

$A \rightarrow N-1$

Ans - 1 \rightarrow (i) Let us suppose that A and B are similar.
Then there exists a invertible 'nxn' matrix P such that $P^{-1}AP = B$.

Now, taking determinant on both sides, we get,

$$\det |P^{-1}AP| = \det |B|$$

$$\det (P^{-1}AP) = \det (B)$$

$$\det (P^{-1}) \det (A) \det (P) = \det (B) \quad \text{--- (1)}$$

$$\begin{aligned} \text{Now, as we know that } \det (M^{-1}) &= \det (M)^{-1} \\ &\Rightarrow \det (M^{-1}) = \frac{1}{\det (M)} \quad \text{--- (2)} \end{aligned}$$

\Rightarrow Replacing (2) in (1), we get,

$$\det (A) \det (P) \left(\frac{1}{\det (P)} \right) = \det (B)$$

$$\Rightarrow \det (A) \det (P) \left(\frac{1}{\det (P)} \right) = \det (B)$$

$$\Rightarrow \det (A) = \det (B)$$

(ii) Now, it is given that $A \sim B$ and from the similarity of A and B, we know that

$$\det (A) = \det (B),$$

Using these properties, we can say that,

The characteristic polynomials prove the equality of eigen values of similar matrices.

Mathematically speaking, "we have,

let $p_B(\lambda)$ and $p_A(\lambda)$ denote the characteristic polynomials of A and B .

$$p_B(\lambda) = \det(B - \lambda I)$$

$$p_A(\lambda) = \det(A - \lambda I)$$

Starting from $p_B(\lambda)$, we have

$$p_B(\lambda) = \det(B - \lambda I)$$

Replacing B by $P^{-1}AP$, we have,

$$p_B(\lambda) = \det(P^{-1}AP - \lambda I)$$

$$p_B(\lambda) = \det((P^{-1}AP - P^{-1}\lambda I P))$$

$$p_B(\lambda) = \det(P^{-1}(A - \lambda I)P)$$

$$p_B(\lambda) = \det(P^{-1}) \det(A - \lambda I) \det(P)$$

Now, since we know that $\det(P^{-1}) = \frac{1}{\det(P)}$, we have

$$p_B(\lambda) = (\det(P))^{-1} \det(A - \lambda I) \det(P)$$

$$p_B(\lambda) = \det(A - \lambda I)$$

$\Rightarrow p_B(\lambda) = p_A(\lambda) \Rightarrow$ Hence proved.

Ans) (2) (i) let $\mathbf{u} = (u_1, u_2, u_3, \dots, u_n)$

Now, we have to calculate the value of $\mathbf{u} \cdot \mathbf{u}$.

Now, by definition of dot product, we have

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + u_3 v_3 + \dots + u_n v_n$$

Now, here, it is \mathbf{u} itself, therefore we have,

$$\Rightarrow \mathbf{u} \cdot \mathbf{u} = u_1 u_1 + u_2 u_2 + u_3 u_3 + \dots + u_n u_n$$

$$\Rightarrow \mathbf{u} \cdot \mathbf{u} = (u_1)^2 + (u_2)^2 + (u_3)^2 + \dots + (u_n)^2$$

$$\Rightarrow \mathbf{u} \cdot \mathbf{u} = \sum_{i=1}^n (u_i)^2$$

Now, taking the "square root" of "square" on the right side, we have,

$$\Rightarrow \mathbf{u} \cdot \mathbf{u} = \sqrt{\sum_{i=1}^n (u_i)^2}$$

$$\Rightarrow \mathbf{u} \cdot \mathbf{u} = \|\mathbf{u}\|^2$$

(where $\|\mathbf{u}\|$ represents the 2-D norm)

Hence Proved.

Ans) (ii) $\vec{x} \rightarrow n \times n$ orthogonal matrix
 $\vec{x} \rightarrow n \times n$ vector
 $I_n \rightarrow n \times n$ Identity matrix

Now, to prove

$$\|\vec{Q}\vec{x}\| = \|\vec{x}\|$$

Taking the left hand side and squaring it, we have

$$\|\vec{Q}\vec{x}\|^2 = \|\vec{Q}\vec{x} \cdot \vec{Q}\vec{x}\|$$

Now, as we know that for a vector \vec{y} ,

$$\vec{y} \cdot \vec{y} = \vec{y}^T \vec{y},$$

Applying this property to the above equation, we have,

$$\|\vec{Q}\vec{x}\|^2 = (\vec{Q}\vec{x})^T \cdot \vec{Q}\vec{x}$$

$$\Rightarrow \|\vec{Q}\vec{x}\|^2 = \vec{Q}^T \vec{x}^T \vec{Q}\vec{x}$$

$$\Rightarrow \|\vec{Q}\vec{x}\|^2 = \vec{Q}^T \vec{Q} \vec{x}^T \vec{x}$$

Now, we know that $\vec{Q}^T \vec{Q} = I_n$

$$\Rightarrow \|\vec{Q}\vec{x}\|^2 = \vec{I}_n \vec{x}^T \vec{x}$$

$$\Rightarrow \|\vec{Q}\vec{x}\|^2 = \vec{x}^T \vec{x}$$

$$\Rightarrow \|\vec{Q}\vec{x}\|^2 = (\vec{x}^T \vec{x})$$

(from inverse transpose property)

$$\|\vec{Q}\vec{x}\|^2 = \|\vec{x}\|^2$$

\Rightarrow Hence proved.

Ans 3) The Gaussian function can be defined as given :-

$$N(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

This can be written as :-

$$N(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Now, in order to find out the maximum value for this function, we have to find out the double derivative which will give the critical points of the function and hence we can find out the maximum value.

$$\Rightarrow \frac{d}{dx} [N(x, \mu, \sigma^2)] = \frac{d}{dx} \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right]$$

Now, since we have to find derivative w.r.t x , we can treat other variables as constants.

$$\Rightarrow \frac{d}{dx} [N(x, \mu, \sigma^2)] = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{d}{dx} \left[e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right]$$

By applying the derivative rule of e^x , we have,

$$\Rightarrow \frac{d}{dx} [N(x, \mu, \sigma^2)] = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \cdot \frac{d}{dx} \left(\frac{(x-\mu)^2}{2\sigma^2} \right)$$

Want small \leftarrow

$$\Rightarrow \frac{d}{dx} [N(x, u, \sigma^2)] = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}} \frac{(-2)(x-u)}{\sigma^2}$$

$$\Rightarrow \frac{d}{dx} [N(x, u, \sigma^2)] = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}} \frac{(-x+u)}{\sigma^2}$$

Now, taking the derivative again, we have,

$$\Rightarrow \frac{d}{dx} \left[\frac{d}{dx} [N(x, u, \sigma^2)] \right] = \frac{d}{dx} \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}} \left[\frac{-x+u}{\sigma^2} \right] \right)$$

