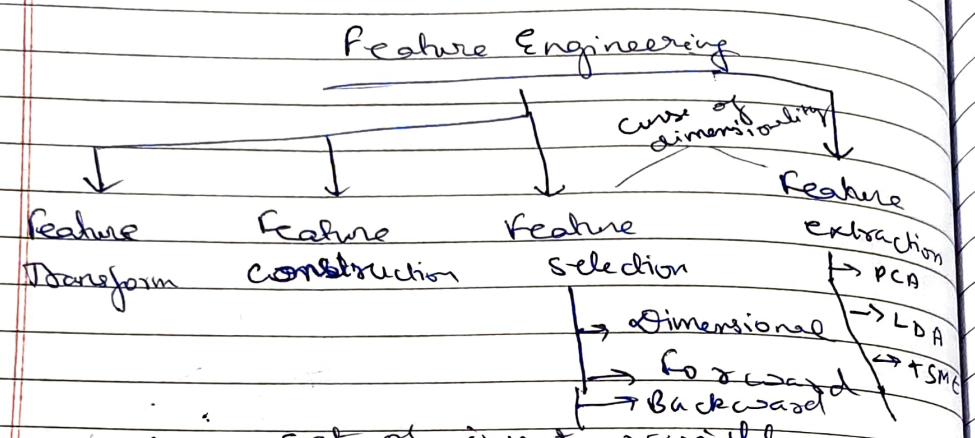
ML

Date :
Page :

Cause of dimensionality Reduction:-



n = set of input variables

1. Performance → Select those attribute that effect it.

2. Visualization → set of input ↓, more the visualization.

$f_1\ f_2\ ----\ f_n$

↓

$F_1\ F_2\ F_3$

new features built on the basis of previous ones.

Mean squared error: $\frac{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}{n}$

1. $F_i = \phi$ $\quad f = \infty$

2. Select feature one by one.
Select F_i, check whether it reduces the error or not.

3. If error ↓, add it to set of features.

1. Set $F_0 = \phi$ & $E(F_n) = \infty$

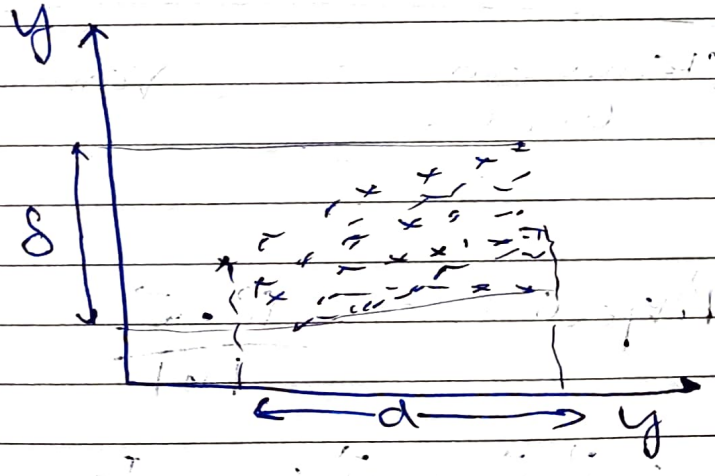
2. for $i = 0$, repeat until $E(F_{i+1}) > E(F_i)$

(a) $F_i \cup \{x_j\}$
Calc $E(F_i \cup \{x_j\})$

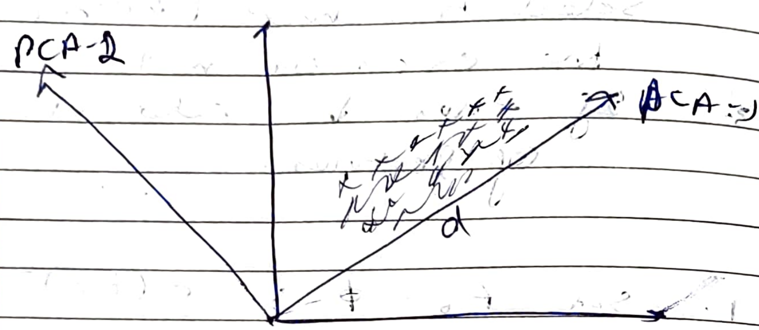
(b) $m = \min_j E(F_i \cup \{x_j\})$

(c) $F(i+1) = F_i \cup (x_j)$

PCA → Principle Component Analysis :-



linearly separable data → both have the same spread,



$$\text{mean} = \frac{-5+5+10}{3}$$

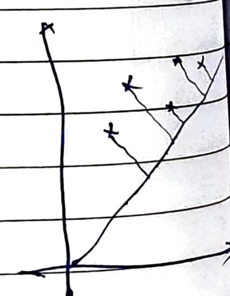
$$\text{Variance} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}$$

