

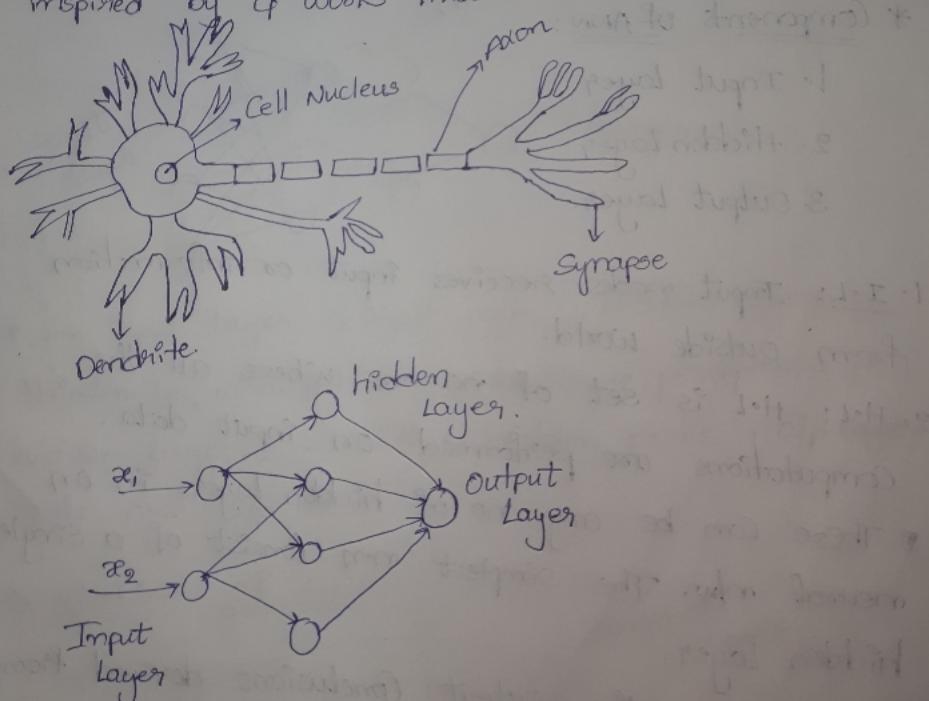
$\frac{2}{3} \log_2 \frac{2}{3}$ )

25/03/23.

Unit-4

### Artificial Neural Network

- \* ANN is inspired by a working of human brain.
- \* the human brain has neurons interconnected to one another, ANN also have neurons that are interconnected to one another in various layers of the N/w. These neurons are known as nodes.
- \* Neural networks are set of algorithm that tries to recognise the patterns, relationships & information from the data through the process which is inspired by & works like the human brain.



### Biological

- \* Dentists
- \* Cell Nucleus
- \* Synapse
- \* Axon

### Components of ANN:

1. Input Layer
2. Hidden Layer
3. Output Layer.

1. I.L: Input nodes receives input or information from outside world.

2. H.L: H.L is set of neurons where all the computations are performed on input data.

3. There can be any no. of hidden layers in an neural net. The simplest n.n consist of a single hidden layer.

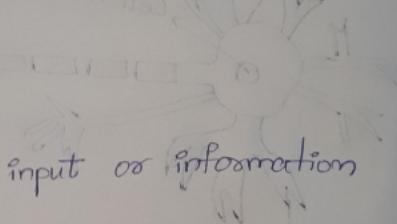
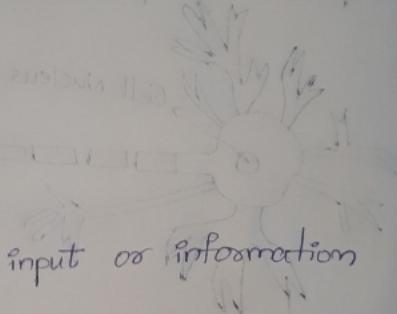
3. O.L: O.L is the output Conclusions derived from all the computations performed

\* There can be single or multiple nodes in the output layer. if we have a binary classification problem the output known is 1. But in case of multi class classification, the output nodes can be more than 1.