Taking out the constants, we have,

$$\Rightarrow \frac{d}{dx} \left[\frac{d}{dx} [N(x, u, \sigma^2)] \right] = \frac{1}{(\sqrt{2\pi\sigma^2})\sigma^2} \frac{d}{dx} \left(e^{-\frac{(x-u)^2}{2\sigma^2}} \left[\frac{-x+u}{\sigma^2} \right] \right)$$

By applying the multiplication rule for derivatives,

we have,

$$\frac{d}{dx} \left[\frac{d}{dx} [N(x, u, \sigma^2)] \right] = \frac{1}{(\sqrt{2\pi\sigma^2})\sigma^2} \left[\left(e^{-\frac{(x-u)^2}{2\sigma^2}} (-x+u) + \text{constant} \right) \frac{1}{\sigma^2} (x-u) \right]$$

Now, putting this $= 0$, we get the critical point as below,

$$\frac{1}{(\sqrt{2\pi\sigma^2})\sigma^2} \left[\left(e^{-\frac{(x-u)^2}{2\sigma^2}} (-x+u) + \text{constant} \right) \frac{1}{\sigma^2} (x-u) \right] = 0$$

Now, as we can see that the only factors that can be zero here are

$$(x-u) \text{ or } (x+u) \quad \text{as} \quad \frac{1}{(\sqrt{2\pi r^2})^{x+u}} \equiv \text{constant} \neq 0$$
$$e^{-b(x)} \neq 0$$

Constant $\neq 0$

$$\Rightarrow (x-u) = 0 \quad \text{or} \quad (x+u) = 0$$
$$\Rightarrow x = u \quad \text{or} \quad x = -u$$

\Rightarrow From here we can say that the maxima occurs at (u)

Hence Proved.

Ans 4) (i) When a fair coin is flipped 3 times, the sample space is {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}

H \rightarrow denotes Head

T \rightarrow denotes Tail

Now, we can see that, for finding the Expected Number of heads, we can take probability of success as getting heads.

$X \rightarrow$ Random variable indicating number of heads,
 $P(X=1) \rightarrow \frac{1}{2} \Rightarrow$ Probability of head

$P(X=0) = \frac{1}{2} \Rightarrow$ Probability of Tail

Now, seeing the sample space, we can say that

$$P(X=0) = \text{all tails} = \frac{1}{8}$$

$$P(X=1) = \text{1 head and 2 tails} = HTT, THT, TTH = \frac{3}{8}$$

\hookrightarrow This can also be written as $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2}$

$$\Rightarrow \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = \frac{3}{8}$$

$$P(X=2) = \frac{3}{8}$$

$$P(X=3) = \frac{1}{8}$$

So now, we can see that the probability of getting a head becomes as

$$P(X) = \frac{1}{2}$$

\Rightarrow Expected value becomes as $\exists 3 \times \frac{1}{2} = \frac{3}{2} = 1.5$

This can also be done by Binomial $E(X)$

where $E(X) = np$ where $n = \text{success (head)}$
 $p = \text{probability of success}$

$$E(X) = 3x$$

$$E(X) = 3 = 1.5$$

(i) for the first toss of coin, we have,

$$P(H) = \text{Probability of } H = \frac{1}{2}$$

$$P(T) = \text{Probability of } T = \frac{1}{2}$$

Now, for the second toss, we have the following outcomes in conjunction with the first toss,

$$\Rightarrow \{HH, HT, TH, TT\}$$

$$\text{Now, } P(H) = TH = \frac{1}{4}, \quad P(T) = TT = \frac{1}{4}$$

$$P(H) = TH = \text{can be rewritten as } \left(\frac{1}{2} \cdot \frac{1}{2}\right)$$

Now, for the third toss, we have the following outcomes in conjunction with the previous two tosses $\Rightarrow \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$

$$\Rightarrow \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$$

$$\Rightarrow P(H) \text{ at } n^{\text{th}} \text{ outcome} = TTH = \frac{1}{8}$$

$$\text{This can be rewritten as } \left(\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2}\right)$$

Now, by following this pattern, we have,

$$P(H) \text{ at } n^{\text{th}} \text{ outcome} = \left(\frac{1}{2}\right)^n \quad |, \quad P(T) = \left(\frac{1-p}{2}\right)^{n-1}$$



This is the probability of failure

This is the probability of success or failure

Now, as we know that the above mentioned pattern is of geometric series, therefore

$$E(H) = \sum_{i=1}^{\infty} n P(X=i)$$

$$E(H) = \sum_{i=1}^{\infty} n (i \cdot p \cdot \cancel{s^{i-1}})$$

$$E(H) = \sum_{i=1}^{\infty} n \cdot \text{sum of G.P} \quad (\text{Here sum of G.P} = \frac{1}{1-p})$$

where $n = \text{interval}$

$$\Rightarrow E(H) = \cancel{n} \cdot \sum_{i=1}^{\infty} (i \cdot \frac{1}{1-p})$$

$$\Rightarrow E(H) = (1-p) (p) \quad \text{where } p \equiv \text{probability of success}$$

$$\Rightarrow E(H) = \left(1 - \frac{1}{2}\right) \left(\frac{1}{2}\right)$$

Now, since we know that the first head terminates the experiment, therefore

$$E(X = \text{Head}) = \frac{x}{1-p} \quad \text{where } x=1, p=\frac{1}{2}$$

steps of heads
many

$$E(X = \text{Head}) = \frac{1}{1-\frac{1}{2}} = \frac{1}{\frac{1}{2}} = 2$$

\Rightarrow first Head will occur at second toss.

(iii) Let the expected Number of coin flips be 'x'.

Now there are 3 possible cases as listed below:

(a) If a tail appears on the first flip of a coin \rightarrow
 $p(x) = \frac{1}{2}$ and no. of steps required = $(x+1)$

(b) If a head appears on the first flip and a tail appears on the second flip.

$$p(x) = \frac{1}{4} \quad \text{and no. of flips} = (x+2)$$

(c) Two consecutive heads on two flips.

$$p(x) = \frac{1}{4} \quad \text{and no. of flips} = 2$$

Now, by using the general law of probability and equations, we have,

$$E(x) = \sum x p(x)$$

$$\Rightarrow X = \left(\frac{1}{2}\right) (x+1) + \left(\frac{1}{4}\right) (x+2) + \left(\frac{1}{4}\right) (2)$$

$$\Rightarrow X = \frac{1}{2}x + \frac{1}{2} + \frac{1}{4}x + \frac{1}{2} + \frac{1}{2}$$

$$\Rightarrow X = \left(\frac{1}{2}x + \frac{1}{4}x\right) + \frac{1}{2} + \frac{1}{2} + \frac{1}{2}$$

$$\Rightarrow X = \frac{(2x+x)}{4} + \frac{3}{2}$$

$$\Rightarrow X = \frac{3x}{4} + \frac{3}{2}$$

$$X - \frac{3x}{4} = \frac{3}{2} \Rightarrow \frac{x}{4} = \frac{3}{2} \Rightarrow (x=6)$$

