Neural Network

* What is Neural Network? Explain Neural Network Application and Limitation.

=> Neural Network:

A Neural Network is a computational model which is inspired by the human brain.

Neural Network contains Neurons, Layer, Weights and its Biases value with its Activation Function.

Neural Network are used for tasks such as Pattern ~~Rco~~ Recognition, classification and decision-making.

Neurons : Neural Network are contain interconnected nodes called as Neurons.

Layers : Neural Network are created into the layer. There are input layer, hidden layer and output layer.

Brain Spot

Weights : Weights are represent the strength of the connection between every neurons.

=> Application of Neural Network :

This are the basic application of Neural Network.

1 Image Recognition :

Neural Network are used for image classification, Object detection and facial recognition etc.

2 Natural Language Processing :

~~Ata~~ Neural Network and such more advanced model are used for the language translation and chatbots.

3 Speech Recognition :

Neural Network used to enabling applications like voice assistants and speech recognition systems.

4 Autonomous Vehicles :

Neural Network are used for

Brain Spot

perform crucial task such as object detection, path planning and decision-making.

5 Healthcare:

Neural Network is used to image recongnition such as MRI devices or CT scans etc.

6 Stock Market Prediction:

Neural Network are applied in Financial sector for perform the task like stock price predication, fraud detection and credit scoring.

7 Sentiment Analysis:

Neural Network is used to identifying public option and trends on Social Media.

8 Gaming:

Neural Network Used in gaming for create character animation.

Brain Spot

9 Cybersecurity:

Neural Network used to detect malware and anomaly in the Network Traffic.

10 Handwriting Recongnition:

A Feed-Forward with 3-Hiddan layer Neural Network are used in Handwriting Recongnition.

=> Limitation of NN:

This are the mainly limitation of Neural Network.

1 Data Dependency:

Neural Network require a large amount of labeled traning data to generalize well.

2 Black Box Approach:

Neural Network can be train to transform Input Pattern to Output Pattern but it is not provide the physics behind the

Brain Spot

transformation.

3 Limited Transfer Learning:

Neural Network might struggle to transfer the information from one domain to another domain when domain is too dissimilar.

4 Traning Time:

Neural Network required large amount of traning time when we have to add complex and large datasets.

5 Low Learning Rate:

For the complex Problem, we have to use large and complex Neural Network architecture.

6 Forgetfullness:

Sometime, Neural Network is forget the old training examples.

Bram Spot

* What is Linear Separability in Neural Network and which are Logic Gates are Linear separable?

=> Linear Separability has ability to separate data points in binary classification problem using a Linear Decision Boundary.

The Perceptron can find weights For classification type of problem that are Linear Separable.

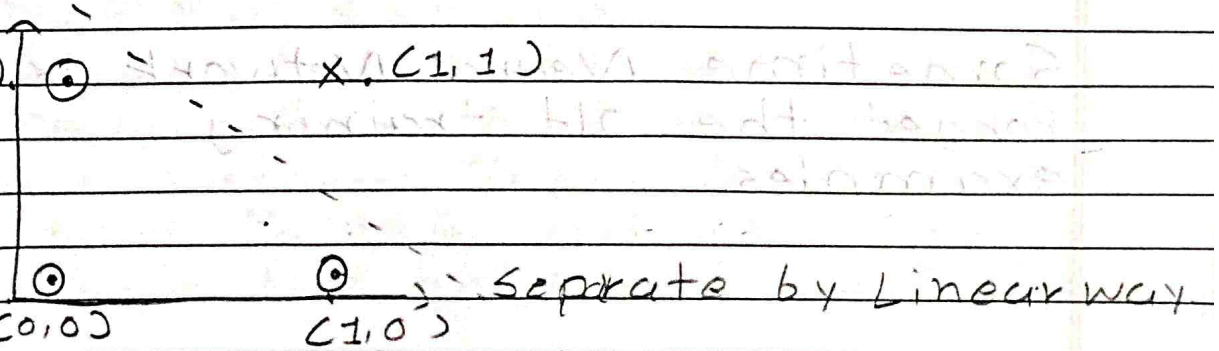
=> Logic Gates : AND, OR, XOR.