### Artificial

- \* Inputs
- \* Nodes.
- \* weights
- \* output.

but although the two calculate function is same i.e. product of input value with weight and sum of all the weights coming with neurons who are marked green and orange and total value is put before



\* perceptron

\*

Con

Con

\* Multilayer Perceptron

\*

thom

mult

Input  
Layer

\* Feed forward

three

hid

with

$\Rightarrow$

$x_1 \rightarrow v$

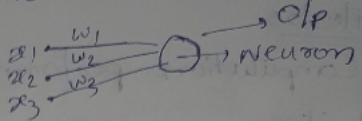
$x_2 \rightarrow w$

$x_3 \rightarrow z$

$z \in$

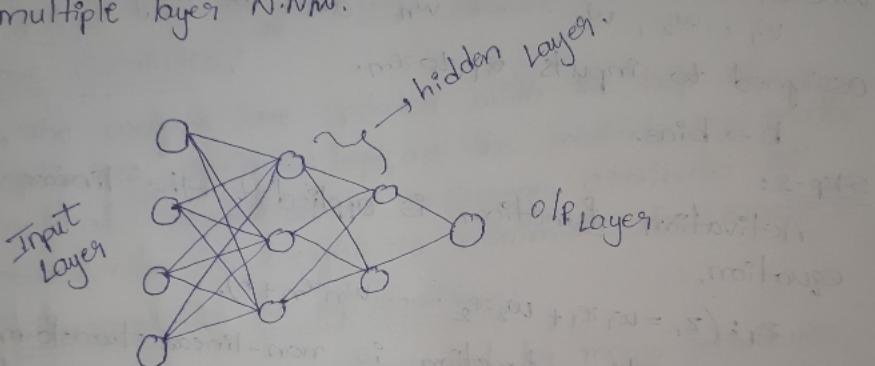
### \* perception:- perception:

\* perception is a simple form of N.N/w & consists of single layer where all the mathematical computations are performed.



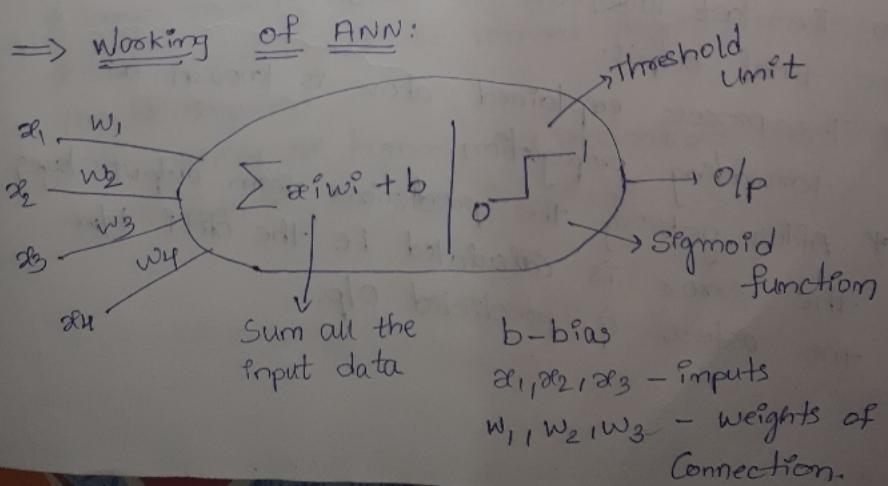
### \* Multilayer perception:

\* M.P is also known as A.N.N consists of more than 1 perception which is grouped together to form multiple layer N.N/w.



\* An input layer, 6 input nodes, Hidden Layer - 1, hidden Layer, with 4 hidden nodes (or) 4 perceptions, hidden Layer - 2 - with 4 hidden nodes, o/p Layer with one o/p node.

### $\Rightarrow$ Working of ANN:



\* bias is the constant that helps the model to fit in the best way possible.

27/03/23

\* Computation perform in hidden layers ~~is~~ done in

2 steps:

1. All the inputs are multiplied by the weight

$$\text{Step-1: } z_1 = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b$$

where,  $w_1, w_2, w_3, \dots, w_n$  are the weights assigned to inputs  $x_1$  to  $x_n$ .

$b \rightarrow$  bias.

Step-2:

Activation function is applied to the linear equation.

$$z_1 : (z_1 = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)$$

\* the activation function is non-linear transformation that is applied to the input before sending it to the next layer of neurons.

Step-3:

The computation results from hidden layer passed to the last layer i.e output layer which gives as the final output.

\* The process explained above is known as forward propagation.

\* After getting the predictions from output layer the error is calculated i.e the diff b/w the actual & predicted op.