Required Answer by $E(X)$ recursive

Ans5) Now, from definition, we know that

$$\text{Cov}[X, Y] = E[(X - E(X))(Y - E(Y))]$$

$$\text{Cov}[X, Y] = E[XY - X(EY) - (EX)Y + (EX)(EY)]$$

$$\text{Cov}[X, Y] = E[XY] - E[X]E[Y] - \cancel{E(X)E(Y)} + \cancel{E(X)E(Y)}$$

$$\text{Cov}[X, Y] = E[XY] - \cancel{E(X)E(Y)} - 0$$

$$\text{Also, } \text{Var}(X) = E[XY] - (E(X)E(Y)) - 0$$

From ① and ② we can say that,

$$\text{Var}(Z) = \text{Cov}(Z, Z)$$

Now replacing $Z = X+Y$, we have,

$$\text{Var}(X+Y) = \text{Cov}(X+Y, X+Y)$$

$$\text{Var}(X+Y) = \text{Cov}(X, X) + \text{Cov}(X, Y) + \text{Cov}(Y, X) + \text{Cov}(Y, Y)$$

$$\text{Var}(X+Y) = \text{Var}(X) + \text{Var}(Y) + 2 \text{Cov}(X, Y)$$

Hence proved.

Ans6)

A → Event A is going to some school $\Rightarrow P(A)$

B → Event B is vaguely recognising those people at the party $\Rightarrow P(B)$

$$P\left(\frac{B}{A}\right) = \frac{1}{2}$$

$$P(A) = \frac{1}{10} \quad [P(X) = 10]$$

$$P(B) = \frac{1}{5}$$

$$P\left(\frac{A}{B}\right) = \frac{P(B/A)}{P(A)}$$

$$P(B)$$

$$P\left(\frac{A}{B}\right) = \frac{\left(\frac{1}{2}\right)\left(\frac{1}{10}\right)}{\left(\frac{1}{5}\right)} = \frac{1}{20} \times 5$$

$$= \frac{1}{4} = 0.25$$

Ans 7) LOGICAL 'NOT' :-

The logical NOT function is as below:-

A	NOT A	A	NOT A
False	True	0	1
True	False	1	0

The prediction algorithm states that

$$\text{Prediction } (y) = \begin{cases} 1 & \text{if } w_1x_1 + b \geq 0 \\ 0 & \text{if } w_1x_1 + b \leq 0 \end{cases}$$

To train the perceptron for Logical NOT gate, we have,

① Initialize weight values and bias.

$$w_1x_1 + b \quad \text{where } w_1 = 1, b = -1$$

$$x_1(1) - 1$$

② Passing NOT logic table, we have,

$$\Rightarrow x_1 = 0 \Rightarrow x_1 - 1 = 0 - 1 = -1$$

↳ Not correct as $w_1x_1 + b \leq 0$,
 $y \leq 0$

③ \Rightarrow let $x_1 = 0$

$$\Rightarrow 0 + 1 = 1 \Rightarrow 1$$

$$y' = 1$$

\Rightarrow rule $w_1x_1 + b > 0$ Lecture work

(4) let $x_1 = 1$

$$w_1x_1 + b \Rightarrow 1 + 1 = 2$$

Now, $w_1x_1 + b > 0 \Rightarrow y' = 1$ but here $y = 2$

∴ Does not work

(5) Now, if we change the $w_1 = -1$, we have,

$$w_1x_1 + b \Rightarrow -1 + 1 = 0 \Rightarrow y' = 0$$

This follows as $w_1x_1 + b \leq 0 = 0$

∴ The NOT Gate can be trained as below:

$$-x_1 + 1 \text{ with } 0 \text{ value for } 1 \text{ and } 1 \text{ value for } 0$$

$$(1) = d, 1 = w, 1 = b \text{ and } 0 = 1 - (1) = x$$



In terms of Graph, we have,

• → fire

0 → NOT fire

$$\begin{aligned} x_1 &= 1 \\ w_1 &= -1 \end{aligned}$$

$$\begin{aligned} w_1 &= 1 \\ x_1 &= 0 \end{aligned}$$

LOGICAL NOR Gate :-

The logical NOR Gate is below:-

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

from the diagram, we can say that

Now, the perceptron algorithm gives us the following eqⁿ $\rightarrow w_1x_1 + w_2x_2 + b \neq 0$

① Initializing w_1 and w_2 as 1 and $b = -1$, we get,

$$x_1(1) + x_2(1) - 1 \\ \Rightarrow x_1 + x_2 - 1$$

② Putting the values from the first row of the truth table, we have,

$$x_1 = 0, x_2 = 0 \text{ in } x_1 + x_2 - 1$$

$$\Rightarrow 0 + 0 - 1 = -1$$

from the perceptron rule, we have

$$w_1x_1 + w_2x_2 + b \geq 0, \text{ then } y' = 0 \text{ But above, } y' = -1$$

\Rightarrow inconsistent

\Rightarrow So, we change $b=1$, we have,

$$0+0+1=1 \Rightarrow \text{rule works}$$

\Rightarrow Now, for Row-2 of the truth table, we have,
 $x_1=0, x_2=1$, we get,

$$0+1+1=2$$

From the perception rule, if $wx+b>0$, then $y=1$,
 \Rightarrow Incorrect now.

\Rightarrow So we will have to make changes to the input as
follows: $x_1=0, x_2=1 \Rightarrow w_1=(-1)$

