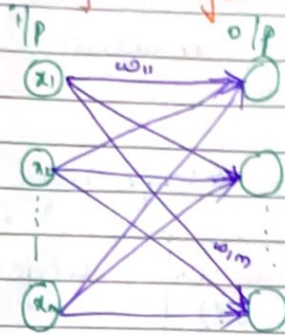


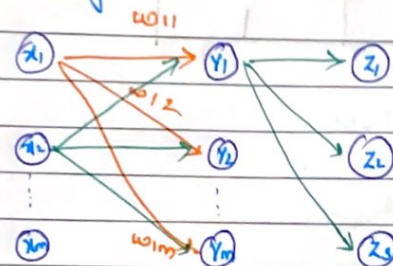
ANN Architecture

DATE 08 10 2021

1. Single layer Feed Forward.

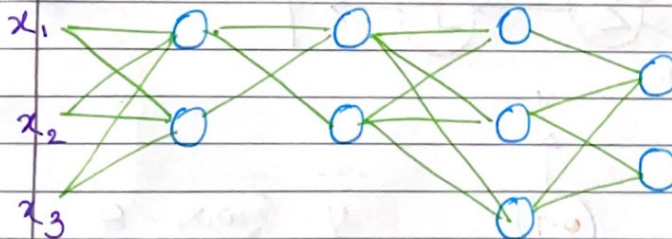


2. Multilayer Feed forward n/w



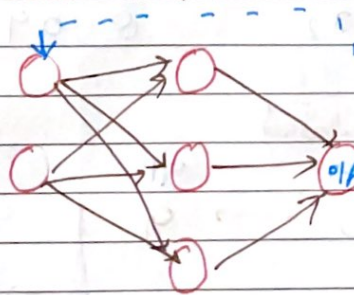
computationally more stronger

3. Multilayer Perceptron (fully three or more layers connected)



To classify non-linear separable data

4. Feed back ANN



Feedback is provided to adjust parameters

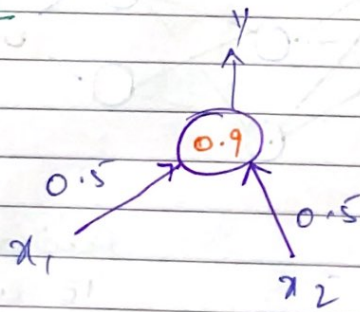
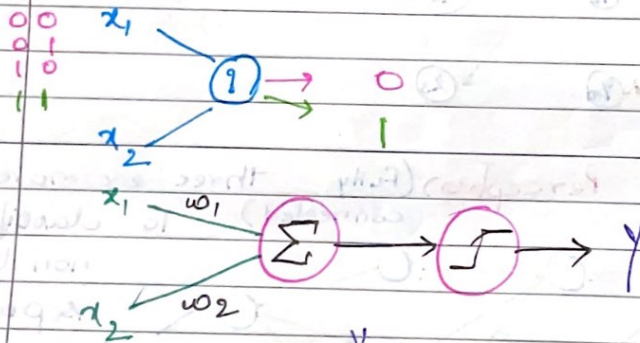
1. Single layer feed forward
2. Multilayer feed forward
3. Recurrent n/w

① \Rightarrow

AND PROBLEM - Single layer feed forward
acyclic.

Inputs		output
x_1	x_2	(Y)
0	0	0
0	1	0
1	0	0
1	1	1

\rightarrow Pattern Recognition



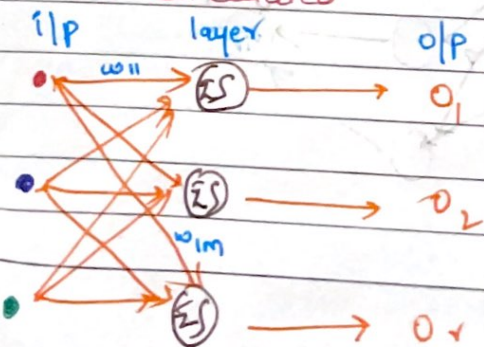
$$Y = \sum w_i x_i - \theta$$

$$w_1 = 0.5$$

$$w_2 = 0.5$$

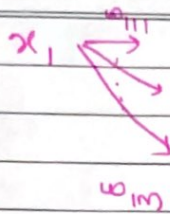
$$\theta = 0.9$$

How θ is calculated



$m \times n$

VISION



- ⇒ 1. Each neuron has same transform function
 2. The weights are the parameters
 3. How n/w can be trained
 4. The Hidden layers give intermediary computation

Modelling SLFFNN (Single Layer Feed Forward Neural Network)