Variance zayda → data points dur dur hai

Take those features variance jiski zayda

$$\text{Standard deviation} = \sqrt{\text{Variance}}$$

$$\text{Projection} = \frac{\vec{u}\cdot\vec{x}}{|u|}$$



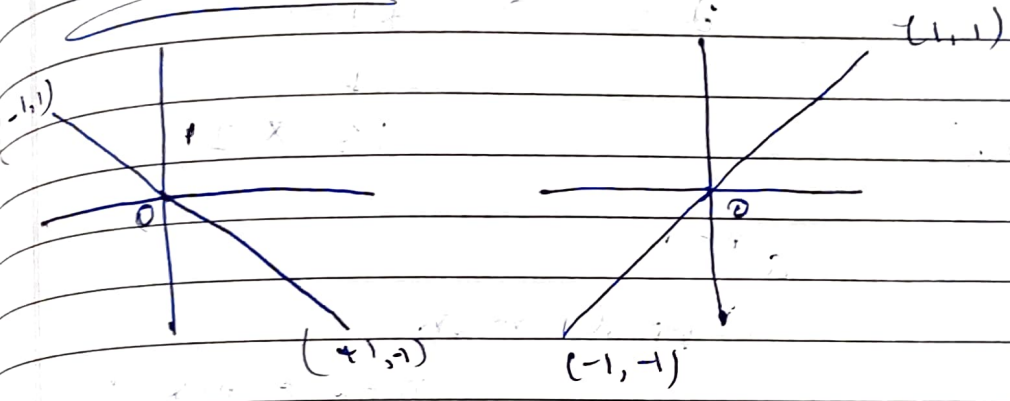
$$= \vec{u}\cdot\vec{x} = \vec{u}\cdot x = u^T x$$

$$[U^T x] = [U^T x_1] \ldots [U^T x_n]$$

$$\text{Variance} = \frac{\sum_{i=1}^{n} (U^T x_i - U^T x_m)^2}{n}$$

$U^T x_m$ → projection of mean

Co-variance :



$$\text{Covariance} = \frac{(-1*1) + 0 + (1*-1)}{3}$$

$$= \frac{-1-1}{3} = -\frac{2}{3}$$

Multiply the matrix and divide by 3

Covariance

	x_1	x_2	x_3
x_1	$Cov(x_1,x_1)$	$Cov(x_1,x_2)$	$Cov(x_1,x_3)$
x_2	$Cov(x_2,x_1)$	$Cov(x_2,x_2)$	$x_{2,3}$
x_3	$Cov(x_3,x_2)$	$Cov(x_3,x_1)$	$Cov(x_3,x_3)$

ML

for non-linear separated data, we cannot apply PCA to it.

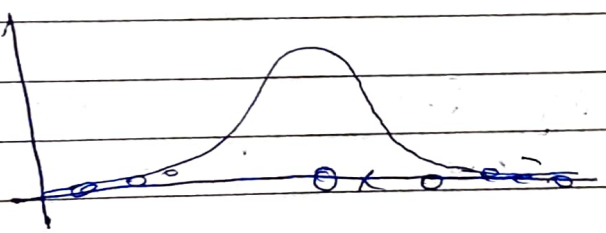
LDA → Supervised Learning
PCA → unsupervised Learning

t-SNE :- for high dimensional data to 2d → 4t is applied in those area where we can't use PCA

t- distributed Stochastic Neighbour Embedding

Score / Sum of all score → the more bigger the ratio, the more it is closer to the point.

t-distribute, the tail is a bit longer than the normal distributed. And the points also lie on the tail.



Date :
Page :

Check whether x_j point is neighbours of x_i or not. for this we use the formula:-

$$P_{j/i} = \frac{\exp(1 - |x_i - x_j|^2 / 2\sigma_i^2}{\sum_{k \neq p} \exp(1 - |x_i - x_j|)^2 2\sigma_i^2}$$

$$a_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|y_k - y_l\|^2)^{-1}}$$

This is for the power of the denominator

$$\sum P_{ij} = 1 \quad \text{for all } i$$

Loss given by KL divergence

KL → Kullback Leibler

$$KL(P \| Q) = \sum_{i \neq j} P_{ji} \log \frac{P_{ji}}{Q_{ji}}$$

If P_{ji} is more which near P_j is the neighbour of P_i

Loss needs to be less for that Q_{ji} should be ~~minimum~~ maximum.

P_{ji} → mi
therefore lo
minimum.

Stack Quest

M.L

Logis

If the points are
no shifting
Bt,
the eqn of lin
by adding o
or adding m
or subtracti

$Ax_1 + B$

$Ax_1 + Bx_2$

$w_1 x'_1 + w$

$y =$

$y_1 =$

P_{ji} → minimum

Therefore log should be minimum.

Stack Quest → Check the video links

M.L

Logistic Regression

If the points are correctly classified, no shifting is done.

But, if they are wrongly classified, the "c" of lines are changed either by adding or subtracting the co-ordinates. On adding the line shifts upwards, on subtraction, it goes below.

$$Ax_1 + Bx_2 + C = 0$$

$$Ax_1 + Bx_2 + Cx_3 + Dx_4 + Ex_5 + C_0 = 0$$

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + x_4 x_4 + w_5 x_5 = 0$$

$$y = w_0 + w_1 x_1$$

$$y = w_0 + w_1 x_1 + w_2 x_2$$

PJI → minimum

therefore log should be minimum.

