Decision Tree learning :-

- It is used in classification kind of algorithms.
- To construct any decision tree earlier we use C4.5 algorithm upon which we are using ID3 algorithm to construct any given tree.
- ID3 algorithm uses entropy as a constraint which describes the impurity, missing value, error rate in any given data set.
- The value of entropy lies between 0 and 1

0 specifies the least or minimum errors and
1 specifies maximum errors.

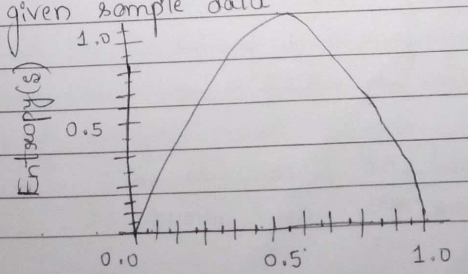
$$Entropy(S) = -p_{\oplus}\log_2 p_{\oplus} - p_{\ominus}\log_2 p_{\ominus}$$

eg :-

S is collection of 14 samples which includes 9 positive and 5 negative examples. The entropy of S is calculated by

$$Entropy([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.940$$

The entropy of above is approximately equal to 1 which clearly tells us the higher error state for the given sample data



Information gain observes the entropy with the least value over the eg sample of data 'yes' with the given formula.

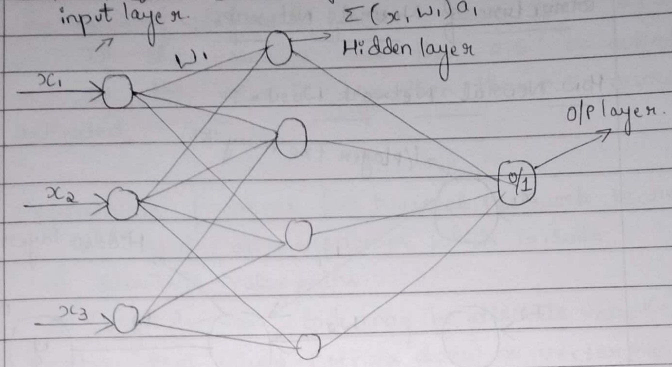
$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Dataset of play tennis

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D_1	Sunny	Hot	High	Weak	No
D_2	Sunny	Hot	High	Strong	No
D_3	Overcast	Hot	High	Weak	Yes
D_4	Rain	Mild	High	Weak	Yes
D_5	Rain	Cool	Normal	Weak	Yes
D_6	Rain	Cool	Normal	Strong	No
D_7	Overcast	Cool	Normal	Strong	Yes
D_8	Sunny	Mild	High	Weak	No
D_9	Sunny	Cool	Normal	Weak	Yes
D_{10}	Rain	Mild	Normal	Weak	Yes
D_{11}	Sunny	Mild	Normal	Strong	Yes
D_{12}	Overcast	Mild	High	Strong	Yes
D_{13}	Overcast	Hot	Normal	Weak	Yes
D_{14}	Rain	Mild	High	Strong	No

Neural Networks

Structure of Neural N/w Hidden layer



input :- Where the i/p is received

Hidden :- Where all Calculation happens

O/P :- May be Binary or Multilayer.

Properties of Neural Networks

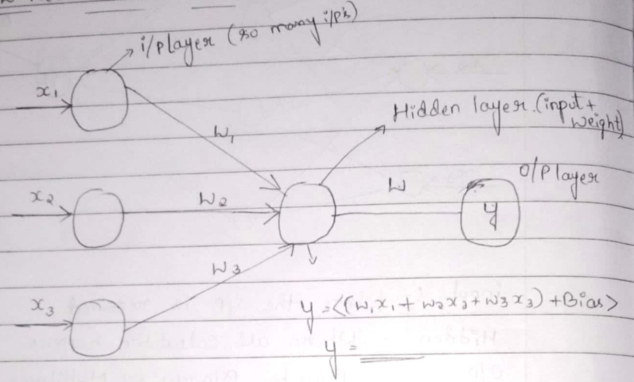
* Many neuron-like threshold switching units
* Highly parallel, distributed.

