Syllabus.

Artificial Neural Network

UNIT 1.

Introduction to ANN.

Introduction to ANN, History of Neural Network, structure and working of Biological Neural Network, Neural Nets architecture, Topology of neural network architecture.

Features: Characteristics, Types, Activation functions, model of neuron - Mc Culloch & Pitts model, Perceptron, Adaline model, Basic learning laws, Applications of neural networks, Comparison of BNN and ANN.

UNIT 2

Learning Algorithms.

Learning & Memory, Learning Algorithms, Number of hidden nodes, Error Correction, Gradient Decent Rules, Perceptron Learning Algorithms, Supervised Learning Backpropagation Multilayer Network Architecture, propagation learning Algorithms, Feed forward & feed back neural networks, example & applications.

- Controlling Water Reservoirs, Rule Extractions
- Medical diagnosis, Automated trading systems.

UNIT-3

Learning Algorithms.

Learning

UNIT-3

Associative Learning.

~~Introduction, Associative learning, Hopfield network, Error, Performance in Hopfield networks, Simulated annealing, Boltzmann machine and Boltzmann learning, State transition, State transition diagram and false minima problem, Stochastic update, Simulated annealing.~~

Basic functional units of ANN for pattern recognition tasks: Pattern association, pattern classification & pattern mapping tasks.

UNIT 4

Competitive Learning Neural Network.

Components blocks of ANN, Architecture, convolution / pooling layers

- Understanging catastrophic Inference in neural nets.
- Translating System for Face-to-Face Dialog & Intelligent Help Systems.

PAGE NO.:
DATE: / /

Components of CL network, Pattern clustering and feature mapping network, ART network Features of ART models, character recognitions using ART network.

Self Organization Maps (SOM): Two basic feature Mapping models, Self-Organization Map, SOM Algorithms, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification.

UNIT-5

Convolution Neural Network

Building blocks of CNN's, Architectures, convolution pooling layers, Padding, Strided convolutions, Convolution over volumes, Softmax regression Deep Learning Framework, training & testing on different distributions, Bias & Variance with mismatched data distribution. Transfer learning multi-task learning, end-to-end deep learning, Introduction to CNN models: LeNet-5, Alex Net, VGG-16 Residual Networks.

Case Study :

PAGE NO.:
DATE: / /

UNIT 6

Applications of ANN.

Pattern classification - Recognition of Olympic games symbols, Recognition of printed characters.

Neocognitron - Recognition of handwritten characters. Net Talk: to convert English text to speech, Recognition of consonant vowel (CV) segments, texture classification & segmentation.

Case Study: Automating language translation.

PAGE NO.:
DATE: / /

30/1/2023
Monday.

Reference Book: Zurada.

Neural Network classification:

characteristics: { color, length, weight }

e.g. Rice, wheat.

color	length	weight
white	7mm	10mg.
⋮	⋮	⋮

forming rules.

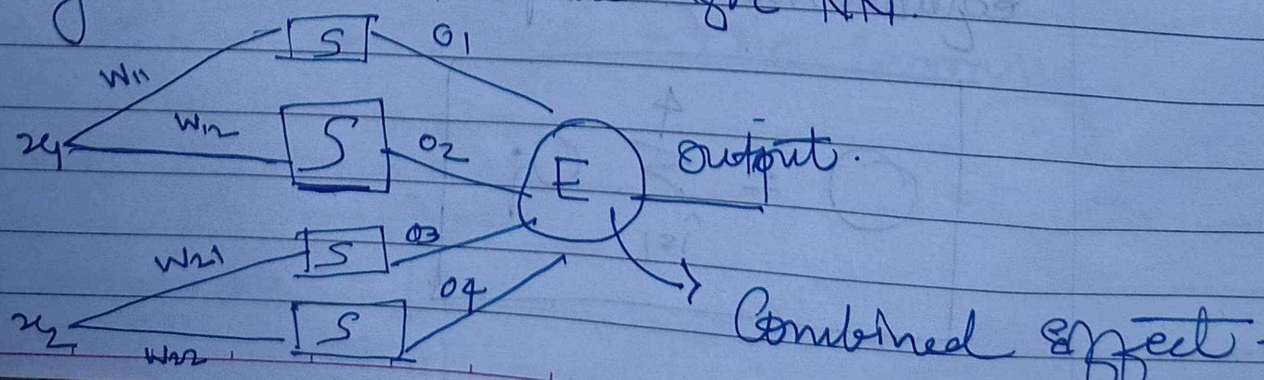
Important / Not Important.

Rules formation: Color → Brown & length > 7

Form Neural Network & train to get required data.

* Optimization for finding the Solution.

Layers choosen Neurons for NN.



Function are Computing

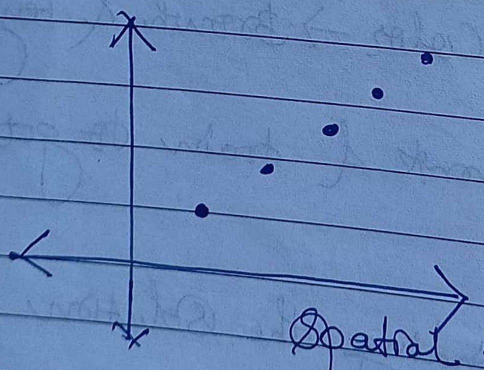
$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

$$(x_1, x_2) = (\quad)$$

$$(x_i, y_i) = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

Wx

$$\boxed{h_1 = (W_{11}x_1 + W_{21}x_2)\ O_1}$$



Spatial Representation

equilibrium state point

$$\boxed{O = \sum_{j=1}^{4} O_j}$$

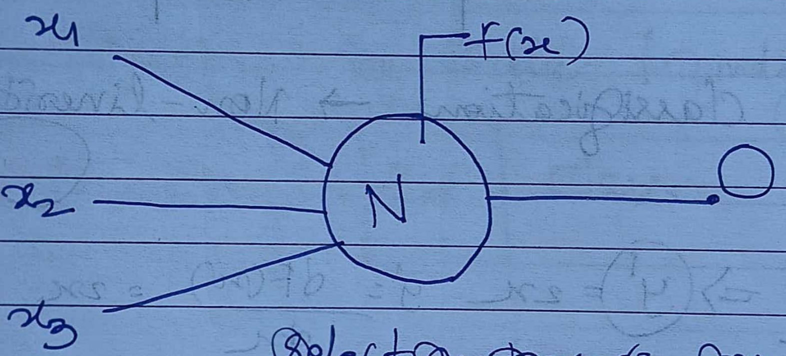
* Skewness of data : Left Skewed & Right Skewed.

1/2/2023
Wednesday.

Activation Functions :

- Binary, Linear, Non-linear. (10+)

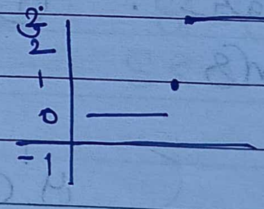
Purpose of A.F : Activate the neuron
keep as it is or fire it or not.



Selection process for achieving higher learning.

Binary : $f(x) \to 1/0$

if $x \geq 0$
fires 1

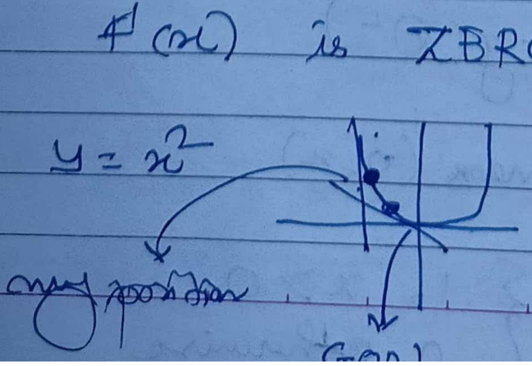


$x < 0$
fires 0

Gradient / Slope & Intercept

$f'(x)$ is ZERO. → No learning.

$y = x^2$



$y = mx + c$ | $m' = m - \delta m$ / fix the direction

m, c

$\downarrow \quad \downarrow$

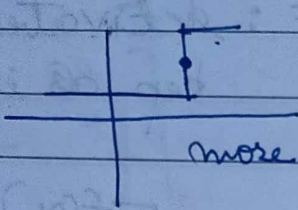
$\delta m \quad \delta c$

$m - \delta m \quad c - \delta c$

$c' = c - \delta c$

learning rate

Step function:



more steepness.

• multiclass classification → Non-linearity.

Softmax (extension of sigmoid)

$y = f(x)$

$y = x^2 \Rightarrow y' = 2x \qquad y' = \frac{dF(x)}{dx} = 2x$

$f(x) = x^2$

Nonlinear functions: ① Sigmoid: $\frac{1}{1+e^{-x}}$

$y'(x) = e^{-x}(1+e^{-x})^{-2}$

② tanh

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

$f'(x) = (1 - g(x)^2)$

③ Relu $\quad f(x) = \max(0, x)$

$f'(x) = \begin{cases} 1, & x > 0 \\ 0 & \text{otherwise} \end{cases}$

$P = VI$

$I = \frac{P}{V} = \frac{60}{200}$

Matplotlib → Plotting Numerical Data
Scipy → high level Scientific Computations
Gradient has reasonable accuracy

PAGE NO.:
DATE: / /

3/2/2023
Friday

Activation Functions.

Non-linearity
- Sigmoid, Tanh, ReLU, Softmax

Works better → Hidden layer (Sigmoid, Tanh, ReLU)

Softmax → Output layer

Equilibrium of Data. (finding equilibrium)

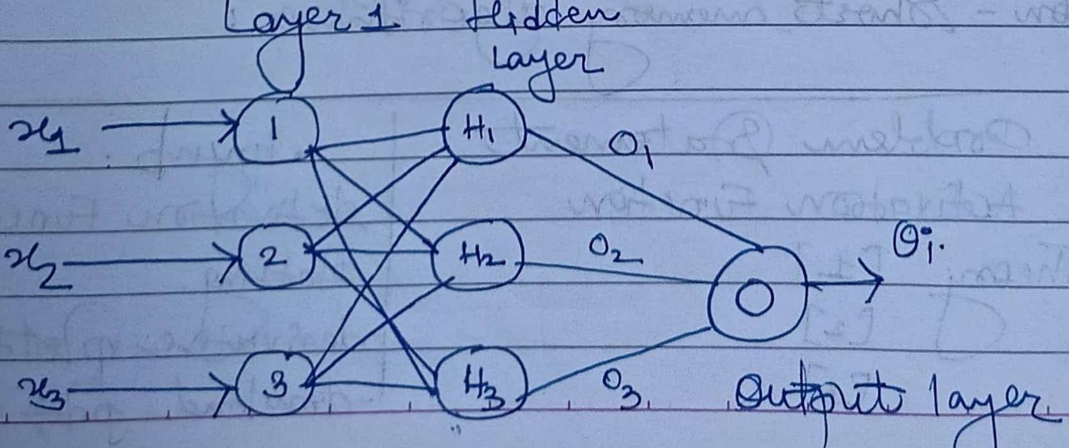
SL, SW,	PL,	PW	Class.
	3		
	2		
	1		
	4		

$\boxed{\frac{1}{N}\sum_{i=1}^{N} P_{i} W_i}$ mean

$\delta_i = (PW_i - \text{mean})^2$

$\delta < 0 \quad \delta > 0 \quad \delta = 0$

$\boxed{\sum_{i=1}^{N} \delta_i}$ $\sqrt{\delta_i}$.



Tensorflow → high performance numerical Computation

Pytorch → optimizes tensor Computations

NP array.

12	13	14
20	28	30
40	38	32

3×3

211	max
201	
25	min

max pooning
min pooning.

Hidden layer finds out the equilibrium stage.

Category : Output should be close to desired class.

Class 1 → 0.99

class 2 → 1.89

class 3 → 3.1

{ Neural Network works with high precision. }

Softmax → Output layer ⇒ we try to maximize Output layer.

$$\boxed{f(z) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_i}}}$$

function for maximizing.

LSTM Non-Short memory algorithms :

Assignment No. 1

Problem Statement
Activation Function
Theory [1]
[2]
[3]

- .ipynb
- Activation Functions plotting.
- derivative plotting
- download pdf

Scikit Learn → Classification Supervized & unsupervized ML.
Open CV → image processing.
Pandas → Data Science

PAGE NO.:
DATE: / /

Laboratory

Library: Scikit Learn
↳ Neural Network.
↳ Classifiers

from Scikit Learn neural_network import MLPclassifier

x_i	y_i	c_i
1	2	0
2	2	1
2	1	1
4	3	0
		1

Binary / multiclass.