\Rightarrow Using the perception rule again, we have,
 ~~$0-1+1=0$~~

\Rightarrow correct

\Rightarrow Now, for Row-3, we have, $x_1=1, x_2=0$, we get,
 $1+0+1=2$

From perception rule, we have,
 $wx+b>0 \Rightarrow y=1 \Rightarrow$ incorrect

Making changes to the values, $w_1=(-1), x_1=0, x_2=1$,

$$\Rightarrow (-1)+0+1=0 \Rightarrow \text{satisfied}$$

Now, for Row-4, we have, $x_1=1, x_2=1,$

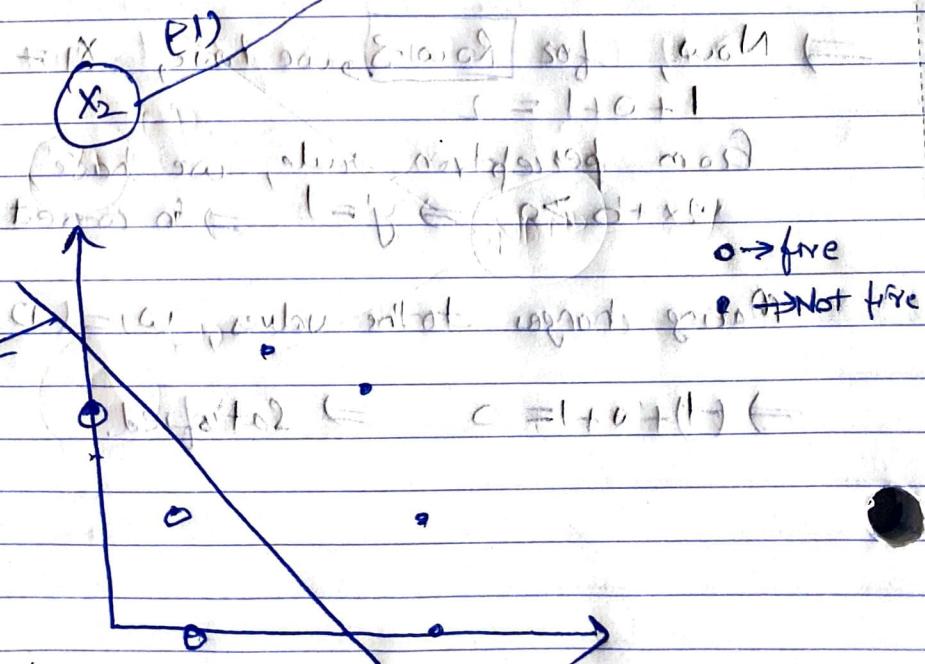
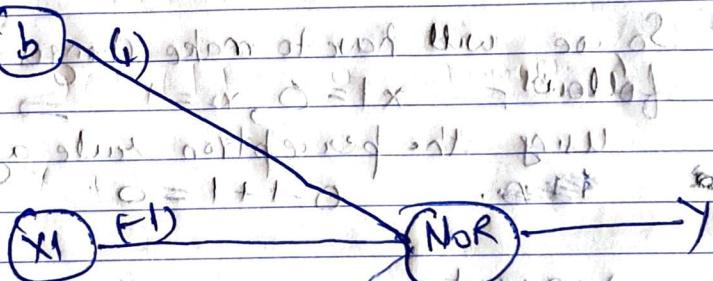
$$f(1)+E(1)+1 = (-1) \quad l = 1+0+0$$

⇒ Satisfies the perception rule.

⇒ The final perception equation for NOR gate becomes as:-

$$-x_1 - x_2 + 1 \quad c = 1+1+0$$

Depicting it graphically, we have:



LOGICAL NAND GATE

From Boolean Algebra, we know that a NAND gate can be written as:

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

From the truth table above, we can say that the NAND gate is 0 only if both inputs are 1.

Now, for Row-1 of the truth table, we have,

$$① w_1x_1 + w_2x_2 + b \rightarrow \text{perception rule}$$

Initializing w_1 and $w_2 = 1$ (and $b = -1$), we get,

$$(x_1) (0) + x_2 (1) = -1$$

Setting $x_1 = 0$ and $x_2 = 0$, we have

$$0 + 0 - 1 = -1 = y'$$

Now, from the perception rule, if $wx+b \leq 0$, then $y' = 0$

\Rightarrow This row is incorrect.

Now, updating the weights, we have,

$x_1 = 0, x_2 = 0, b = 1$, we have,

$$0 + 0 + 1 = 1 \Rightarrow \text{works for the perception rule.}$$

Now, for Row-2, we have

if $x_1=0, x_2=1$, we have overload (max 1 in million) and on step

$$0+1+1=2 \Rightarrow 0+1+1=2$$

works

Row-3, we have

$$\Rightarrow 0+1+1=2 \Rightarrow \text{works}$$

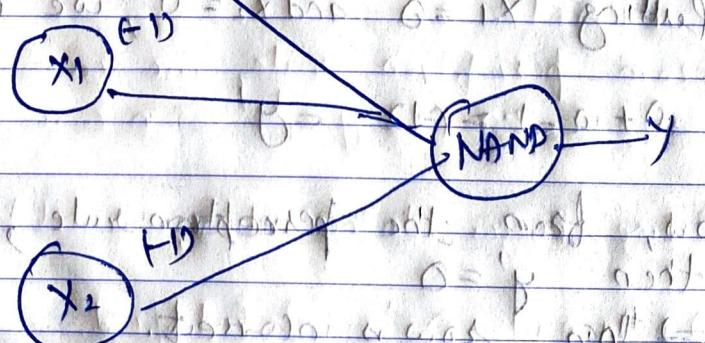
Row-4, $1+1+1=3 \Rightarrow \text{Not expected}$

$$\Rightarrow (-1)+1+2=0 \Rightarrow \text{works}$$

$$\Rightarrow -x_1 - x_2 + 2$$

$$\text{using normalizing } c = x_1W + x_2W \quad (1)$$

(b) (2)



Ans 8) Now, as we all know that there are 3 inputs, the table of inputs and outputs is given below:

A	B	C	Output
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

The Graphical Representation for this is given below:

x_0

w_0

x_1

w_1

x_2

w_2

x_3

$f(x)$

$\rightarrow y$

Now, the rules of perceptron learning algorithm states that

$$y = 1 \text{ if } \sum_{i=1}^n w_i x_i + b > 0 \quad \text{or} \quad y = 0 \text{ if } \sum_{i=1}^n w_i x_i + b \leq 0$$

and

$$\Delta w_i = \eta (y^k - t^k) x_i^k$$

let the initial weights be

$$w_0 = (-0.05)$$

$$w_1 = (-0.2)$$

$$w_2 = (0.2)$$

$$w_3 = (0.3)$$

putting these values with x^i from Row 1, we get,

$$w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 \Rightarrow (-1) \times (-0.05) + 0 + 0 + 0 \\ \Rightarrow (0.05)$$

~~Perceptron has seen that it is correct, so we will update the weights~~

~~we have to update~~

\Rightarrow for the first 4 rows, we can say that it is ' $y = t$ '.

Now, for row 5, we will have to update the weights.

$$w_0 = 0.2 - 0.25 (0-1)x' = (-0.05)$$

$$w_1 = -0.02 - 0.25 (0-1)x'_1 = 0.25$$

$$w_2 = 0.02 - 0.25 (0-1)x'_2 = 0.25$$

$$w_3 = 0.3 - 0.25 (0-1)x'_3 = 0.05$$

\Rightarrow for the next 3 rows, we can say that

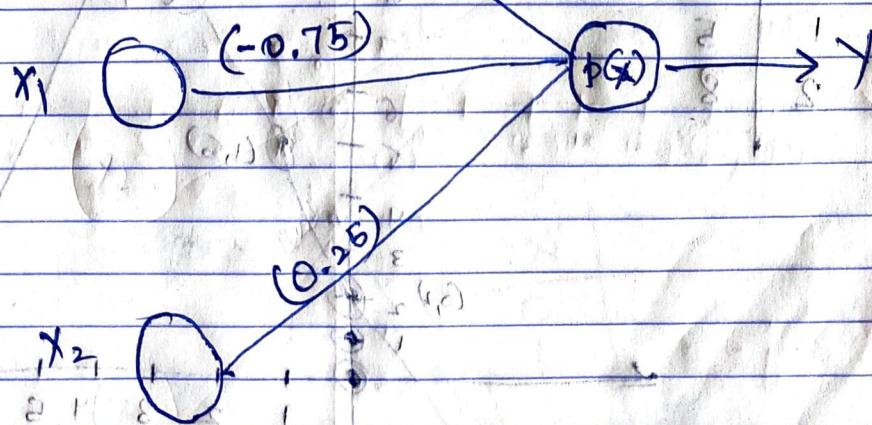
$$y \neq t$$

\Rightarrow perceptron cannot learn.

