x_2$$

y_3 is dependent upon 2 i/p x_1 & x_2



Eg:-



Note:-

→ The straight line curve fitting is possible in less no:- of i/p's.

→ The " " " " is not Appropriately in more no:- of i/p's.

so we Comeup with (non-linear model.)

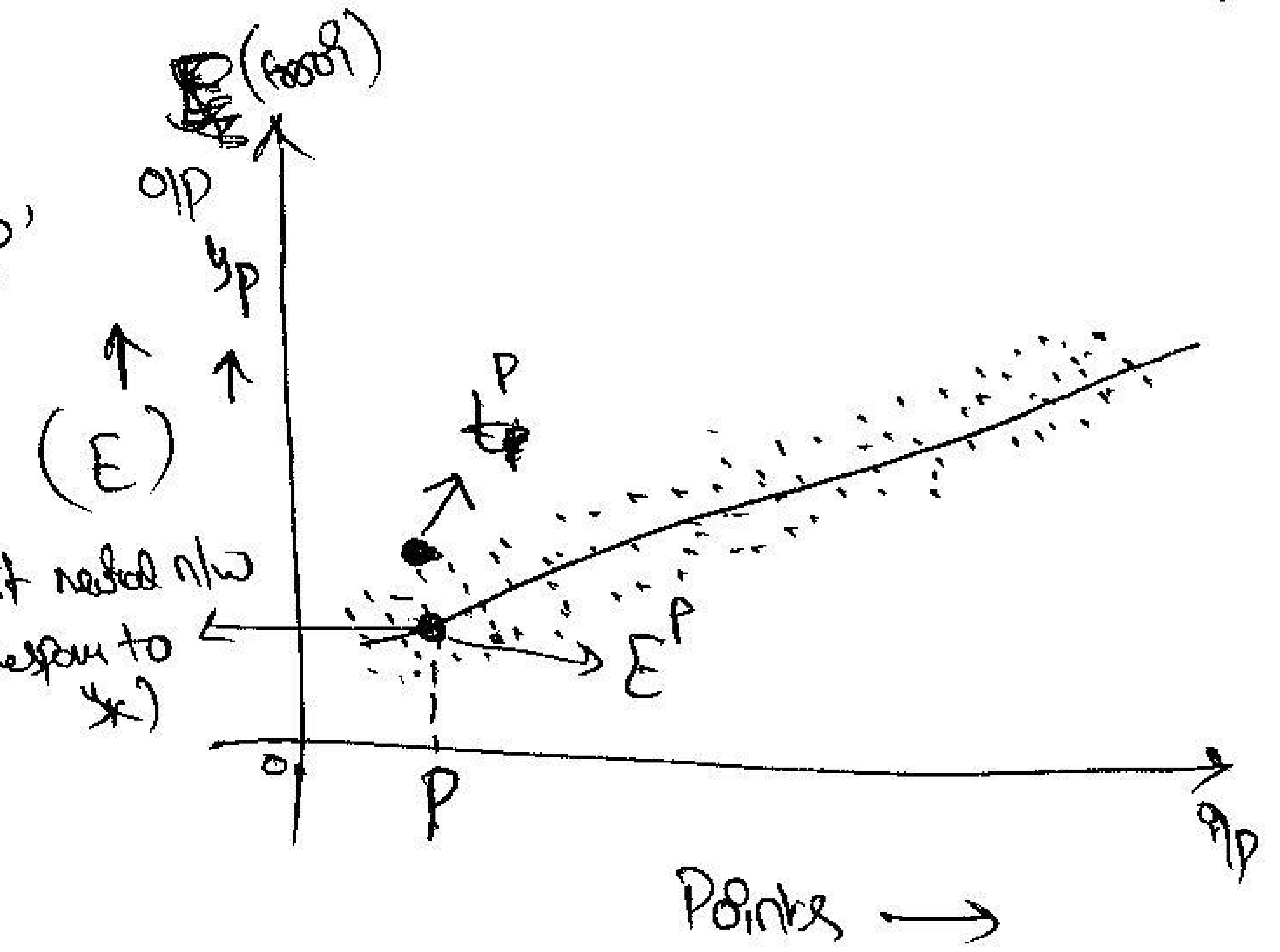
↳ Binary | Step | unipolar | Threshold | hard limit | Sigmoid functions

(i) Represent an Error :- (E)

let E^P → Error for Particular Point 'P'

t^P → target o/p for Point 'P'

y^P → Fitted o/p (or) neural n/w Response curve for neural n/w Job (Respon to 'x')



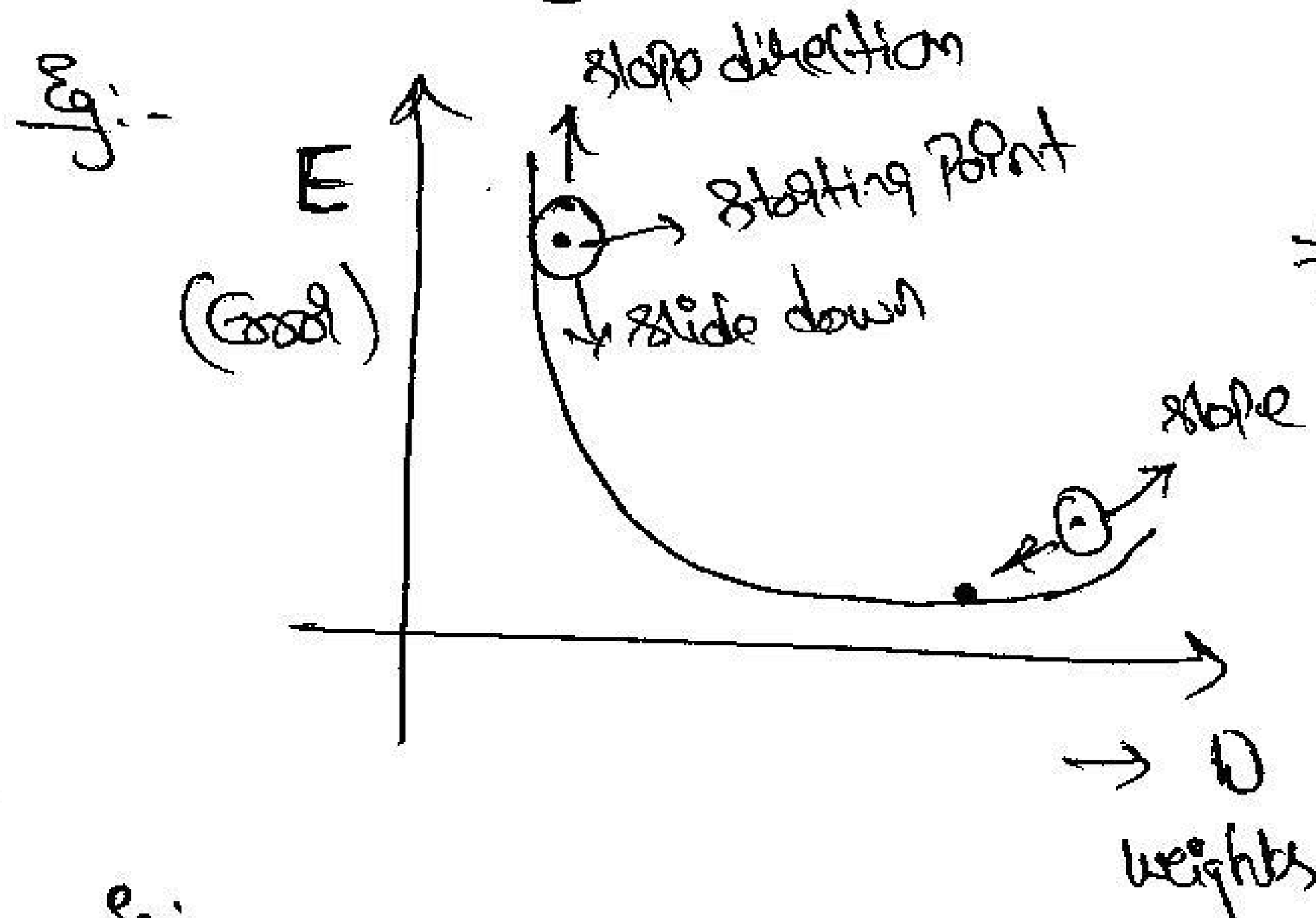
$$\therefore \quad E^P = (t^P - y^P)^2 \rightarrow (1)$$

∴ Total error of all set up points are curve is

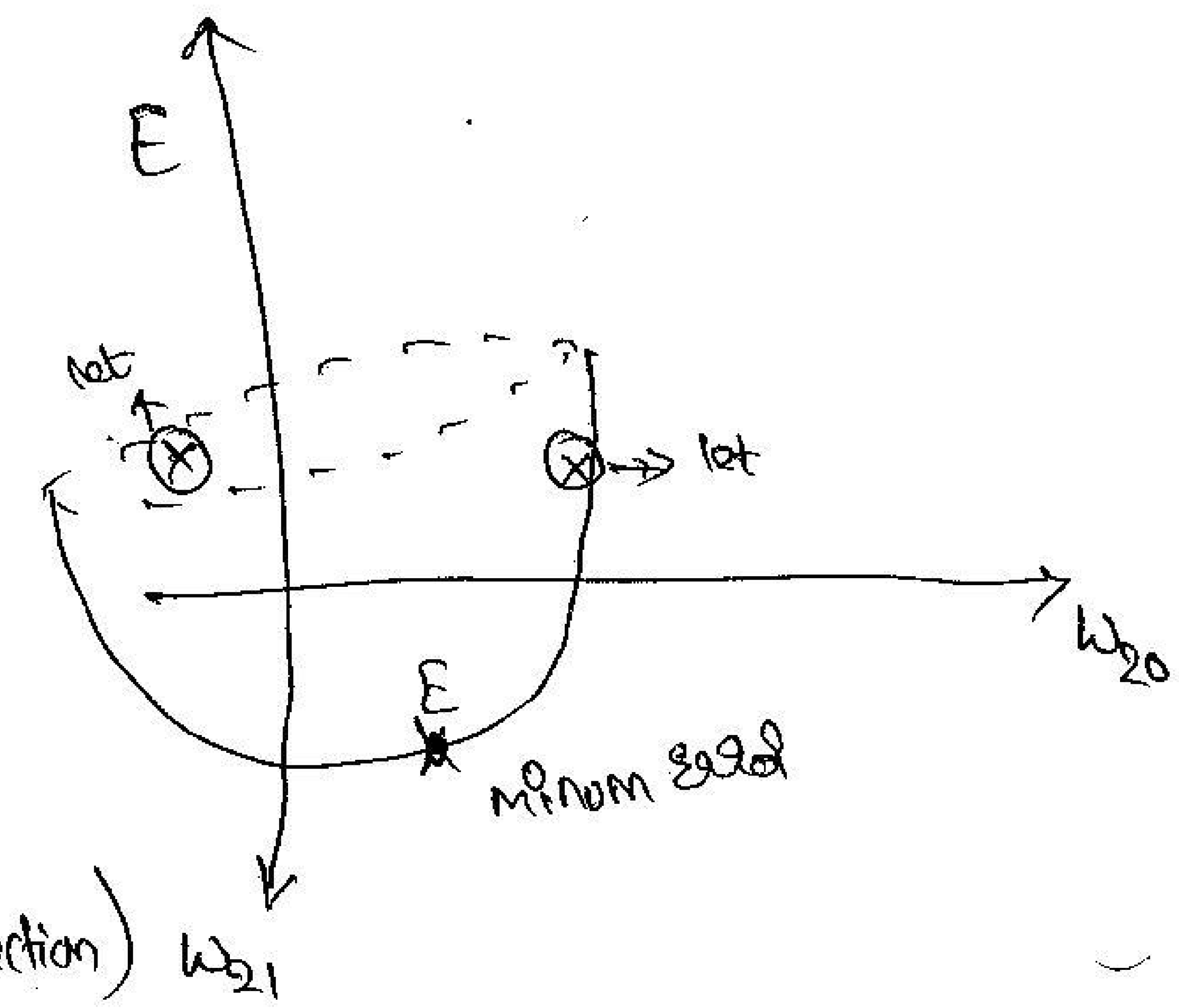
$$E = \sum_P E^P = \sum_P (t^P - y^P)^2 \rightarrow (2)$$

The main object is to find a minimum error in slope plane.

Eg:-



$$\Rightarrow \frac{\partial E}{\partial \omega}$$



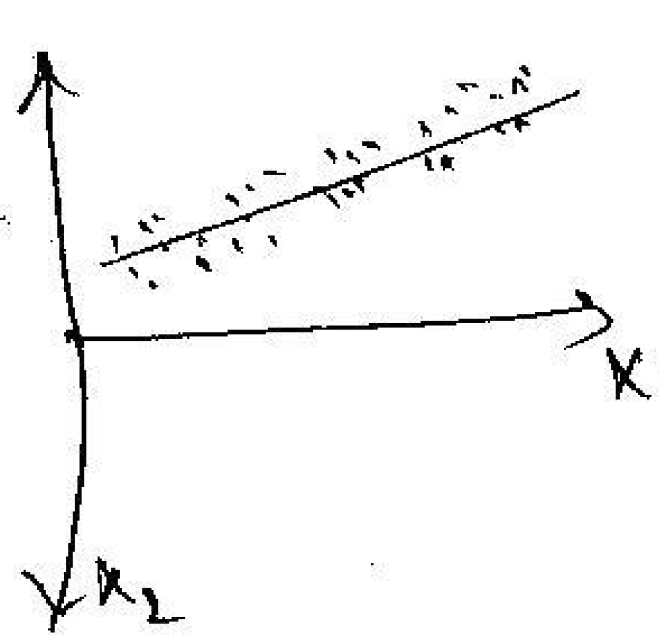
Eg:- (Ball in net rolled across slope direction)

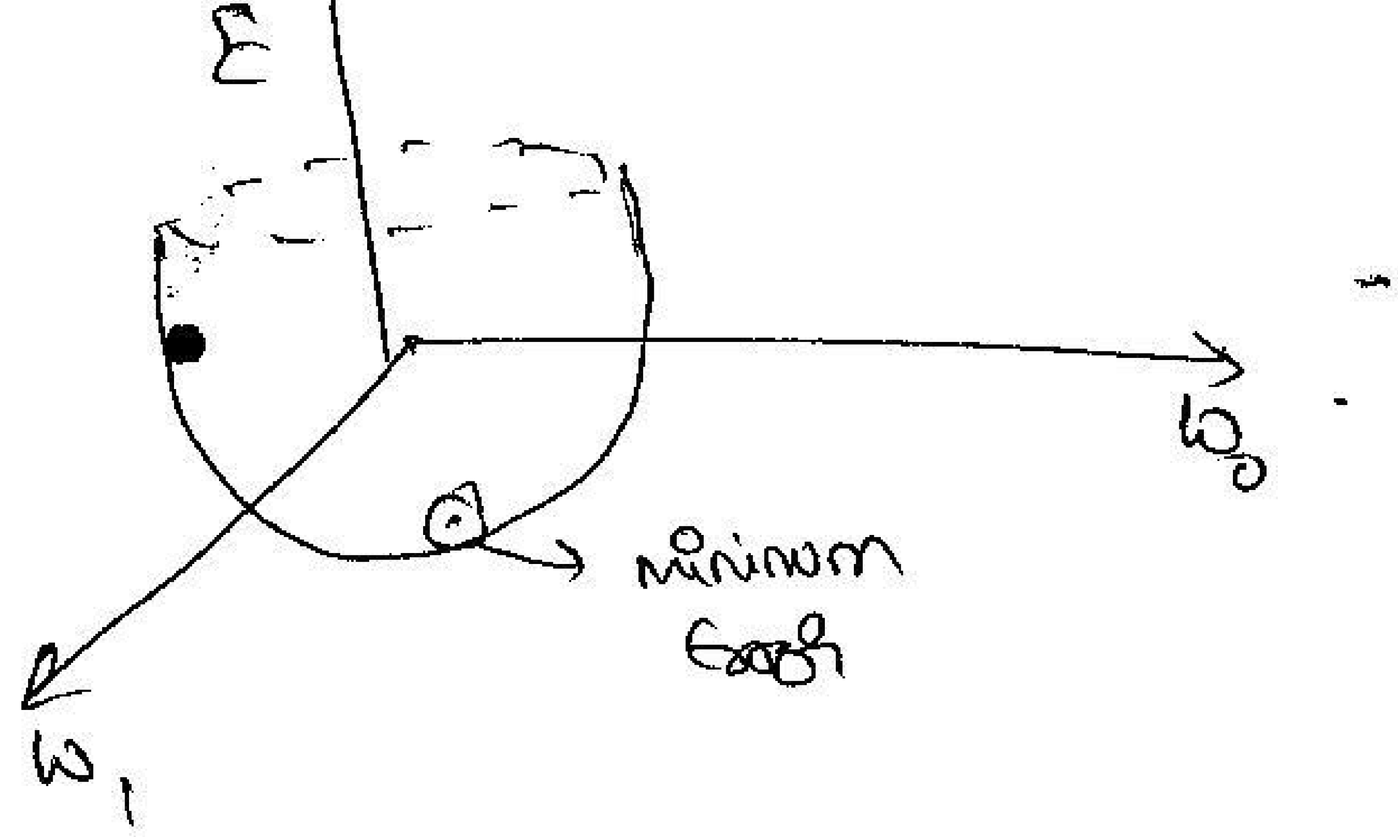
→ This technique is known as steepest descent technique.