Truth Table :

Defined
0 as ⊙
1 as X

X	Y	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

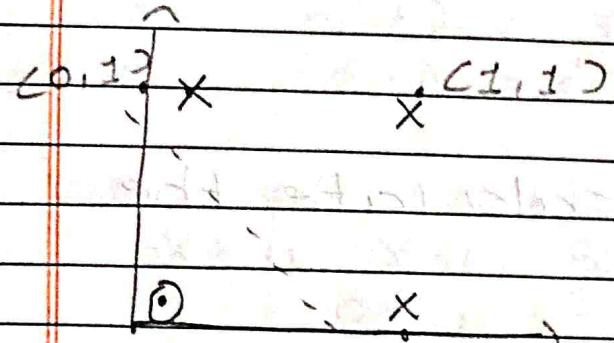
-> For AND Gate :



Brain Spot

So, AND Gate is Linear Separable.

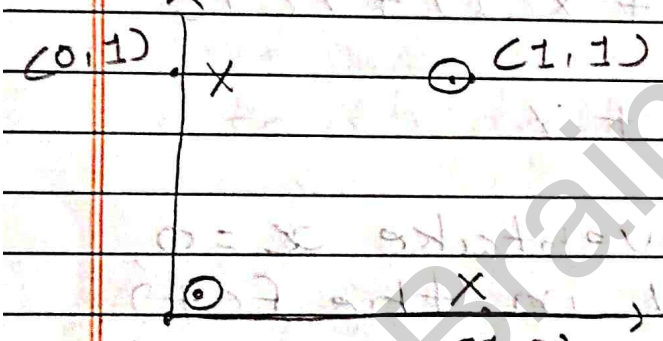
→ For OR Gate:



Separate by Linear Way.

So, It is Linear separable.

→ For XOR Gate:



We can not Separate by Linear Way.

So, It is not Linear Separable.

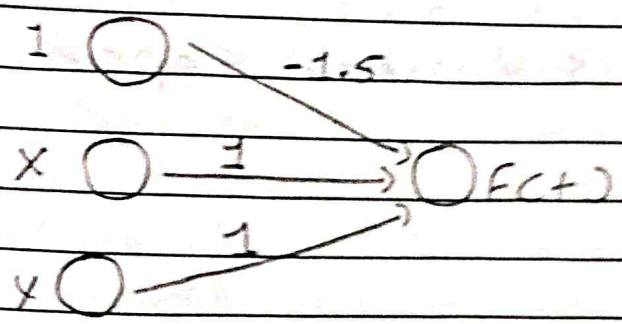
* Consider a Single Perceptron with weight as given in the Figure and F(t) designed as,

$$F(t) = \begin{cases} 1.0, & t > 0 \\ 0, & t \leq 0 \end{cases}$$

, Solve which Kind of Problem

a) OR b) AND c) XOR d) All

Brain Spot



First we have to calculate the value of F(t).

So, $F(t) = \sum_{i=1}^{n} w_i x_i$

$= 1(-1.5) + x(1) + y(1)$

$F(t) = -1.5 + x + y$

After that we have to take $x=0$, $x=1$ or $y=0$, $y=1$ in the F(t).

For $x=0, y=0 \Rightarrow F(t) = -1.5 + 0 + 0$
$= -1.5$

For $x=0, y=1 \Rightarrow F(t) = -1.5 + 0 + 1$
$= -0.5$

For $x=1, y=0 \Rightarrow F(t) = -1.5 + 1 + 0$
$= -0.5$

For $x=1, y=1 \Rightarrow F(t) = -1.5 + 1 + 1$
$= 0.5$

Brain Spot

So, we have to take F(t) value according to given question.

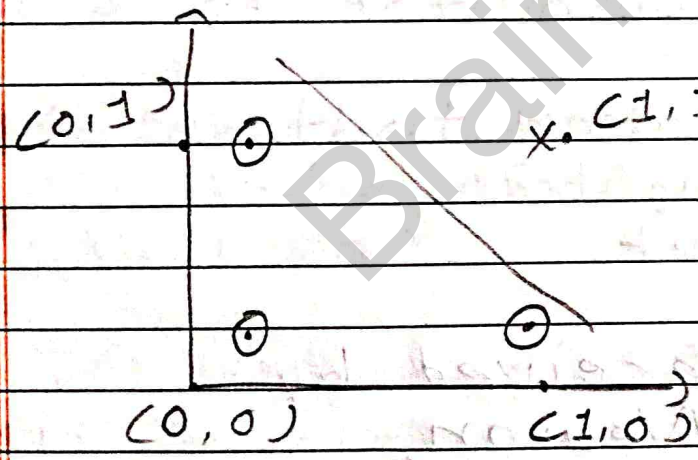
For, $F(t) = 1, \quad t > 0$

$F(t) = 0, \quad t \leq 0$

So,

X	Y	F(t)	F'(t)
0	0	-1.5	0 ⊙
0	1	-0.5	0
1	0	-0.5	0
1	1	0.5	1 X

Here, we can see that F'(t) is a truth table of AND Gate.