Ans 10).

	x_1	x_2	Class
	0	0	1
	0	1	1
	1	0	0
	1	1	0

(a)



(b) The equation for the perceptron from the above question becomes as :-

$$-0.75x_1 + 0.25x_2 + 0.5 = 0$$

Now, writing the above equation in the slope intercept form, we have,

$$-0.75x_1 + 0.25x_2 + 0.5 = 0$$

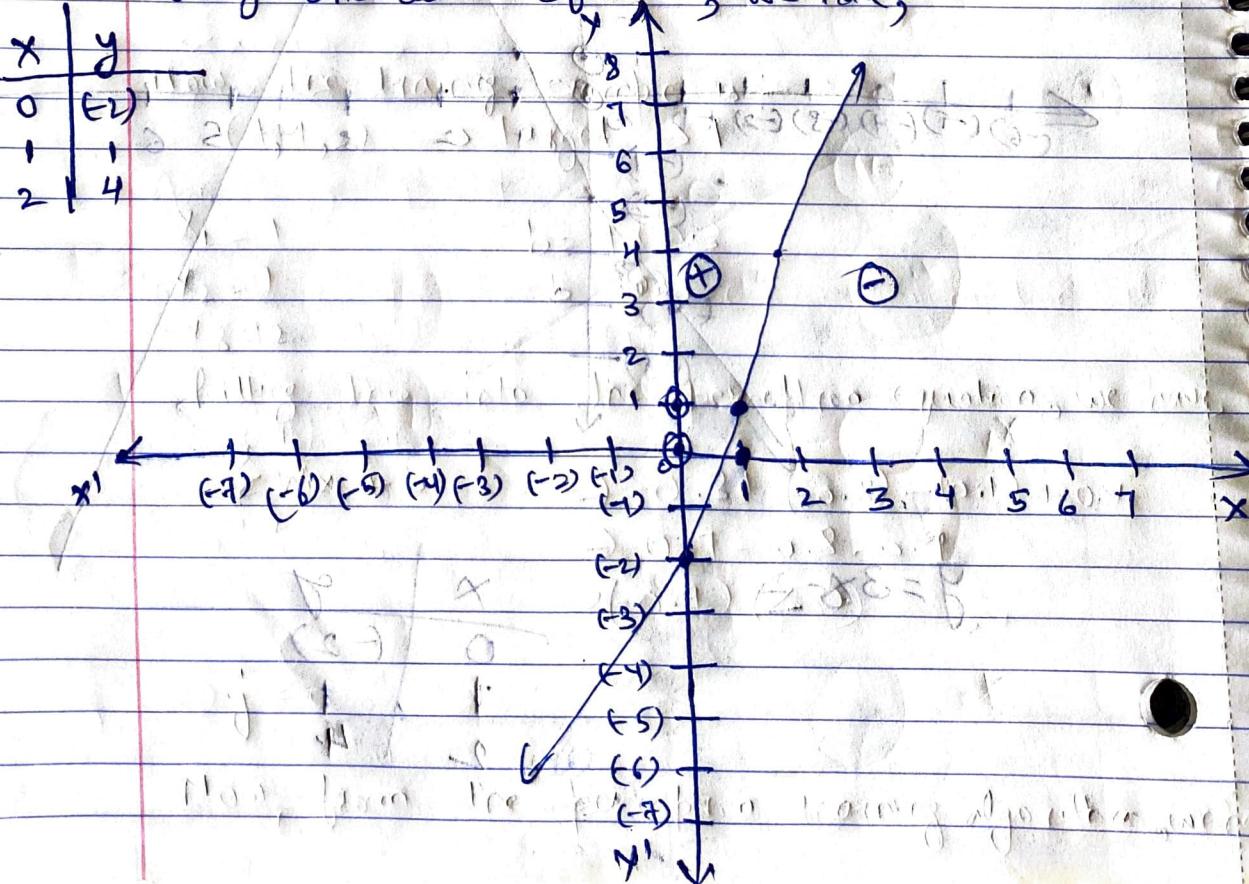
$$-0.75x_1 + 0.25x_2 = -0.5$$

$$0.25x_2 = 0.75x_1 - 0.5$$

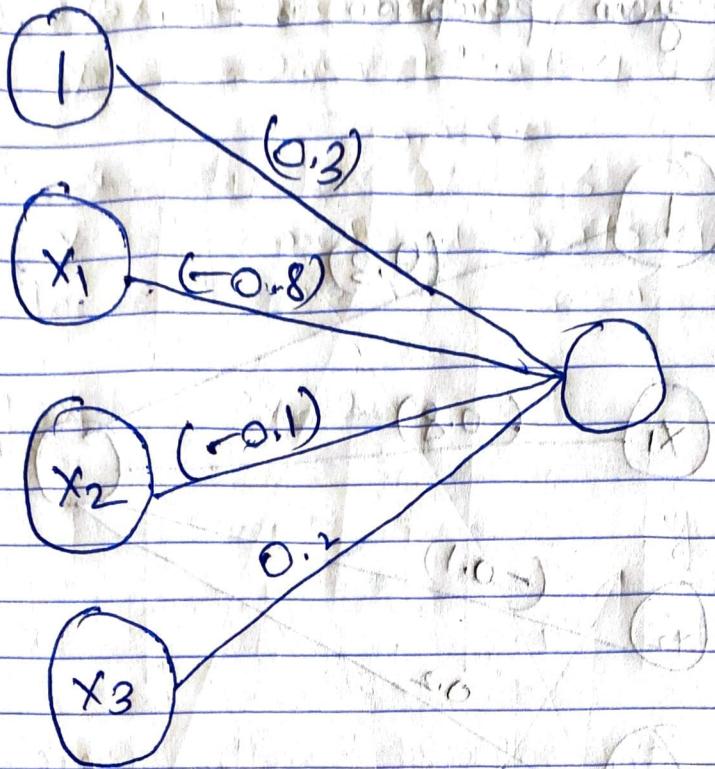
$$x_2 = \frac{3}{0.25}x_1 - \frac{0.5}{0.25}$$

$$x_2 = 3x_1 - 2$$

Plotting the above equation, we have,



(c)