Introduction :-

* Around 1950's, Scientist called Rosen Blatt came up with the concept of perceptron
* Perceptron is a single neural network Based model, which acted as a basis to the development of deep learning modules.
* Scientist called jeffory hinten came up with the concept of Backpropagation algorithm which added as a fundamental algorithm for the area of artifical neural networks.
* The objective of deep learning was to mimic the human brain with same functionalities

Structure of Neural Networks :-

How Neural Network Works :-



$y = \langle (w_1x_1 + w_2x_2 + w_3x_3) + Bias \rangle$

$y =$

$Z = Act(y)$

Steps in Network function :-

Step ① :- Hidden Layer gets the input of the summation of input x_i and w_i (that is $\sum_{i=1}^{n} w_ix_i$)

Step ② :- Activation function is applied at the hidden layer with the values $y = w_1x_1 + w_2x_2 + w_3x_3 + bias$

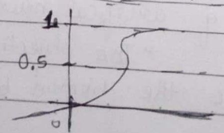
Step ③ :- The output z is obtained by applying the activation function on y. $\therefore Z = Act(y)$

Step ④ : Activation function is of different types, few of them are :-

Step function

Sign function

* Sigmoid function :- $\boxed{Sigmoid(x) = 1/\left(1+e^{-y}\right)}$ → Formula



The value of Si
If the v
If the v
activated

Appropriate Pr
is based
a) attribute -
b) target func
marks), Real v
(Obtaining rel
c) Training
d) Training
of iterations

Architectures

Ther

ANN
a) Single laye
b) Multi-laye
c) Recurrent
d) Mesh Ne

a) x_1 –
x_2 –
x_3 –
i/p la

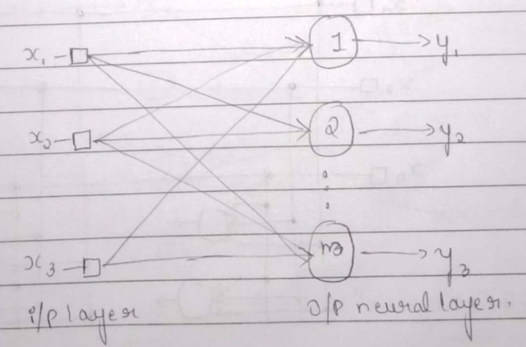
The value of sigmoid function varies from 0 to 1
If the value is less than 0.5 no activation.
If the value is greathen than 0.5 neuron is activated.

Appropriate Problems for Neural Network learning is based on scenarios which include
a) attribute-value pairs
b) target function output may be discrete valued (exam marks), real valued (sensor data) or vector valued (obtaining vehicle registeration number in a traffic image)
c) Training examples which may contain errors.
d) Training example which may require more number of itterations

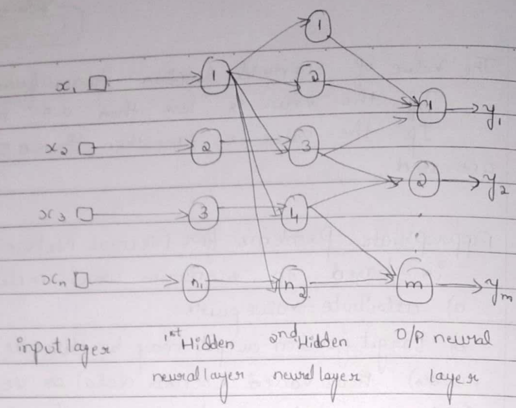
Architectures of Artificial Neural Networks.