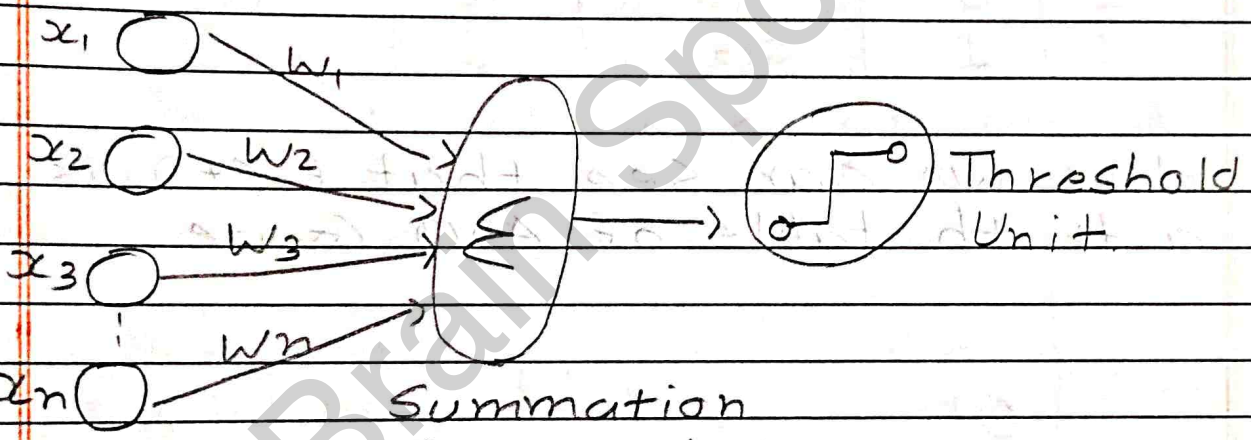
So, the Answer is AND Gate.

Brain Spot

* Explain Simple Model of Artifical Neural Network.

=> Artifical Neural Network mainly consist one Input Layer, which takes or contain Neuron.

This Neuron is contain the weightes. Every weights are connected with output.



Here, Total Received by Cell Body of Neuron,

$$T = w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

$$\therefore I = \sum_{i=1}^{n} w_i x_i \quad \text{— (1)}$$

Brain Spot

After the generates final output of the sum, which is passed on Non-Linear Filter (ϕ) which is called Activation Function.

Activation Function is also known as Transfer Function, which is Release the Final Output.

Activation Function $Y = \phi(I)$

Threshold Function is commanly use as Activation Function.

After that we have to Compare the value of I to the Threshold value (θ)

If value of $I > \theta$ then Output is 1

If value of $I < \theta$ then Output is 0.

So, $$Y = \phi\left(\sum_{i=1}^{n} W_i x_i - \theta\right)$$

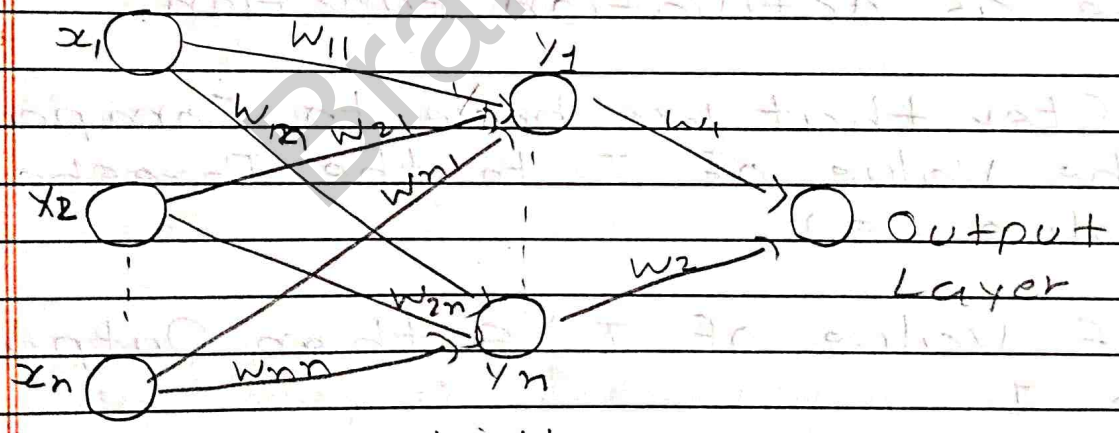
Brain Spot

* Explain Algorithm of Backpropagation

) Backpropagation is a Supervised Learning Algorithm used for training artifical neural networks.