$X = [[1,2],[2,2]\ \ldots]$

$C = [0, 1, 0, 1, 1 \ldots]$
(class)

Traning Data

MLPClassifier. fit (X, C)

Predict ([4, 5, 6],

McCulloch & Pitts model:

$$\boxed{u = \sum_{j=1}^{N} w_j y_j + \theta}$$

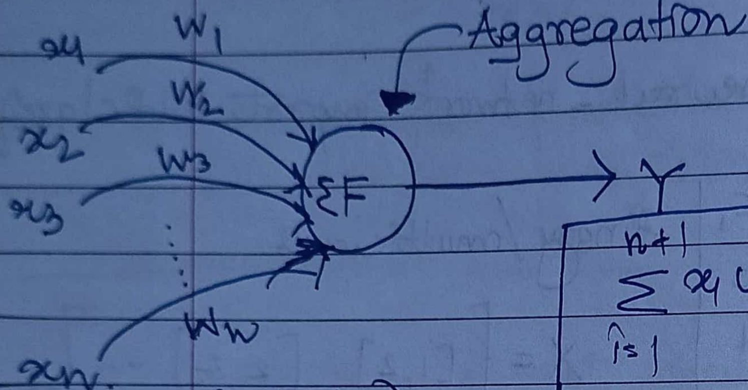
$w_j = 1 \leq j \leq N$

x_1 x_2
1 1

Aggregation ≥ 2 → AND gate

6/2/.. Monday

Mc Culloch Pitts Model
(MP model)



* No Backpropagation

$$\sum_{i=1}^{n+1} x_1 w_1 + x_2 w_2 + x_n w_n \ldots$$

Bias (1/-1/0)

Threshold (T)

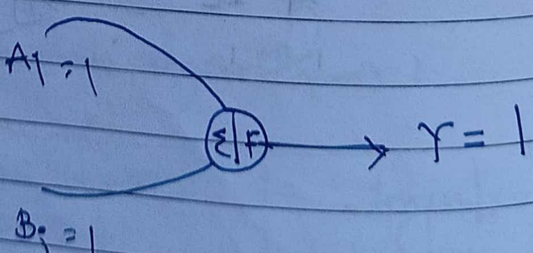
Value = X

if X > T
gives 1. Excitatory Stage

· if X < T Inhabitatory Stage
gives 0

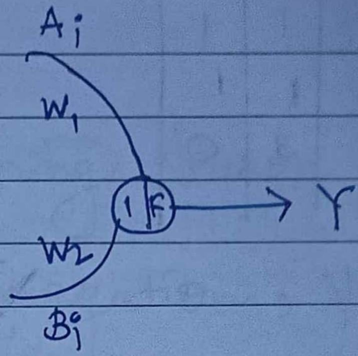
GATES : AND
OR
XOR
AND NOT
NOT

A_i	B_i	Y_i
0	0	0
1	0	0
0	1	0
1	1	1

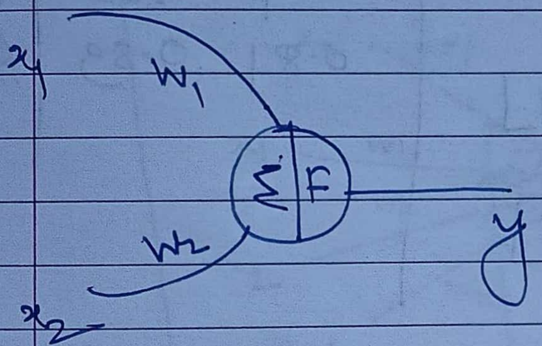


OR GATE

A_i	B_i	Y
0	0	0
0	1	1
1	0	1
1	1	1



XOR Gate : $\bar{A_i} \cdot B_i + A_i \cdot \bar{B}$

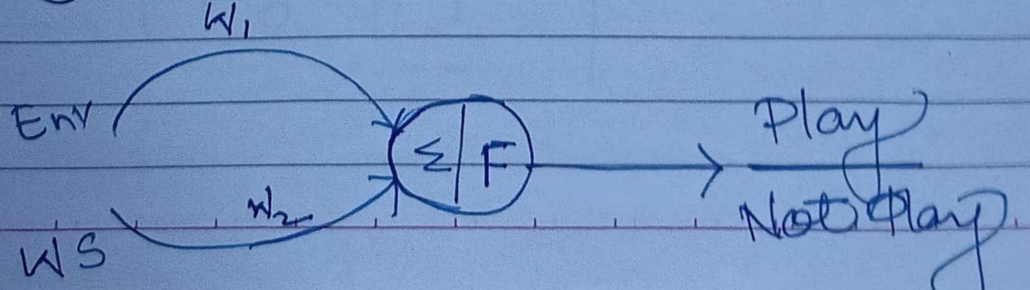


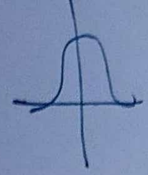
$$\Sigma(x_1 W_1 + x_2 W_2) = 1$$

* Weather Forecasting Data set :

Environment	Windspeed	class
sunny (1)	low (1)	Play — 1
rainy (2)	medium (2)	Not play
sunny (1)	high (3)	Not play

} 0





A_i	B_i	Y_i
1	1	1
2	1	1
2	3	0

$Y_3 \, [1, 0.5, 2.5]$

$$\frac{l}{\sqrt{l^2+m^2+n^2}}, \frac{m}{\sqrt{l^2+m^2+n^2}}, \frac{n}{\sqrt{l^2+m^2+n^2}}$$

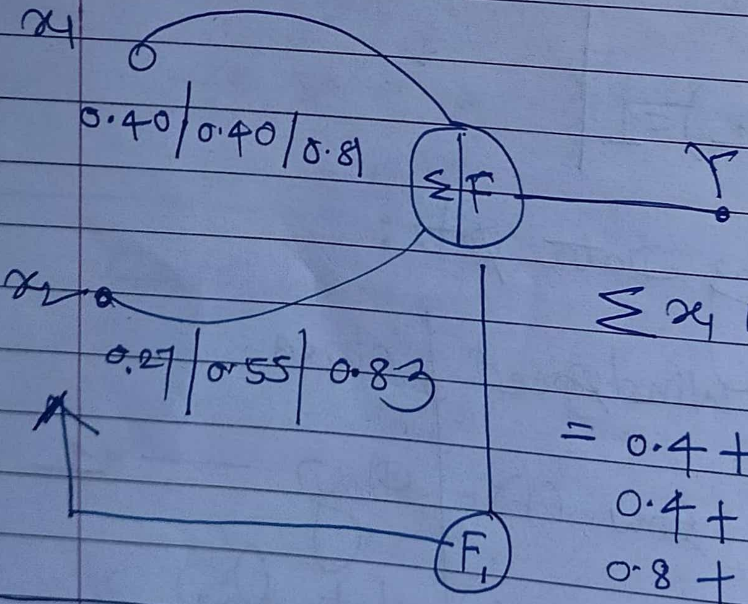
$Y_1 = [1, 1, 2]$

$Y_2 = [1, 2, 3]$

$V_1 = [0.40, 0.40, 0.81]$

$V_1 = [0.27, 0.55, 0.83]$

Y_1	Y_2	
0.4	0.27	1
0.4	0.55	1
0.81	0.83	0



$\Sigma x_1 w_1 + x_2 w_2$

$= 0.4 + 0.27 = 0.67 \neq 1$

$0.4 + 0.55 = 0.75 \neq 1$

$0.8 + 0.83 = 1.63 \neq 1$

Adaline Model.

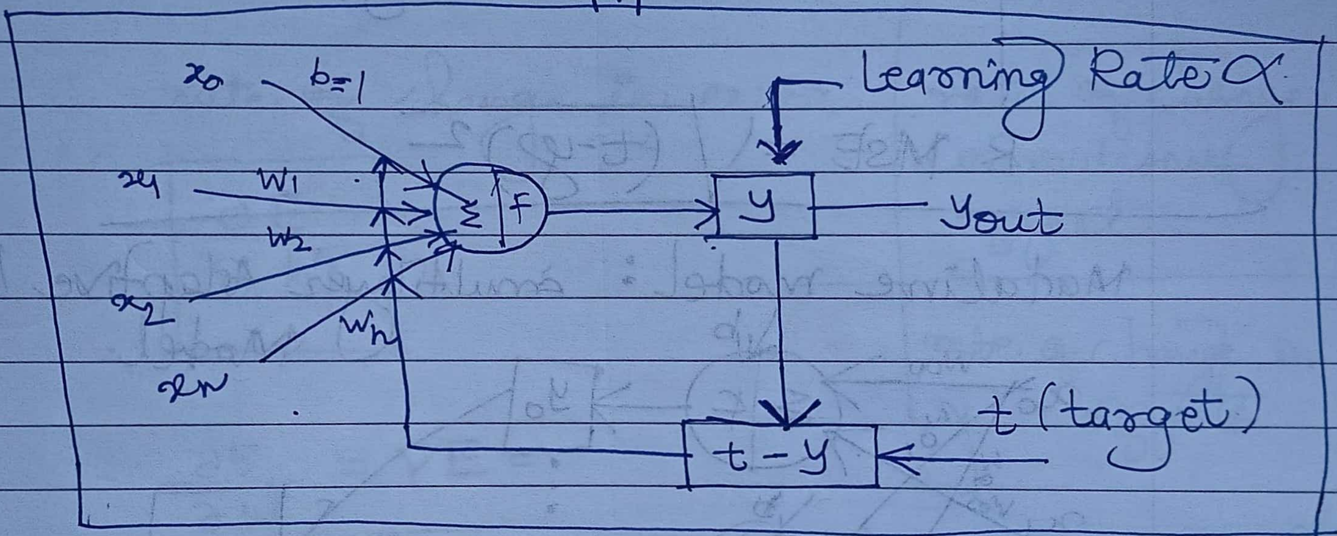
7/2/2023
Tuesday.

* Adaptive Linear Learning Neuron

MP model ➡ No Backpropagation, No increase in efficiency.

- Linear Approach.

#



$x_1 = 1$ & $x_2 = 1$ $t = 1$

Error → $\boxed{t-y}$, weights Non-zero

$x_1 = 1$ $x_2 = 1$ $t = 1$

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

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

$$\boxed{W_{new} = W_{old} + \eta(t-y) \cdot x_i}$$

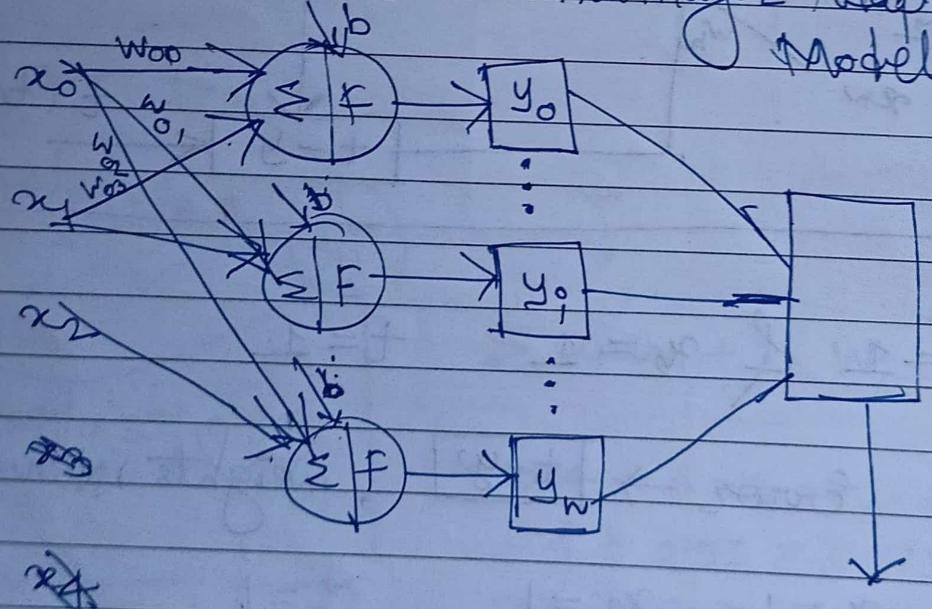
$$W_{i(new)} = w_i(old) + \eta(t-y) \cdot x_i$$

$(t-y) \approx .005$ (example) → Tolerance

$+$ | $-$ $\cancel{error} = e = (t-y)^2$ MSE

$$R\ MSE = \sqrt{(t-y)^2}$$

Madaline model: multilayer Adaptive Linear Model.



* Bipolar Activation functions :

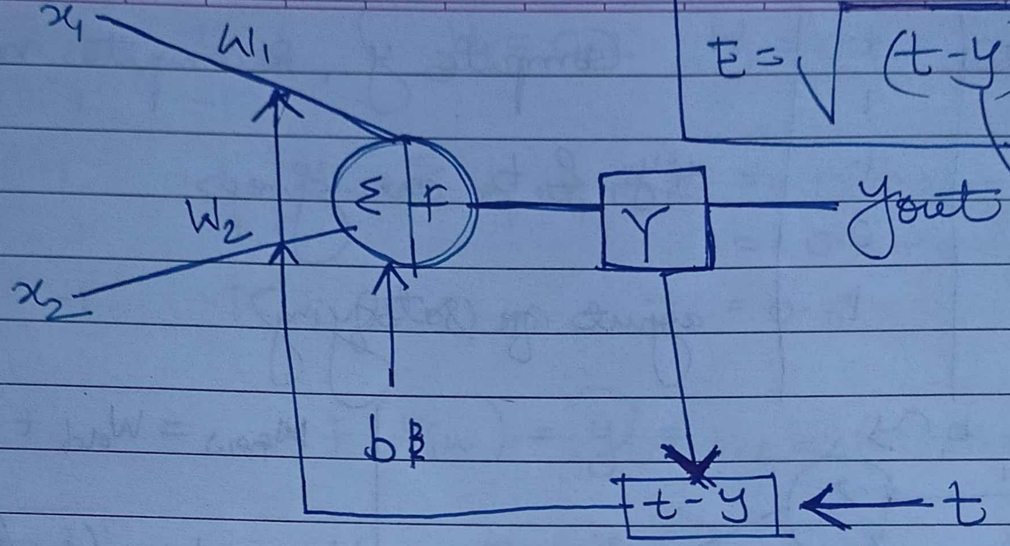
Adaline , Madaline

Learning Rules Data Rule .

change in weight till it reaches threshold .

8/2/2023
Wednesday

*



$$E = \sqrt{(t-y)^2}$$

→ Cummulative effect.

rate of change in error.

$$\boxed{\frac{\partial E}{\partial W_i}}$$

rate of change in Cummulative effect.

rate of change in y → effects

$$\frac{\partial E}{\partial W} = \nabla E =$$

$$\frac{\partial}{\partial W_1} = (x_1 W_1 + x_2 W_2 + b)$$

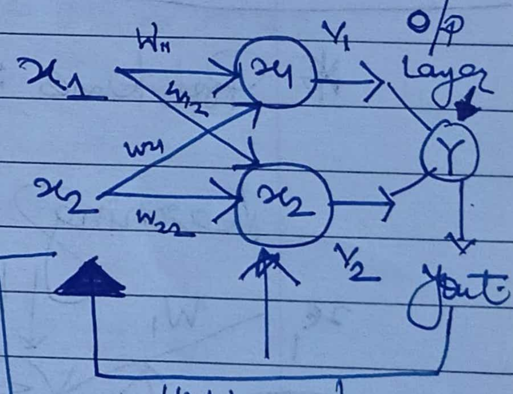
$$\nabla E = (t-y)^2 = t^2 - 2ty + y^2$$

$$\therefore \quad 0 - 2t + 2y$$

$$\nabla E = -2(t-y)$$

$$\nabla E = -\alpha (t-y)$$

$$\boxed{\nabla E_{ij} = -\alpha (t - y_i)}$$



Hidden layer

$$\nabla E_{ij} = \frac{\partial E}{\partial W_i} = \frac{\partial}{\partial W_i} [t-y]^2$$

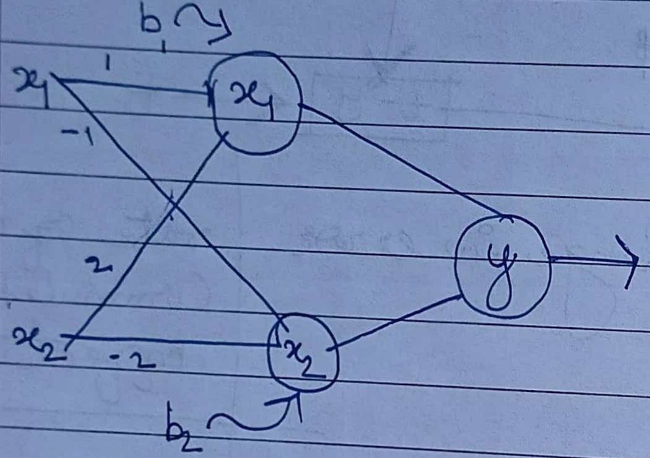
$$\nabla E_{ij} = -2 [t - j]$$

x_1	x_2	t
-1	-1	1
1	-1	1
1	0	0
0	1	1

Compute y, Recompute new weights

x_1, y, t are given

adjust for satisfying



$$\boxed{\alpha = 0.1 \text{ to } 1}$$

$W_{new} = W_{old} + \alpha(t-y)x_i$

$W_{new} = 1 + 1(1-1) - 1 =$

y will depend & calculated on gate values.