*) Gradient Descent Algorithm :-

$y = F(x)$

$y = F(x_1, x_2)$





Gradiant Descent algorithm:- $G = \frac{\partial E}{\partial w}$

Let $P \to$ no: of Points/observations for no: of points

$E \to$ Combined Error

w.k.t $\boxed{E = \sum_P E^P} \to (1)$ where $\boxed{E^P = \frac{1}{2} \sum_o (t_o^P - y_o^P)} \to (2)$

$O \to$ no of different outputs

$t_o^P \to$ target o/p

$y_o^P \to$ o/p of system Actually

$\Big\}$ @ 1 - o/p unit

for several o/p

$y_1 = F_1(x_1, x_2 \ldots x_n)$

$y_2 = F_2(x_1, x_2 \ldots x_n)$

$\vdots$

$y_m = F_n(x_1, x_2 \ldots x_n)$

$\therefore$ the gradiant descent is related as $G = \frac{\partial E}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_P E^P \to (3)$

$$G = \sum_P \frac{\partial E^P}{\partial w_{ij}} \to (4)$$

Applying chain rule of differentiation.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E^P}{\partial y_o} \cdot \frac{\partial y_o}{\partial w_{oi}} \to (5)$$

from Eq (2) derivating with ∂w_{oi}

$$\therefore \frac{\partial E}{\partial y_o} = -(t_o^P - y_o^P) \to (6)$$

w.k.t $y_o = \sum_j w_{oj} \cdot x_j \to (7)$

from Eq (7) derivating Eq wrt ∂w_{oi}

$$\frac{\partial y_o}{\partial w_{oi}} = \frac{\partial}{\partial w_{oi}} \cdot \sum_j w_{oj} \cdot x_j$$

$$\Rightarrow \frac{\partial y_o}{\partial w_{oi}} = x_j \to (8)$$

$\therefore$ now Eq (5) is written as

$$\frac{\partial E}{\partial w_{oi}} = -\left(t_o^p - y_o^p\right) \cdot x_i \rightarrow (9) \rightarrow \text{(for -ve direction)}$$

$\therefore$ on correction representation is

$$\boxed{\Delta w_{oi} = \eta \left(t_o - y_o\right) \cdot x_i} \rightarrow \text{correction.} \rightarrow (10)$$

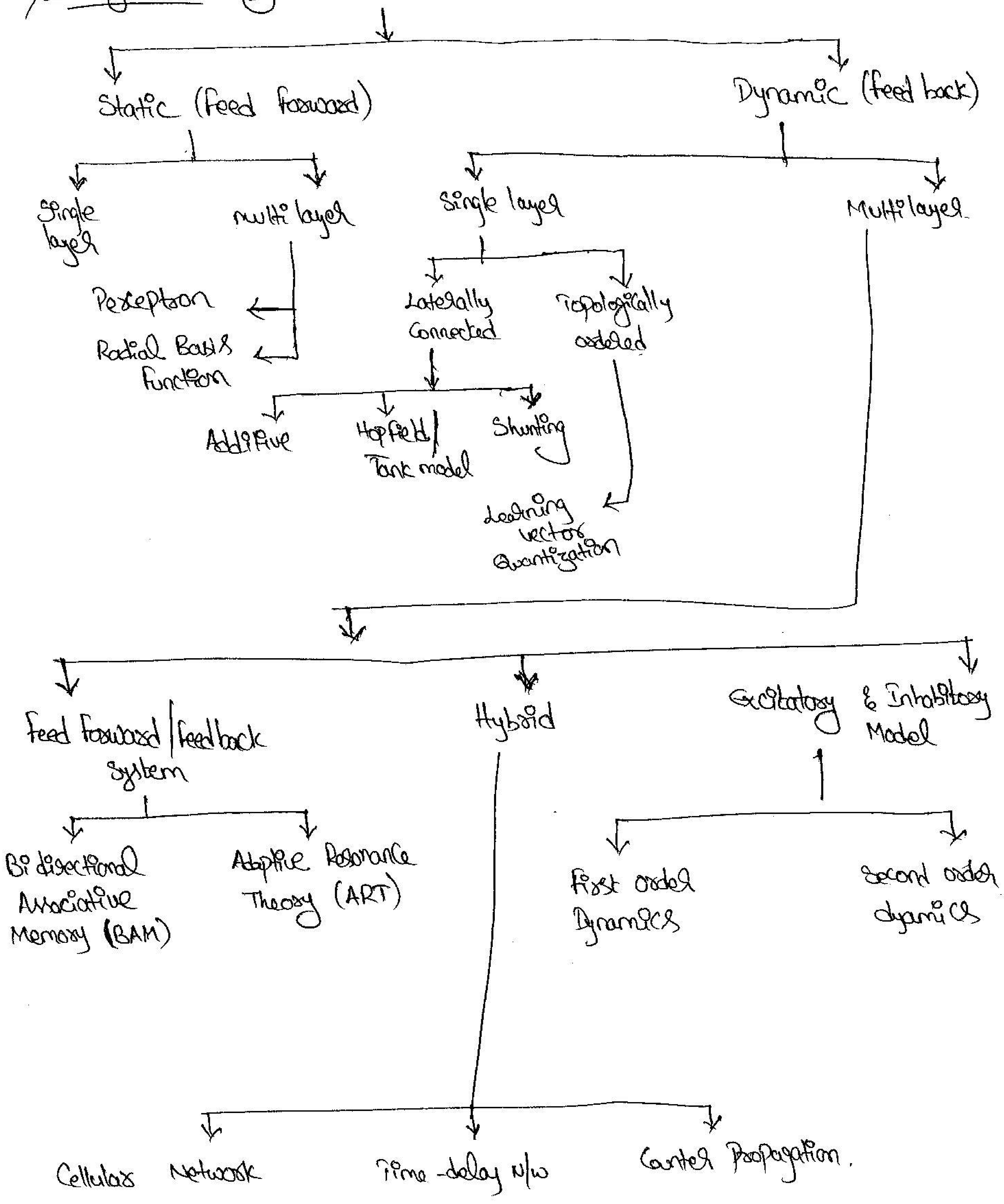
$\eta \rightarrow$ rate at which it is falling is controlled by η

$\eta \rightarrow$ learning rate

$$\therefore \boxed{\underset{}{w_{oi}} = \underset{(old)}{w_{oi}} + \underset{(new)}{\Delta w_{oi}}} \rightarrow (11)$$

$$F(v_k) = \frac{1}{1 + e^{(-a v_k)}}$$

12) Classification of Artificial Neural network :-



13) Neuron Modeling for Artificial Neural Systems:-

McCulloch - Pitts model of neuron is characterized by its formalism and its elegant, Precise mathematical defination.

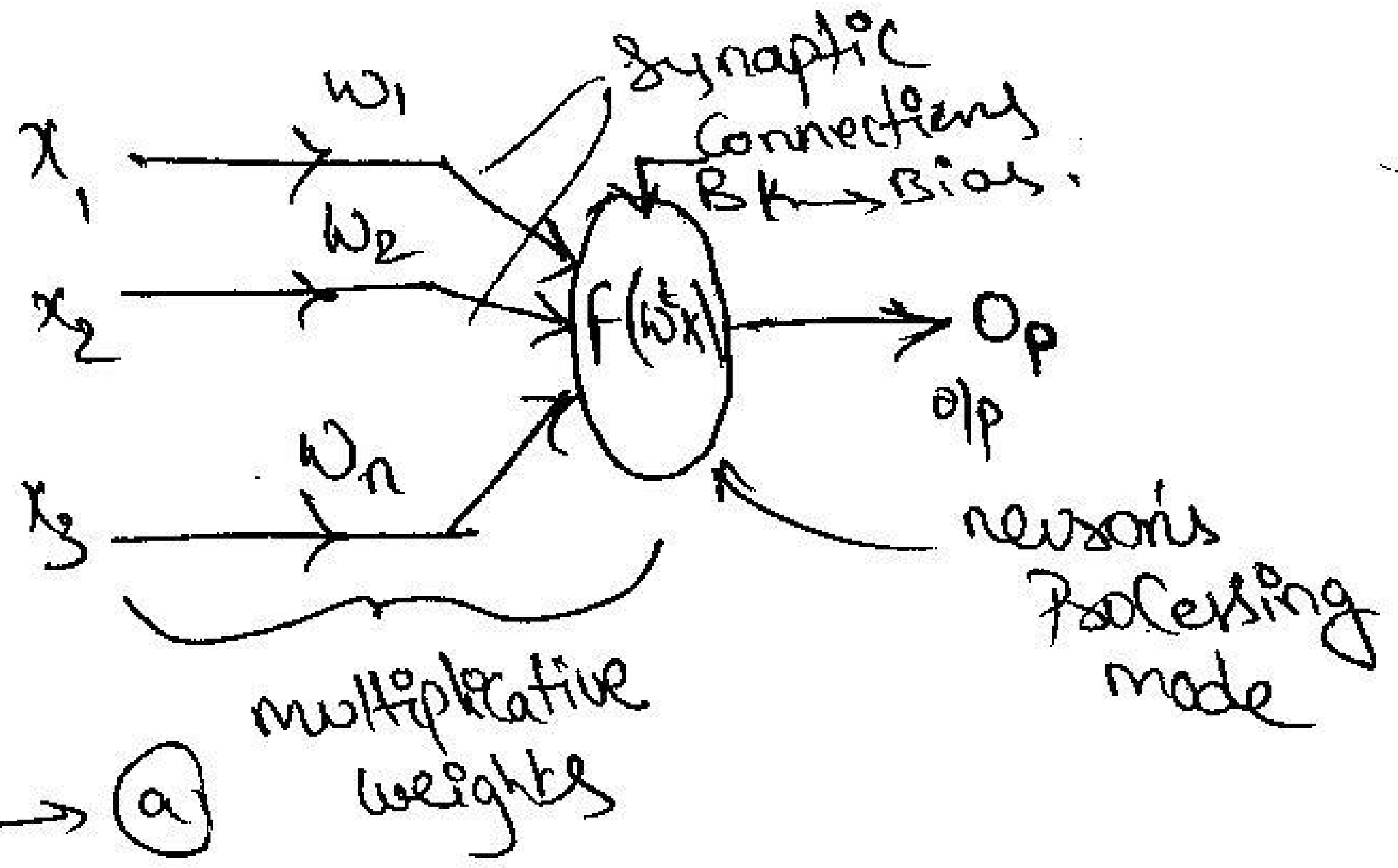
→ Model makes several drastic simplifications

→ It allows 0, 1 states only. operates under a discrete time assumptions. in larger n/w.

→ Weights and neurons threshold are fixed in model and no interaction among n/w neurons take place except signal flow.

the i/p $x_1, x_2 \cdots x_n \rightarrow x_i'$

$\omega_1, \omega_2 \ldots \omega_n \rightarrow \omega_i$



Symbolic Representation of o/p is

$$O_p = f(w^t . x) \quad (or) \rightarrow (a)$$

$$O_p = f\left(\sum_{i=1}^{n} w_i x_i \right) \rightarrow (b)$$

where weighted vector defined as $W = [w_1 \; w_2 \; \ldots \; w_n]^t$

and X is i/p vector

$$X = [x_1 \; x_2 \; x_3 \ldots x_n]^t$$

($t \rightarrow$ denotes transposition)

where function $f(w^t x) \rightarrow$ Refered as Activation function.

2 - Models Mainly introduced:-

(i) neural n/w are meant to be artificial neural n/w consisting of neuron models.

(ii) neurons are meant to be artificial neuron models.

The typical activation used are

$$f(O_p) = \frac{2}{1+e^{(-\alpha O_p)}} - 1 \rightarrow ①$$

and

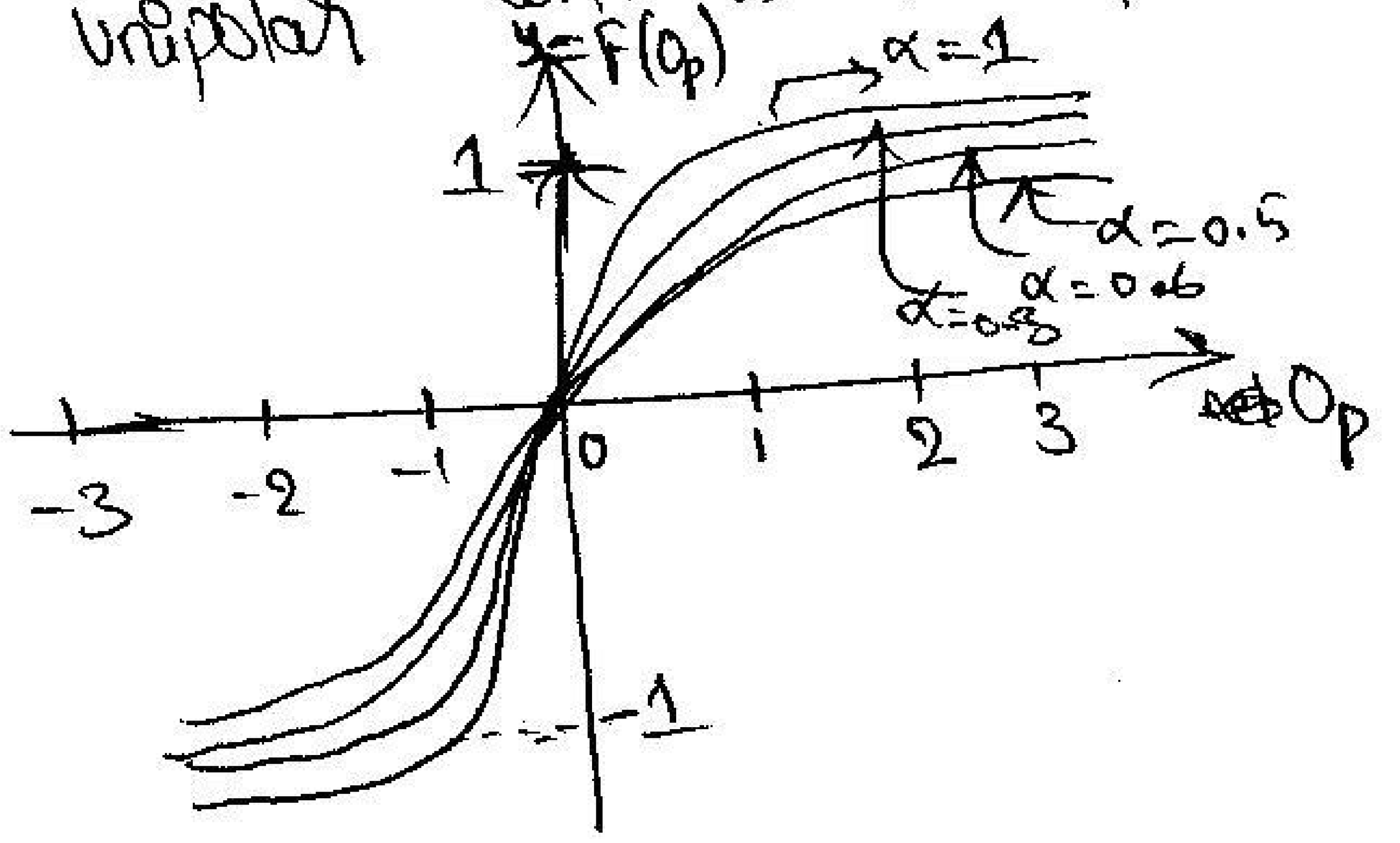
$$f(O_p) = \operatorname{Sgn}(O_p) = \begin{cases} +1, & O_p > 0 \\ -1 & O_p < 0 \end{cases} \rightarrow ②$$

where $\alpha > 0$ in Eq ① is proportional to neuron gain determined the steepness of continous function $f(O_p)$ near $O_p = 0$

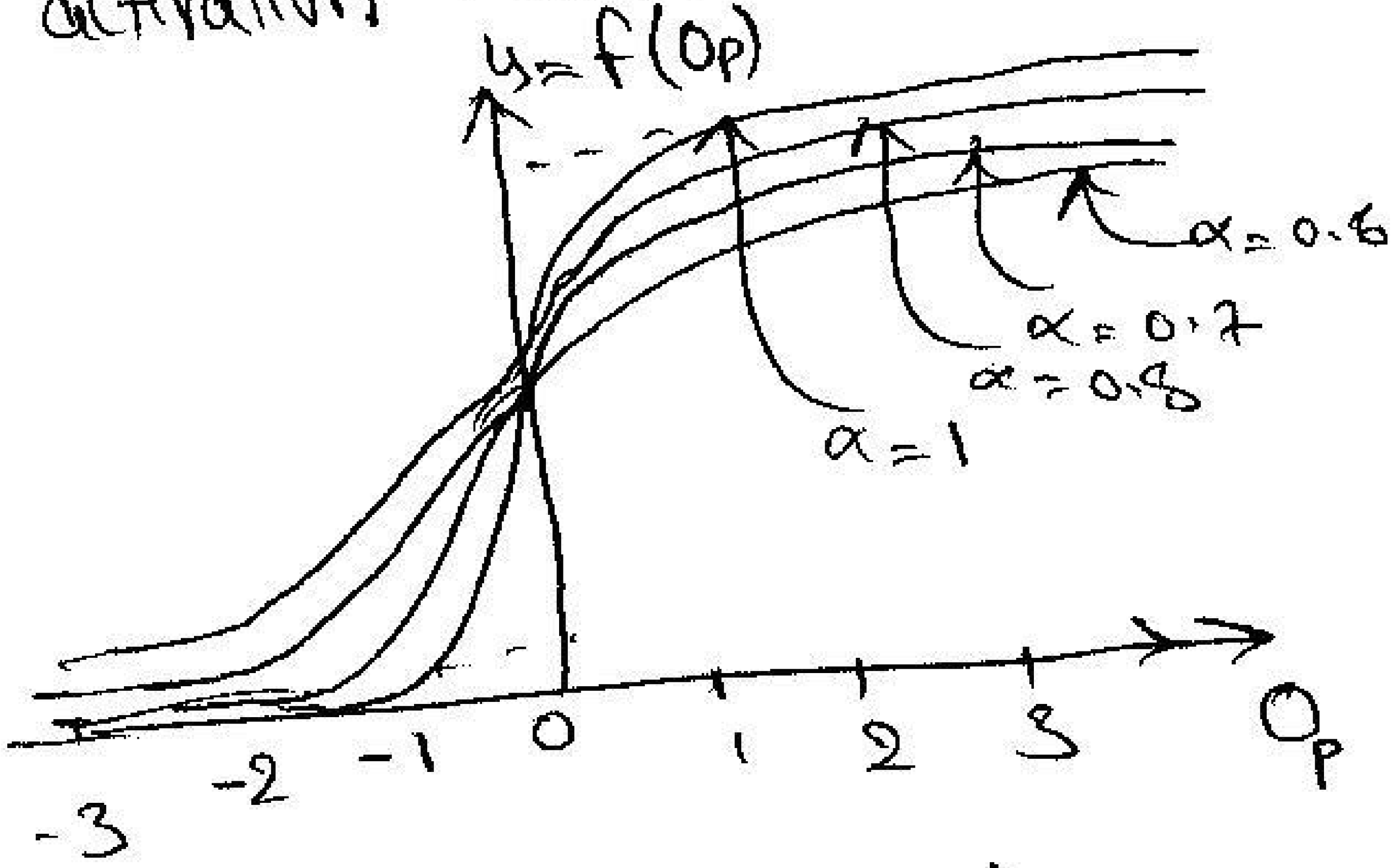
If $\alpha \rightarrow \infty$ (infinite) then limit continous function becomes the Sgn (O_p) function defined in Activation are called bipolar functions. (or) bipolar binary function.

bipolar used to point out both (+ve & -ve) response of neuron.

→ By shifting and scaling the bipolar activation function defined by unipolar continous & unipolar binary activation function obtained.