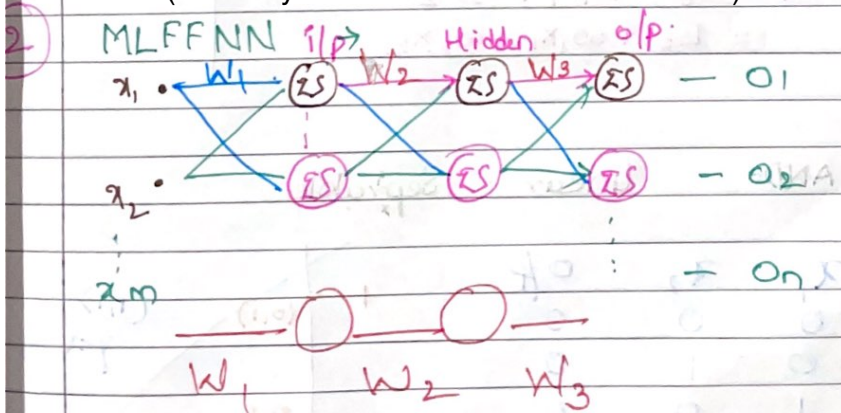
$$W = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ w_{21} & & w_{2m} \\ \dots & & \vdots \\ w_{m1} & & w_{mn} \end{bmatrix}$$

o/p of k^{th} neuron

$$O_k = F_k \left(\sum_{i=1}^m (w_{ik} x_i) - \theta_k \right), \quad k = 1, 2, 3, \dots, n$$

↓
Threshold Value.

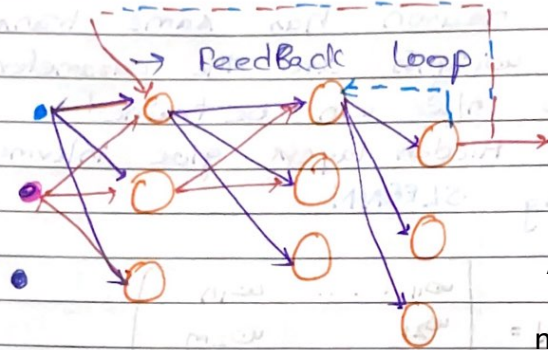
(Multi Layer Feed Forward Neural Network)



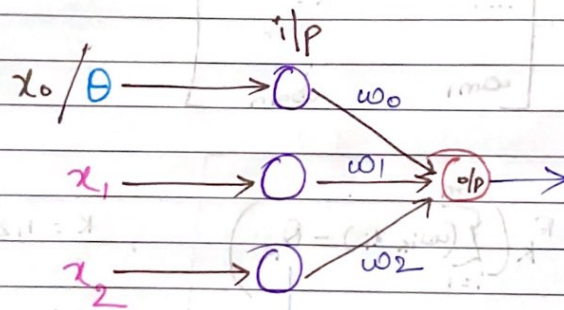
Modelling

$$O_i^l = f_i^l \{ \omega_i^l x_i + \theta_i^l \}$$

③ Recurrent neural network



A recurrent neural network (RNN) is a special type of an artificial neural network adapted to work for time series data or data that involves sequences. Ordinary feed forward neural networks are only meant for data points, which are independent of each other.

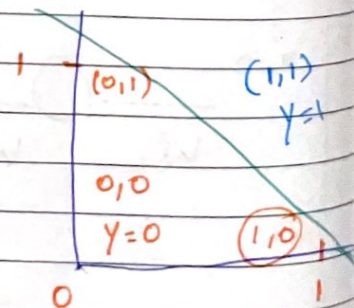


$$f = \omega_0 \theta + \omega_1 x_1 + \omega_2 x_2$$

$$= b_0 + \omega_1 x_1 + \omega_2 x_2$$

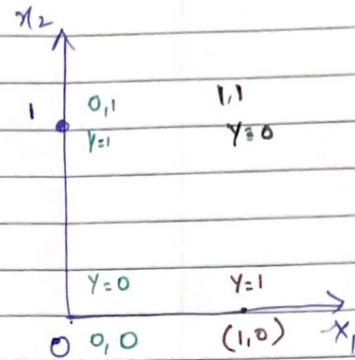
AND= linear Separable

x_1	x_2	o/p
0	0	0
0	1	0
1	0	0
1	1	1

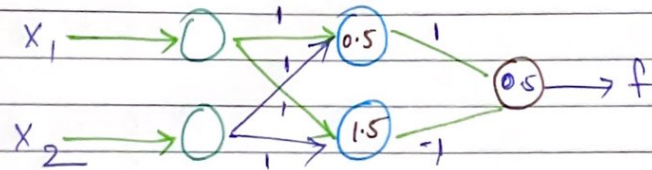


XOR

x_1	x_2	O/P
0	0	0
0	1	1
1	0	1
1	1	0



\Rightarrow linear non-separable.



\Rightarrow MLFFNN should be used