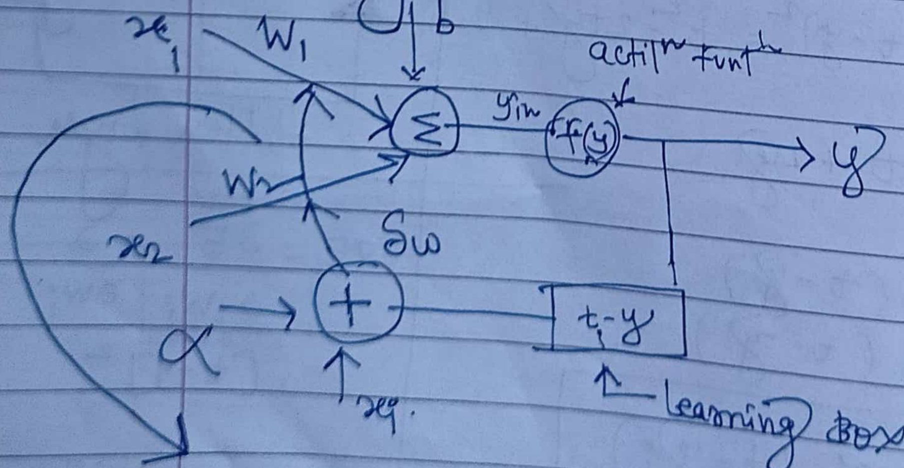
---

* Problems: Adaline / madaline

Backpropagation

Learning Rules.

10/2/2023. Friday.



$\delta w + W_1$

$\delta w + w_2$

$\boxed{x_i + w_i + b}$

Delta Rule: rate of change in weight.

If output comes zero we will stop.

PAGE NO.:
DATE: / /

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

$W_1 = 0.2$, $W_2 = 0.2$, $b = 0.2$, $\alpha = 0.2$

$y_{in} = \sum x_i w_i + b = 1\times 0.2 + 1\times 0.2 + 0.2$
$= 0.2 + 0.2 + 0.2$
$= 0.6$

$f(y_{in}) = y =$ $e = (-1-0.6) = -1.6$

Epoch

$E = (t_i - y)^2 = (-1.6)^2$

Current Error $= 2.56$

minimize Mean Valued Error.

Now modify the weights:

$nW_1 = 0.2 + \alpha(t_i - y)\cdot x_i$

1-

$= 0.2 + 0.2(-1.6)\cdot 1 = 0.2 + 0.32$
~~$= -0.30$~~
$nW_1 = -0.12$

$y = 1\times 0.2 + (-1)0.2 + 0.2 = 0.2$

~~$nW_2 = 0.2 +$~~

$e = (1-0.2) = 0.8$ | $E = 0.64$

$nW_1 = 0.2 + 0.2(0.8)1 = 0.36$
$W_2 = 0.2 + 0.2(0.8)(-1) = 0.04$

1.6
0.2
32
00
0.32

0.32
-0.20
0.12

-0.16
0.20
0.16

0.8
0.2
1.6

$y_{in} = -1\times 0.2 + 1\times 0.2 + 0.2 = 0.2$
$e = (-1-0.2) = -1.2$, $E = 1.44$
$w_1 = 0.2 + 0.2(-1.2)\cdot 1 =$ ~~0.36~~ 0.44
$w_2 = 0.04$

0.16
0.20
36

1.60
0.20
1.80

1.2
0.2
0.24

∇ Nebla

PAGE NO.:
DATE: / /

$x_1 = 1, x_2 = -1, t = -1.$

$y_{in} = -1 \times 0.2 + (-1) \times 0.2 + 0.2 = -0.2$

$e = -0.8 \mid E = 0.64$

$W_1 = 0.36, W_2 = \cancel{0.34}\ 0.36$

~~W~~ Aggregated Error $= 2 \times 0.56 + 0.64 + \cancel{0}\ 1.44 + 0.64$

$E =$ Aggregated Error $= 5.28$

~~ΔW~~ = W_1	W_2	t
-0.12	-0.12	-1
-0.36	0.04	1
-0.44	-0.04	-1
-0.36	0.36	1

$e = (1 - 0.2) = 0.8$

$y_{in} = -0.12 \times$

$nw_1 = -0.12 + 0.2 ($

$y_{in} = 1 \times \cancel{(-0.12)} + 1 \cancel{(-0.12)} + 0.2 = 0.2$

$E = (0.8)^2 = 0.64$

$f(y_{in}) = y = 0.64$

$W_1 = -0.12 \times (0.2) \cdot (-1) 1$

$= -0.12 \times -(0.2)$

$= 0.024$

$W_2 = -0.\cancel{34} \times (0.2)(-1)$

$= 0.008$

$E = (1$

$y_{in} = 1(0.36) + 0.2(1 -$

$E = E_1 + E_2 + E_3 = 0.92 + 0.0...$

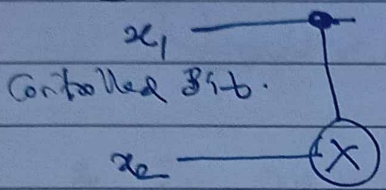
0.12
0.2
0.24
$.000$
0.024

0.36
0.2
0.072

0.04
0.2
0.8

PAGE NO.:
DATE: / /

Assignment No. 2 (Mc Culloch Pitts model)



if x is 1 then

Implementation of AND NOT

4 tables

x_1	x_2	t
1	1	0
0	1	1
1	0	0
0	0	0

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

$y = f(y_{in})$

$1 \times 1 + 1(-1) = 0$
$\rightarrow 0$

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

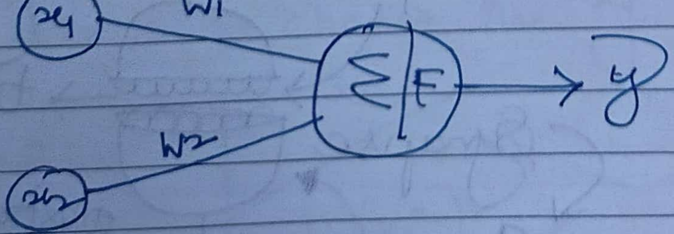
$0 \times 1 + 1(-1) = -1 < 0$
$\rightarrow 0$

Vector $[x_1, x_2]$, $[w_i, w_j]$

@

x @ w, for aggregator (x, w)

```
r = x @ w
if r >= 1
    return 1
else:
    return 0
```



PAGE NO.:
DATE: / /

Summary

$x_1\ x_2\ t.$

$W_1 = 0.2$, $W_2 = 0.2$, $b = 0.2$, $\alpha = 0.2$

$$y_{in} = \sum x_i w_i + b$$

$$\boxed{f(y_{in}) = y = e = t_i - y}$$

Current Error $\quad E = (t_i - y)^2$

$$n_1 W_1 = W_1 + \alpha\,(t_i - y)\cdot x_i$$

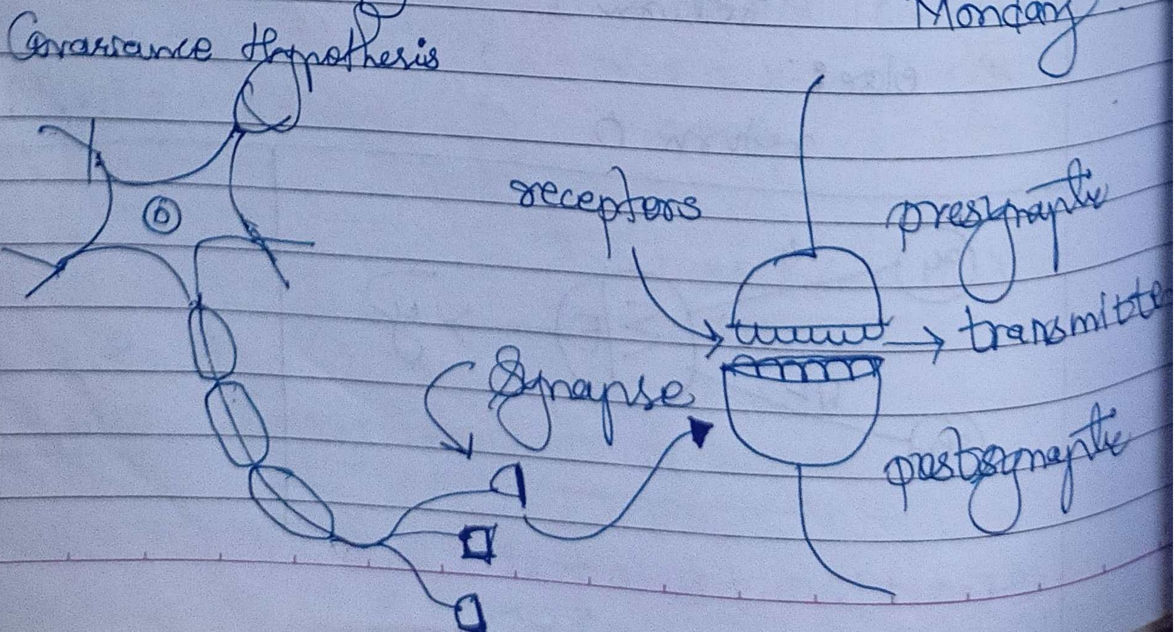
$$n_2 W_2 = W_2 + \alpha\,(t_i - y)\cdot x_i$$

E = Aggregated Error = $E_1 + E_2 + E_3$

$$RMSE = \sqrt{(t-y)^2}$$

* Hebbian Learning (Hebb Rule) 13/2/2023 Monday

- Covariance Hypothesis



1. If neurons to the size of synapse, synchronously changes then there is a positive increase a synapse.

2. If neurons of the side of synapse, works asynchronously then it changes negatively.

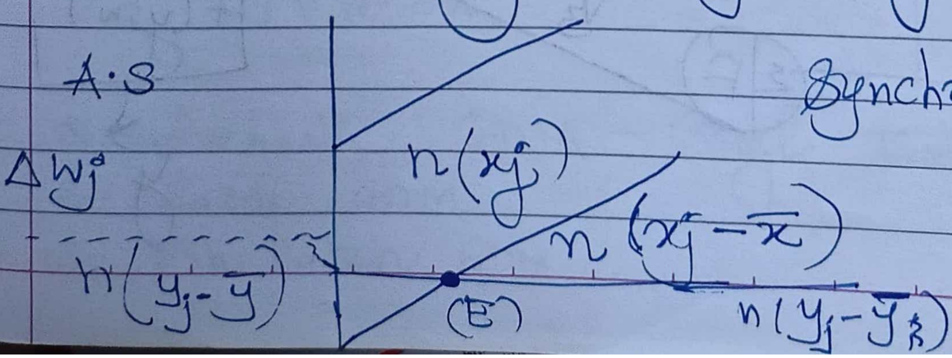
(Carries reaction – potential)

Presynapse x_j signal	Postsynapse y_j signal
$n \rightarrow$ learning rate	
$\Delta w_j = n x_j y_j$	

$$\Delta w_j = n (x_j - \bar{x})(y_j - \bar{y})$$

$$\Delta w_j = n F(x_j, y_j) \longrightarrow \text{Hebbien Hypothesis}$$

There is change in x_j & y_j.



Hebbian $\longrightarrow$ Synchronous Correlated
AntiHebbian $\longrightarrow$ ~~Hebbian~~ asynchronous. Corre
non-Hebbian $\longrightarrow$ anasynchronous. non correlated.

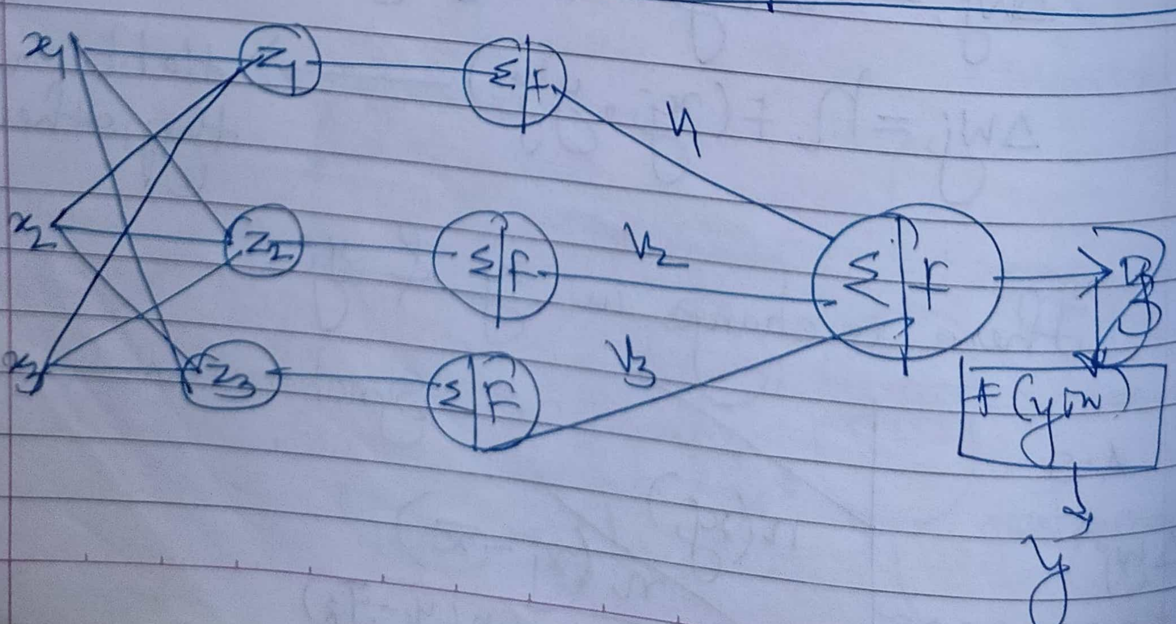
$$\Delta w_j = \eta (x_j - \bar{x})(y_j - \bar{y})$$

$\bar{x}, \bar{y}$ mean | Discrete.

x_1	x_2	x_3	$x_1 \otimes x_2$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$\boxed{t - y_{in}}$ Considered actual o/p
No fired o/p.