Backpropagation is to minimize the error between the predicted output and the target output by adjusting the weights.

There are three layer in Backpropagation structure: Input Layer, Hidden Layer and Output Layer.



It Compares generated output to the desired output and generates error.

Brain Spot

If result does not match the generated output vector, then it adjusts the weights according to the error to get your desired output.

=> Algorithm:

1 Inputs X, arrive through the preconnected path.

2 The input is modeled using true weights W. Weights are usually chosen random.

3 Calculate output goes into Output Layer.

4 Calculate Error = Actual Output - Desired Output

5 For reduce the error, Go back to the hidden layer to adjust the weights.

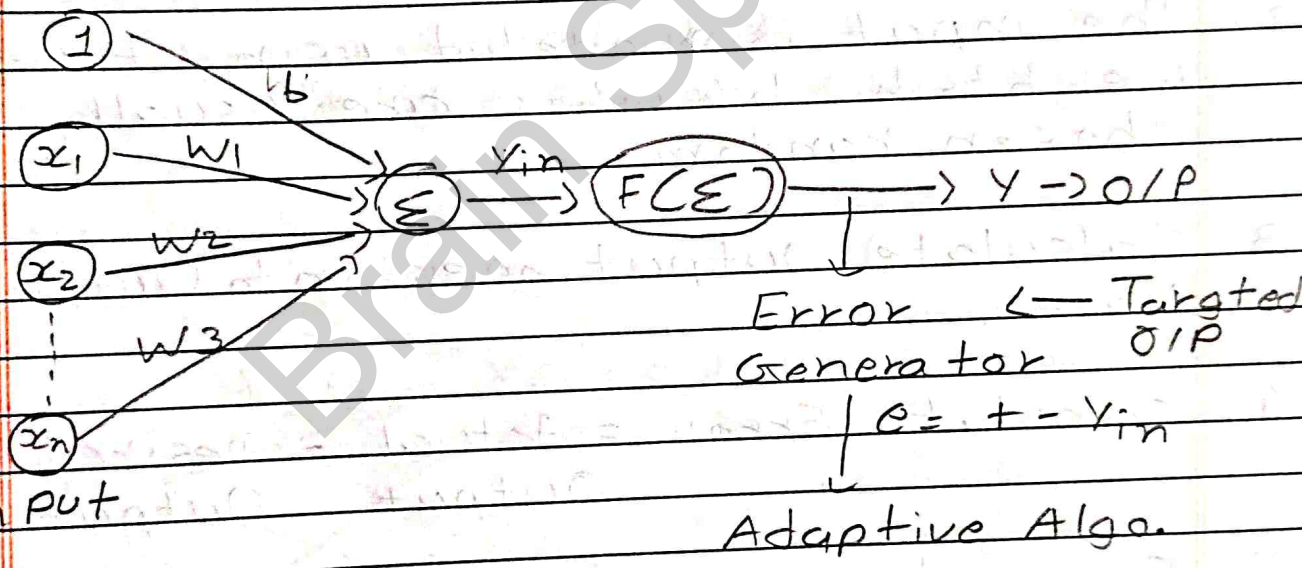
6 Repeat the process until the Desired Output is achieved.

Brainspot

* Explain ADLINE Architecture.

=> ADLINE stands For Adaptive Linear Neuron Structure which is type of single-layer neural network.

ADLINE is use Linear Activation Function For Target its input and output.



In ADLINE structure, First we have to apply input to the Activation Function and compare its output to the original output

IF Both output are equal then produce the output else send an error.

Brain Spot

After that network update the weight according to the error.