Bipolar Continous



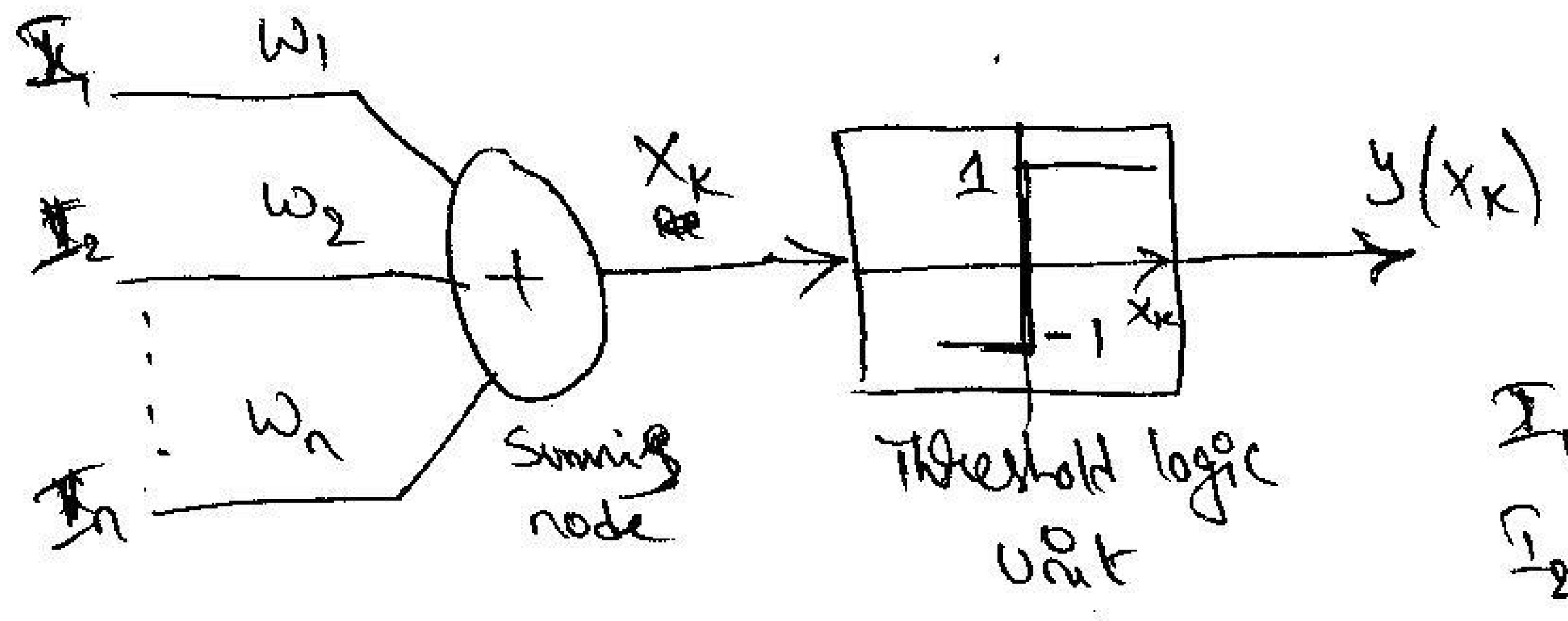
unipolar Continous.

respectively:-

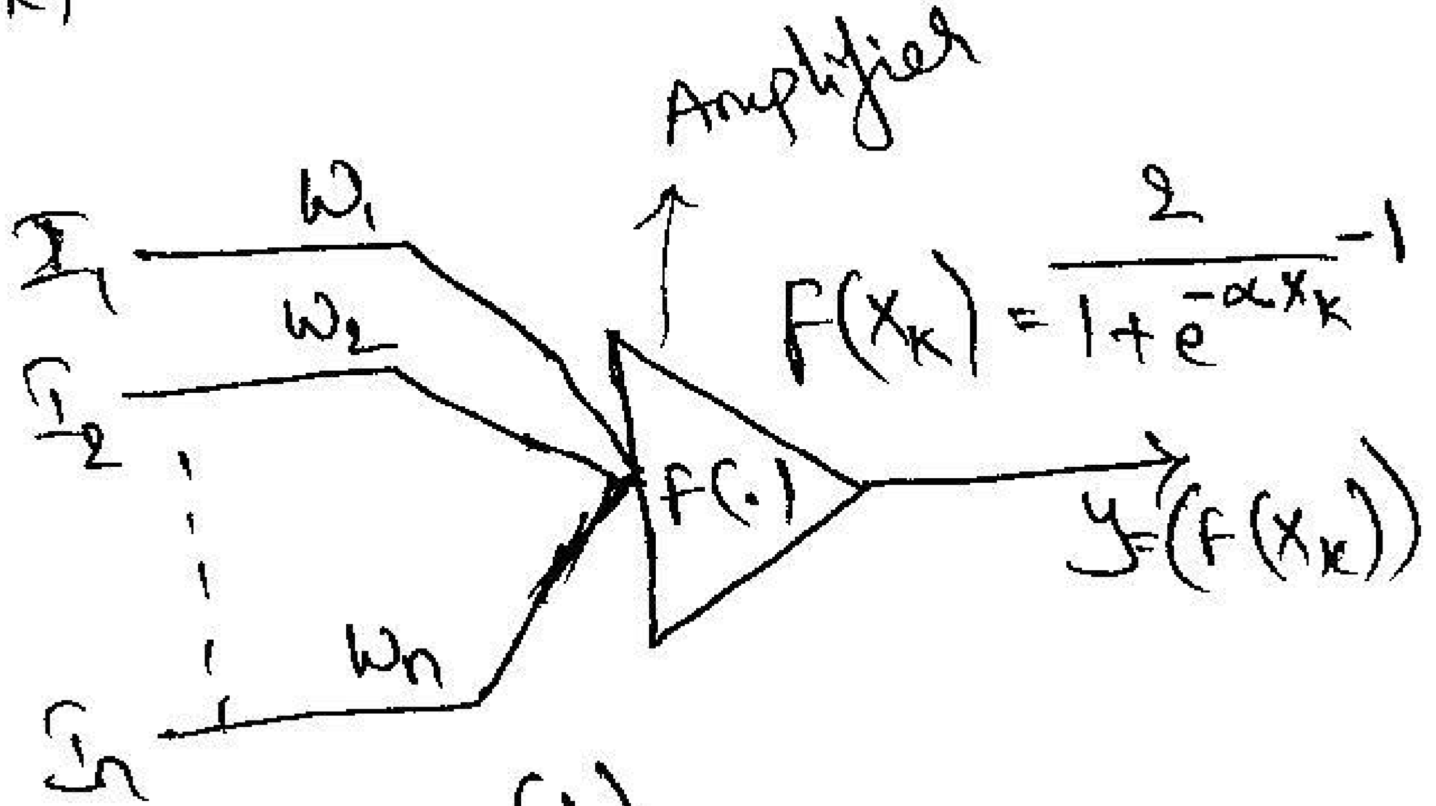
$$f(O_p) = \frac{1}{1+e^{(-\alpha O_p)}} \rightarrow ③$$

$$f(O_p) = \begin{cases} 1 & O_p \geq 0 \\ 0 & O_p < 0 \end{cases} \longrightarrow ④$$

Eq ① & ④ are Sigmoid function. characteristics. as opposed to hard limiting function given in Eq ② & ③.



(a) hard limiting neuron
(binary perceptron)



(b)
Soft limiting neuron
(continous perceptron)

→ In case of continous activation function, above figure-(a) is model used and shown in (b). The neuron is depicted as a summing high gain saturating amplifier which amplifies its i/p signal '$w^t X$'.

These two model are called discrete (binary) & continous perceptrons.

→ The discrete perceptron introduced by Rosenblatt was first learning machine.

If neuron o/p are discrete (binary) (or) continous given a layer of 'm'-neurons their o/p's $O_1, O_2 \ldots O_p$ are arranged in layer output vector.

$$O_p = [O_1 \; O_2 \; \ldots \; O_m]^t$$

where O_p is o/p signal of i^{th} neuron. The domain of vector 'O' are defined in m-dimensional space as follows $i = 1, 2 \ldots m$

$$(-1, 1)^m = \{O \in R^m \; ; \; O_i \in (-1, 1)$$

$$(0, 1)^m = \{O \in R^m \; ; \; O_i \in (0, 1)\}.$$

① Models of ANN :-

(i) feed forward n/w :-

let $I = [I_1, I_2 \dots I_n]^t \rightarrow (a)$; $W_{kj} = [\omega_{k1}, \omega_{k2} \dots W_{km}]^t$

$y = [y_1, y_2 \dots y_n]^t \rightarrow (b)$

as w.k.t for 1 - neuron $y = f(V_k) = f[\omega_k^t . I] \rightarrow (c)$ for $j = 1 \rightarrow n$.

let us consider a non-linear matrix operation denoted by 'Γ'.

$$\therefore y = \Gamma[W . I] \rightarrow (d)$$

$$\therefore W = \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \omega_{23} & \dots & \omega_{2n} \\ \vdots & \vdots & & & \vdots \\ \omega_{m1} & \omega_{m2} & \omega_{m3} & \dots & \omega_{mn} \end{bmatrix} \rightarrow (e)$$

$j = 1 \rightarrow n$
$k \rightarrow 1 - m$

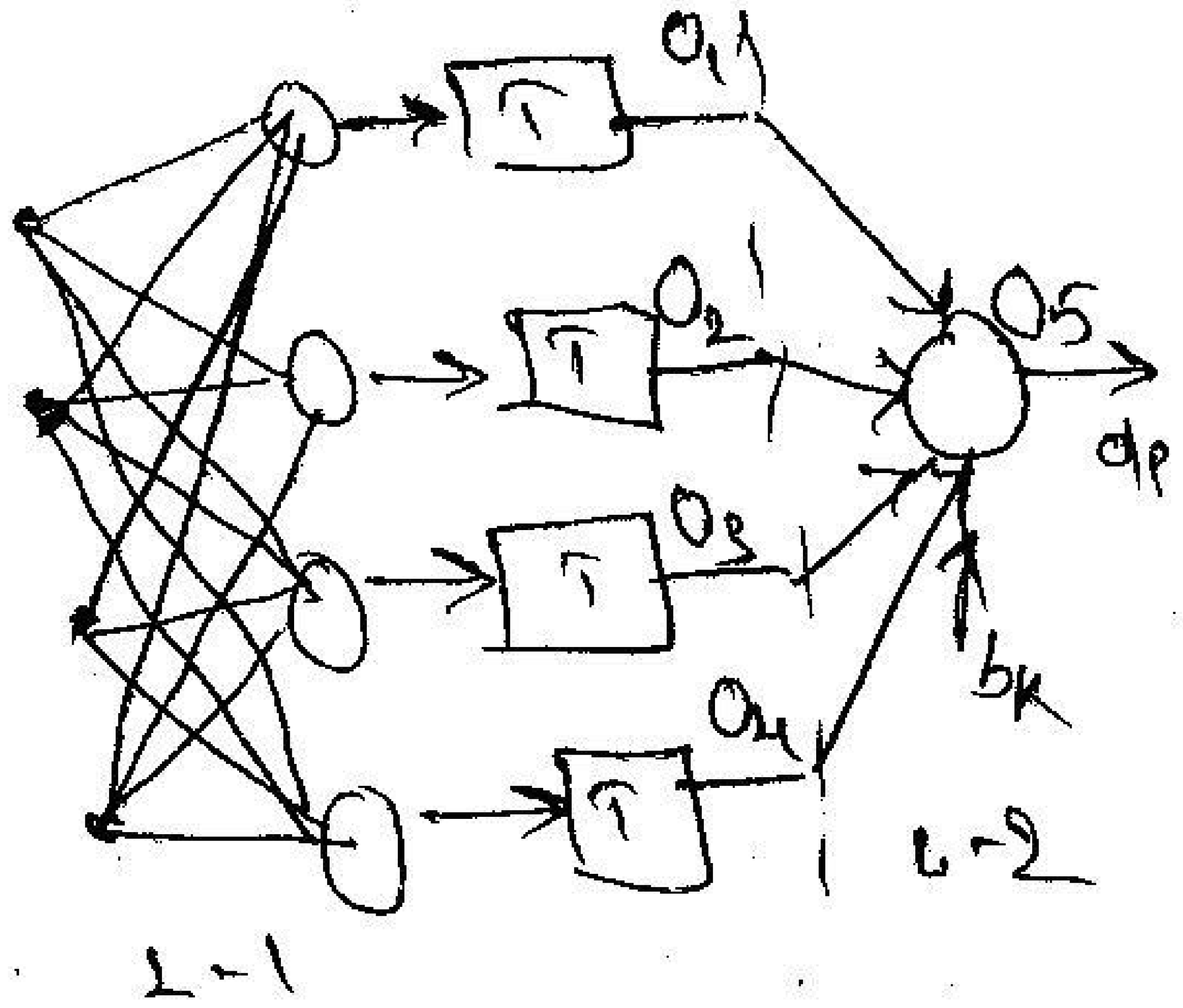
and $\therefore y = \Gamma(\cdot) = \begin{bmatrix} \Gamma(\cdot) & 0 & \dots & 0 \\ 0 & \Gamma(\cdot) & \dots & 0 \\ 0 & 0 & \Gamma(\cdot) \dots & 0 \\ \vdots & \dots & \dots & \Gamma(\cdot) \end{bmatrix} \rightarrow (f)$

$\therefore$ I & y are i/p & o/p patterns involves in time delay for activation

$$\therefore \boxed{y(t) = \Gamma[\omega . I(t)]} \rightarrow (g)$$

Eg :- multilayer n/w :-

WKT $f(net) := sgn(net) = \begin{cases} 1 & net \geq 0 \\ 0 & net < 0 \end{cases} \rightarrow (a)$



$\therefore$ The response should satisfy

$$O_5 = sgn(O_1 + O_2 + O_3 + O_4 - b_k) \rightarrow (b)$$

where response should be $O_1 + O_2 + O_3 + O_4 + O_5 = 1$.

(Competative learning)

expressed $y = f(X_k)$

where weights W_j contain weights leading toward the k^{th} neuron

o/p node is $W_k = [W_{k1}\ W_{k2} \dots W_{km}]^t \rightarrow (e)$

W.K.T in general Activation function o/p is given as

$$y = \mathrm{Sgn}(U_k) = \frac{1}{1+e^{-aU_k}} \rightarrow \text{(c)} \quad \left(O_p = \frac{2}{1+e^{-aU_k}} - 1 \right)$$

typical Activation function

∴ in multilayer the bipolar activation function obtain for first layer is

$$O_p = \begin{bmatrix} \frac{2}{1+e^{(1-I_1)\cdot\lambda}} - 1 \\ \frac{2}{1+e^{(I_1-2)\lambda}} - 1 \\ \frac{2}{1+e^{(3-I_2)\lambda}} - 1 \\ \frac{2}{1+e^{(I_2-3)\lambda}} - 1 \end{bmatrix} \rightarrow 1^{st} \text{ layer bipolar AC func}$$

∴ for second layer $O_5 = \frac{2}{1+e^{(3.5-O_1-O_2-O_3-O_4)\cdot\lambda}} - 1$ //.

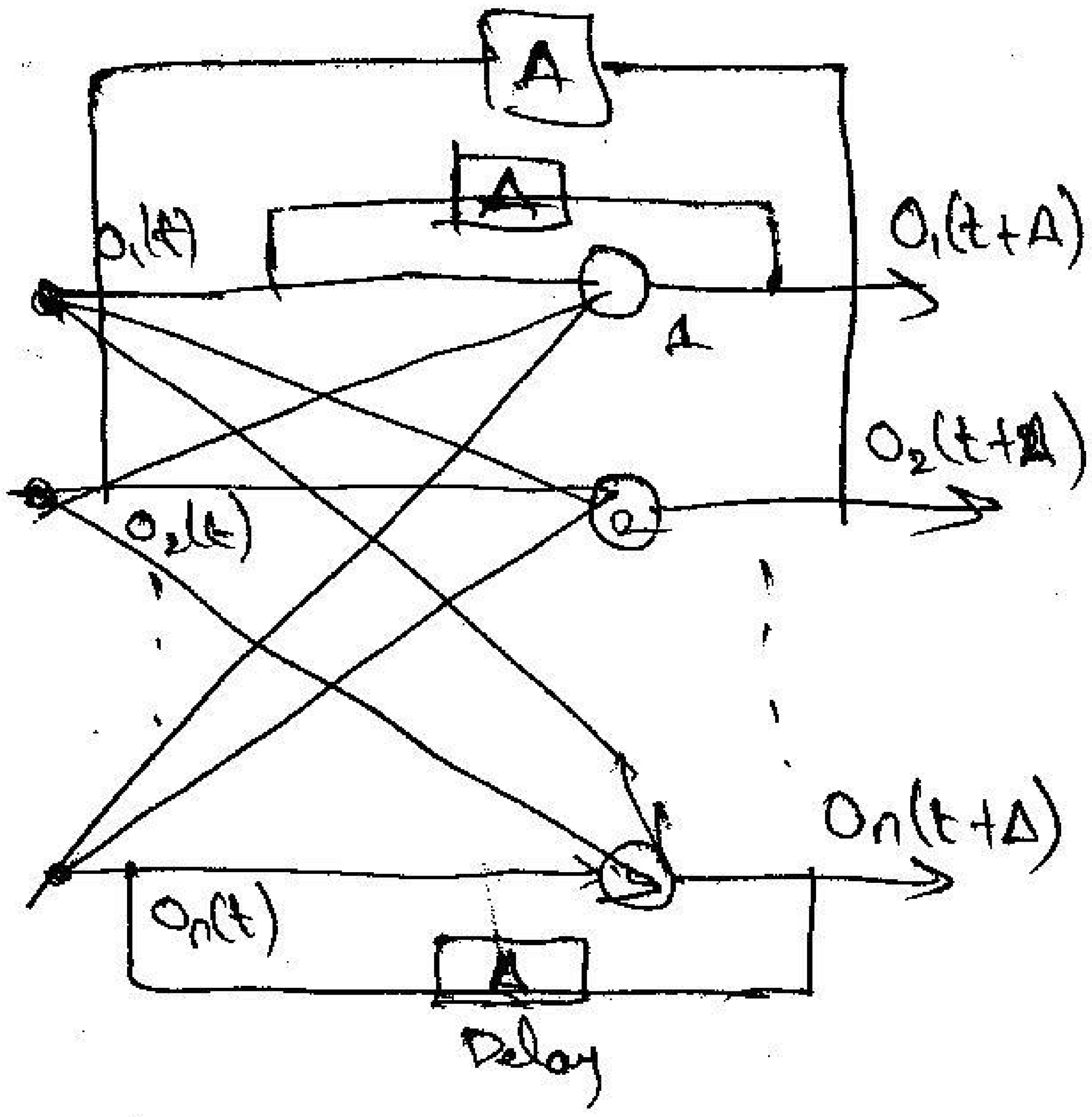
(2) feed backward :-

The control of o/p O_p through o/p $J=1,2\ldots m$

Control is especially meaningful if present o/p

say $O(t)$ say $O(t)$ control i/p as $O(t+\Delta)$

$t \rightarrow$ time Δ-delay



if we have discrete variable performance

than $\Delta, 2\Delta, 3\Delta \ldots$ called discrete time

for discrete Aron converted

$$O^{k+1} = \Gamma\left[W O^{k}\right] \text{ for } k=1,2\ldots$$

Eg:- $O^1 = \Gamma\left[W x^0\right]$

$O^2 = \Gamma\left[W \Gamma\left[W x^0\right]\right] \ldots\ldots O^{k+1} = \Gamma\left[W \Gamma\left[\Gamma\left(W x^0\right)\right]\ldots\right]$

4) Models of Artificial neural networks:-

→ An introductory to artifical neural n/w is a physical cellular networks that are able to acquire, store and utilize knowledge has been related to network capabilities and performance.