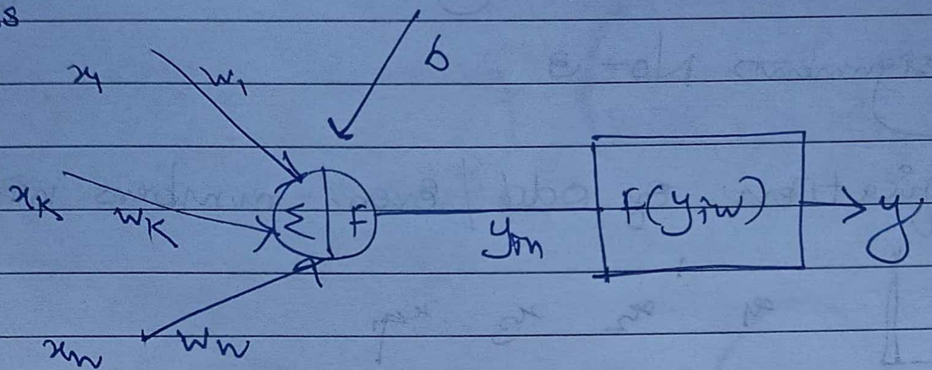
$x_1 \otimes x_2$ 1/0 if x_2



* Error Rule & Gradient Descent Method

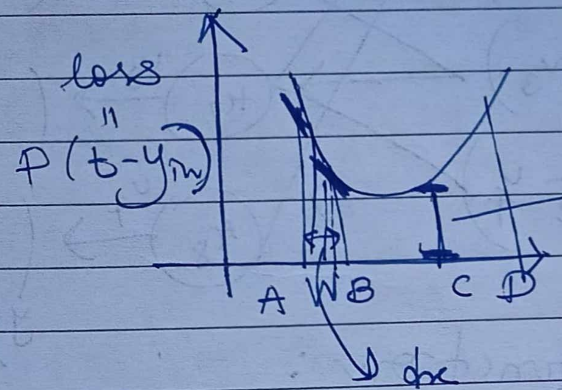
14/2/2023
Tuesday

Loss



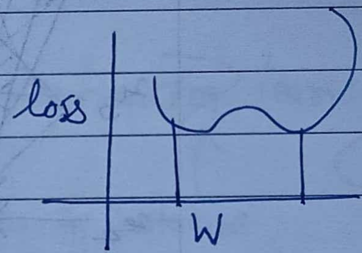
$$x_1 w_1 + x_2 w_2 + \cdots x_k w_k \cdots + x_n w_n + b$$

E → Loss = $t - y_i$



Single Minima

local Minima & global minima.

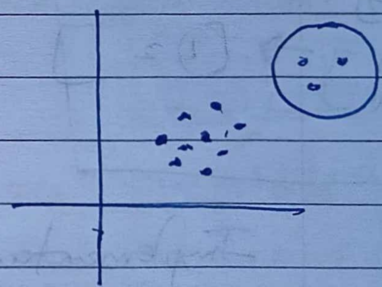


$$E = (t - y_{in})^2 \longrightarrow MSE$$

$$\frac{\partial E}{\partial W} = -2\,(t - y_{in})$$

$$\delta W = -\alpha\,\underbrace{(t - y_{in})}_{e}$$

$$W = W - \delta W$$



pdf

1. Stochastic GDM – Single Instance.
2. Batch GDM
3. Min Batch GDM.

Class A $\boxed{y < 0.5}$ $\boxed{y > 0.5}$ Class B

PAGE NO.:
DATE: / /

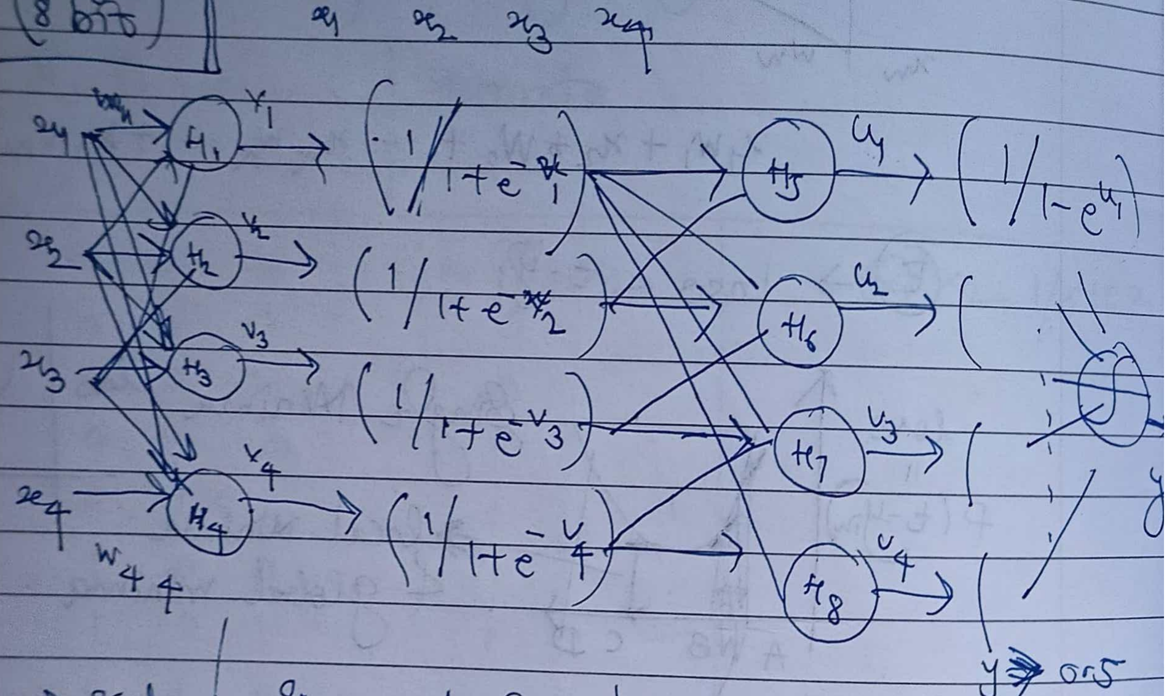
Software Laboratory - 2

14/2/2023
Tuesday

Assignment No - 3

* Identification of odd / even numbers with NN.

(8 bit)



Vector → scalar
[·0]
[1 1; 2 1] → (1) 2

Sigmoid function:

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

Implementation: X [[0, 0, 0, 0], ... [1, 1, 1, 1]]

W_{L_1} = [[$w_{11}, w_{21}, w_{31}, w_{41}$] .... [$w_{14}, w_{24}, w_{34}, w_{44}$]]

$x @ W_{L_1} = S_{11}, S_{12}, S_{13}, S_{14}$

y_1 = sigmoid (S_{11})

- - -

- - -

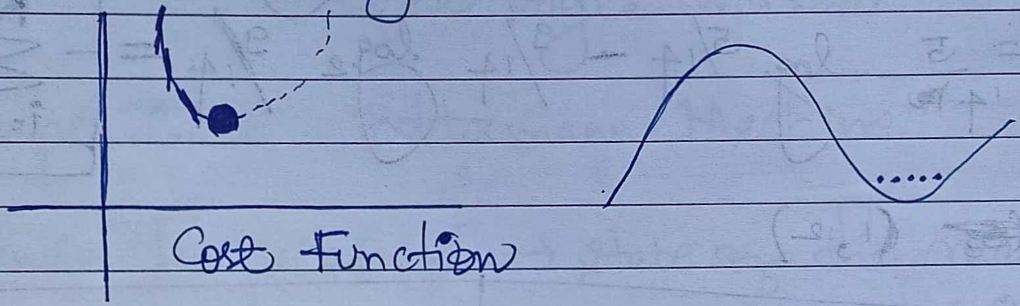
$\alpha = 0.001$, $b = 1$

* Gradient Descent Rule :

15/2/2023
Wednesday

* Competative Learning (neurons compete with each other)

* Boltzman Learning

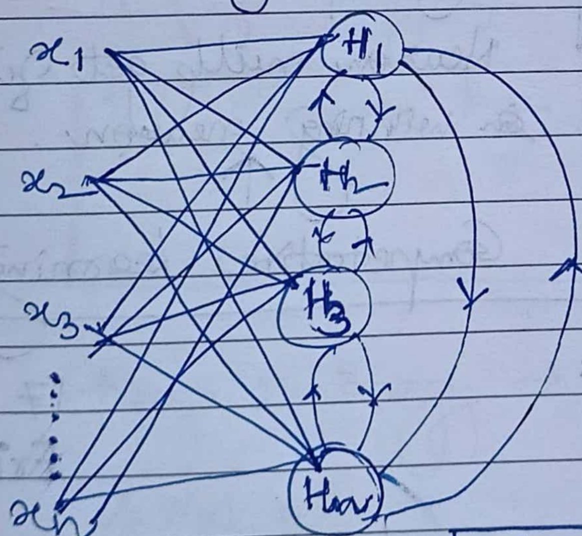


Cost Function

$\boxed{\text{faces problem of sadding}}$

enhancing learning

Sadding points



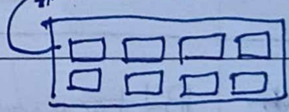
$\boxed{\text{Winner} \rightarrow 1}$

Layer 1

issues : Selection of instances.

{ Excitatory → ①
{ Inhabitatory → other

input supplied → Instances.



~~Problem~~ Process : It selects & process one active & one deactive

	x_1	x_2	x_3	
i_1	1	2	3	$\rightarrow V_1$
i_2	2	1	3	$\rightarrow V_2$
i_3	2	2	3	$\rightarrow V_3$
⋮	⋮	⋮	⋮	

Vector Distance :
$D(V_1, V_2)\ (V_1 V_3)\ (V_2 V_3)$

Orthonomal Leads to zero.

$A \rightarrow K$
$B(n-K) \rightarrow K$

PAGE NO.:
DATE: / /

Class A, Class B,
weight factor changes.

Great Learning :

k/n & $\frac{n-k}{n}$

$$E(D) = \frac{k}{n}\log_2 k/n - \frac{n-k}{n}\log_2 \frac{n-k}{n}$$

$$E(D) = \frac{5}{14}\log \frac{5}{14} - \frac{9}{14}\log_2 \frac{9}{14} = \boxed{-\sum_{i=1}^{K} p_i \log_2 p_i}$$

Potential (1, 2)

$x_1 \rightarrow 1 \rightarrow A$
$x_4 \rightarrow 1 \rightarrow B$

Information Gain
↑
highest potential
↑
Neuron will get fired
winning neuron.
↑
Competitive learning.

---

* Boltzmann Learning :

17/2/202
Friday

Energy Rule :

$$E = (t - y_i)^2$$

output / Actual value

target value

x_j & x_k → neurons.

$$\boxed{E = -\frac{1}{2}\sum_j \sum_k W_{kj} X_k X_j \quad \text{where } j \neq k}$$

w_{kj} is synaptic weight from neuron k to j.

Visible → interact with system & environment.

Hidden → free style workers.

Temperature Boltzmann Machine

Pseudo Temperature → state of the neuron (1, active / -1 inactive)

Probabilistic Form: +, − (x_j^+, x_j^-)

$P(x_j^+ \to x_j^-) = 0.6$, $P(x_j^- \to x_j^+) = 0.4$

$$\boxed{P(x_j^+ \to x_j^-) = \frac{1}{1 + e^{-\Delta E / T}}}$$ probabilistic answer.

Change in Energy → change in weight.

$\Delta w_{kj} =$

+, − → Correlation S_{kj}^+ | S_{kj}^- scalers.

α (learning rate)

$$\Delta w_{kj} = \alpha\left(S_{kj}^+ - S_{kj}^-\right)$$

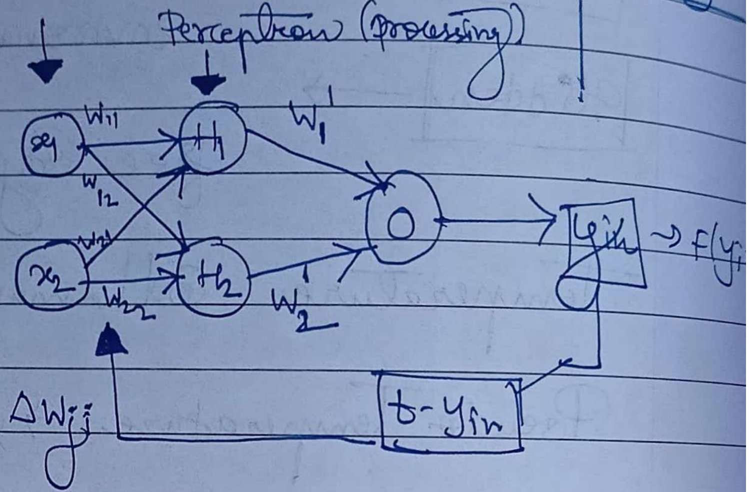
Correlation $= S_{kj}^+ = x_k^+ \cdot x_j^+$ | S_{kj}^- $x_k^- \cdot x_j^-$

* Memory Based Learning

20/2/2023
Monday.

x[1,-1]

BIBO System.



memory

$$\begin{array}{cc} e_1 & e_2 \\ \begin{bmatrix} W_{11} \\ W_{12} \\ W_{21} \\ W_{22} \end{bmatrix} & \begin{bmatrix} W_{11}+\Delta W_{11} \\ W_{12}+\Delta W_{12} \\ W_{21}+\Delta W_{21} \\ \Delta_{22}+\Delta W_{22} \end{bmatrix} \end{array}$$

$H_1 \longrightarrow W_{11}$ & W_{21}

$H_2 \longrightarrow W_{12}$ & W_{22}

Neighbourhood.

$H_1 = ((x_1 x_2), d_1)$

$H_2 = ((x_1 x_2), d_2)$

factor di – Euclidean ~~Difference~~ Distance

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 2 \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$d(x_1, x_2) =$

negligible change

$d(X_1, X_2) \to \Delta W_{ij}$

$$\{\Delta W_{ij}\}^{k}$$

$$\boxed{N_h = \frac{N_s}{(N_o + N_i)\alpha}}$$

N_h = No of hidden layers

N_s = No of samples.