Calculate weights,

$$\Delta w_i = \alpha(t - Y_{in})x_i$$

where,

$\alpha \rightarrow$ Learning Rate

$x_i \rightarrow$ I/P Vector

$t \rightarrow$ Targeted O/P

Total Input $Y_{in} = \sum x_i w_i + b$

where,

$x_i \rightarrow$ I/P

$w_i \rightarrow$ Weight

$b \rightarrow$ bias value (1)

Calculate Error (E) $= \sum (t - Y_{in})^2$

$\Rightarrow$ Calculate New weights after the error,

$$w_{(new)} = w_{(old)} + \Delta w_i$$

$$b_{(new)} = b + \Delta b$$

Brain Spot

=> Algorithm :

1 ~~Initize weight~~

1 Initialize weightes randomly but weightes can be not zero and Set α (Lerning Rate).

2 For each Traning Set,

Set, activation of i/p unit $x_i = s_i$ (For $i = 1$ to n)

Calculate net o/p

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

3 Updates the weights and bias For $i = 1$ to n

$$\Delta w_i = \alpha (t - y_{in}) x_i$$

$$\Delta b_i = t - y_{in}$$

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

$$b_{i(new)} = b + \Delta b$$

$$error = (t - y_{in})^2$$

4 When predicated output and the value is same then weights are not change.

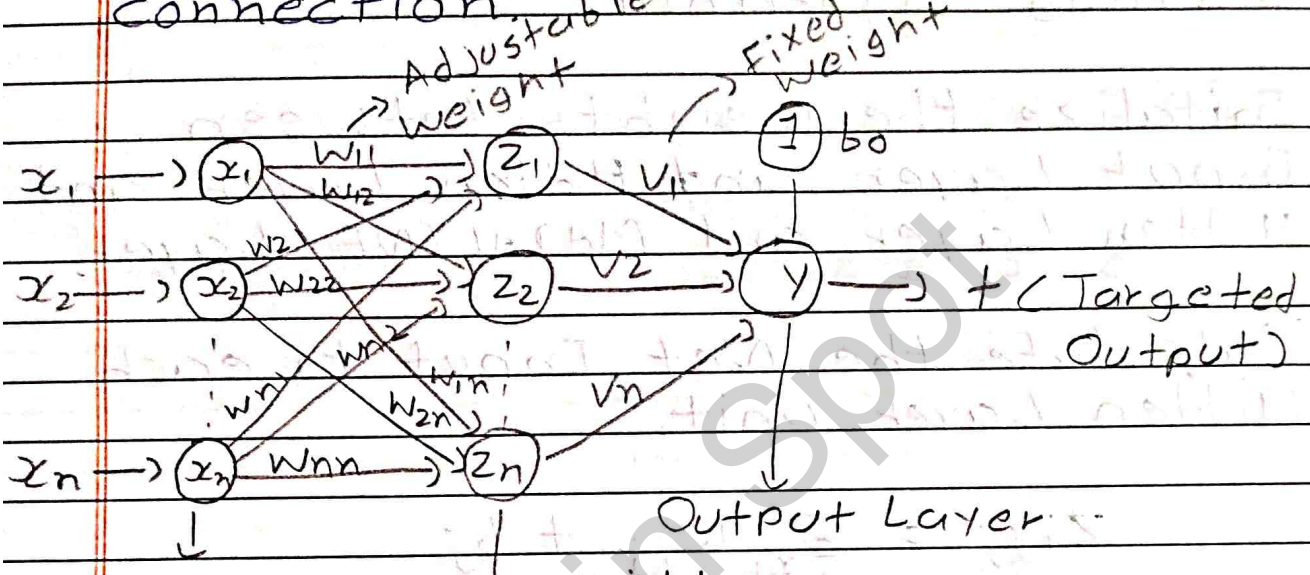
5 else change the weights and continue to step 2.

Brain Spot

* Explain MADALINE Architecture.

=> MADALINE stands for Multiple Adaptive Linear Neuron Architecture.

MADALINE model consist of ~~may~~ many ADALINE in parallel connection.



The weights between Input layer and Hidden Layer are Adjustable in training process.

The weights between Hidden layer and Output Layer are Fixed in training process.

Output Layer have a baise of Input value 1 connected with them.

Brain Spot

Activaction Function For MADALINE is a bipolar Step Function:

$$F(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

=> Training Algorithm:

1. Initialize the weights between Input Layer and Hidden Layer and Hidden Layer and MADALINE Layer.

2. Calculate the Net Input to each Hidden Layer unit,

$$z_{ijj} = \varepsilon_{xi} w_{ij} + b_j$$

3. Calculate the Output of each Hidden Unit

$$z_j = f(z_{in})$$

4. Find the output of Network

$$y_{in} = b_0 + \sum_{J=1}^{n} z_j \cdot v$$

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

BrainSpot

Teacher's Signature

5 If Targeted Output t = Y then
No weight Updation

else, Update the weights.

$$w_{ij(new)} = w_{ij(old)} + \alpha (t - z_{inj}) x_i$$

$$b_{j(new)} = b_{j(old)} + \alpha (t - z_{inj})$$

6 Reapeat the Process until the no change in weights is required.

BrainSpot

2 Explain Characteristics of Neural Networks.

=) This are the basic characteristic of Neural Networks.