$$x_1 = (-1)$$

$$x_2 = 1$$

$$x_3 = (-3)$$

Using the perceptron algorithm, we have,

$$\sum_{i=1}^n w_i x_i + b \Rightarrow \text{class 1}$$

$$\Rightarrow w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0 x_0$$

$$\Rightarrow 0.3 + 0.8 + (-0.1) + (-3) \times 0.2$$

$$\Rightarrow 0.3 + 0.8 - 0.1 - 0.6 = 0.4$$

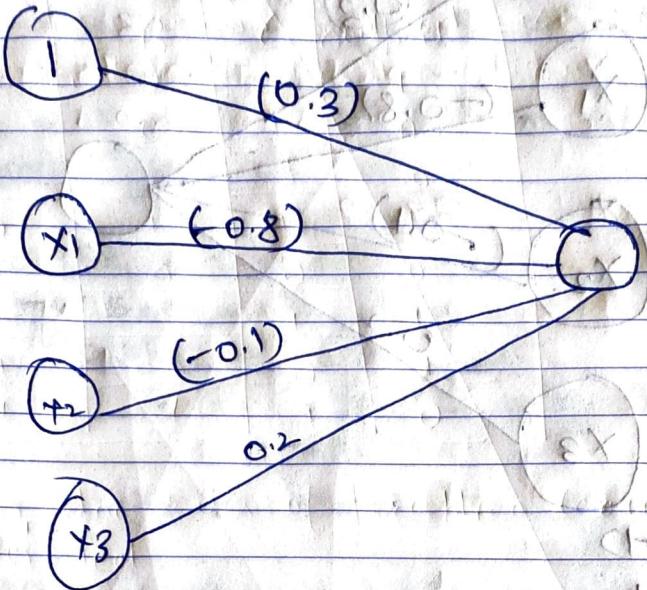
$$\Rightarrow 1.1 - 0.7$$

$$\Rightarrow 0.4 > 0 \Rightarrow y > 0$$

$$\Rightarrow \text{class 1 (by)} \quad y = \sum_{i=0}^n x_i w_i + b \quad \begin{cases} = 1, & y > 0 \\ = 0, & y \leq 0 \end{cases}$$

using multiclass general model, we'll need 401

(d) The given perception is :-



Now, The training example is :-
 $(1, 1, 2) \rightarrow \text{target} \rightarrow 1$

$$x_1 = 1$$

$$b = 1 \times 0.3$$

$$x_2 = 1$$

$$b = 0.3$$

$$x_3 = 2$$

$$b = 0.3 + (-0.8) + 0.2$$

Putting this into the perception equation, we have,

$$\sum_{i=1}^3 w_i x_i + b \Rightarrow 0.3 + (-0.8) - (0.1) + 0.2$$

$$\Rightarrow 0.7 - 0.8 - 0.1$$

$$\Rightarrow (-0.2) \quad (10)$$

$$\Rightarrow y < 0$$

Now, from the perception learning algorithm, we have

$$\sum_{i=1}^n w_i x_i + b = \begin{cases} 1 & \text{if } w_i x_i + b > 0 \\ 0 & \text{if } w_i x_i + b \leq 0 \end{cases}$$

Now the target was 1 but we got 0.
 ⇒ weights must be corrected & another iteration will be required.

→ Now, the training example's record is:

$$(1, -2, -2) \rightarrow \text{target} \rightarrow 0$$

$$\Rightarrow x_1 = 1 \quad b = 1 \times 0.3$$

$$x_2 = -2 \quad b = 0.3$$

$$x_3 = -2$$

Putting these in the perceptron eqⁿ, we have

$$\sum_{i=1}^n w_i x_i + b \Rightarrow 0.3 + (-0.8) + 0.2 - 0.4$$

$$\Rightarrow 0.3 + 0.2 - 0.8 - 0.4$$

$$\Rightarrow 0.5 - 0.12 \Rightarrow 0.38 \Rightarrow y' < 0$$

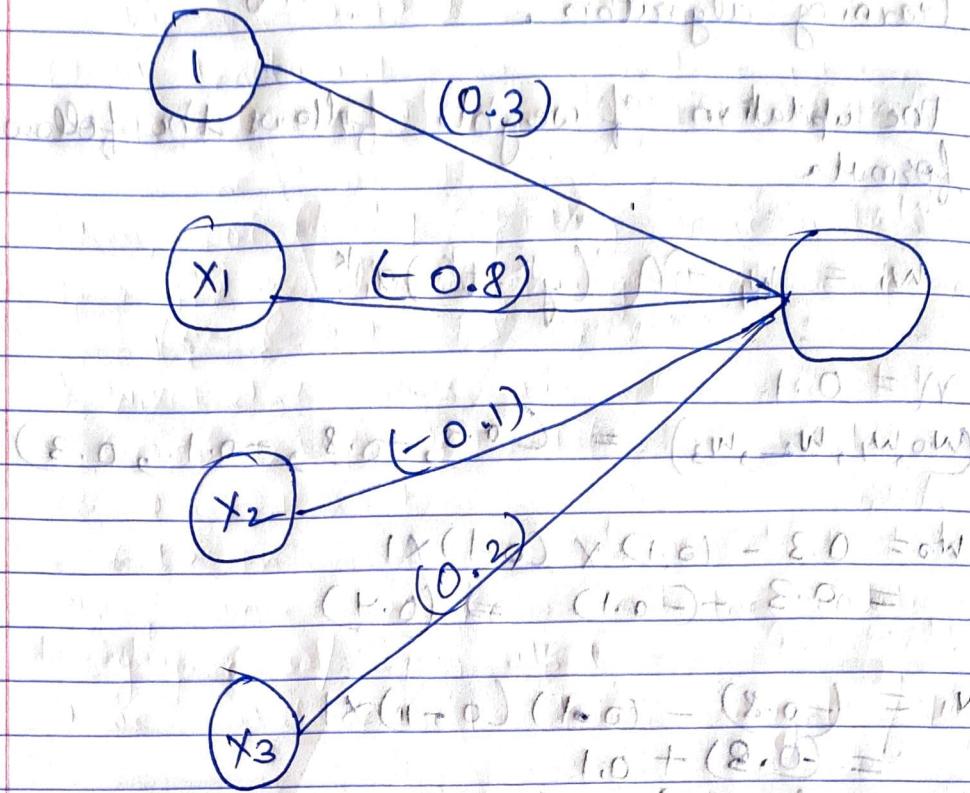
$$\Rightarrow \sum_{i=1}^n w_i x_i + b = 0 \text{ for target } 0.$$

(e) Now, as we can see from the above question, we can see that for first training set, the perceptron wrongly predicts and for the second training set, the perceptron correctly predicts the target.

Therefore, we can say that the accuracy is 50%.

$$\text{Correct } \% = \frac{1}{2} \times 100 = 50\%$$

(Q) The perception given is incorrect.