(i) feedforward network:-

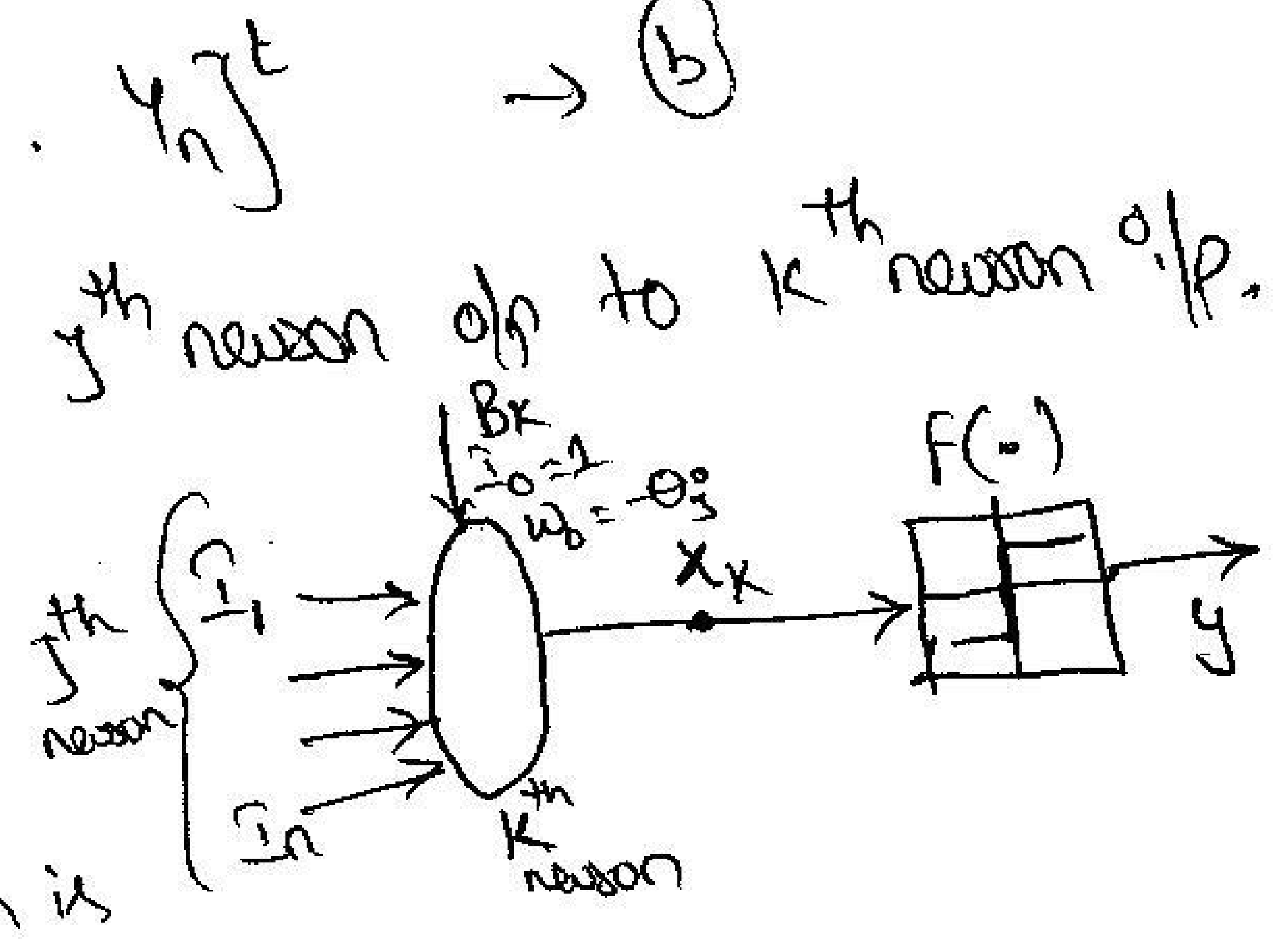
Let us consider an elementary feedforward architecture of m-neurons are receiving & n-inputs. Its o/p & i/p vectors are respectively be

$$I = [I_1, I_2 \dots I_n]^t \rightarrow (a)$$

$$y = [y_1\ y_2 \dots y_n]^t \rightarrow (b)$$

weights of synaptic be W_{kj} connects j^{th} neuron o/p to k^{th} neuron i/p.

W_{kj}: k → destination, j → source



we thus write activation of k^{th} neuron is

$$\boxed{X_k = \sum_{j=1}^{n} I_j \cdot W_{kj} + B_k} \text{ for } j = 1, 2 \dots m. \rightarrow (c) \quad \left(\because B_k = -\theta_j,\ B_k = 0 \right)$$

The non-linear transformation involving the activation function $y = F(X_k)$ for $j = 1, 2 \dots m$ completes processing of I. the transformation performed by m neurons. In n/w is strongly non-linear mapping expressed as

$$y = F(X_k) = f(W_k^t \cdot I) \rightarrow (d) \text{ for } j = 1, 2 \dots m$$

where weights W_j contain weights leading toward the k^{th} neuron o/p node is

$$W_k = [W_{k1}\ W_{k2} \dots W_{km}]^t \rightarrow (e)$$

By introducing a non-linear matrix operation 'Γ', the mapping of i/p space $\underline{I}$ to o/p space y implemented by n/w expressed.

$$y = \Gamma[W\underline{I}] \rightarrow (f)$$

where W is weight matrix also called connection matrix

$$W = \begin{bmatrix} W_{11} & W_{12} & W_{13} & \cdots & W_{1n} \\ W_{21} & W_{22} & W_{23} & \cdots & W_{2n} \\ W_{31} & W_{32} & W_{33} & \cdots & W_{3n} \\ \vdots \\ W_{m1} & W_{m2} & W_{m3} & \cdots & W_{mn} \end{bmatrix} \rightarrow (g) \qquad \begin{array}{l} J = 1 \rightarrow n \\ \cancel{K} = 1 \rightarrow m \end{array}$$

and

$$y = \Gamma(\cdot) = \begin{bmatrix} f(\cdot) & 0 & \cdots & 0 \\ 0 & f(\cdot) & \cdots & 0 \\ \vdots \\ 0 & 0 & \cdots & f(\cdot) \end{bmatrix} \rightarrow (h)$$

The matrix of function operator 'Γ' operates component wise on activation values X_K of each neuron.

The i/p & o/p vectors $\underline{I}$ & Y called as i/p & o/p patterns

$\because$ It involves timedelay b/w $\underline{I}$ & Y so we can write eq (f) as

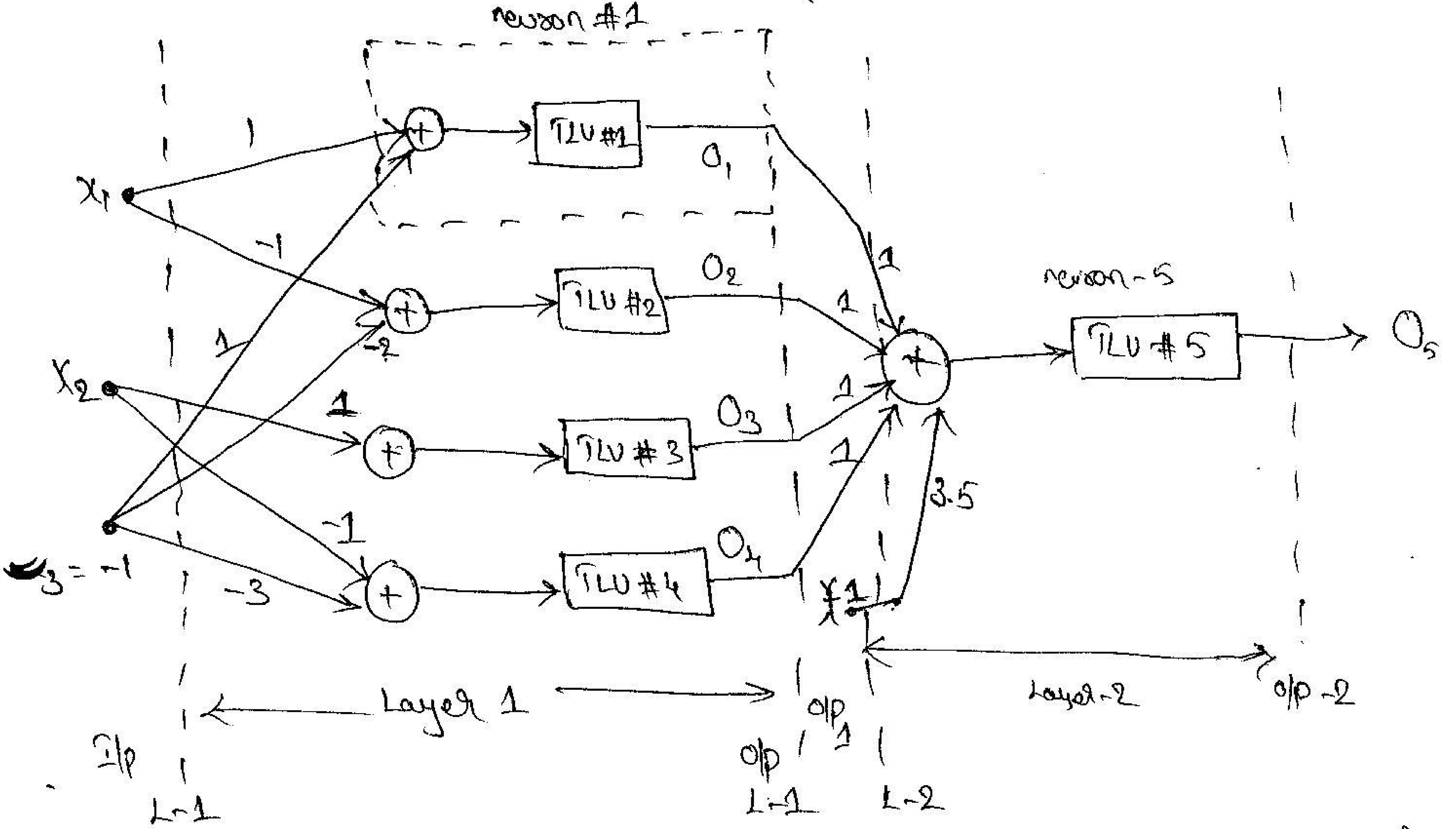
$$\boxed{Y(t) = \Gamma[W\underline{I}(t)]} \rightarrow (i)$$

→ The generic feedforward n/w characterized by lack of feedback.

→ This type of connection n/w can be connected in cascade to create a multilayer n/w.

Example 1:-

w.k.t $f(net) \triangleq Sgn(net) = \begin{cases} +1 & net \geq 0 \\ -1 & net < 0 \end{cases} \rightarrow (a)$



$\therefore$ The response of neuron-5 $O_5 = Sgn(O_1 + O_2 + O_3 + O_4 - 3.5)$

the O_5 neuron respond $+1$ if $O_1 = O_2 = O_3 = O_4 = 1$

$\therefore$ The bipolar activation function obtain for first layer.

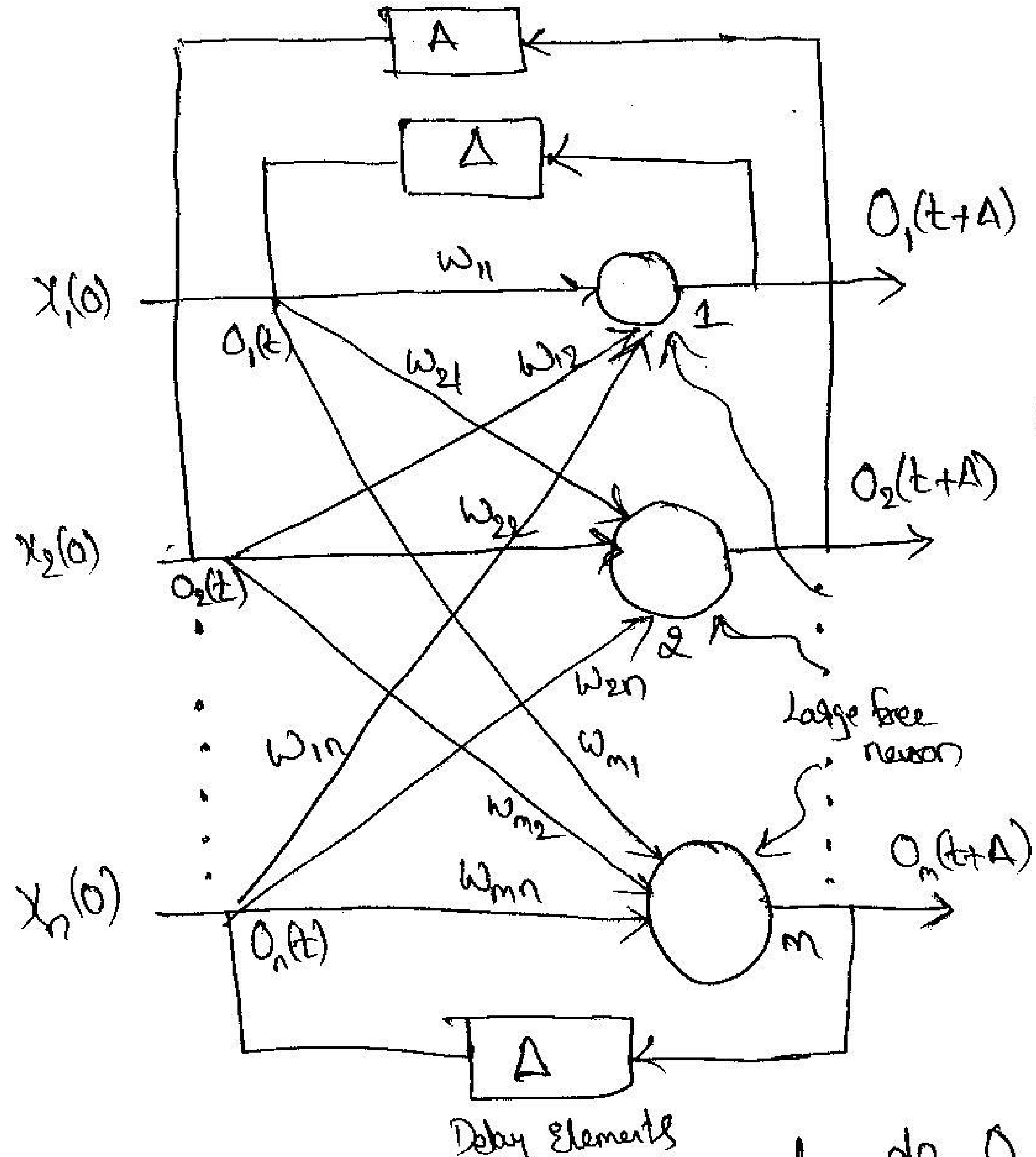
$$O_p = \begin{bmatrix} \dfrac{2}{1+e^{(1-x_1)\lambda}} - 1 \\ \dfrac{2}{1+e^{(x_1-2)\lambda}} - 1 \\ \dfrac{2}{1+e^{(-x_2)\lambda}} - 1 \\ \dfrac{2}{1+e^{(x_2-3)\lambda}} - 1 \end{bmatrix}$$

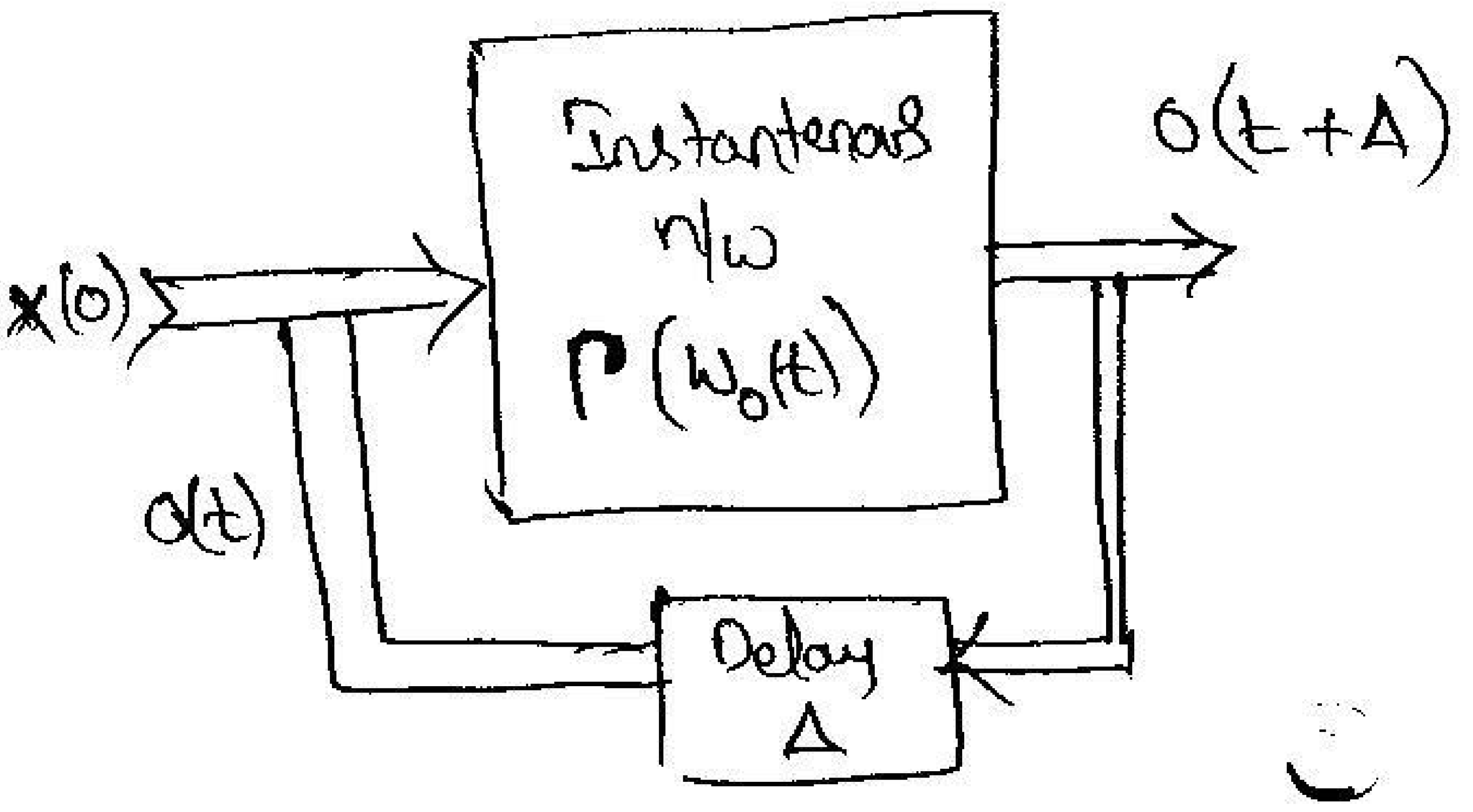
$\therefore$ for second layer.

$$O_5 = \frac{2}{1+e^{(3.5-O_1-O_2-O_3-O_4)\lambda}} - 1$$

(ii) Feedback Network :-

Can be obtained from the feed forward n/w neuron o/p to their i/p's.





→ Recurrent n/w.

The Control of o/p O_i through o/p O_j for $j = 1, 2 \ldots m$ such control is especially meaningful if present o/p say $O(t)$ control o/p as $O(t+Δ)$ The time Δ (delay) elapsed b/w t and t+Δ is introduced by the delay elements in feedback loop.

feed forward n/w mapping of $O(t)$ into $O(t+Δ)$ is

$$O(t+Δ) = \Gamma[W_o(t)]$$

There are ~~two mainly~~ two main categories of single layer feedback n/w. If we consider time as discrete variable than n/w performance at discrete time instants Δ, 2Δ, 3Δ .... the system is called discrete time.

for discrete time artificial neural system, we have converted to

$$O^{k+1} = \Gamma\left[W_o^{k}\right], \text{ for } K = 1, 2 \ldots \quad \rightarrow (a)$$

K → is instant number.