1 Adaptability: Neural Network can adapt and learn from data and allowing them to improve their performance over the time.

2 Parallel Processing: Neural Network can process the multiple inputs simultaneously.

3 Non-Linearity: Neural Network enables to model complex relationships and capture patterns that linear model is not represent.

4 Fault Tolerance: In Neural Network, if some neurons or connections are damaged or missing, the network can still performe.

Brain Spot

5 Leaning From Data : Neural Network can learn from the old data rather than relying on explicit programming.

6 Scalability : Neural Network can be scaled to handle large and complex datasets.

7 Applicability to Various Tasks : Neural Network can be applied to a wide range of tasks including Recognition or Processing.

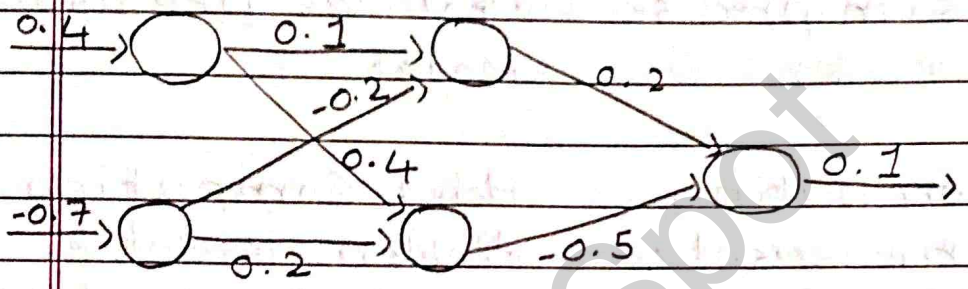
8 Backpropagation : Neural Network algorithm can supervised learning algorithm that adjusts the weights connection based on the error.

9 Generalization : Neural Network can predicte the new outcomes from the past trends.

Brain Spot

SMVS
Page No.
Date : / /

4 Find out Error of the give network where two inputs and one output the values lies between -1 to 1, hence there is no need to normalize the value, Assume two neurons in the hidden layer.



=> Initialize Input:

$$\{O\}_I = \{I\}_I = \begin{Bmatrix} 0.4 \\ -0.7 \end{Bmatrix} \quad - (1)$$

Initialize the Weights:

$$[V]^0 = \begin{bmatrix} 0.1 & 0.4 \\ -0.2 & 0.2 \end{bmatrix} \quad - (2)$$

$$[W]^0 = \begin{bmatrix} 0.2 \\ -0.5 \end{bmatrix} \quad - (3)$$

=> Find The I/p For Hidden Layer

Brain Spot

$\{I\}_H$ = Transpose of V * I/P

$= [V]^T * \{O\}_I$

$= \begin{bmatrix} 0.1 & -0.2 \\ 0.4 & 0.2 \end{bmatrix} \begin{bmatrix} 0.4 \\ -0.7 \end{bmatrix}$

$= \begin{bmatrix} 0.18 \\ 0.02 \end{bmatrix}$ — (4)

$\{O\}_H = \begin{Bmatrix} 1/1+e^{-2I} \\ 1/1+e^{-2I} \end{Bmatrix}$

$= \begin{Bmatrix} 1/1+e^{-1(0.18)} \\ 1/1+e^{-1(0.02)} \end{Bmatrix}$

$= \begin{Bmatrix} 1/1.83 \\ 1/1.98 \end{Bmatrix}$

$= \begin{Bmatrix} 0.5448 \\ 0.505 \end{Bmatrix}$ — (5)

Brain Spot

=> Input For Output Layer:

$$\{I\}_O = \{W\}^T * \{O\}_H$$

$$= \begin{bmatrix} 0.2 & -0.5 \end{bmatrix} * \begin{bmatrix} 0.5448 \\ 0.505 \end{bmatrix}$$

$$= (0.2 \times 0.5448 + (-0.5)0.505)$$

$$= -0.1445$$

-> Find out Activation Function:

$$= \frac{1}{1+e^{-2I}} = \frac{1}{1+e^{-(-0.1445)}}$$

$$= 0.464$$

Error Finding $= (T_o - O_o)^2$

$$= (0.1 - 0.464)^2$$

$$= (-0.3642)^2$$

$$= 0.13264$$

Brain Spot

5 Consider the following Patterns:
$A1 = (-1, 1, -1, 1)$, $A_2 = (1, 1, 1, -1)$
$A_3 = (-1, -1, -1, 1)$

(a) Recognition of Stored Pattern:

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

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

$$T = A^T * A$$

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

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

Brain Spot

SMVS
Page No.
Date : / /

=> For Pattern:

-> $A_2 = [1 \quad 1 \quad 1 \quad -1] \begin{bmatrix} 3 & 1 & 3 & -3 \\ 1 & 3 & 1 & -1 \\ 3 & 1 & 3 & -3 \\ -3 & -1 & -3 & 3 \end{bmatrix}$

$= [10 \quad 6 \quad 10 \quad -10]$

$= [1 \quad 1 \quad 1 \quad -1]$

This Pattern is Recognition

-> $A_1 = [-1 \quad 1 \quad -1 \quad 1] \begin{bmatrix} 3 & 1 & 3 & -3 \\ 1 & 3 & 1 & -1 \\ 3 & 1 & 3 & -3 \\ -3 & -1 & -3 & 3 \end{bmatrix}$

$= [-8 \quad 0 \quad -8 \quad 7]$

$= [-1 \quad 1 \quad -1 \quad 1]$

This Pattern is Recognition

-> $A_3 = [-1 \quad -1 \quad -1 \quad 1] \begin{bmatrix} 3 & 1 & 3 & -3 \\ 1 & 3 & 1 & -1 \\ 3 & 1 & 3 & -3 \\ -3 & -1 & -3 & 3 \end{bmatrix}$

Brain Spot

$= [-1 \quad -1 \quad -1 \quad 1]$

This Pattern in Recognition.

=> Recognition of Noisy Pattern

$A' = [1 \quad 1 \quad 1 \quad 1]$

$A' - A_1 = (1, 1, 1, 1) - (1, 1, 1, -1)$
$= (2, 0, 2, 0) = 4$

$A' - A_2 = (1, 1, 1, 1) - (-1, -1, -1, 1)$
$= (0, 0, 0, 2) = 2$

$A' - A_3 = (1, 1, 1, 1) - (-1, -1, -1, 1)$
$= (2, 2, 2, 0) = 6$

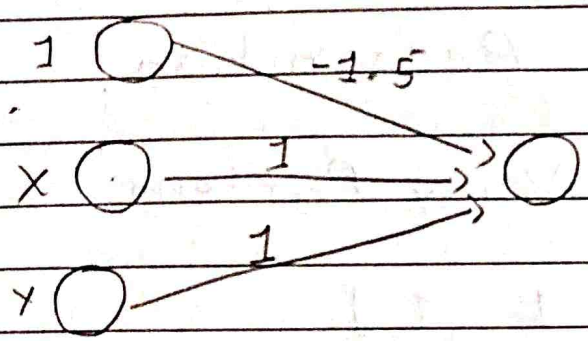
So, A_2 Pattern is Noisy Pattern.

6 Consider a Single Perceptron with weight as given in the Figure and $F(t)$ designed as,

$$F(t) = \begin{cases} 1, & t > 0 \\ 0, & t \leq 0 \end{cases}$$

Brain Spot

Can solve which kind of problem,
A) OR B) AND C) Ex-Or D) All



Here, $F(t) = \sum_{i=1}^{n} w_i x_i$

$= 1(-1.5) + 1(x) + 1(y)$

$= -1.5 + x + y$

After that we have to take $x=0$, $x=1$ or $Y=0$, $Y=1$.