→ FIRST TRAINING EXAMPLE ↳

$$x_1 = 1 \quad b = 1 \times 0.3 \quad \text{target} = 1$$

$$x_2 = 1 \quad -b = 0.3$$

$$x_3 = 2 \quad w_3(1-0)(1.0) = 0.2 = 0.2$$

Now, applying this in the perception equation, we have,

$$\sum_{i=1}^n w_i x_i + b \Rightarrow w_1 x_1 + w_2 x_2 + w_3 x_3 + b \Rightarrow (-0.8) + (0.1) + 0.4 + 0.3 = 0.2 \quad (ii)$$

$$\Rightarrow (-0.9) + (0.7) = 0.2$$

$$\Rightarrow (-0.2) < 0$$

$\Rightarrow y' < 0 \Rightarrow \text{incorrect.}$

Now since the target is wrongly predicted, we will have to update the weights using Perceptron learning algorithm.

The updation of weights follows the following formula

$$w_i = w_i + \eta (y_k - t_k) x_i^k$$

$$\eta = 0.1$$

$$(w_0, w_1, w_2, w_3) = (0.3, -0.8, -0.1, 0.3)$$

$$\begin{aligned} w_0 &= 0.3 - (0.1) \times (0-1) \times 1 \\ &= 0.3 - (-0.1) \Rightarrow (0.4) \end{aligned}$$

$$\begin{aligned} w_1 &= (-0.8) - (0.1) (0-1) \times 1 \\ &= (-0.8) + 0.1 \\ &= (-0.7) \end{aligned}$$

$$w_2 = (-0.1) - (0.1) (0-1) \times 1$$

$$w_3 = 0.2 - (0.1) (0-1) \times 2$$

$$\text{new weights } \Rightarrow (0.4, -0.7, 0, 0.4)$$

SECOND TRAINING EXAMPLE

(ii) Now, for the second case, we will take the new weights as below:

$$x_1 = 1$$

$$x_2 = (-2)$$

$$x_3 = (-2)$$

$$(w_0, w_1, w_2, w_3) = (0.4, (0.7), 0, 0.43)$$

Again, using the formula used in (1) part,

$$y = (1 \times 0.4) + (1 \times 0.7) + (-2 \times 0) + (-2 \times 0.4)$$

$$y = 0.4 - 0.7 + 0 - 0.8$$

$$y = (-1) + (0.1)$$

$$y = (-1.1) < 0$$

$$\text{border} \Rightarrow y = 0$$

weights after epoch are $\{0.4, 0.7, 0, 0.4\}$

⑨ Since the new weights after one epoch of training worked for the second training data

\Rightarrow Accuracy = 100%.

Ans 11) In the perceptron learning algorithm, we have the variables as weights and the perceptron should learn to improve the accuracy by updating the weights in each iteration (epochs).

This updating of the weights will mean that the perceptron is learning with each passing input set in each iteration.

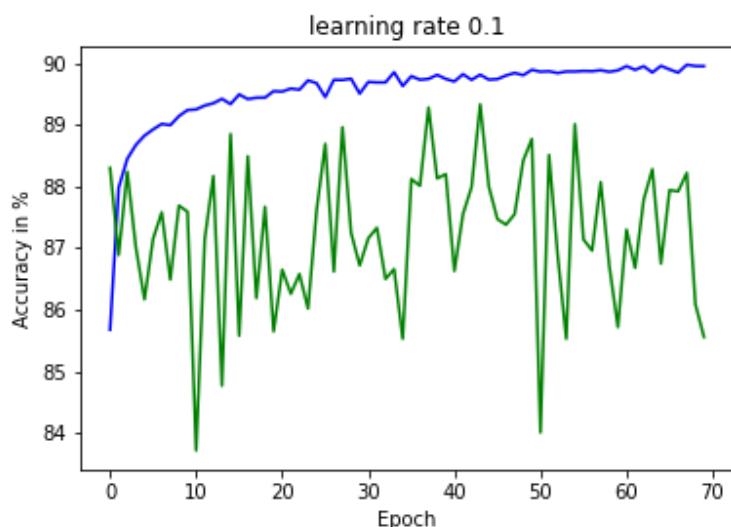
Now, in order to achieve good accuracy, we have in this experiment, used three learning rates as 0.1, 0.01, 0.001.

Experiment-1: -

In this experiment, I have used learning rate as 0.1. Here, as the initial weights are used with the input data sets, the perceptron starts learning. With each passing epoch, the accuracy starts increasing indicating that the perceptron is making decisions as per the target more accurately.

The more correctly the perceptron is predicting the outcomes, the more the accuracy increases.

In this particular experiment, the accuracy percentage plateaus at 90% for 70 epochs which can be seen as per the graph below.



This Experiment also gives us a confusion matrix which is plotted below

CONFUSION MATRIX OF TRAIN SET : learning rate 0.1										
[[5689 3 31 24 11 60 41 10 32 22]]										
[1 6458 55 23 23 23 14 19 100 26]										
[36 59 5304 168 35 52 88 67 134 11]										
[16 36 148 5321 9 242 11 57 216 85]										
[12 5 57 10 5311 50 50 59 38 253]										
[71 36 41 261 20 4588 77 11 247 64]										
[34 7 76 22 55 92 5580 3 44 3]										
[9 26 62 63 48 18 3 5714 26 298]										
[43 105 161 177 60 236 51 22 4905 87]										
[12 7 23 62 270 60 3 303 109 5100]]										
CONFUSION MATRIX OF TEST SET : learning rate 0.1										
[[961 0 15 4 4 12 18 3 14 10]]										
[0 1122 38 1 3 3 4 11 26 3]										
[1 2 736 8 3 0 7 19 7 1]										
[2 3 162 933 7 56 2 29 194 18]										
[0 0 6 2 778 5 4 10 11 4]										
[8 3 19 36 2 777 53 2 105 7]										
[2 3 12 1 16 8 864 0 8 1]										
[3 0 6 10 1 3 0 859 4 8]										
[1 1 27 5 12 17 5 4 570 1]										
[2 1 11 10 156 11 1 91 35 956]]										

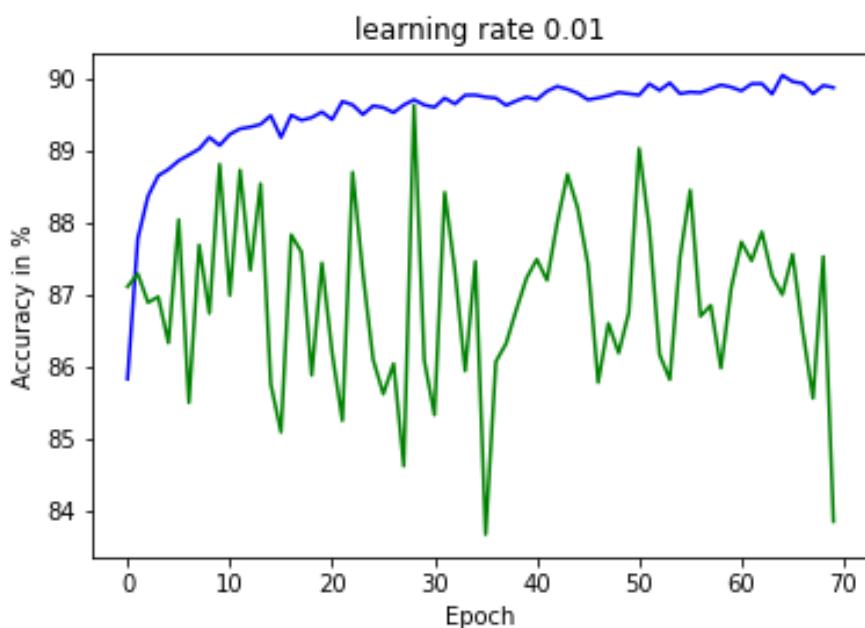
As we know that the accuracy is above 90% but is not near to 100%, the confusion matrix is diagonally dominant but not completely diagonally dependent.