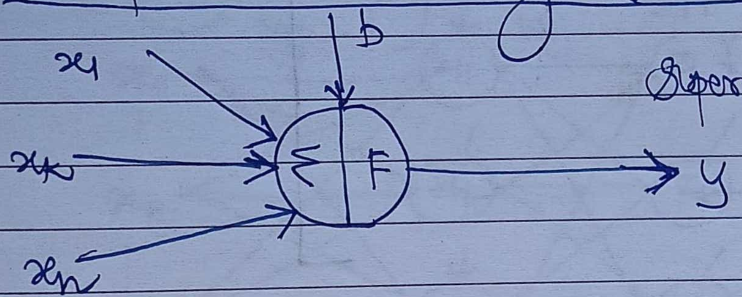
α = scaling factor.

---

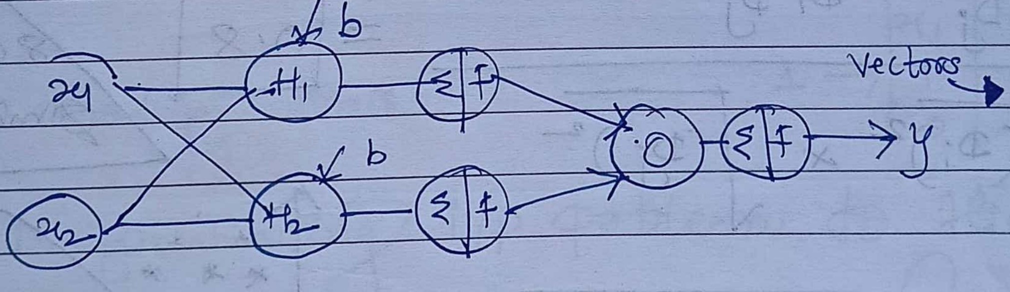
* **Perceptron Learning & Backpropagation** | 22/2/2023 Wednesday

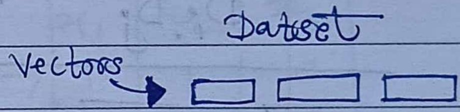
Supervized Activity

Case 1:



Case 2.



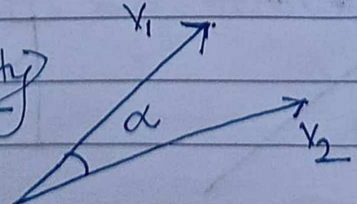


$x_0 w_0$ $x_0 = 1$, $W_0 = 0.1$ to 1

$$b \rightarrow \boxed{W_0 + \sum_{i=1}^{n} x_i w_i}$$

	F_1	F_2	F_3	
1.	[2	4	6]	v_1
2.	[4	8	7]	v_2

Cosine Similarity



v_1 v_2	$\cos\alpha$

$$D_1 = \{ W_1, W_2, W_3 \cdots\cdots W_k \}$$

$$D_2 = \{ W_1, W_2 \cdots\cdots W_j \}$$

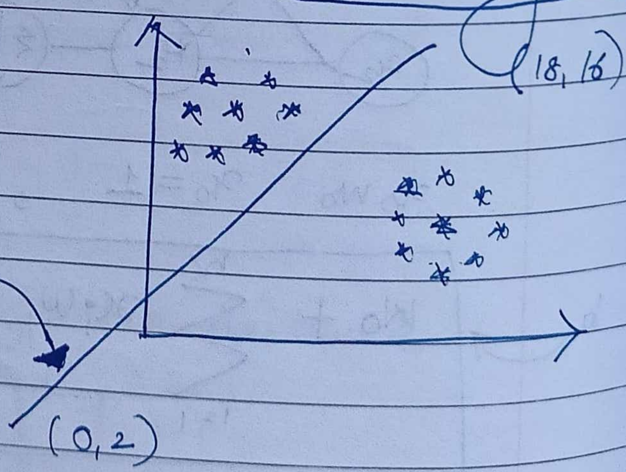
$$\vdots$$

$$\begin{array}{c|cccccc} & V_1 & V_2 & V_3 & \cdots\cdots & (V_{k+j+l+m}) & V_N \\ \hline D_1 & 50 & 25 & 0 & \cdots\cdots & & 0 \\ D_2 & 45 & 24 & 2 & & & 5 \\ D_3 & \vdots & \vdots & \vdots & & & \vdots \\ D_4 & & & & & & \end{array}, \quad N > k, j, l, m$$

$$\frac{D_i^T . D_j \quad D_i D_j}{\sqrt{(D_i)^2} \times \sqrt{(D_j)^2}} = \boxed{0.8}$$

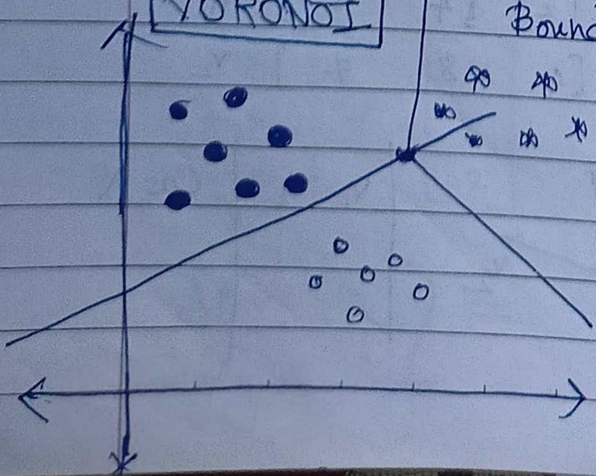
80 % Similarity

$$\sqrt{D_1^2} = \sqrt{(50)^2 + (25)^2}$$

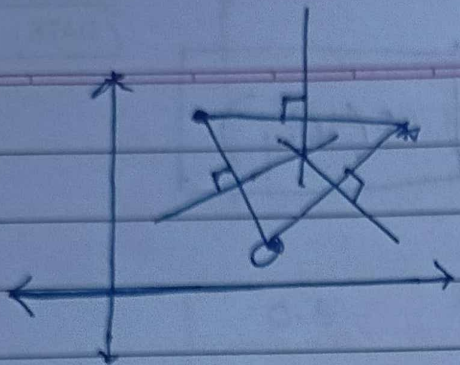


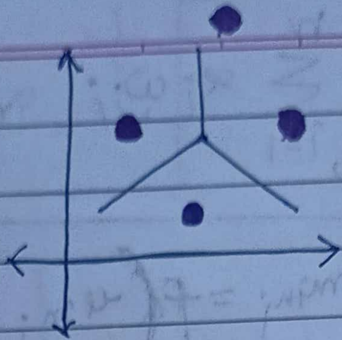
Decision Boundary

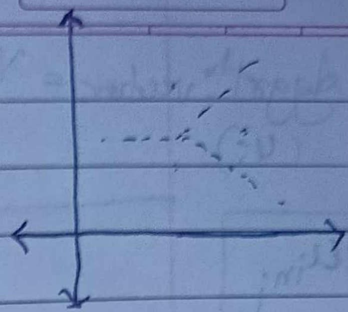
VORONOI



VORONOI DIAGRAM





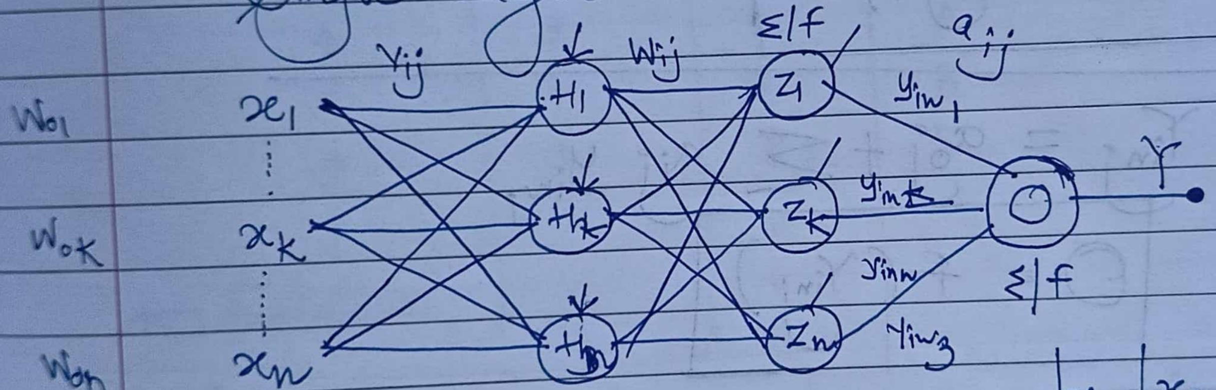


Categorization of instances.

27/2/2023
Monday.

* BackPropagation :

Single-Layer Network :



	x_1	x_2	x_3
1 →	1	1	0
	0	1	0

modify bias

1. forwarding i/p pattern to Hidden layer
2. Computing aggregation & applying activations to decides the closest need of the pattern.
3. Back Propagation error to hidden layers to adjust weight ~~on~~ vectors.
4. Σ

$$\left[\frac{2}{1+e^{-x}} - 1\right]$$ Bipolar Sigmoid $2\,sig(x) - 1$

PAGE NO.:

DATE: / /

$$\underset{(u_i)}{\text{Aggrtn vector}} = v_{oi} + \sum_{j=1,\, i=1}^{n} x_i w_{ij} = \boxed{f(u_{in_i})}$$

$\boxed{u_{in_i}}$

$$m_{in_j} = f(u_{in_i})$$

$$Z_{in_j} = W_{oi} + \sum_{i=1}^{n} m_{in_j} W_{ij}$$

$$\boxed{Y_{in_j} = f(Z_{in_j})}$$

$$Y_{in_j} = a_{oj} + \sum a_{ij}\, y_{in_i}$$

$$\boxed{O = f(Y_{in_j})}$$

$$\delta_k = (t_k - y_{in_k}) \cdot m_{in_j in}$$ ← second hidden layer

$$\Delta W_{ij}\ \alpha\ d_k \cdot f'$$

$$\delta_i = (t_i - u_{in_j}) \cdot x_i$$ ← First hidden layer

$$v_{ij} = \alpha\ d_i\ f'$$

$$\Delta V_{ij} = V_{ij\,old} + \Delta V_{ij}$$

method union. weight range :- 0.5 to 0.5

n : # inputs

p : Perceptrons in hidden layer

β : scaling factor.

28 feb absent.

PAGE NO.:

DATE: / /

$$\boxed{\beta = 0.7\,(p)^{\frac{1}{n}}}$$

$$\boxed{\frac{0.5}{\|W_{ij}\|} \quad \beta}$$

1/3/2023
Wednesday.

* Neural Network (Selective Backpropagation)

attr1	attr2	attr3	Class.
4	3	2	A
3	4	6	B
3	3	~~8~~ 2	A
⋮	⋮	⋮	
100 instances.			

Data → Dataset D → rows × Column

D over the field of $\mathbb{R}^{m \times n}$ $\quad R^{100 \times 4}$

Potential data

A B
4 ← 10 (5, 5)
5 ← 3 (~~3~~, ~~7~~)
3 ← 7 (3, 4)
20

$E(D) = -\sum_{i=1}^{k} P_i \log P_i$

probability

55 — A
45 — B

$$\boxed{E(D) = -\frac{55}{100}\log \frac{55}{100} - \frac{45}{100}\log\frac{45}{100}} = 0.924$$

$\text{Info}_{\text{attr1}}(D) = \frac{10}{2}\left(-\frac{5}{10}\log_2\frac{5}{10} - \frac{5}{10}\log_2\frac{5}{10}\right)$

(4)

Name of attr1

$$+\frac{7}{20}\left(-\frac{3}{7}\log\frac{3}{7}-\frac{4}{7}\log\frac{4}{7}\right)$$

$$+\frac{3}{20}\left(-\frac{1}{3}\log\frac{1}{3}-\frac{2}{3}\log\frac{2}{3}\right)$$

$$=0.724$$

$$\text{Gain}(attr_1) = info(D) - info_{attr_1}(D)$$

$$= 0.924 - 0.724$$

$$\text{Gain}(attr_1) = 0.200$$

Software Laboratory-2

Multiclass classification

Data Set: Iris. Setosa
Verginica
Versicolor

150

70% 30%

XOXO 1
XXOO 2
XXOX 3

Y_{ij}, $H_1 = [Y_{11}, Y_{21}, Y_{31}, Y_{41}]$

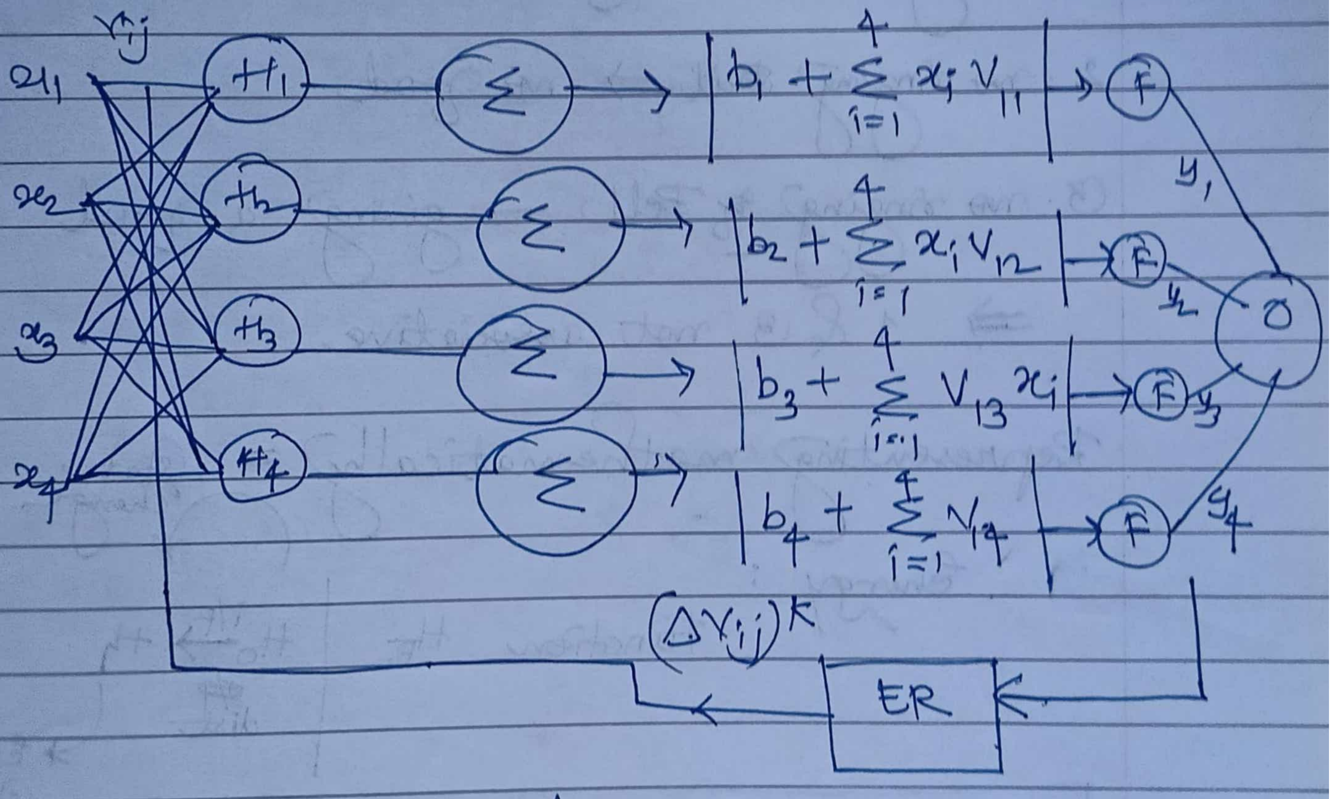
$H_1: \quad \Delta ij = [\Delta_{11}, \Delta_{21}, \Delta_{23}, \Delta_{24}]$