- \* If the error is large, then the steps are taken to minimize the error & for the same purpose, Back Propagation is performed.
- done in Back propagation:
- \* It is process of updating & finding the optimal values of weights (or) co-efficients which helps to minimize the error, i.e. diff b/w the actual & predicted value.
  - \* How the weights are updated & new weights are calculated?  
→ The weights are updated with the help of optimises, optimises (opt), are the methods (or) mathematical formulation to change attributes of neural n/w i.e. weight to minimises the error.
  - \* Characteristics of B.P Algorithm:
    - \* Instances are represented by many attributes.
    - \* The target function output (make) discrete value, real value, (or) a vector of several real (or) discrete value attributes.
    - \* The training examples may contain error.
    - \* Large training times are acceptable.
    - \* fast evalution of the learned target function may be required.
    - \* The ability of humans to understand the learned target function is not important.

27/05/93

### \* Perceptron learning Algorithm:

1. Initialise weights & threshold.
- \* Define  $w_i(t)$ , ( $i \in n$ ), to be the weight from input  $i$  at the time  $t$ , &  $\theta$  to be the threshold value in the output node.
- \* set  $w_0$  to be  $-\theta$ , the bias, &  $y$  to be always 1.
2. present input & desired output.  
\* present input  $x_0, x_1, x_2, \dots, x_n$  & desired o/p  $d(t)$ .
3. Calculate actual o/p.

$$y(t) = f_m \left[ \sum_{i=0}^n w_i(t)x_i(t) \right]$$

4. Adapt weights.

if correct  $w_i(t+1) = w_i(t)$

\* if o/p, 0, should be 1 (class A)  $w_i(t+1) = w_i(t) + \eta x_i(t)$ .

\* if o/p 1, should be 0 (class B)  $w_i(t+1) = w_i(t) - \eta x_i(t)$ .

→ Adapt weights - modified version.

\* if correct,  $w_i(t+1) = w_i(t)$ .

\* if output 0, should be 1 (class A),  $w_i(t+1) = w_i(t) + \eta x_i(t)$ .

\* if output 1, should be 0 (class B),  $w_i(t+1) = w_i(t) - \eta x_i(t)$ .

where  $0 \leq \eta \leq 1$ , a positive gain term that

Controls the adoption rate.

\* The error term  $\Delta$  can be written

$$\Delta = d(t) - y(t).$$

→ Adapt weights - Widrow-Hoff delta rule.

$$\Delta = d(t) - y(t).$$

$$w_i(t+1) = w_i(t) + \eta \Delta w_i(t)$$

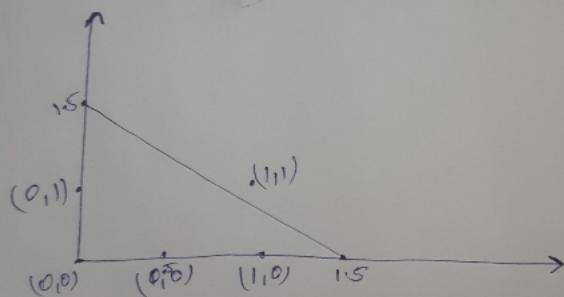
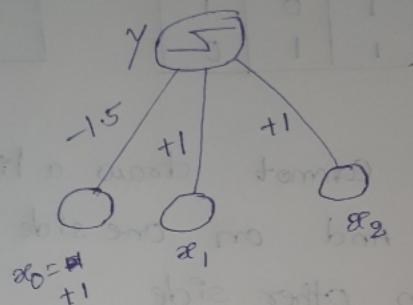
$$\Delta(t) = \begin{cases} +1, & \text{if input from class A} \\ 0, & \text{if input from class B.} \end{cases}$$

\* Input of  $\Delta p$  for the and function:

Learning Boolean function:

AND

$x_1$	$x_2$	$d$
0	0	0
0	1	0
1	0	0
1	1	1



\* In a Boolean function,

The perceptron that implements And and

its geometric interpretation.

\* In a Boolean function, that inputs are binary

& output is 1 if the corresponding function

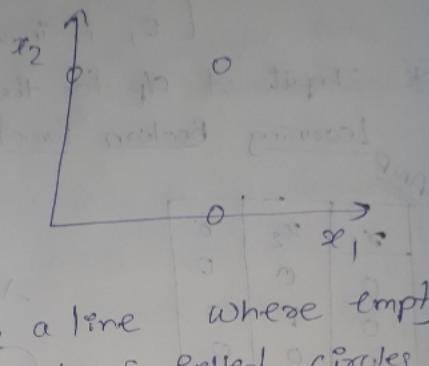
value is true (& the inputs are zero otherwise)

therefore, it can be seen as a 2 class classification problem.

\* Input of o/p for XOR function

XOR.

$x_1$	$x_2$	$\sigma$
0	0	0
0	1	1
1	0	1
1	1	0



\* we cannot draw a line where empty circle and on one side & filled circles are on other side.

\* XOR process is not linearly separable.