∴ response at K+1 instant depends on entire history of n/w starting at K=0.

$$O^1 = \Gamma[W x^0]$$

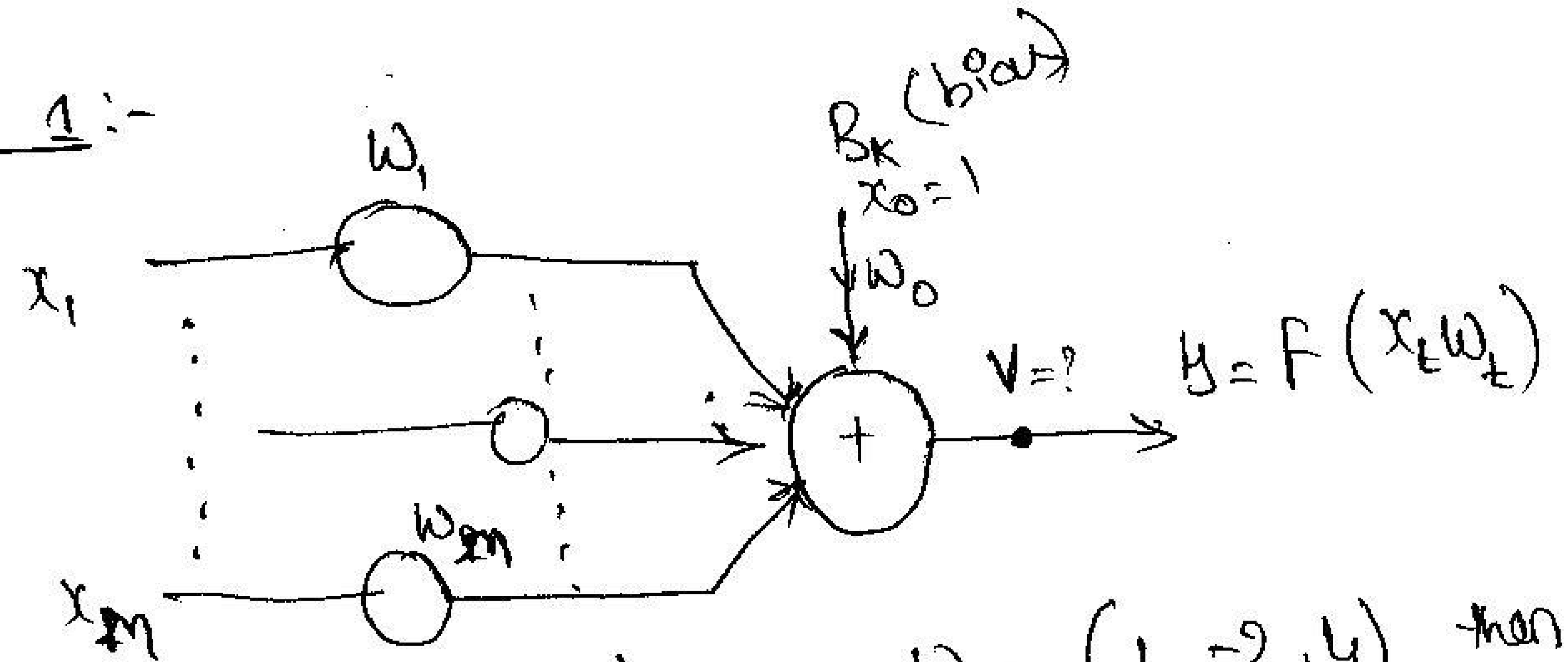
$$O^2 = \Gamma[W\Gamma[Wx^0]]$$

$$\vdots$$

$$O^{k+1} = \Gamma\{W\Gamma..[\Gamma[Wx^0]]...\}$$

A system with discrete time inputs and discrete data representation is called an automaton.

Example 1:-



Let $X = (0, 1, 1)$; $W = (1, -2, 4)$ then

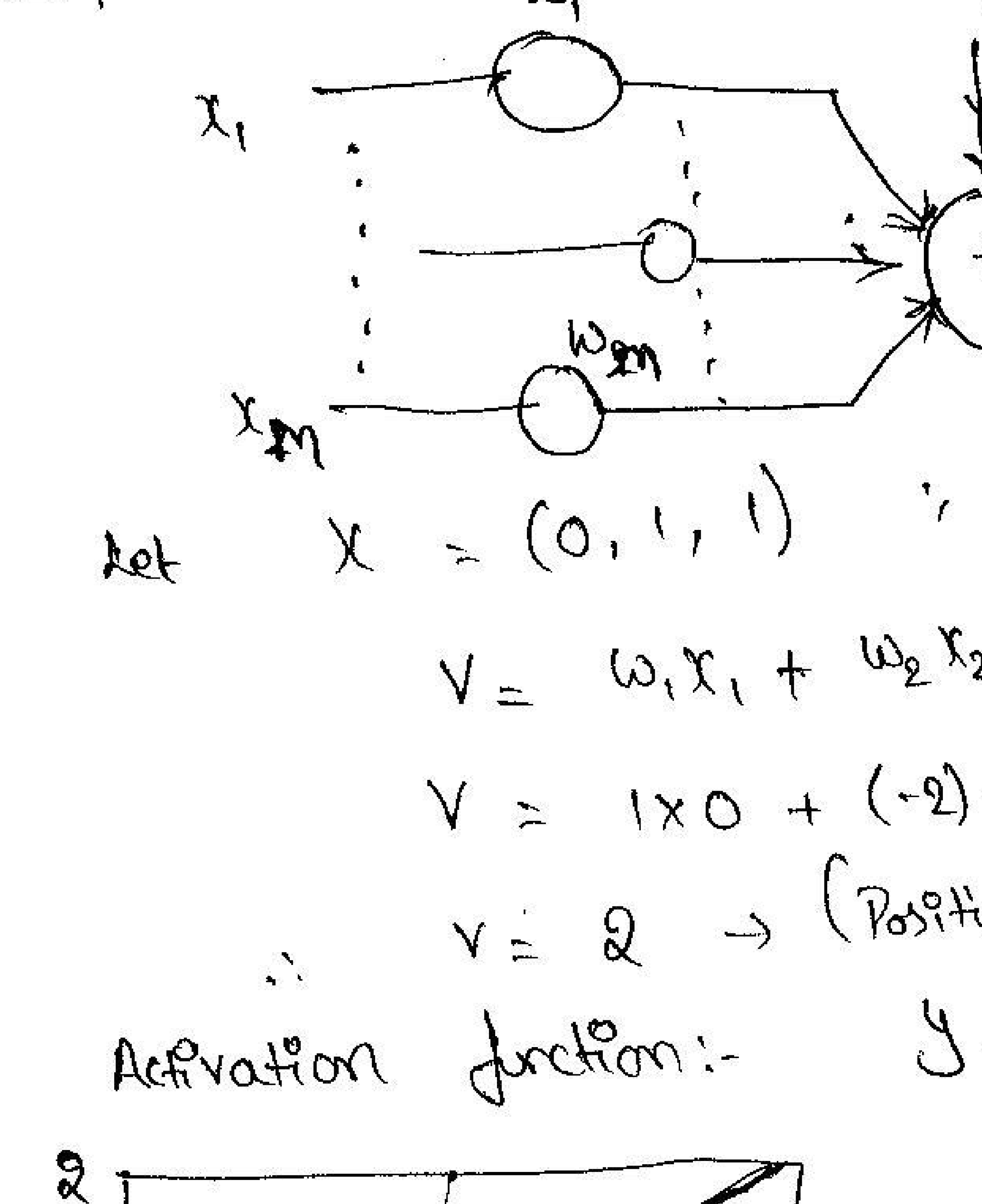
$$V = w_1 x_1 + w_2 x_2 + w_3 \cdot x_3 + \dots w_m x_m$$

$$V = 1 \times 0 + (-2) \cdot 1 + (4) \cdot 1 = 2$$

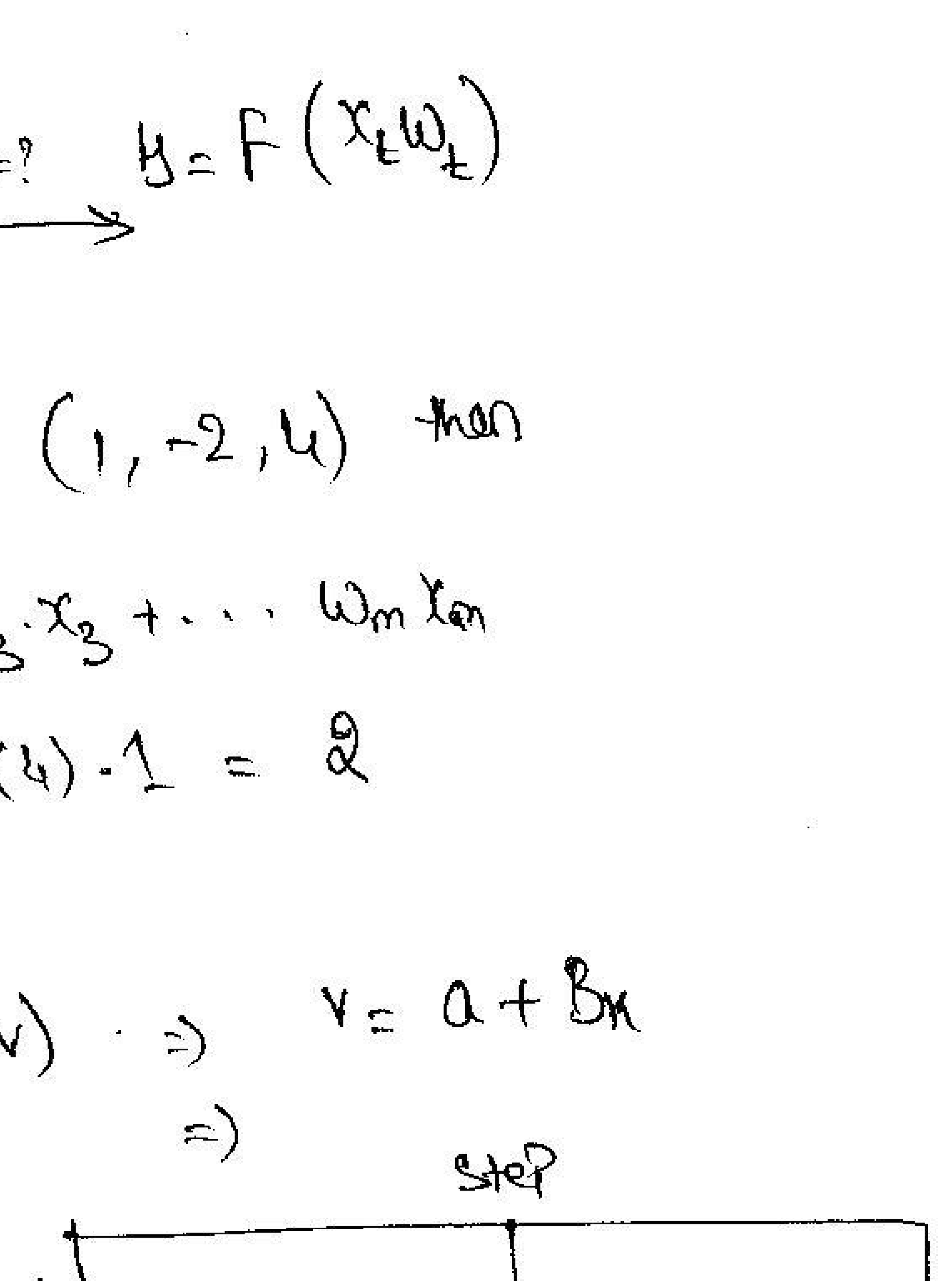
$$\therefore \quad V = 2 \rightarrow \text{(Positive)}$$

Activation function:- $y = f(v) \Rightarrow V = a + Bx$

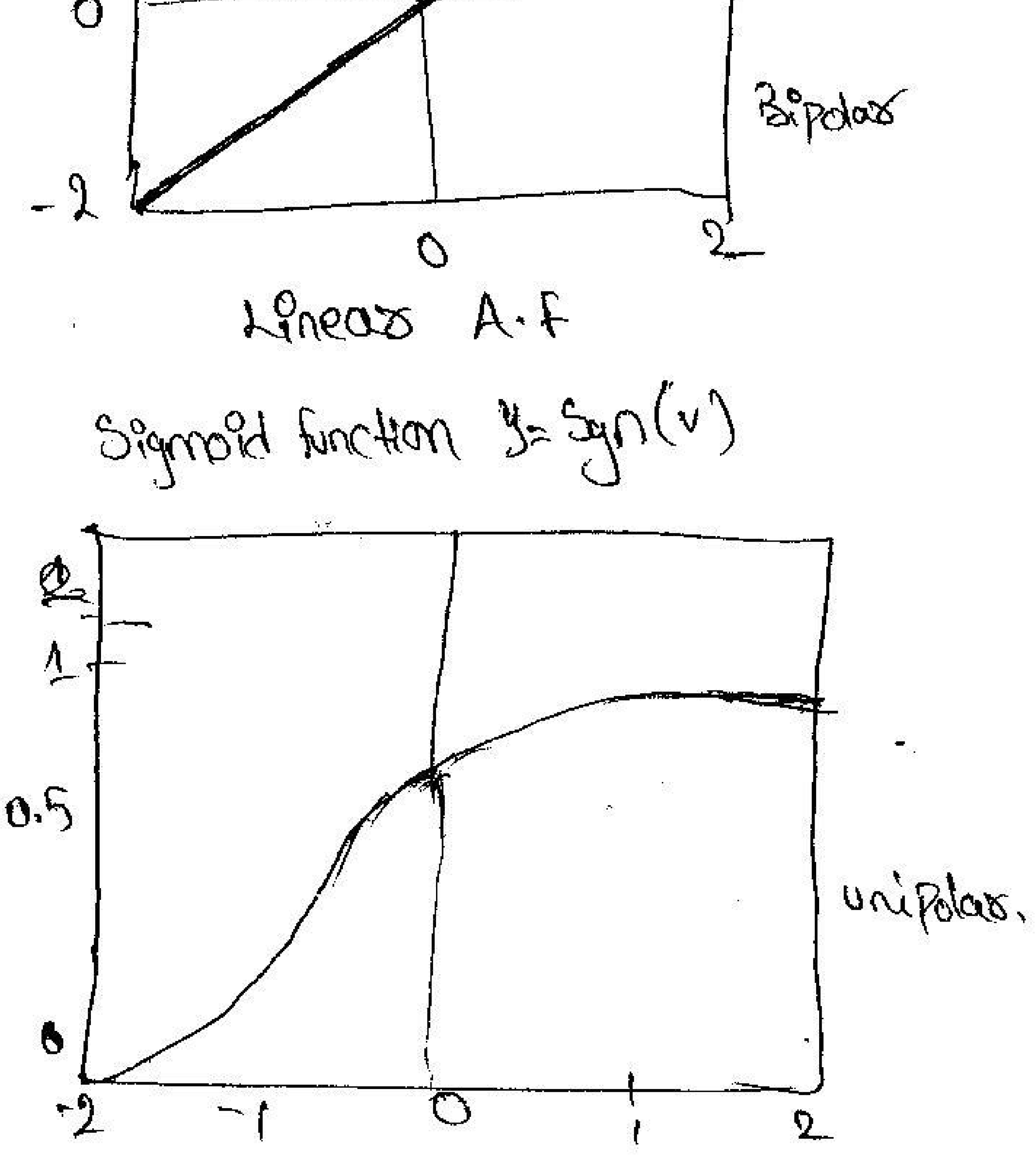
$\Rightarrow$



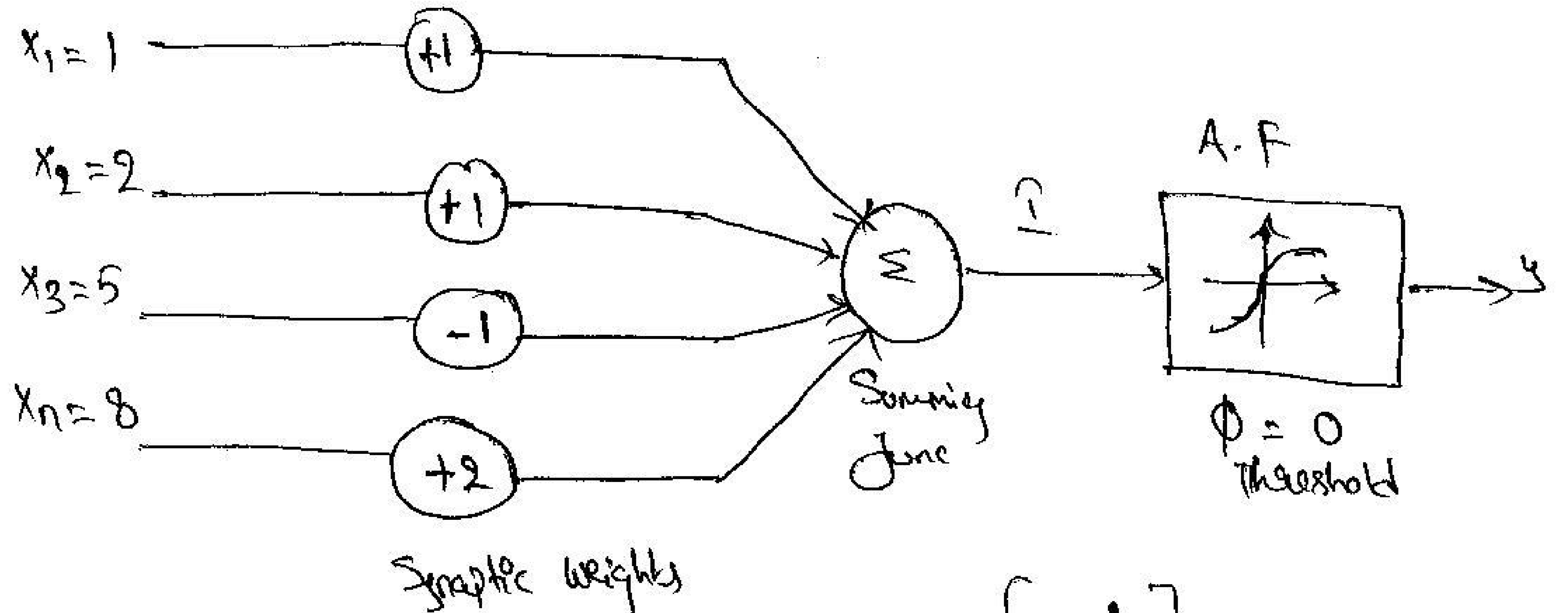
Linear A.F



Sigmoid function $y = \text{Sgn}(v)$



Eg:- 2



$$I = X^T \cdot W = \begin{bmatrix} 1 & 2 & 5 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1\times1 + 2\times1 + 5\times(-1) + 8\times2 \end{bmatrix} = \begin{bmatrix} 14 \end{bmatrix}$$

with a binary A.f the o/p is '1'.