$H_2 = [V_{12}, V_{22}, V_{32}, V_{42}] \quad [\Delta_{12}, \Delta_{22}, \Delta_{32}, \Delta_{42}]$

$E_K \quad k = 1 \text{ to } 4$

$\delta_j = (t_j - y_{inj}), \quad \Delta V_{ij} = \alpha \, \delta_j \, |z_j|$

$V_{11}() = V_{11} \text{ old} + \Delta V_{11}, \quad V_{12}() = V_{12} \text{ old} + \Delta V_{12}$



$\Delta V_{11}, \Delta V_{12}, \Delta V_{13}, \Delta V_{14} \quad V_{22}$	0.5 Set
$V_{ij}(new) = V_{ij}(old) + \Delta V_{ij}$	0 Vin − 0.5 Vin

$\Delta V_{oj} = \alpha \delta_j$

$V_{oj}(new) = V_{oj}(old) + \Delta V_{oj}.$

8/3/2023

* Hopfield model

- Associative Learning
  - Associative memory

Dog + Bell Experiment

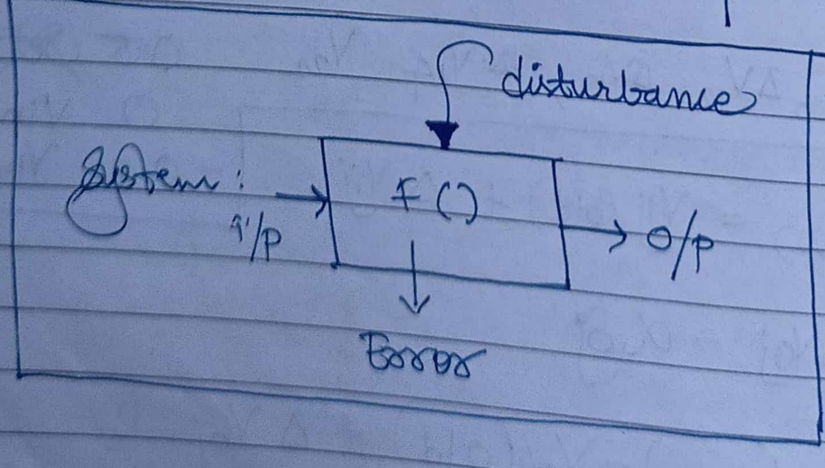
1. Ring Bell → give a food
2. no ringing Bell → no food.
3. no ringing of Bell → giving a food

⟹ 1 & 3 not associative.

Representing mathematically = State change

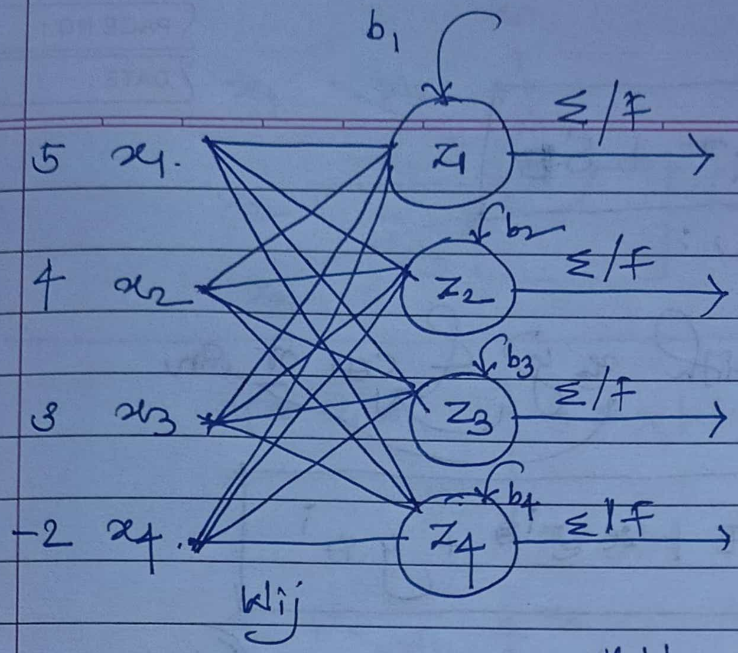
Energy :

Function H_E | $H_0 \xrightarrow[\text{dist}]{i/p} H_1$ → Excited State

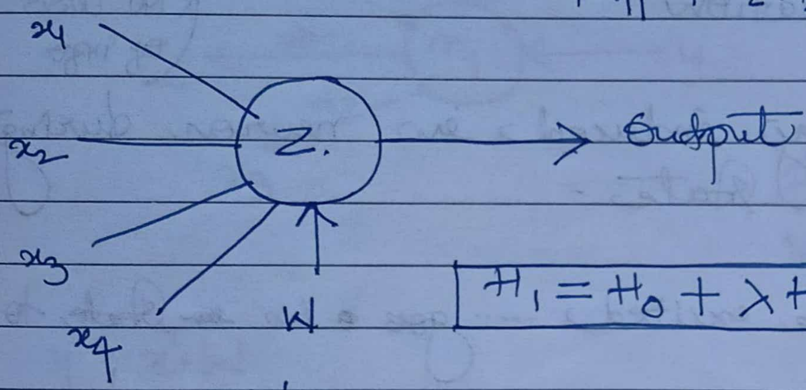


activation ~~vector~~ scaler

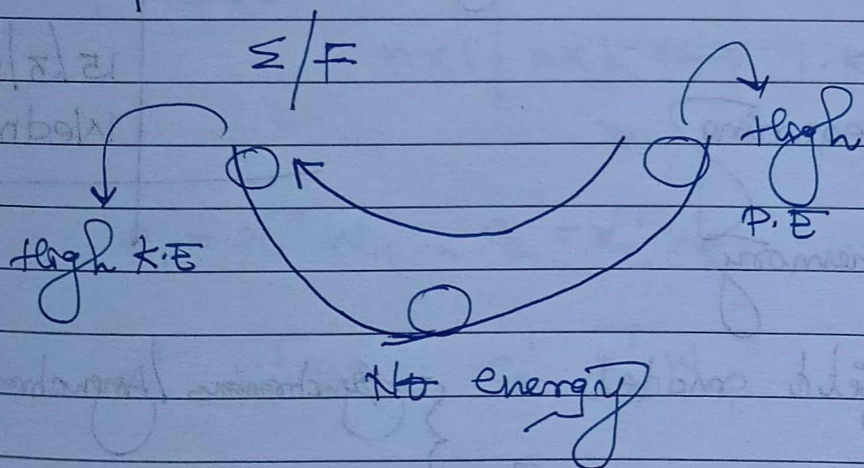
$H_1 = H_0 + \lambda H$ → Universal State



$$x_1 W_{11} + x_2 W_{21} + x_3 W_{31} + x_4 W_{41} + b_1$$



$$\boxed{H_1 = H_0 + \lambda H}$$



$b_{oij} + \Sigma \boxed{x_i W_{ij}}$

$\cancel{\boxed{x^T}} \xrightarrow{F} y$

$\boxed{W \times T} \to y \to x$

Reverse operation on transponse we get x

$E = \frac{-1}{2} W x^T$

$1/2 \to$ scaling factor

$E = -WX^T$

Every neuron subsystem has energy state

CO $e^{-i\theta} = \cos\theta + i\sin\theta$

PAGE NO.:
DATE: / /

$$\boxed{E = -WX^T + \delta E}$$

Vector representation:

association with x, y & cos or sin

$$\boxed{E = -Wx^T + \delta E + x e^{-i\theta} + y\theta^i}$$

Perturbation

(No loss of info x_1, x_2 5, 4, -2)

* Little Energy induced in neuron during excited state.

• → once excited ..... goes in state to run

15/3/2023
Wednesday

Associative Learning

Associative memory

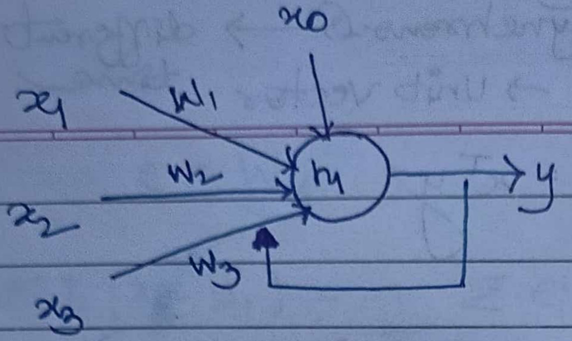
– Hopfield model. { Synchronous / Asynchronous

Input = $[x_1, x_2, x_3 \cdots x_n]$

Output = $[y_1, y_2, y_3 \cdots y_n]$

Weight matrix : $\begin{bmatrix} W_{11} & W_{21} & W_{31} & \cdots & W_{n+1} \\ W_{12} & W_{22} & W_{32} & \cdots & \\ & & & & \\ W_{1n} & \cdots & & \cdots & W_{nn} \end{bmatrix}$

PAGE NO.:
DATE: / /

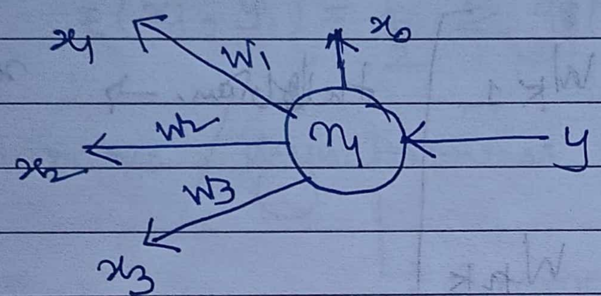


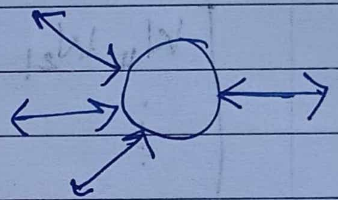
$$\boxed{y = \sum x_i W_{ij}}$$

$y_1 = x_1 W_{11}$ | $y = xW$

$x = yW$

Energy difference beth input & output layer





Biderectional Associative Memory (BAM).

$y = x^T \cdot W$

$n \times 1 \mid n \times K \rightarrow 1 \cdot K$

$e_1 = x_1^T W_{11}$, $e_2 = x_2^T W_{21}$ ....

e_1' $y_1^T W_{11}$, e_2' $y_2^T W_{21}$

$E = 0 - \sum x_i^T W$

$E = \boxed{\frac{-1}{2}} \sum x_i^T W$

Scaling factor

$Z = \text{sign}(x^T W)$ $-/+$

$x^T W_{ij}$

Stopping Condition: 2/3 epochs leads to eq same equilibrium state.

Synchronous → at a time

x norm is 1   Asynchronous → different time

$\|x\|^2$ unit time → unit vector

$x^T W y \;\Big|\; W = x^T y$

$$z = \text{sign}(x^T W) = \text{sign}(x^T x^T y)$$

$$= \text{sign}(\|x\|^2 y) = y$$

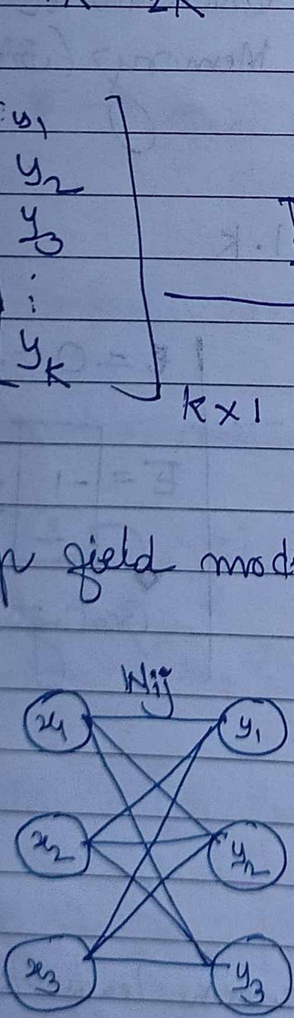
$[x_1, x_2, x_3 \cdots x_n]_{1\times n}$

$$\begin{bmatrix} W_{11} & W_{21} & W_{31} & \cdots & W_{k1} \\ \vdots & & & & \\ W_{1k} & W_{2k} & \cdots & & W_{nk} \end{bmatrix}_{n\times k}$$ Hebbian → xWy

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_k \end{bmatrix}_{k\times 1}$$ → probability is high if works synchronously. $\boxed{n\times 1}$

17/3/2023
Friday

- Hop field models.



$$\boxed{E(x_i\, y_i) = -\frac{1}{2} x W y^T}$$

$$\underbrace{[x_1, x_2, \cdots x_n]_{1\times n} \begin{bmatrix} W_{11} & \cdots & W_{k1} \\ \vdots & & \vdots \\ W_{1k} & \cdots & W_{nk} \end{bmatrix}_{n\times k} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}_{k\times 1}}_{e_i}$$

$$e = W y^T$$

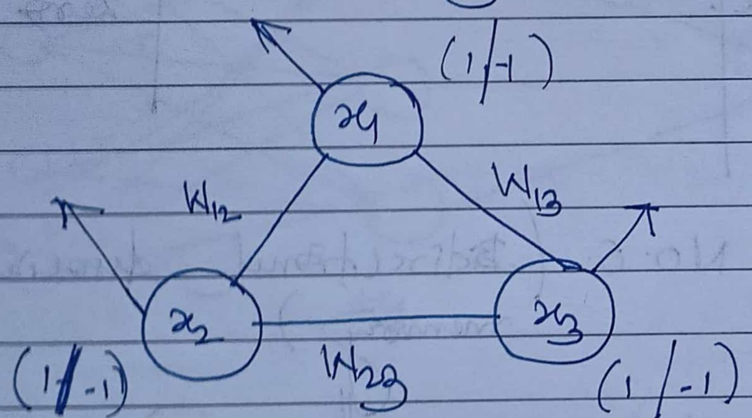
$$\Sigma E(x_i, y_i) = \frac{-1}{2} \Sigma e_i x_i$$

$$E(x_i, y_i) = \frac{-1}{2} \Sigma e_i x_i$$

$$E(x_i, y_i) = \frac{-1}{2} \Sigma g(x_i)$$

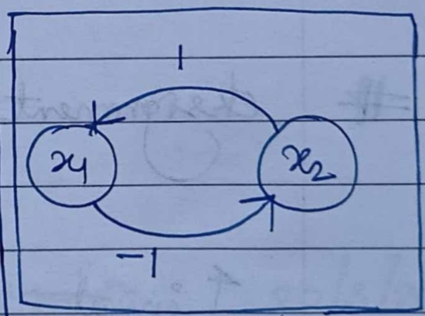
Synchronous / asynchronous system

↳ Symmetric Asymmetric

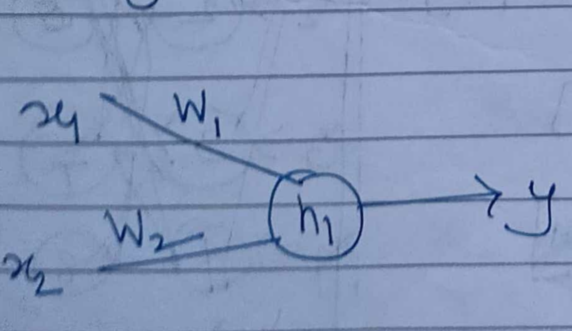


System - H_1

$E(H_1)$



Zero / cancel out



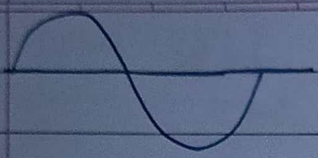
$x_1' = y W_1$

$x_2' = y W_2$

neuron will in the state till new inputs are supplied for some time.