2) Experiment-2

In this experiment, I have used learning rate as 0.01. Here, as the initial weights are used with the input data sets, the perceptron starts learning. With each passing epoch, the accuracy starts increasing indicating that the perceptron is making decisions as per the target more accurately.

The more correctly the perceptron is predicting the outcomes, the more the accuracy increases.

In this particular experiment, the accuracy percentage plateaus at 90% for 70 epochs which can be seen as per the graph below.



The confusion matrix for this experiment is given below: -

CONFUSION MATRIX OF TRAIN SET : learning rate 0.01

```
[[ 5689    1   34   26   15   56   35   11   35   22]
 [    1 6446   64   22   20   21   16   20 108   25]
 [  38   74 5284 177   40   53   90   62 133   12]
 [ 16   39 132 5315   10 231     6   58 235   86]
 [ 12    6   57     9 5332   55   47   53   32 241]
 [ 62   28   39 251    16 4585   94   17 258   69]
 [ 38    5   82   17   49   96 5579     6   43   2]
 [   8   25   63   58   38   18     3 5723   23 302]
 [ 42 110 178 184    57 240    45   21 4882   94]
 [ 17    8   25   72 265    66     3 294 102 5096]]
```

CONFUSION MATRIX OF TEST SET : learning rate 0.01

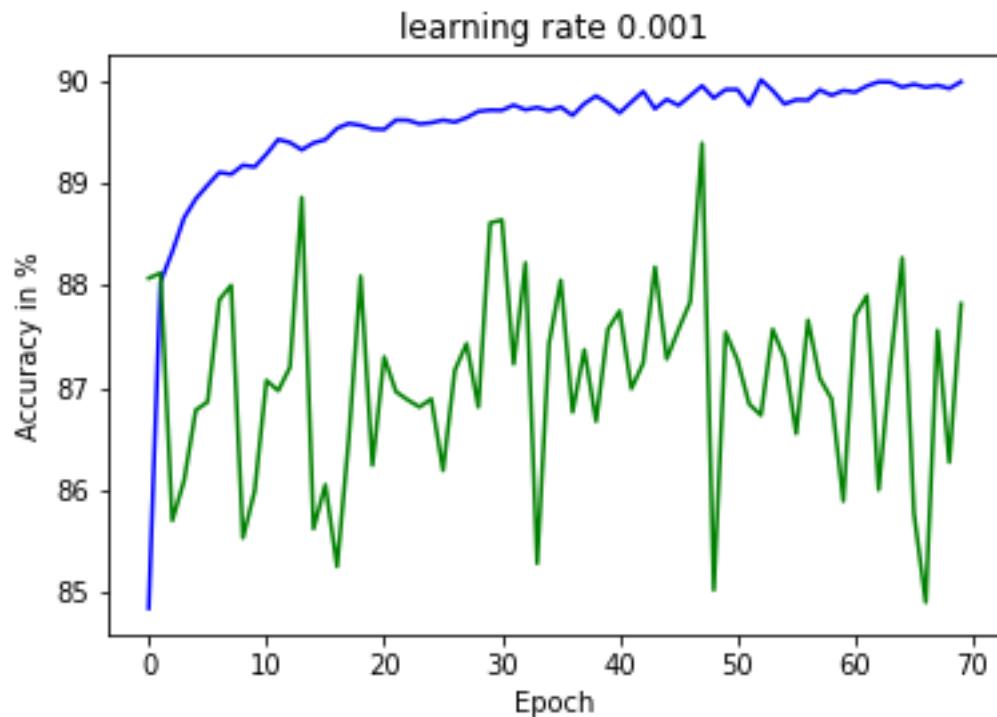
```
[[ 963    0   11    3    3   15   24    3    8    9]
 [    0 1116   23    1    1    6    4   12   20    6]
 [    3    1 734     8    5    0   19   18    4    1]
 [    5    9 214 975   11 173     7   65 273   30]
 [    0    1    9    2 789   12    5   10   10    5]
 [    4    2    3    5    0 623   41    2   19    3]
 [    1    3   10    0    8    6 854     0    6    1]
 [    2    1    5    7    2    4     0 780     2    3]
 [    2    2   16    4   15   35     4    6 603     2]
 [    0    0    7    5 148   18     0 132    29 949]]
```

2) Experiment-2

In this experiment, I have used learning rate as 0.001. Here, as the initial weights are used with the input data sets, the perceptron starts learning. With each passing epoch, the accuracy starts increasing indicating that the perceptron is making decisions as per the target more accurately.

The more correctly the perceptron is predicting the outcomes, the more the accuracy increases.

In this particular experiment, the accuracy percentage plateaus at 90% for 70 epochs which can be seen as per the graph below.



The confusion matrix for this experiment is given below:-

CONFUSION MATRIX OF TRAIN SET : learning rate 0.001

```
[[5678    1   34   25   18   53   40    9   35   26]
 [  2 6444    68   25   27   21   12   19 105   22]
 [ 33   64 5305 169   35   56   82   60 136   19]
 [ 15   36 134 5334 13 234    4   52 220   86]
 [ 13    7   59   10 5303 53   54   47   40 257]
 [ 75   35   34 264   17 4589   96   14 237   61]
 [ 37    5   76   21   54   95 5584    4   44   2]
 [ 10   25   64   49   41   16    3 5736   21 299]
 [ 43 114 157 163   58 241   39   31 4910   90]
 [ 17   11   27   71 276   63    4 293 103 5087]]
```

CONFUSION MATRIX OF TEST SET : learning rate 0.001

```
[[ 967    0   23   15    4   28   20    3   20   10]
 [  0 1123   39    1    3    7   5 16 42   8]
 [  2   3 850   27    3    4   8 25 21   1]
 [  1   1 60 889    4   32   1 15 95 10]
 [  1   0 10    4 883   14   6 16 15 18]
 [  2   1   5 36    0 730   9   2 26   5]
 [  2   4 19    5 21   30 907   0 22   1]
 [  3   1   5 11    1   5   1 871   8   9]
 [  2   2 14    9   5 25   1   4 690   5]
 [  0   0   7 13   58 17   0 76 35 942]]
```