There are different types of architecture of ANN
a) Single layour feed forward network
b) Multi-layour
c) Recurrent or Feed back Network
d) Mesh Networks

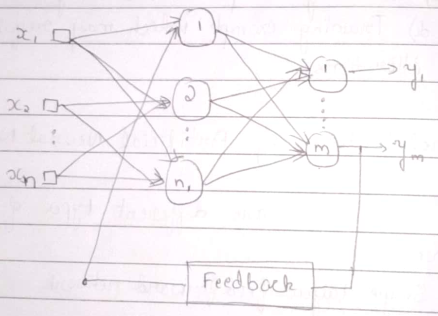
a)



b)



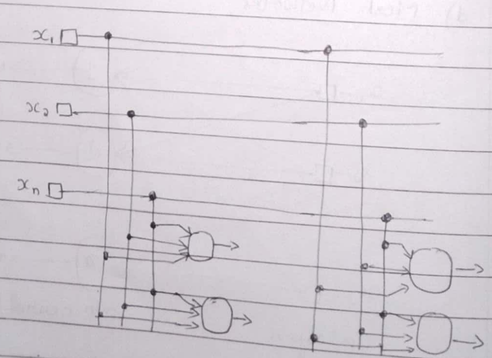
c)



Note :-

Perceptron is a single layer neural network

d)



Perceptron :-

The output of the perceptron is computed with the inputs x_1 to x_n

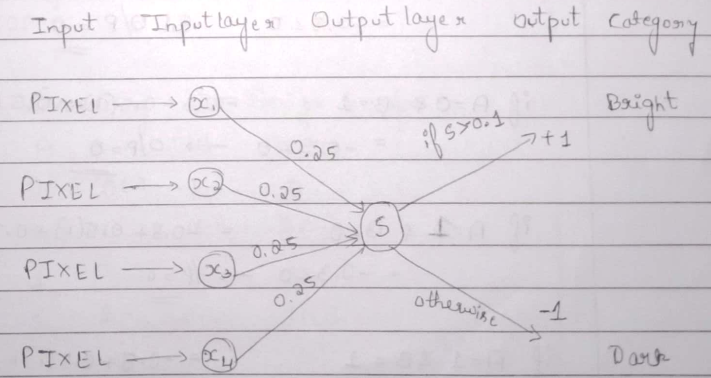
$$o(x_1, \ldots x_n) = \begin{cases} 1 & \text{if } w_0 + w_1x_1 + \cdots + w_nx_n > 0 \\ -1 & \text{otherwise.} \end{cases}$$

In Perceptron model the threshold weight (w_0) is increased when the obtained output is not equal to the desired output.

The difference b/w change in the weights with one itteration to other is called as learning rate of a neural network.

Final obtained value of a neural network must has less variance with more bias and least learning rate.

Eg :- Consider the given scenario in which we have to classify a given image is dark or bright.



$S = 0.25^4 \times 1 + 0.25^4 \times 2 + 0.25^4 \times 3 + 0.25^4 \times 4$

Problems on Perceptron Based model :

① How does a single perceptron can be used to represent Boolean functions such as and & or