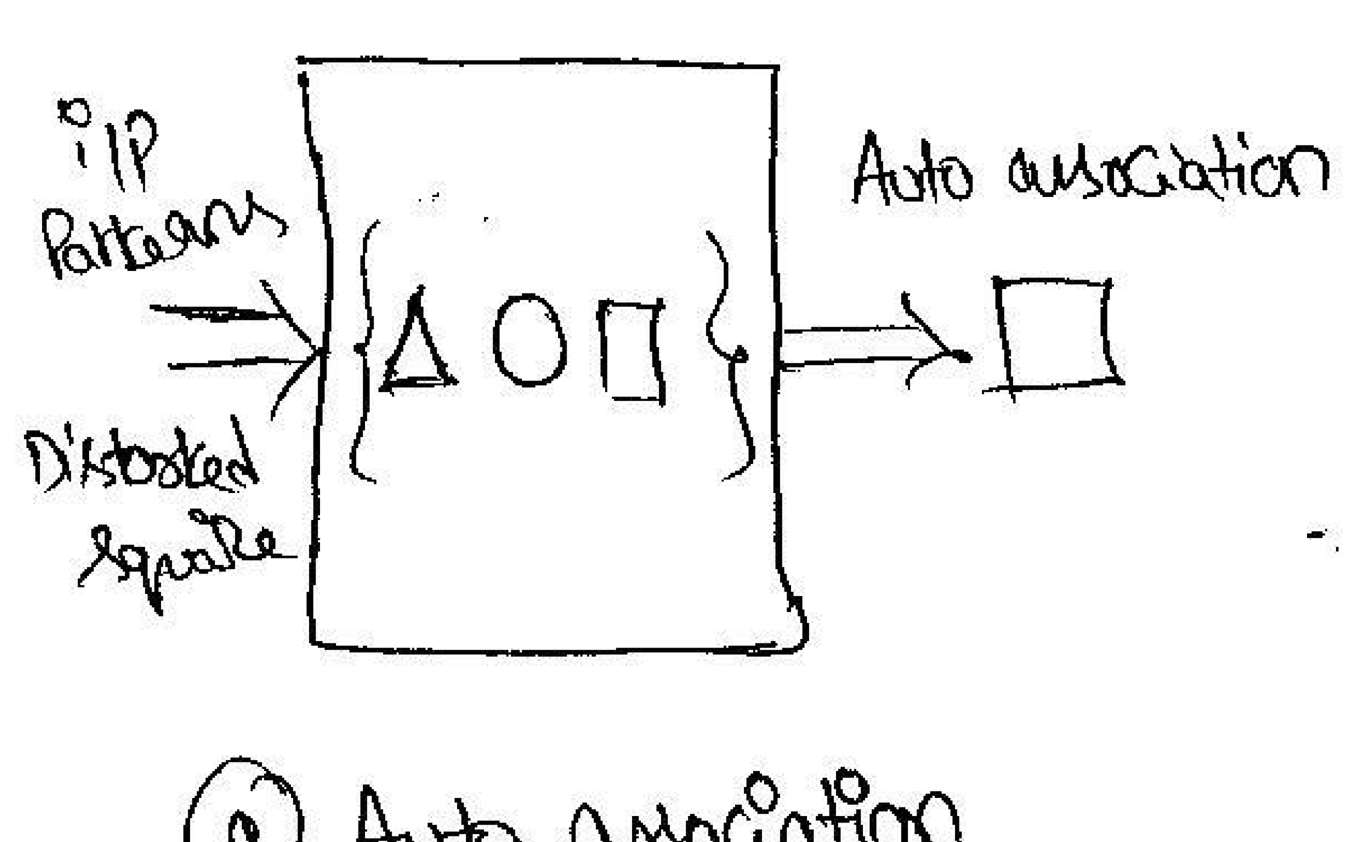
14) **Neural Processing:-**

* The process of computation of O/p for a give X performed by the n/w is known as Recall.

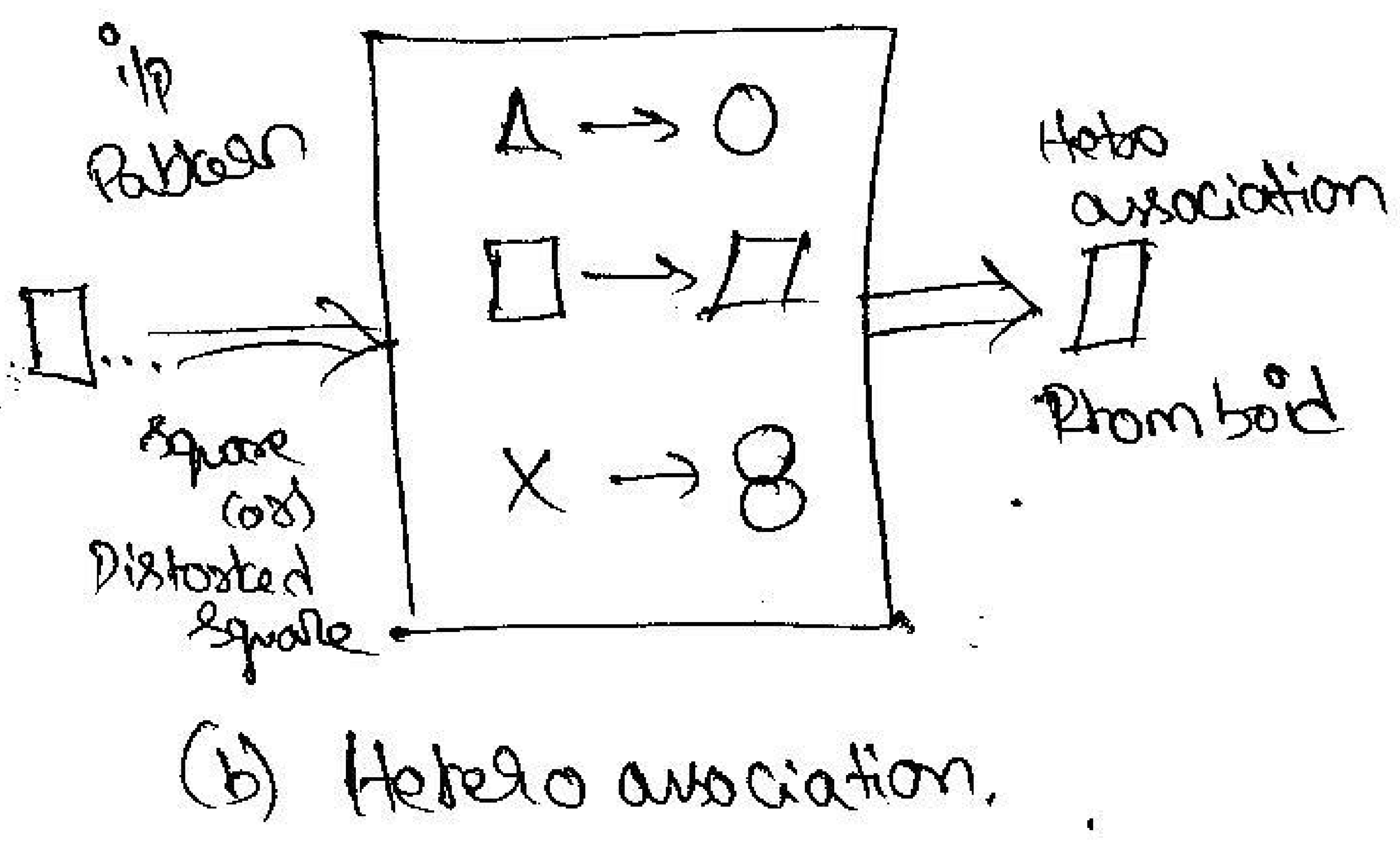
Recall is a proper processing phase for neural n/w. and its objective is to retrive information.

Recall corresponds to decoding of stored content which may have been encoded in a n/w previously.

Assume that a set of pattern can be stored in n/w. Later if network is presented with a pattern similar to member of stored set it may associate the input with the closest stored pattern. The process is called autoassociation.



(a) Auto association.



(b) Hetero association.

→ In hetero association processing, the association b/w pair of patterns are stored. (I-35)

→ A square i/p pattern presented at i/p results in rhomboid at o/p. It can be inferred at that the rhomboid and square constitute one pair of stored patterns.

→ Classification can be understood as a special case of hetero association. where association is now b/w (i/p & o/p pattern.) and second member of hetero associative pair which is supposed to indicate the i/p's class members.

→ If n/w desired response is class member but the i/p pattern doesnot exactly corresponds to any pattern in set, the processing is called recognition.

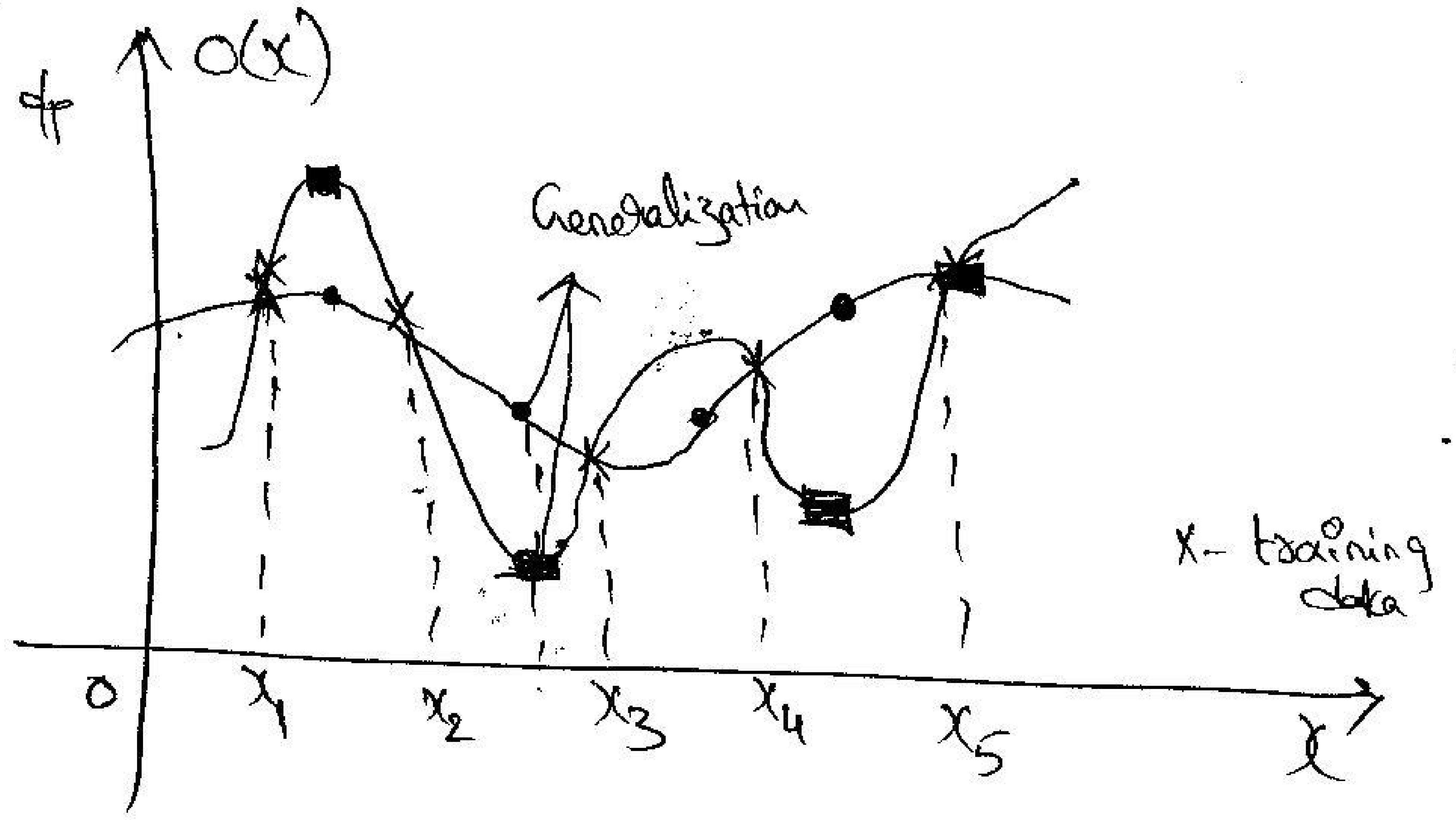
→ When class membership for one of pattern in the set is recalled, recognition becomes identical to classification.

→ The distinct strength of neural n/w is their ability to generalize.

→ The n/w is said to be generalized well when it sensibly interpolate i/p pattern that are new to n/w.

→ Assume that n/w has been trained using data i/p $x_1, x_2 \ldots x_5$ and o/p $O(x)$ then.

O → Good generalization
□ → Bad "



(xi) Training methods of ANN:-

Consists 2-Parts. (1 is Activation State
2 is dynamic synaptic weights)

Terms:- Short term ; long term

① Short term → neural n/w is modeled by Activation State of n/w
② long term → " " " Encoded by information in synaptic weights due to learning.

Def:- of learning:-
is a Process of neural n/w are adapted through a Process of Simulation by Environment in which n/w is Embedded.

(i) Supervised learning:- requires Pairing of each i/p vector $(I)^t$ with target vector $(t)^t$ representing desired o/p y_k. all these are called training Pair.

→ usally n/w is trained over number of training pairs.

→ An i/p vector is applied, the o/p ~~vector~~ of network is calculated and Compared to corresponding target vector and difference (error) is fed back through the n/w & weight are changed according to algorithm and tends to minimize error.

supervised learning:- An External Signal known as teacher controls learning and incorporates information.

(ii) un supervised learning:- no External signal (teacher) is used in learning Process. the neural n/w relies upon both internal and local information.

→ unsupervised learning is more plausible model of training in biological system
it requires no target vector for the o/p. and hence no comparision to predetermined ideal responses. the training set consists of solely of i/p vectors.

→ The training algorithm modifies n/w weights to Produce o/p vector to Produce in same Pattern.

<u>Types of basic learning mechanisms</u>:- (EMHCB)

(i) Error Correction mechanism
(ii) Memory Based learning "
(iii) Hebbian learning
(iv) Competitive learning
(v) Boltzmann learning.

(a) <u>Error Correction learning</u>:- (Recognization)

A <u>n/w is trained</u> so <u>that application</u> of set of <u>i/p produces desired o/p's</u>. Each such i/p (or) o/p is referred to vector. <u>Training is accomplished by</u> sequentially <u>applying i/p vectors</u> <u>while adjusting n/w weights according to predetermined procedure</u>. <u>During training the n/w weights gradually converge</u> to values <u>such that each i/p vector produces</u> the <u>desired o/p vector</u>.

Let us Consider:- n time index, then the error at n^{th} instant case obtained as function of actual o/p of o/p layer be $O_k(n)$ for k^{th} neuron and desired target response (or) target o/p of n/w be $d_k(n)$ for k^{th} node. denoted by $e_k(n)$.

$\therefore$ O/p layer error signal

$$e_k(n) = d_k(n) - O_k(n) \rightarrow (a)$$

The error signal $e_k(n)$ is used to adjust the synaptic weight of neuron k the <u>Cost function</u> (or) <u>Performance index</u> $E(n)$ & given interms error signal

$$E(n) = \frac{1}{2} \cdot e_k^2(n). \rightarrow (b)$$

→ This kind of learning based on many learning rules which are derived in order to update the weights of n/w. like

(i) Perceptron learning rule
(ii) Delta learning rule
(iii) Widrow Hoff learning rule.

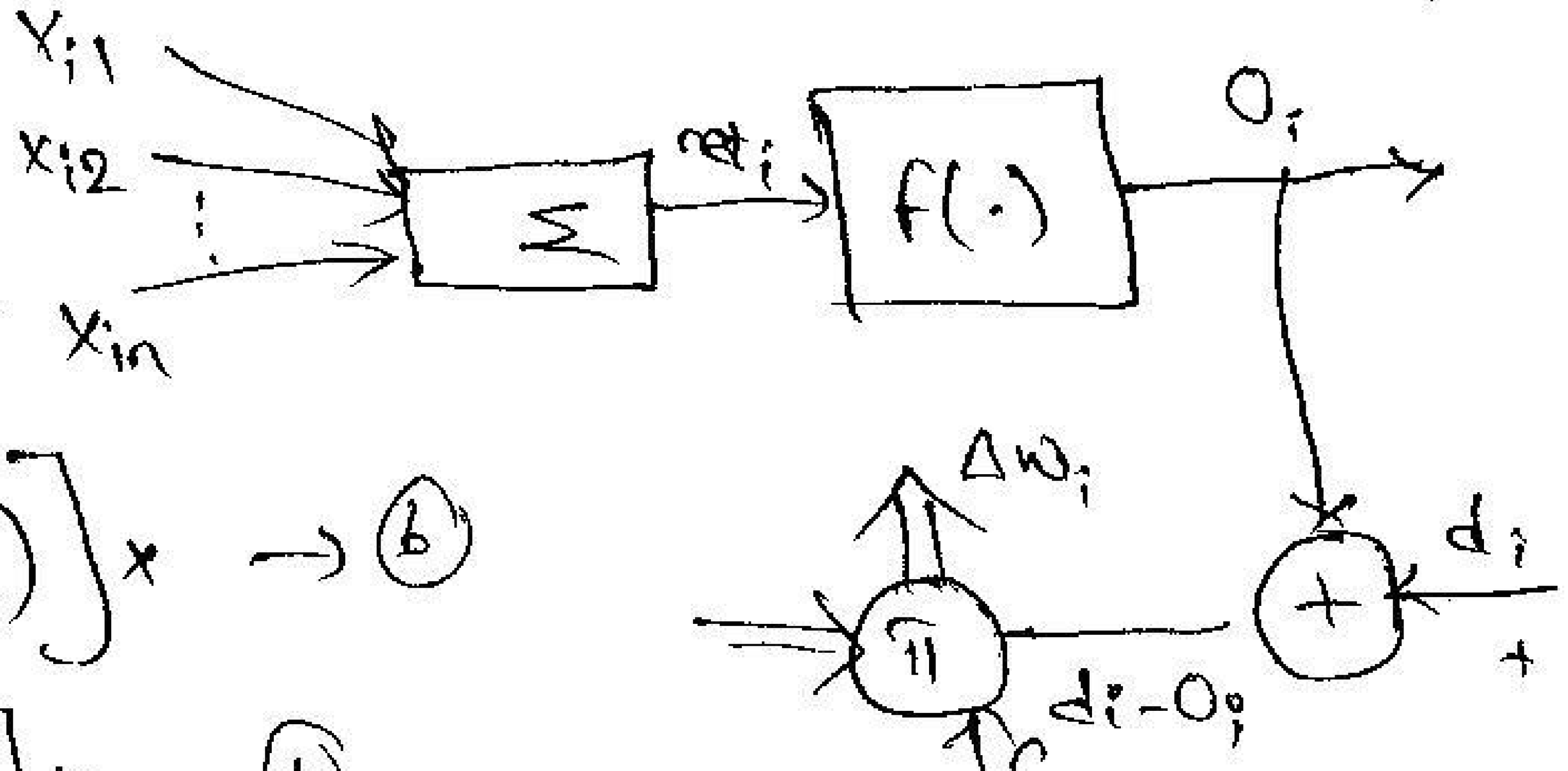
(i) Preceptron learning rule :- (Rule is applicable only for binary neuron response)

Learning signal known as error signal defined as difference b/w the desired and actual neuron response

$$e_i = d_i - O_i \rightarrow (a)$$

where $O_i = \text{Sgn}(W_i X)$ and $d_i \rightarrow$ desired response

weight adjustments in this method ΔW_i and ΔW_{ij} are defined as



$$\begin{cases} \Delta W_i^1 = c\left[d_i - \text{Sgn}(W_i X)\right]x \rightarrow (b) \\ \Delta W_{ij} = c\left[d_i - \text{Sgn}(W_i X)\right]x_j \rightarrow (c) \end{cases}$$

for $J = 1 \rightarrow n$.

the relationship (b) & (c) express the rule for bipolar binary case.

under this rule weights are adjusted if and only if incorrect.

since the desired response is either 1 (or) -1 the weight adjustment reduces to

$$\Delta W_i^1 = \pm 2cX$$

$C \rightarrow$ learning constant.

(ii) Delta learning rule :-

→ introduced for training a neural n/w.

→ This learning rule is applicable only for continous activation function.

→ This rule can be derived from condition of least square error b/w 'd_i' and 'O_i'.