For $F(t) = 1, \quad t > 0$

$F(t) = 0, \quad t \leq 0$

x	Y	F(t)	F(t)
0	0	-1.5	0
0	1	-0.5	0
1	0	-0.5	0
1	1	0.5	1 X

Brain Spot

SMVS
Page No.
Date : / /

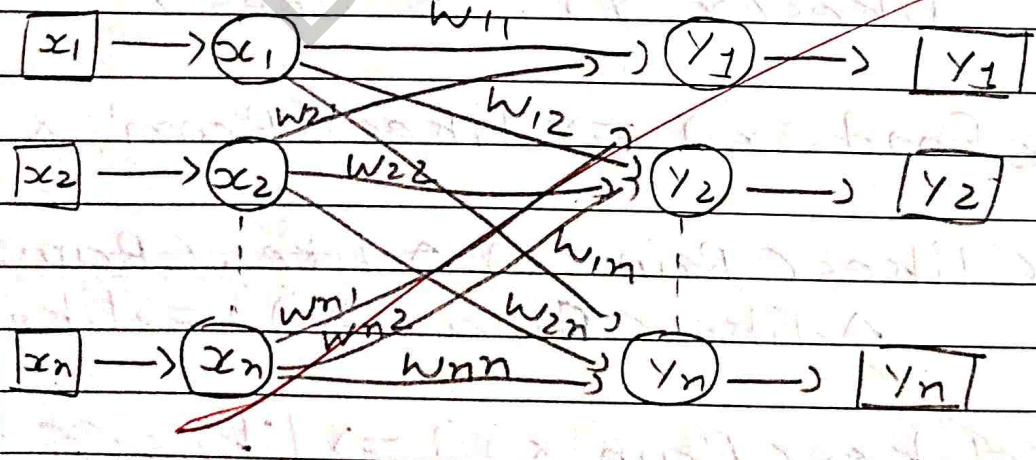
5 Explain Auto associative and Hetero associative Memory.

=> Auto Associative Memory:

Auto Associative Memory is a type of a single layer neural network.

In Auto Associative Memory, Input Training vector and the target output vector are the same.

Auto Associative memory makes a parallel search with the stored patterns.



Auto Associative memory network has n number of input training

Brain Spot

vectors and similar n number of output target vectors.

→ Training Algorithm:

1 Initialize all the weights to zero as $w_{ij} = 0$, $i = 0$ to n, $j = 1$ to n

2 For each input vector,

Active each input unit $x_i = s_i$ ($i = 1$ to n)

Active each output unit $y_i = s_j$ ($j = 1$ to n)

3 Adjust the weights,

$$w_{ij(new)} = w_{ij(old)} + x_i y_i$$

where, x_i = Input Vector
y_i = Output Vector
w = Weights between Input and Output Vector

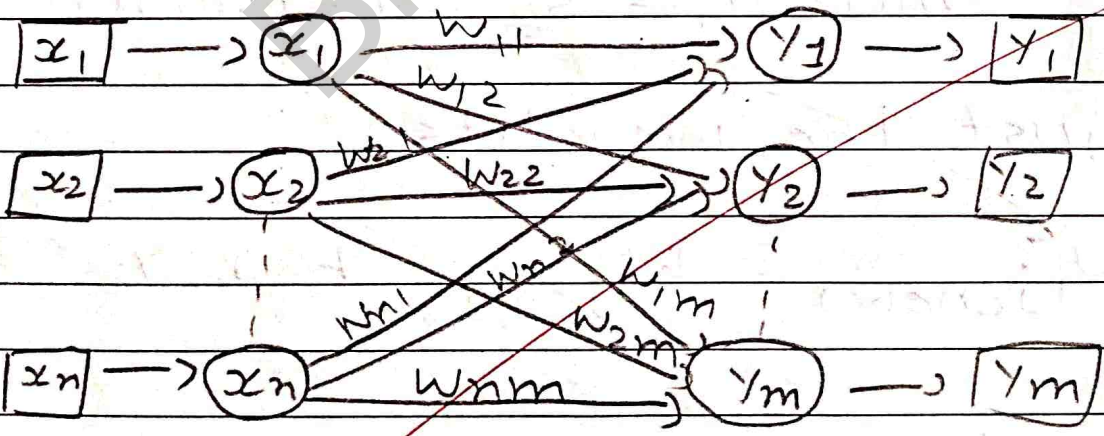
Brain Spot

⇒ Hetero Associative Memory:

Hetero Associative Memory is a type of a single layer neural network.

In Hetero Associative Memory, The Network input training vector and output target vector are not same.

In this network weights are determined so, that the network stores a set of pattern.



Hetero Associative Memory network has n number of input training vector and m number of output target vectors.

Brain Spot

Hetero Associative Memory network is use Hebb or Delta ~~trc~~ learning rule.

-> Training Algorithm:

1 Initialize all the weights to zero as $W_{ij} = 0$, $i = 1$ to n, $j = 1$ to m.

2 For each input vector,

Activate each Input Unit ($x_i = s_i$ $(i = 1$ to $n)$

Activate each Output Unit ($y_j = s_j$ $(j = 1$ to $m)$

3 Adjust the weights,

$$W_{ij(new)} = W_{ij(old)} + x_i y_j$$

Where, x_i = Input Vector
y_j = Output Vector
W = Weights between Input and Output Vector.

Brain Spot