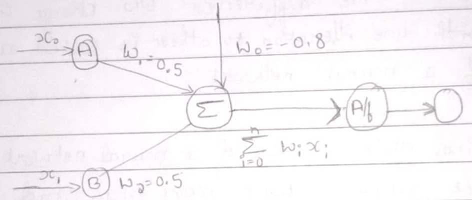
→ AND

A	B	A∩B
0	0	0
0	1	0
1	0	0
1	1	1

Let $W_0 = -0.8$

$W_1 = 0.5$

$W_2 = 0.5$



if A=0 & B=0

$= W_0 + w_1 x_1 + w_2 x_2$

$= -0.8 + 0.5x_1 + 0.5x_2$

$= -0.8 + 0.5(0) + 0.5(0)$

$= -0.8 < 0$ So O/P = 0

if A=0 & B=1 $= -0.8 + 0.5(0) + 0.5(1)$

$= -0.3 < 0 \longrightarrow$ O/P = 0

if A=1 & B=0 $= -0.8 + 0.5(1) + 0.5(0)$

$= -0.3 < 0 \rightarrow$ O/P = 0

if A=1 & B=1 $= -0.8 + 0.5(1) + 0.5(1)$

$= 0.2 > 0 \rightarrow$ O/P = 1

A	B
0	0
0	1
1	0
1	1

if A=

if A=

if A

if A

②i) Design
boolean

ii) Design
XOR

→ Const

A	B	A∪B
0	0	0
0	1	1
1	0	1
1	1	1

let $W_0 = -0.3$
$W_1 = 0.5$
$W_2 = 0.5$

if $A=0$ & $B=0$ $\quad -0.3<0 \quad \rightarrow$ O/P = 0

$= -0.3+0.5(0)+0.5(0)$

$= -0.3$

if $A=0$ & $B=1$

$= -0.3+0.5(0)+0.5(1)$

$= 0.2 > 0$ $\quad$ O/P = 1

if $A=1$ & $B=0$

$= -0.3+0.5(1)+0.5(0)$

$= 0.2>0$ $\quad$ O/P = 1

if $A=1$ & $B=1$ $\quad = -0.3+0.5(1)+0.5(1)$

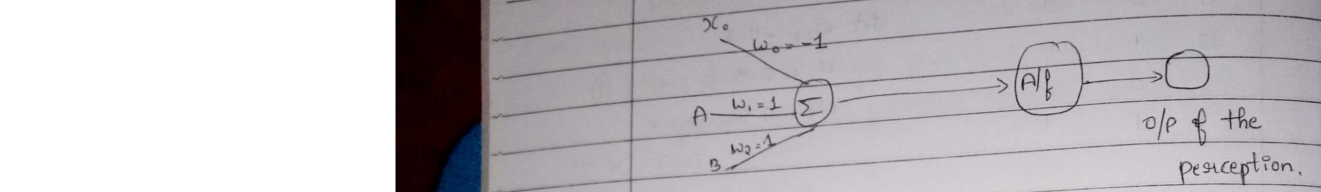
$= 0.7>0$ O/P = 1

(2) i) Design a two input perceptron that implements the boolean function $A \cap \neg B$.

ii) Design a two layer perceptron that implements A XOR B.

→ Constructing the Design surface.

A	B	¬B	A∩¬B
0(-1)	0(-1)	1	0
0(-1)	1	0(-1)	0(-1)
1	0(-1)	1	1
1	1	0(-1)	0(-1)



* From the above perceptron the values of A and B are 1 (True) or -1 / 0 (false condition)
* Observing the decision surface in the design of perceptron the line crosses 'a' axes at 1 and 'b' axes at -1.
* The weights are $w_0 = -1$, $w_1 = 1$ and $w_2 = -1$

ii) A xor B cannot be calculated by a single perceptron, so build a 2 layer network of perceptrons.

(3) Consider 2 perceptrons define by the threshold expression $w_0 + w_1 x_1 + w_2 x_2$. Perceptron 'A' has weight values $w_0 = 1$, $w_1 = 2$, $w_2 = 1$
Perceptron B has weight values $w_0 = 0$, $w_1 = 2$, $w_2 = 1$.
Justify the above statement and check whether perceptron A is more general than perceptron B.

→ Perceptron A :- $w_0 = 1$, $w_1 = 2$, $w_2 = 1$

$$A(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

$$= 1 \qquad = 1 + 2x_1 + x_2 > 0$$

$$= 1 + 2 + 1 > 0$$

$$B(x_1, x_2) = 1 \quad \& \quad w_0 = 0, \; w_1 = 2, \; w_2 = 1$$

$$0 + 2x_1 + x_2 > 0$$

$$= 0 + 2 + 1 > 0$$

Problems on Decision Tree learning :-

Dataset of play tennis :-

(1) Give the Decision Tree for the following set of Training Examples :-

→

$$Entropy(S) = -P_{\oplus} \log P_{\oplus} - P_{\ominus} \log P_{\ominus}$$

$$Gain(S) = Entropy(S) - \sum_{v \in values} \frac{|S_v|}{|S|} Entropy(S_v)$$

* Note :-

Entropy (S) = 0 (Same class) either yes or no

Entropy (S) = 1 [Contains equal no. of Positive & Negative Samples]

Entropy of S : P.S = 09

NS = 05

$$Entropy[9+, 5-] = -\left(\frac{9}{14}\right)\log\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right)\log\left(\frac{5}{14}\right)$$

$$= 0.940$$

Information Gain of attribute outlook.