→ The squared error may be defined as

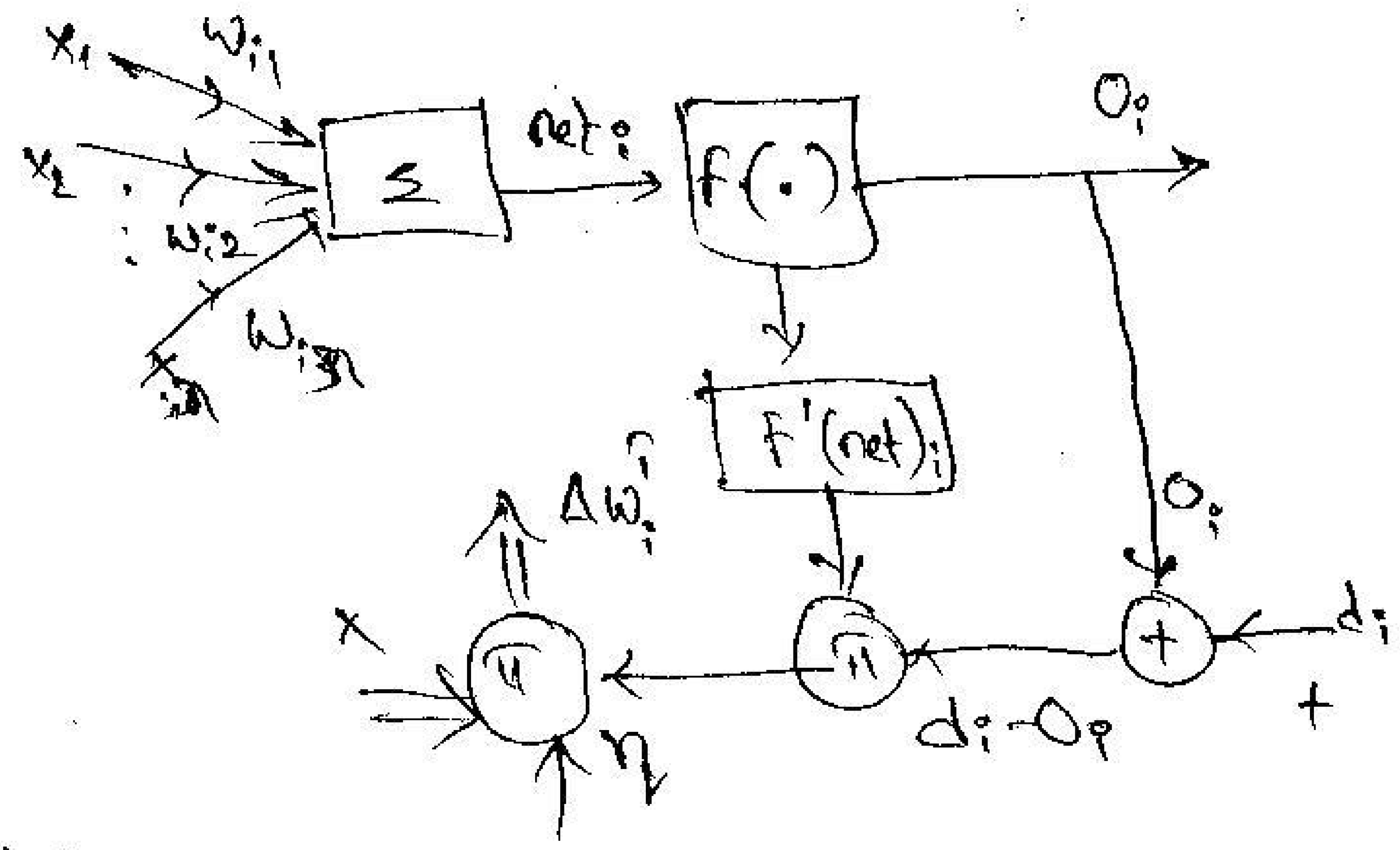
$$E = \frac{1}{2}(d_i - O_i)^2 = \frac{1}{2}\left[d_i - f(W_i X)\right]^2 \rightarrow (a)$$

→ The error gradient vector with respect to W_i may be written as

$$\nabla E = -(d_i - O_i)\, f'(W_i X)\, x \rightarrow (b)$$

The elements of gradient vector are $\frac{\partial E}{\partial W_{ij}} = -(d_i - O_i)\, f'(W_i X)\, x_j$ for $J = 1 \rightarrow n$. $\rightarrow (c)$

In order to minimize the cost function E, it requires weight changes to be in '-ve gradient direction'.



$$\therefore \Delta W_i = -\eta \nabla F \rightarrow (d)$$

where η is a positive constant

from eq (b) & (d) we write obtain

$$\Delta W_i^T = \eta (d_i - O_i) F' (W_i X) \cdot X = \eta (d_i - O_i) f'(net_i) X \rightarrow (e)$$

since $net_i = W_i X$ and single weight adjustment becomes

$$\Delta W_{ij} = \eta (d_i - O_i) f' (net_i) x_j \quad \text{for } i = 1, 2 \ldots n$$

for this rule, the weights can be initialized to any value. ~~The~~

(iii) Windrow-Hoff Learning rule :-

this rule is independent of activation function of neurons. It minimizes the squared error b/w the desired o/p d_i and the neurons activation value $net_i = W_i X$

$$\therefore \text{Error} \quad e_i = d_i - O_i = d_i - W_i X \rightarrow (a)$$

using learning rule change in weight vector

$$\Delta W_i = \eta (d_i - W_i X) X \rightarrow (b)$$

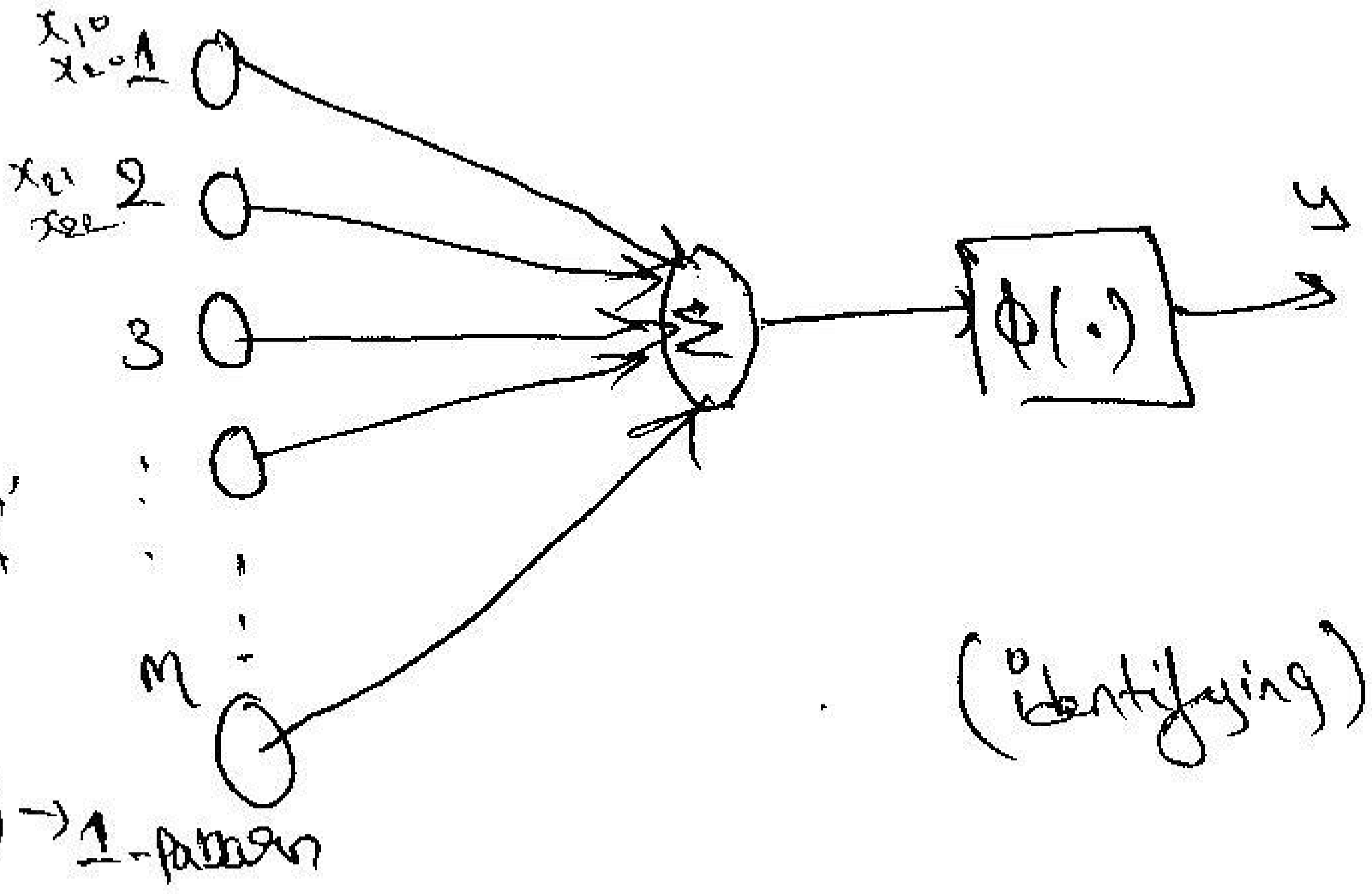
single weight adjustment written as

$$\Delta W_{ij} = \eta (d_i - W_i X) \quad \text{for } j = 1 \rightarrow n \rightarrow (c)$$

This rule is special case of delta learning rule i.e $f(net) = net$ and $f'(net) = 1$

This rule is also called as least mean square (LMS) learning rule.

(b) Memory based learning :-



let Conside 1 - Pattern of x the $\bar{X}$ is vector and of o/p of x' is d_i → vector. $\{\bar{X}_i, d_i\}_{i=1}^{N}$ → 1-pattern

$\{\bar{X}_i\}_{i=1}^{N}$ =

(identifying)

Let an i/p be $\bar{X}_{test}$ and what is o/p of system. out of these vectors $\bar{X}_{test}$ searches for nearest vector among m vector i/p.

if 1 of is closest among 'm'.

$x'_N \in \{\bar{x}_1, \bar{x}_2, \bar{x}_3 \dots x_m\}$

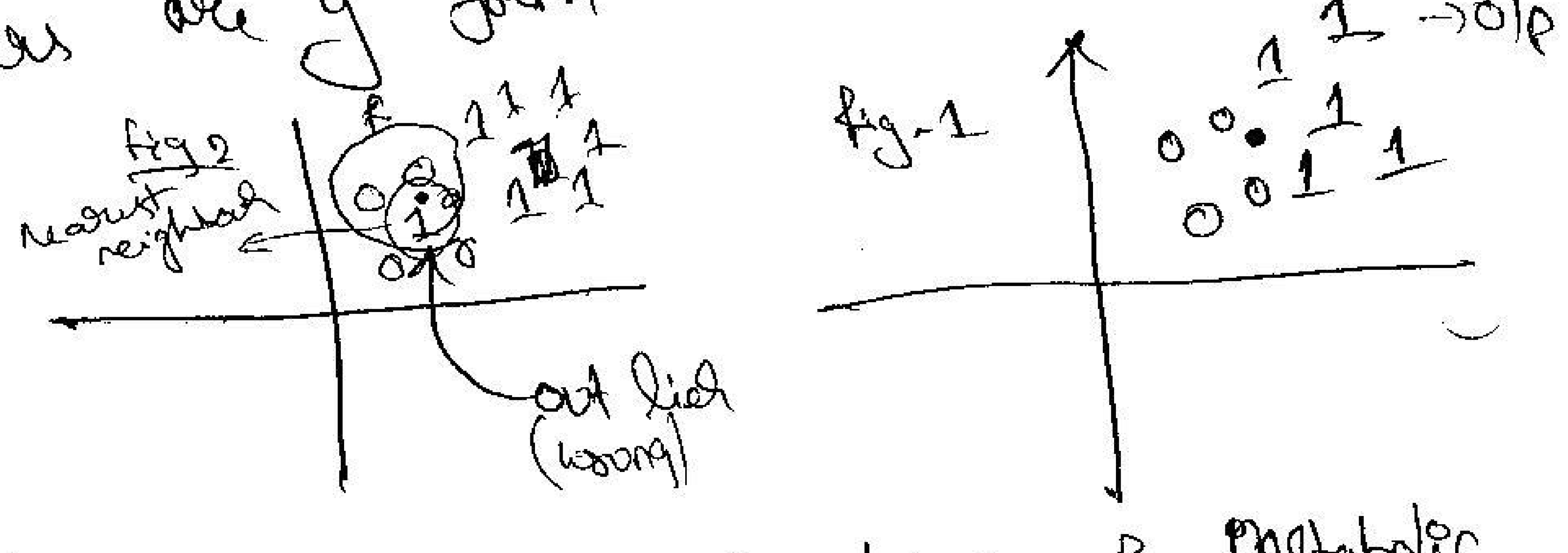
↓ nearest neighbor of $\bar{x}_{test}$ if minimum of euclidean distance b/w

$d(\bar{X}_P$ & $\bar{X}_{test})$ is searched by varying $P = 1 \to N$

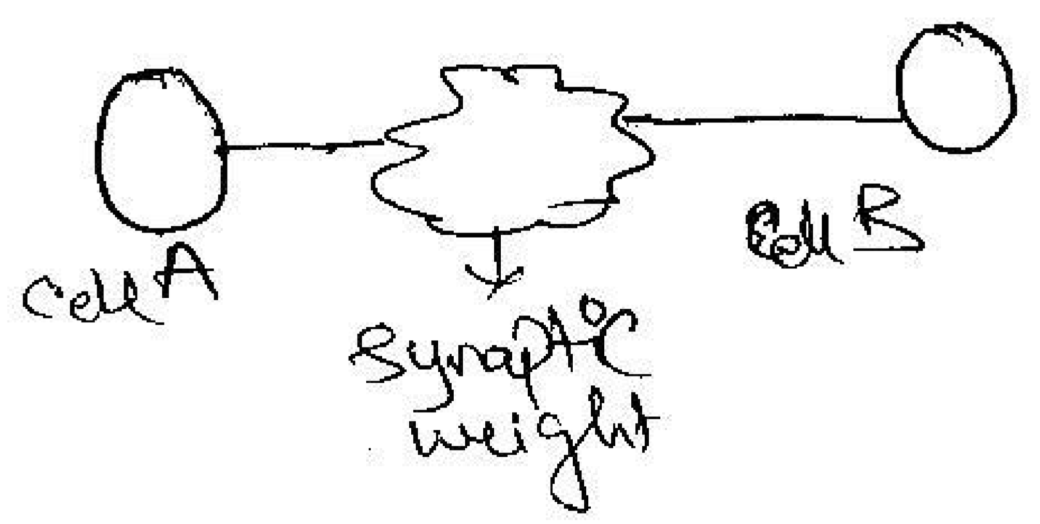
$\min d(\bar{X}_i, \bar{X}_{test}) = d(\bar{X}'_N, \bar{X}_{test})$

let us Consider Patterns are of form in classes. like '0' & '1'

[K-nearest neighbour classification]



(c) Hebbian learning :-



A's efficiency in firing B metabolic changes happen so that efficiency of A firing B increases.

Hebbian synapses → 3-characters

(a) very much time dependent (Synchronous should be)

(b) local

(c) strongly interactive

If two neurons on either side of synapse (connection) are activated (1-31) simultaneously (i.e synchronously) then the strength of that synapse is selectively increased

If two neurons on either side of a synapse are activated asynchronously, then that synapse selectively weakened (or) eliminated.

→ Hebbian synapse defined as synapse that uses a time dependent, highly local, and strongly interactive mechanism to increase synaptic efficiency as a function of correlation b/w presynaptic & post synaptic activity.

→ Key Properties:-

(i) Time dependent mechanism:- refers modification in a hebbian synapse depend on exact time of occurance of presynaptic & post signals.

(ii) Local mechanism:- A synapse is transmission site where information-bearing signals are in spatiotemporal contiguity.

(iii) Interactive mechanism:- change of hebbian synapse depends on signals on b.s of synapse. i.e hebbian learning depends on a true interaction b/w presynaptic & post synaptic signals in the sense that we can not make a prediction from either one of these two activities by itself.

(iv) Conjuntional (or) correlational mechanism:- According to one of Hebbian interpretation the co-occurence of pre & post synaptics signals is sufficient to produce synaptic modification. for another interpretation of hebbian learning one may think of interactive mechanism charatarizing a hebbian synapse an statistical term.

(ii) Mathematical relation:- Consider a synaptic weight w_{kj} of neuron 'k' with pre & post synaptic signals by x_j and O_k respectively.

∴ the change of weight is $\Delta w_{kj}(n) = f(O_k(n), x_j(n)) \rightarrow (a)$

Post synaptic   Pre synaptic

Simple form of Hebbian is given as

$\Delta w_{kj}(n) = \eta\, O_k(n) \cdot x_j(n) = \eta\, f(w_j x)\, x_j(n)$ for $j = 1 \rightarrow n$ → (b) where $\eta \rightarrow$ learning rate & Positive constant.

{ +ve ly correlated leads synaptic strengthing. } (binary Activation)
{ -ve ly " (or) un correlated synpatic weakining }

(iii) Classification of synaptic modification:-

(i) Hebbian :- synapse increases its strength with +ve correlation

(ii) Anti hebbian :- " " " " but " -ve "

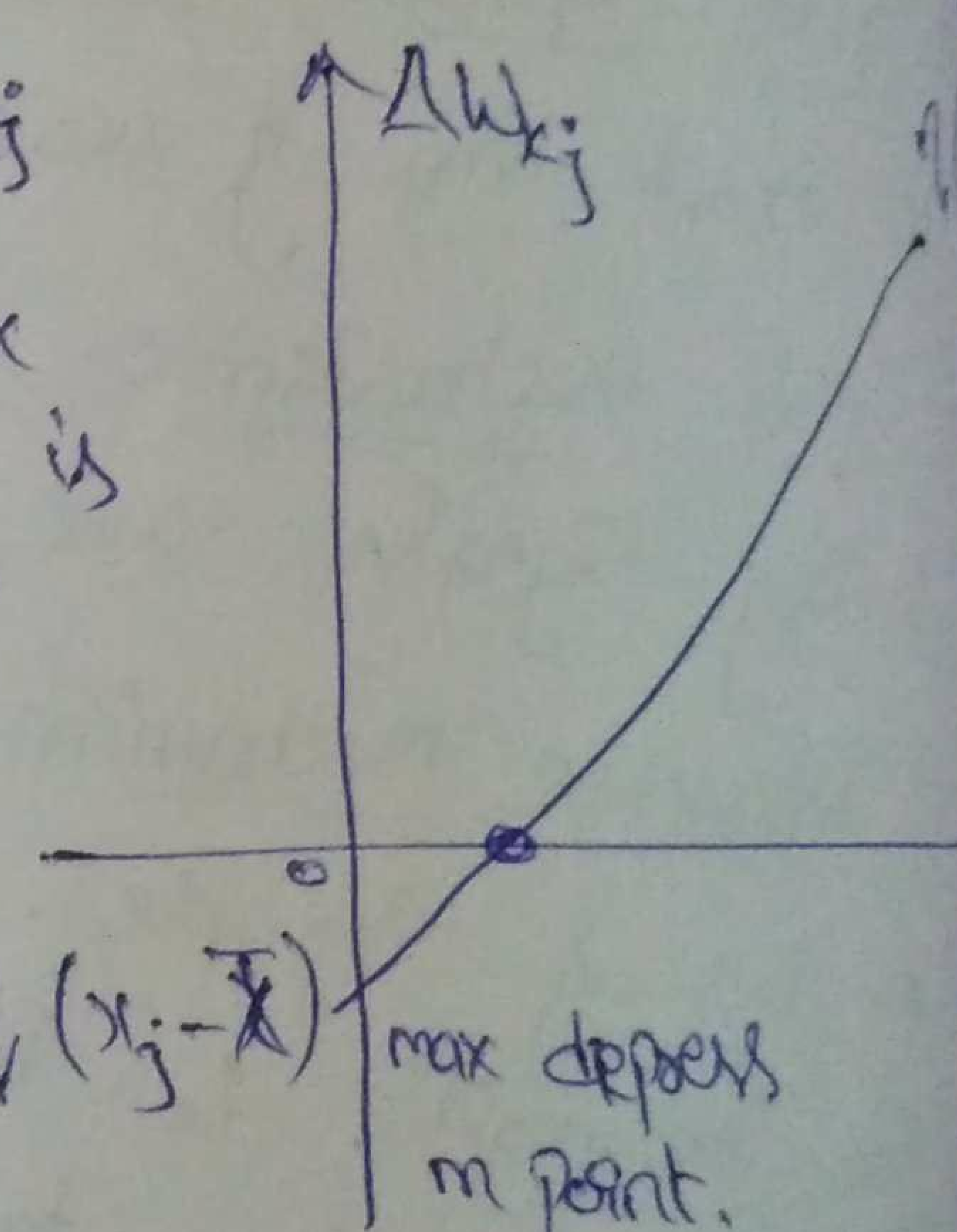
(iii) non- " :- which doesn't involve hebbian mechanism of either kind.