Asynchronous → Good / stable system
Synchronous → mechanical system

PAGE NO.:
DATE: / /



$$E(x, y) = \frac{-1}{2} \sum x_i \, g(x_i)$$

$$W y^T = W(1) = W$$
$$W y^T = W(-1) = -W$$

$$E(x', y) =$$

Asynchronous	Synchronous	
Active neuron only → system will be governed.	Whole system → all neurons	If system is zero then some neurons are active but combined effect is zero

Assignment No. 6 (Bidirectional associative memory)

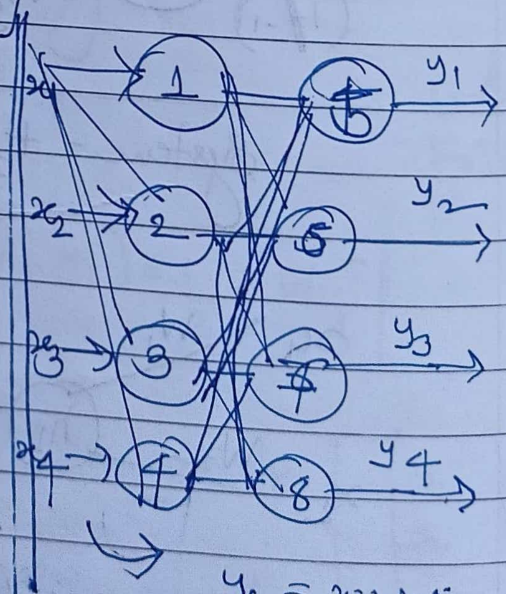
4 input vectors

x_1 [1, 1, 1, -1, -1, -1, ~~1~~]

x_2 [-1, 1, 1, -1, 1, 1]

x_3 [1, 1, 1, 1, 1, 1, ~~1~~]

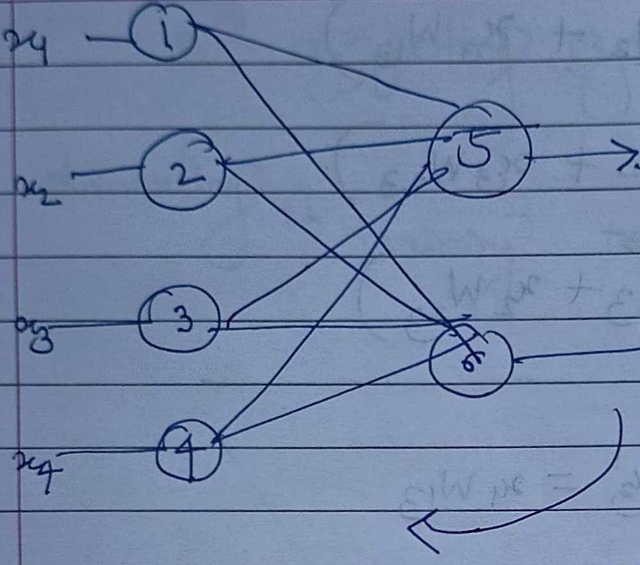
x_4 [1, 1, -1, 1, -1, 1]



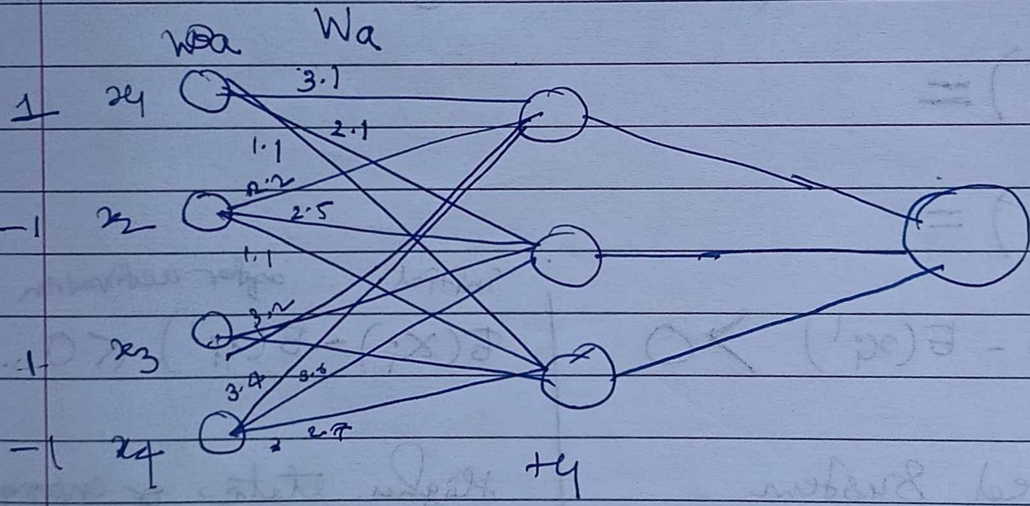
$y_i = x_i W_{ij}$
$x_i = W_{ij} \, y_{ij}^T$

T[1 1] T[-1 0] T[-1, -1]

T[0 -1] T[1, -1] T[1 0]

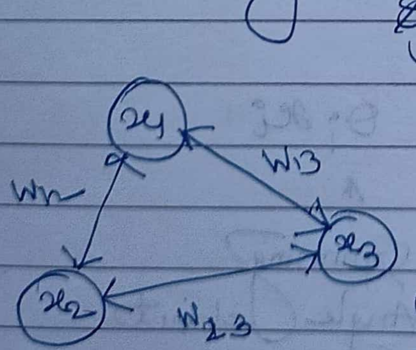


$$x_i = W_{ij}\, y_i^T$$



Hopfield Model ~~symmetric~~ / ~~Assymmetric~~ 20/8/2022
Working. Monday

Synchronous / Asynchronous



Energy Computation

$e_1 \quad x_2 = x_1 W_{12} + x_3 W_{23}$

$e_2 \quad x_1 = x_2 W_{12} + x_3 W_{23}$

$e_3 \quad x_3 = x_1 W_{13} + x_2 W_{12}$

$$\Sigma e_i = e_1 + e_2 + e_3$$

$$E = -\frac{1}{2}(e_i x_i)$$

PAGE NO.:
DATE: / /

$$E(x_1) = -\frac{1}{2}(x_2 W_{12} + x_3 W_{13})$$

$$E(x_2) = -\frac{1}{2}(x_1 W_{12} + x_3 W_{23})$$

$$E(x_3) = -\frac{1}{2}(x_1 W_{13} + x_2 W_{23})$$

$x_2' = x_1 W_{12}$ & $x_3' = x_1 W_{13}$

$x_1' = x_2 W_{12}$ & $x_1 = x_3 W_{13}$

$E(X_1') =$

$E(X_2') =$

	initial / after activation
$E(x_i) - E(x_i') > 0$	$E(X_i) - E(x_i') < 0$
Deactivated System	Higher state of energy
Agg/acts State change energy is lowered	Activated System

$$E(x_i, y_i) = -1/2 \sum g(x_i) + \theta_i x_i$$

Spinning Angle about x_i

$$E(x, y) = -1/2 \sum_{i=1}^{N} g(x_i) + \theta_i x_i + H^{*}$$

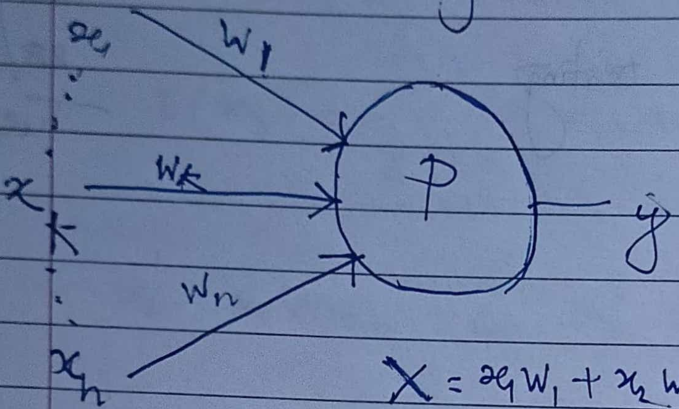
Energy applied to system external

(1) { 4 − 1 = 3   (2) 4 − 1 = 3
3 − 1 = 2

PAGE NO.:
DATE: / /

Assignment No. 7

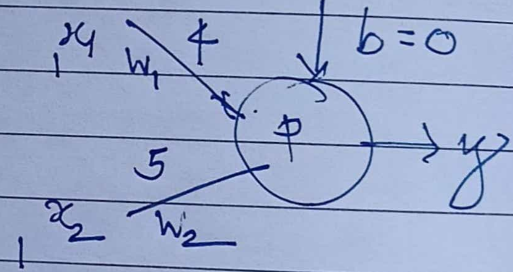
* Perceptron learning law & graphical representation



$X = x_1 W_1 + x_2 W_2 + x_3 W_3 + \dots x_3 W_k + \dots + x_n W_n$

$F(X) = y$

I/P	x_1	x_2	t (OR)
	0	0	0
	0	1	1
	1	0	1
	1	1	1



$x = (x_1 W_1 + x_2 W_2)$

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

Setting threshold Th

y < Th | 0 } assumption
y > Th | 1

y < 0.5 | 0
y > 0.5 | 1

$X = 1 \times 4 + 1 \times 5 + 0 = 9$

$y = \frac{1}{1 + e^{-9}} = 1.000$

if y < Th
W = W + x
y > Th
W = W − x

ART 2

PAGE NO.:
DATE: / /

* $\cos\alpha = \dfrac{x \cdot W^T}{\|x\|\,\|W^T\|}$ * $\cos\alpha' = \dfrac{x \cdot W'^T}{\|x\|\,\|W'\|}$

* Adaptive Resonance Theory (ART_1) 12/4/2023

Supervised Learning Algorithms:

- pattern matching

Class: Setosa, verginica, versicolor,

Spatial Distance Measurement:

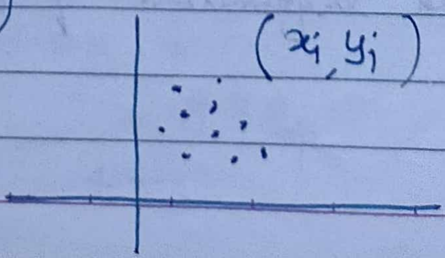
$v_1 = [\dots]$
$v_2 = [\dots]$

1. Hamming distance
2. Eucledian distance
3. Minkowski distance
4. Mahalanobis distance

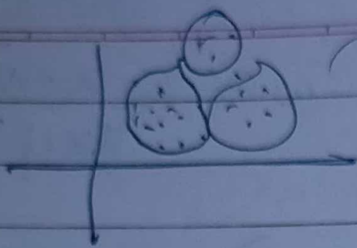
Eucledian distance $v_1 = x_1, x_2, x_3, x_4$, $v_2 = (y_1, y_2, y_3, y_4)$

$$E.\,dist(v_1, v_2) = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + (x_3-y_3)^2}$$

Threshold value



(x_i, y_i) Scatter plot



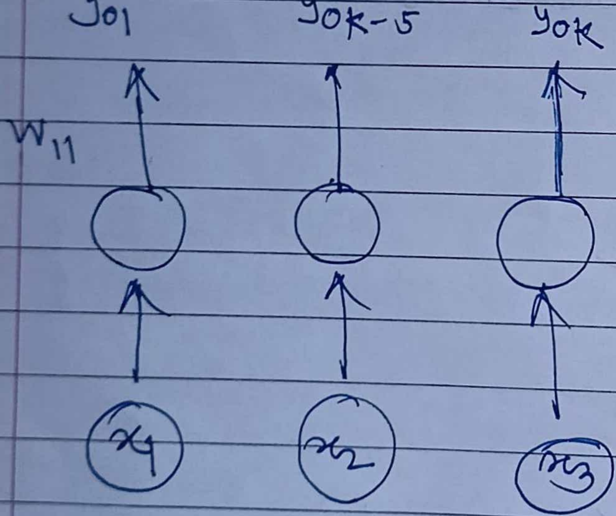
controlled grouping

Inputs x_1 x_k x_n

W_{11} ... k hidden Perceptrons

α W_{1k} W_{kk} W_{nk} × input lines

weight matrix



k hidden perceptrons

agg computing | activation applying

W_{1k} W_{kk} W_{nk}

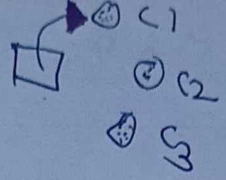
$$\boxed{y_{0k} = \sum_{i=1}^{n} W_{ik} x_i}$$

y_0

Collected information | pattern has learned

$w_{ij} = b_{ij}$

$w_{ji} = t_{ij}$



* Adaptive Resonance Theory (ART-I) (unlabelled data)

21/4/2023 Friday

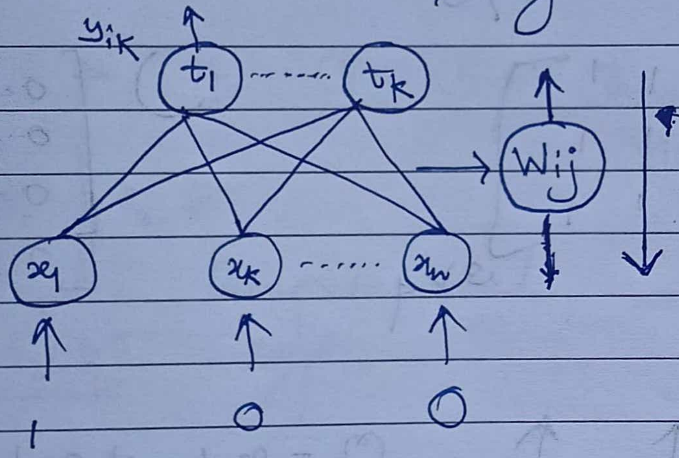
Input Supplied.