Value (outlook) = Sunny, Overcast, Rain

$S_{Sunny} \rightarrow [2+, 3-]$

$S_{Overcast} \rightarrow [4+, 0-]$

$S_{Rain} \rightarrow [3+, 2-]$

$$Gain(S, outlook) = Entropy(S) - \left[\left(\frac{5}{14}\right) Entropy(S_{Sunny}) + \left(\frac{4}{14}\right) Entropy(S_{Overcast}) + \left(\frac{5}{14}\right) Entropy(S_{Rain})\right]$$

$$\text{Entropy}(S_{Sunny}) = -\left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)$$

$$= 0.970$$

$$\text{Entropy}(S_{Overcast}) = 0$$

$$\text{Entropy}(S_{Rain}) = -\left(\frac{3}{5}\right)\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}$$

$$= 0.970$$

$$= 0.940 - \left[\frac{5}{14} * 0.9709 + 0 + \left(\frac{5}{14}\right) 0.9709\right]$$

$$= 0.246$$

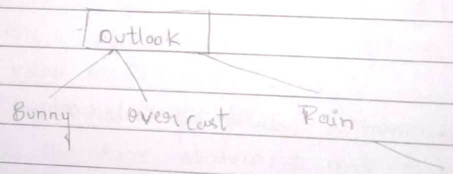
So Gain (S, outlook) = 0.246

Gain (S, Temp) = 0.029

Gain (S, Humidity) = 0.151

Gain (S, Wind) = 0.029

So Root Node will be outlook



{D1, D2, D8, D9, D11} {D3, D7, D12, D13} {D4, D5, D6, D10, D14}

[2+, 3-] [4+, 0-] [3+, 2-]

$S_{Sunny} = \{D_1, D_2, D_8, D_9, D_{11}\}$

$$\text{Gain}(S_{Sunny}, \text{Humidity}) = 0.970 - \left[\frac{3}{5} \times 0 + \frac{2}{5} \times 0\right]$$

$$= 0.970$$

$$Gain(Sunny, Temp) = 0.970 - \left[\left(\frac{2}{5}\right) \times 0 + \frac{2}{5} \times 1 + \left(\frac{1}{5}\right) \times 0\right]$$

$$= 0.970 - 0.4$$

$$= 0.570$$

$$Gain(Sunny, Wind) = 0.970 - \left[\left(\frac{3}{5}\right) 0.98 + \left(\frac{2}{5} \times 1\right)\right]$$

$$= 0.970 - 0.9508$$

$$= 0.0192$$

Entropy (S_{Rain}) = 0.970

(0.75)(0.4150) + 0.25(2.0)

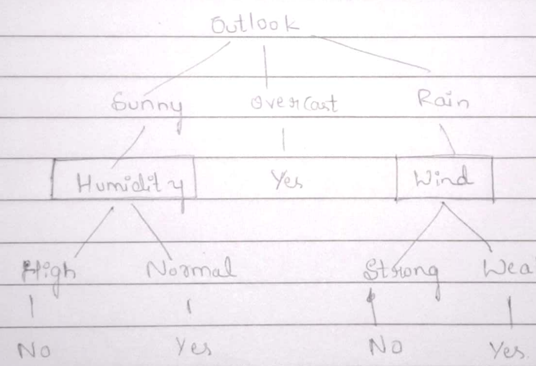
0.313 + 0.050

= 0.363

Gain (Rain, Temp) = 0.0198

Gain (Rain, Wind) = 0.970

Gain (Rain, Humidity) = 0.0198.



0.4286 (1.2224) + 0.5714 (0.8074)

= 0.5239 +

= 0.9852 = 0.4426

= 0.8571 (0.2224) + 0.1429 (2.8074)

+ 0.4012

0.5918 = 0.2959