(iv) Covariance hypothesis :- In this hypothesis Pre- & Post synaptic signals are replaced by deviation of Presynaptic & Post synaptic signals from their respective average values over a certain time interval.

Let $\bar{x}$ = is time averaged value of x_j
$\bar{y}$ = " " " " " y_k

The adjustment applied to synaptic weight w_{kj} is

$$\Delta w_{kj} = \eta (x_j - \bar{x})(y_k - \bar{y})$$



$\therefore$ the hebbian rule represents Purely feedforward, unsupervised learning.

(i) $w_{kj} \uparrow$ if $x_j > \bar{x}$ & $y_k > \bar{y}$

(ii) $w_{kj} \downarrow$ if either → (a) $x_j > \bar{x}$ & $y_k < \bar{y}$
→ (b) $x_j < \bar{x}$ & $y_k > \bar{y}$

(iii) $w_{kj} = 0$ no operating (it is within its average) → no change in weights.

Competitive Learning :-

(P-41)

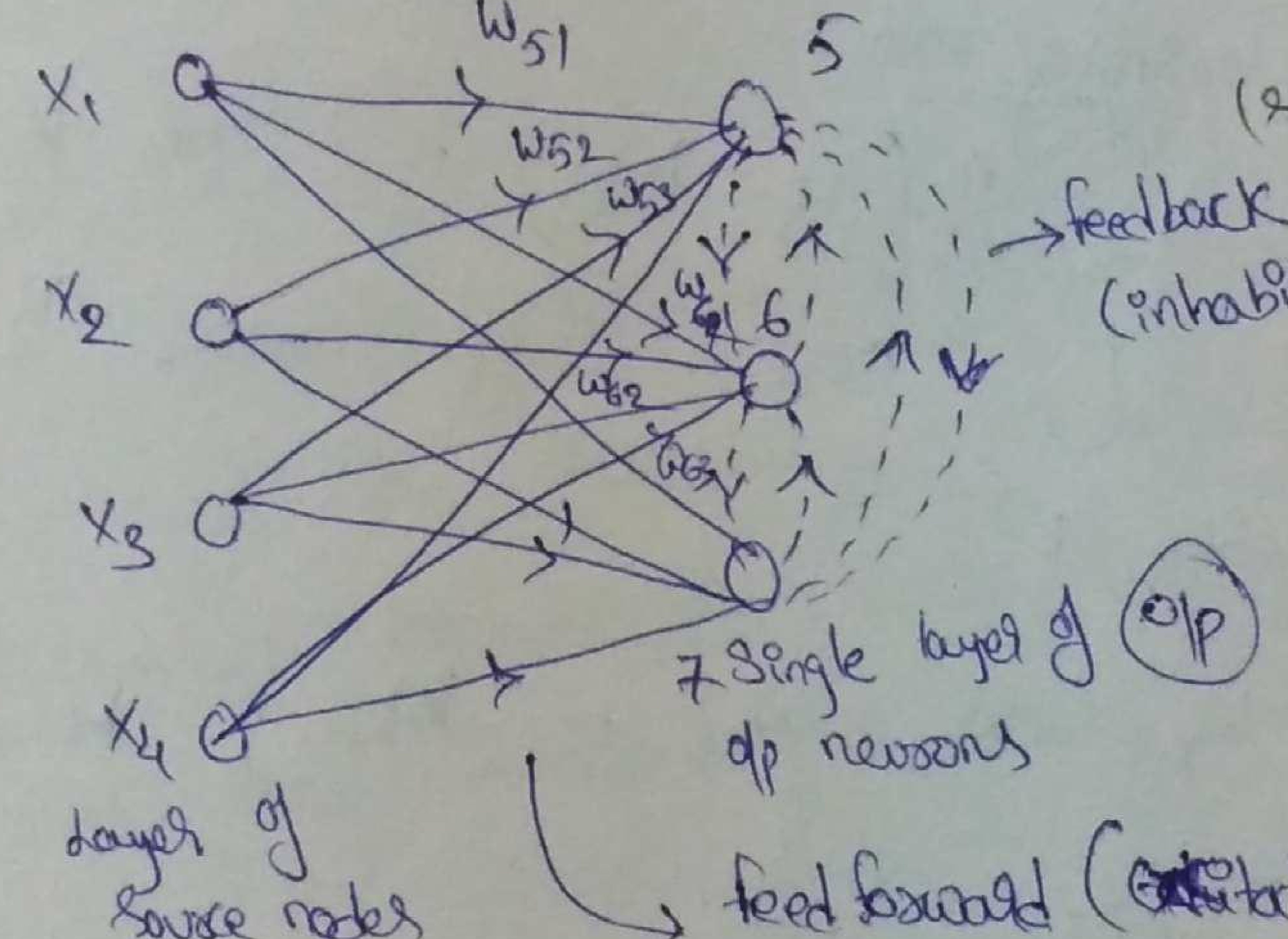
In Competitive learning only a single o/p neuron is active at any one time.

→ A set of neurons that are all the same except for some randomly distributed synaptic weights, which therefore respond differently to given set of i/p patterns

→ A limit imposed on strength of each neuron

→ A mechanism that permits 2 neurons to compete for the right to respond to a given subset of o/p such that only one o/p neuron (or) only one neuron per group is active (i.e on) at a time.

→ The neuron that wins the competition is called a winner take all neuron.



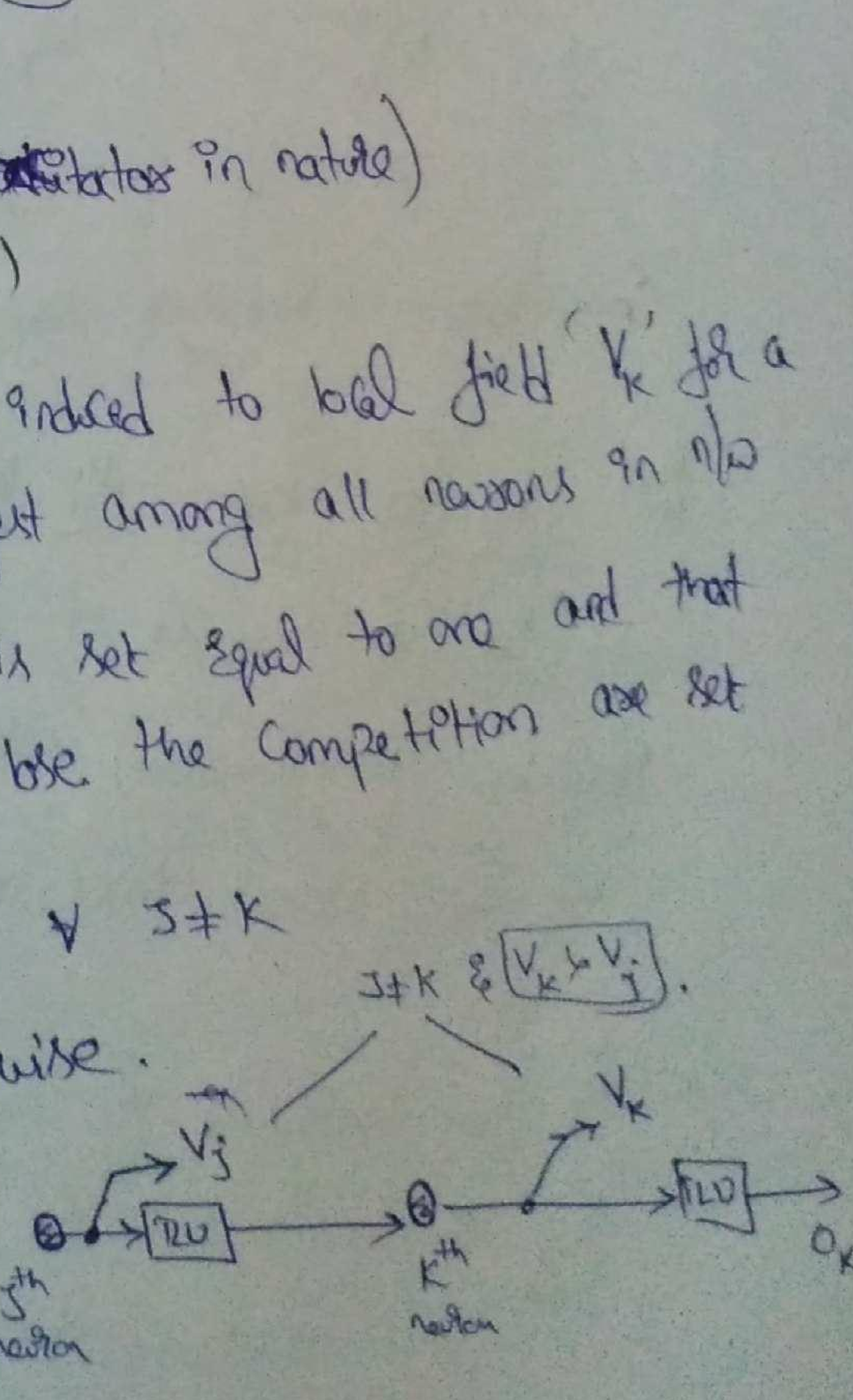
(i) Mathematical relation :-

For a neuron 'k' to be winning neuron its induced to local field 'V_k' for a specified i/p pattern 'x' must be the largest among all neurons in n/w

→ the o/p signal y_k of winning neuron 'k' is set equal to one and that the o/p signal of all the neurons that lose the competition are set equal to zero

$$y_k = \begin{cases} 1 & \text{if } V_k > V_j \quad \forall\ j \neq k \\ 0 & \text{otherwise.} \end{cases}$$

$j \neq k$ & $\boxed{V_k < V_j}$.



where local field V_k represents Combined action of all the forward and feedback i/p to neuron 'k'.

→ Let W_{kj} denote synaptic weight connecting o/p node $J \rightarrow k$ neuron.

If each neuron is alloted a fixed amount of synaptic weight (i.e all synaptic weights are positive) which is distributed among its i/p node i.e

$$\sum_j W_{kj} = 1 \quad \text{for all 'k'.}$$

→ The neuron then learn by shifting weights from inactive to input nodes.

→ If neuron does not respond to a particular o/p pattern, no learning place in neuron

from Competitive learning rule :-

$$\Delta W_{kj} = \begin{cases} \eta (x_j - w_{kj}) & \text{if neuron } k \text{ wins} \\ 0 & \text{if neuron } k \text{ loses} \end{cases}$$

moving synaptic vector 'k'.

$$\vec{W_k} = [W_{k1}, W_{k2}, W_{k3} \ldots W_{km}] \quad k \rightarrow m-1$$

$$\vec{X} = [X_1, X_2 \ldots X_m]$$

Competative learning defines moving $\vec{W_k}$ towards i/p pattern $\vec{X}$,

(ii) Winner take All learning rule :-

It is Example of Competitive learning and is used for unsupervised training (feed forward) network..

→ The winner take all learning is used for learning statistical properties of o/p's

→ The learning is based on premise that one of neuron in the say the m^{th}, has the maximum response due to o/p 'x' Even this neuron is declared the winner. As a result of wining weight vector W_m.

$$W_m = [W_{m1}, W_{m2} \dots W_{mn}].$$

The weights adjusted given as unsupervised learning step.

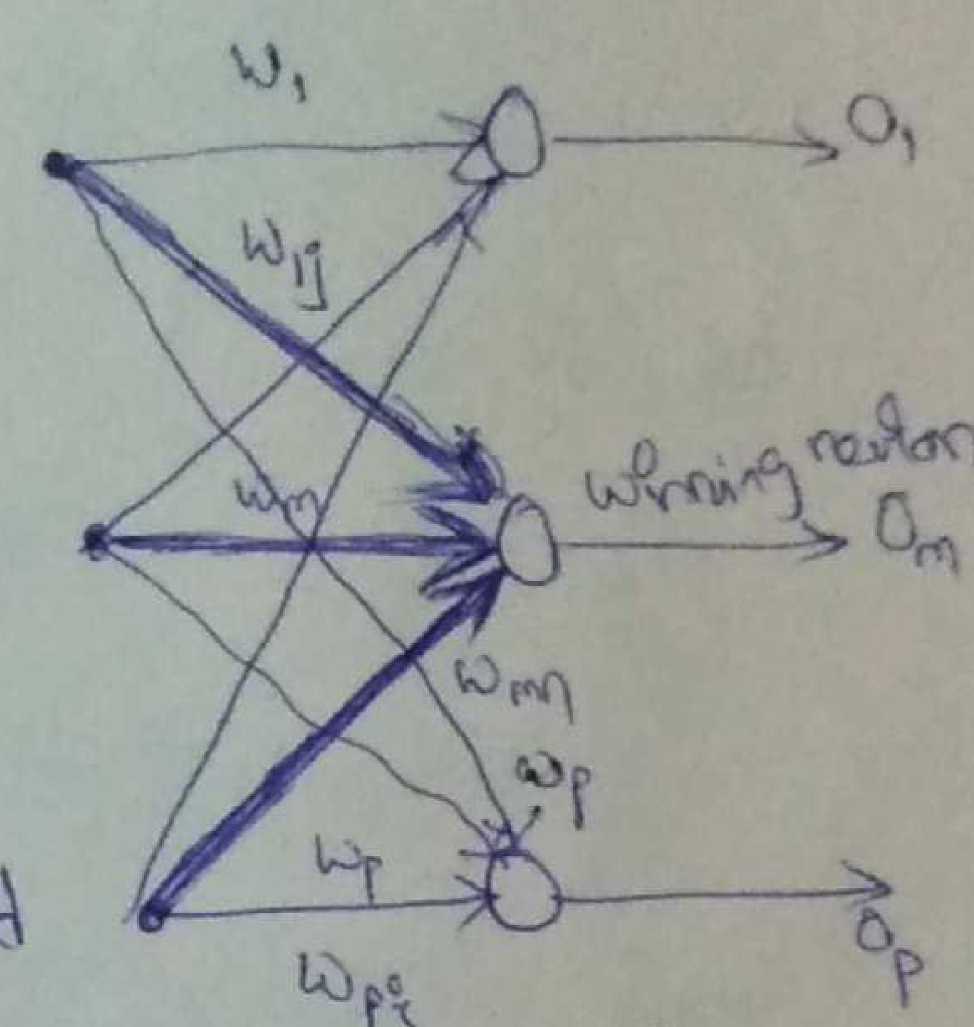
Computed as $\Delta W_m^T = \eta (X - W_m^T) \rightarrow$ (b)

for individual weight adjustment $\Delta W_{mj} = \eta (X_j - W_{mj})$ for $j = 1 \rightarrow n \rightarrow$ (c)

$\eta \rightarrow$ small learning constant.

winner selection is based on criterion of maximum activation among all P- neurons participating in a competition.

$$W_m X = \max(W_i X) \quad i = 1, 2 \dots P.$$



(adjusted weights are highlighted).

(e) Boltzmann learning:-

is a statistical learning algorithm derived from ideas rooted in statistical machines.

In Boltzmann machine the neurons form a recurrent structure and they operate in a binary manner. denoted as +1 / -1 (ON / OFF)

→ In Boltzmann machine no hidden layer is present.

→ This machine is characterized by an energy function 'E'.

$$E = -\frac{1}{2} \sum_j \sum_k W_{kj} X_k . X_j \rightarrow \text{(a)}$$

$$j \neq k$$

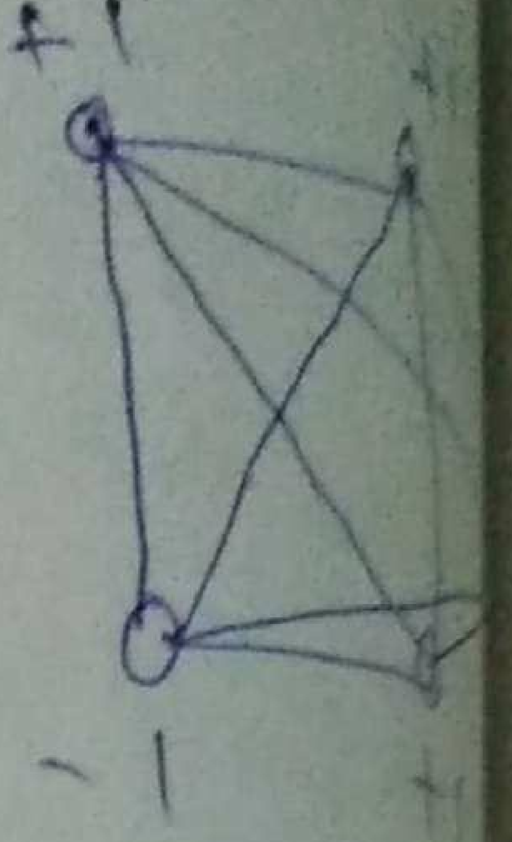
X_j is state of neuron j and W_{kj} is synaptic weight connecting neuron j to 'k'. The fact that $j \neq k$ mean simply none of neurons in machine has self feedback.

→ Boltzmann learning defined particular state of neuron we will have energy function.

if i pickup particular neuron 'k' change state (x_k)

from $(-1) \to (+1)$ it is said to be recompute

∴ The probability of neuron flip 'k' is x_k



$$P(x_k \to -x_k) = \frac{1}{1+e^{(-\Delta E_k/T)}}$$

ΔE_k → change of Energy resulting from such a flip (-1) to $(+1)$ (or) $(+1$ to $-1)$

T → Pseudo temperature.//

The neurons of Boltzmann machine Partition in two groups

1) visible → Provide interface b/w network (Available at o/p)

2) hidden → it operates freely (in vailable at o/p).

there are 2-modes of operation.

→ clamped condition in which visible neurons are all clamped specific state determined by Environment. (either +1/-1)

→ free running condition in which all neurons (visible & hidden) are allowed to operate freely.

Let ρ^+_{kj} = correlation between neuron k and neuron j in clamped condition

are binary quantities of (+1 (or) -1)

ρ^-_{kj} = correlation b/w neurons in free running condition.

$$\Delta w_{kj} = \eta(\rho^+_{kj} - \rho^-_{kj}) \quad j \neq k$$