Numeric Input → Binary | Continuous

ART-I | ART-II

0/1 | 0, 1, 2, 3, 4 .... n

input vector $\in \{0, 1\}^*$ ~~String~~



input size (n), number of clusters (m), learning parameter (α), vigilance parameter (ρ)

initialization of weight matrix $W_{ij}(b_{ij}) = \left\{\frac{1}{1+n}\right\}$

e.g. if $n = 4$, $b_{ij} = \frac{1}{5} = 0.2$, $m = 3$

$$b_{ij} = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & & \\ 0.2 & & \\ 0.2 & & 0.2 \end{bmatrix}$$

~~3×4~~

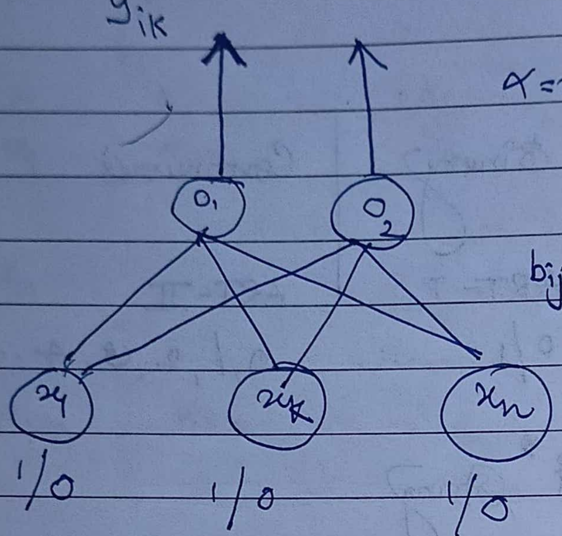
4×3

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

3×4

PAGE NO.:
DATE: / /

Assignment No. 8 ART-1 Neural Network 21/4/2023



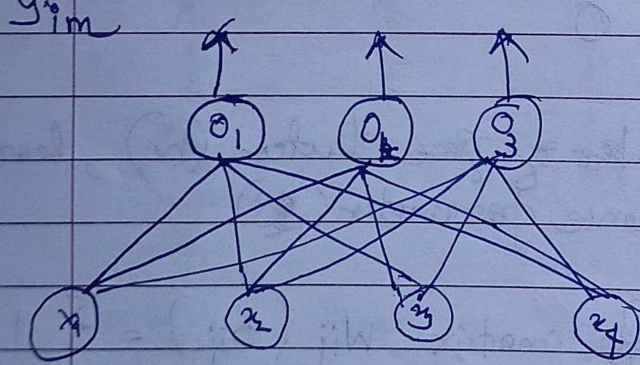
$\alpha = 2$, $\rho = 0.4$, $m = 3$, $b_{ij} = 0.2$

Activation Functions

$b_{ij} \,/\, T_{ji}$ $F_1 \uparrow$ $F_2 \downarrow$

$$t_{ji} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}_{3 \times 4}$$

$$\begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & & \\ 0.2 & & \\ 0.2 & & \end{bmatrix}_{4 \times 3}$$



$O_1 = x_1 b_{11} + x_2 b_{21} + x_3 b_{31} + x_4 b_{41}$

$O_2 = x_1 b_{12}$

$O_3 = x_1 b_{13}$

$O_4 = x_1 b_{14}$

0.2

inputs
[1 0 0 1]
[0 1 0 1]
[1 0 0 1]
[1 0 0 1]

1 > 0.4 ✓ 0.2 > 0.4 ✗ (11 → 19)

$J_{1} = 11 \quad 12 \quad 13 \quad (b_{ij})$

$11 \quad 12 \quad 13 \quad 14 \quad (t_{ji})$

$$\frac{\alpha x_i}{(\alpha-1)+|x_i|}$$

* ART-2 Network (Continuous values)

IRIR dataset

4 4 5 5 Setosa / versicolor / Virginica

f_1 | f_2

$f_1(a)$ $f_2(b)$ Computational layer

input layer

1.] 1 2 1 2

Step 0. 2.] 6↑ 9↑ 8↑ 9↑ normalize

Steps: ① Initializing parameters

$\{a, b, \theta, e, d, \alpha, \rho\}$

~~# n = no of input~~
~~# m = no of clusters~~

a, b are fixed weights, used in F_1 layer

Sample value of $a, b = 10$ | $a, b_1 = 0$

c: fixed weight, used in testing model.

d: Activation value of ~~main~~ winning F_2 unit. has to satisfy the inequality

$$\boxed{\frac{c \cdot d}{1-d} \leq 1}$$

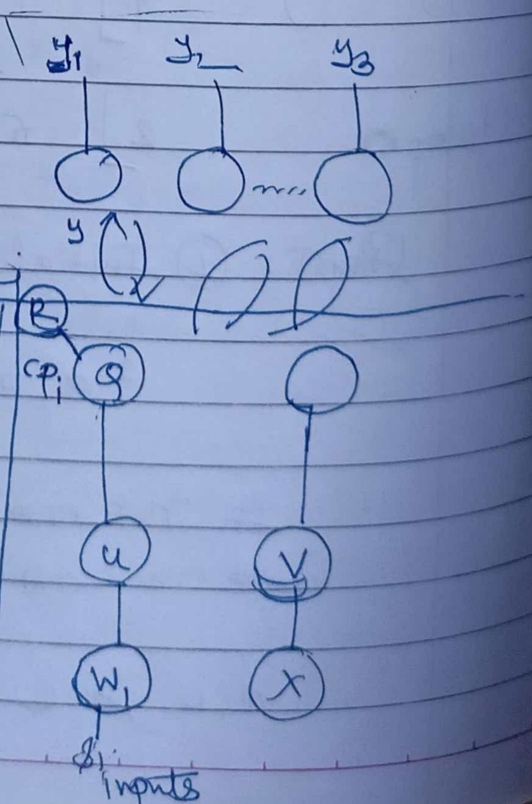
e: Small parameter used to prevent. divide by zero

θ: noise suppression parameter:

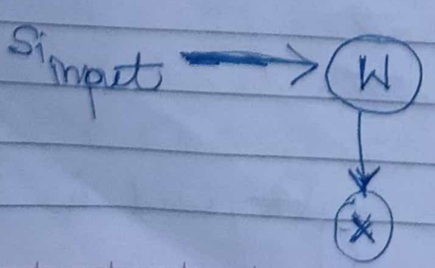
$$\theta = \frac{1}{\sqrt{n}}$$ (to normalization)

α: learning rate

ρ: vigilance parameter.



$u_i = 0$, $u_i = \dfrac{v_i}{e + \|v\|}$



PAGE NO.:
DATE: / /

$W_i = \underline{S_i + a u_i}$, $P_i = u_i + d t_{Ji}$

$x_i = \dfrac{W_i}{e + \|W\|}$, $q_i = \dfrac{P_i}{e + \|p\|}$, $V_i = f(x) + b f(q_i)$

(The activation function used is
$f(x) = \begin{cases} x & \text{if } x > \theta \\ 0 & \text{if } x < \theta \end{cases}$)

$$\left\{ \begin{array}{l} u_i = 0,\ W_i = S_i,\ P_i = 0 \\ x_i = \dfrac{S_i}{e + \|S\|},\ q_i = 0 \\ V_i = f(x_i) \end{array} \right\}$$

updating F_1 layer,

$u_i = \dfrac{V_i}{e_i + \|V\|}$, $P_i = u_i$

$x_i = \dfrac{W_i}{e + \|W\|}$, $q_i = \dfrac{P_i}{e + \|p\|}$

$V_i = f(x_i) + b f(q_i)$

Computation in F_2 layer

$y_j = \sum b_{ij} P_i$

PAGE NO.:
DATE: / /

if reset is ~~true~~ TRUE

$y_i \rightarrow \rho$

$u_i = \frac{v_i}{e + \|V\|}$, $p_i = u_i + d t_{Ji}$, $r_i = \frac{u_i + c p_i}{e + \|e\| + c\|p\|}$

if $\|r\| \geq \rho - e$ then

$w_i = s_i + a u_i$

$x_i = \frac{w_i}{e + \|w\|}$

$q_i = \frac{p_i}{e + \|p\|}$

$v_i = f(x_i) + b f(q_i)$

if reset is FALSE

updating weights of winning neuron / unit J

$t_{Ji} = \alpha d u_i + \{1 + \alpha d (d-1)\} t_{Ji}$

$b_{iJ} = \alpha d u_i + \{1 + \alpha d (d-1)\} b_{iJ}$

updating F_1 activation

$u_i = \frac{v_i}{e + \|v\|}$, $w_i = s_i + a u_i$

Self Organizing Maps.

PAGE NO.:
DATE: / /

$$p_i = u_i + d t_{J_i}, \quad x_i = \frac{W_i}{e + \|W\|}, \quad q_i = \frac{p_i}{e + \|p\|}$$

$$v_i = f(x_i) + b f(q_i)$$

Stopping Condition

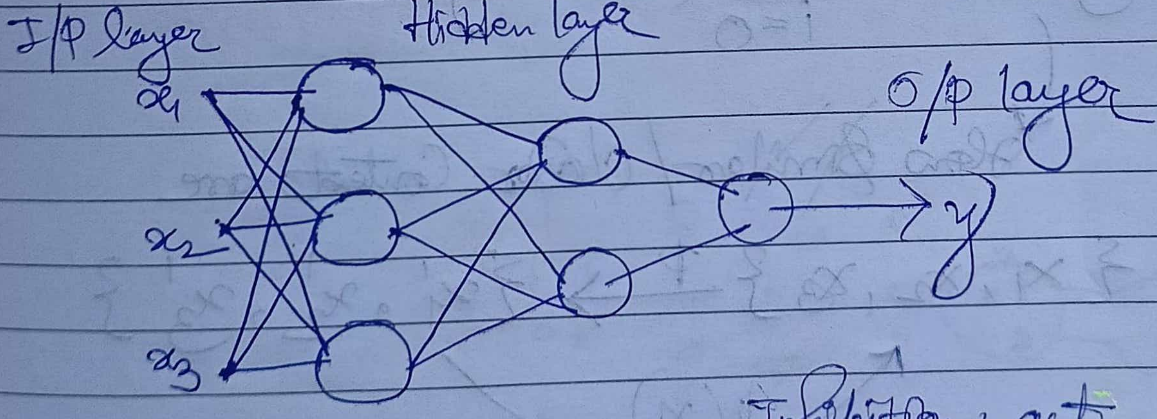
10/5/2023

* Self Organizing Maps (SOM)

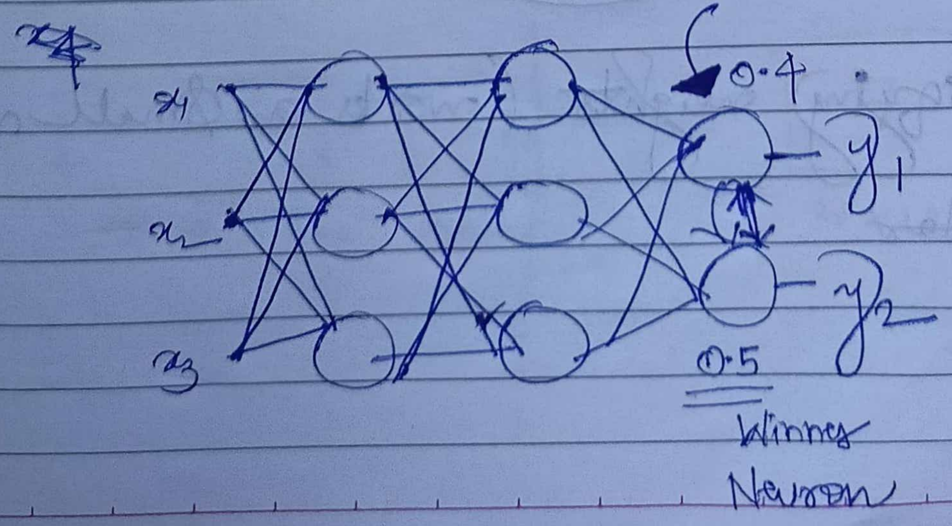
inhibition:

{audio, play, act} $\xrightarrow{\text{Stored}}$ Sequential manner in cartoon. Content.

Content, Association



Inhibition act



PAGE NO.:
DATE: / /

Lateral Inhibition

input : vector } dot → act → result
weights : vector }

input vector $X = [x_1, x_2, x_3 \cdots x_n]$

$$W = \begin{bmatrix} w_{11} \cdot w_{21} \cdots w_{n1} \\ \\ w_{nm} \end{bmatrix}$$

$d_j(X)$ → Squared Euclidean distance.

X, W

$$d_j(x) = \sum_{i=0}^{p} (x_i - w_{ij})^2$$

→ How similar / closer context are

$$\{x_1, x_2, x_3\} \xrightarrow{P} \{x_1', x_2', x_3'\}$$

$d_j(x)$

modifying weights (works on small weights)

~~Similar~~

Convolution layer

ANN → matrix multiplication.

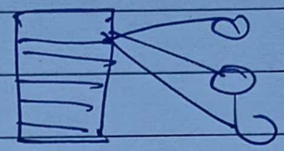
CNN → CN layer | pooling | FC layer

why CNN →

image data

Pixel → 2D grid.

2D → 1D.



Overfitting

CNN intuition:

edges.