StackQuest → Checks the video links

ML

Logistic Regression

If the points are correctly classified, no shifting is done.

But, if they are wrongly classified, the "eq" of lines are changed either by adding or subtracting the co-ordinates. On adding the line shifts upwards, on subtraction, it goes below.

$$Ax_1 + Bx_2 + C = 0$$

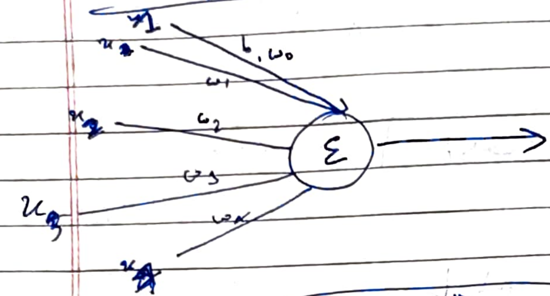
$$Ax_1 + Bx_2 + Cx_3 + Dx_4 + Ex_5 + \omega = 0$$

$$\omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + x_4 x_4 + \omega_5 x_5 = 0$$

$$y = \omega_0 + \omega_1 x_1$$

$$y = \omega_0 + \omega_1 x_1 + \omega_2 x_2$$

Perceptron



$$y = \sum_{i=1}^{n} w_i x_i + b$$

$y = w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_4$

↓

$1 \cdot b$

$= b + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots$

$$y = \sum_{i=1}^{n} w_i x_i + b$$

extra column added →

x_0	x_1	x_2	y
1	..	. .	0 Yes
1	-	.	1 No
1	-	-	1 No

$w_{new} = w_{old} -$ Gra

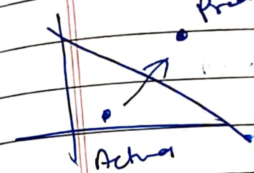
↳ $2x + 3y + 5$
$7x + 5y + 6$

Gradient descent =

Program :-

for i in range (
randomly select
if $x_i \in N$

Case 1:



If it should be in negat

Thus A, B, C

$w_{new} =$

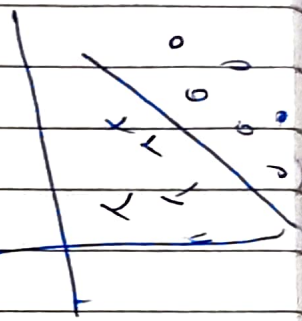
$w_{new} = w_{old} -$ Gradient learning rate

$2x + 3y + 5 = 0$
$7x + 5y + 6 = 0$

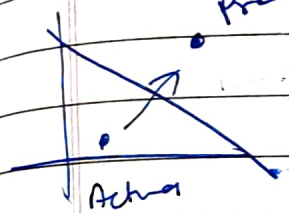
Gradient descent = new value − old value × learning rate

Program :-

for i in range (epochs)
random select point = 0
if $x_i \in N$ and $\sum w_i x_i > 0$



Case 1:



it should be in negative

$y > 0$
↓
this is positive in actual

Thus A, B, C needs to be changed

$w_{new} = w_{old} - \eta x_i$

↓
learning rate

Case 2 :-

If $x_i \in P$ & $\sum_{i=1}^{n} w_i x_i < 0$

↓ in positive region actual

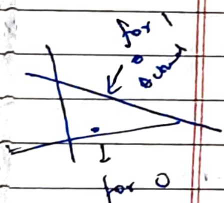
$y < 0$ ↑ but wrong out to be in negative region

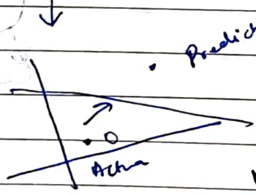
thus to make it positive we add

$w_{new} = w_{old} + \eta x_i$

$$w_{new} = w_{old} + \eta (y_i - \hat{y}_i) x_i$$

	Actual	Predicted
	y_i	$\hat{y}_i$
$w_{new} = w_{old}$	1	1
$w_{new} = w_{old}$	0	0
	1	0
	0	1





$w_{new} = w_{old} - \eta x_i$

$w_{new} = w_{old} + \eta x_i$

$\hat{y} = \sum_{i=0}^{n} w_i$ …

$\hat{y} < 0 \rightarrow$

$\therefore \hat{y} > 0$

desmos.c…

maximum likelihood

Cross entropy or log loss function → $ML = - y_1 \log($…

$y_1 = 1$ …

for $y_1 = 0$

$\log(ab) =$ …

G → green

B → Black

$$\hat{y} = \sum_{i=0}^{n} w_i x_i$$

$\hat{y} < 0 \rightarrow$ below the -ve region

$\hat{y} > 0 \rightarrow$ in the region.

desmos.com

maximum likelihood

loss entropy or log loss function

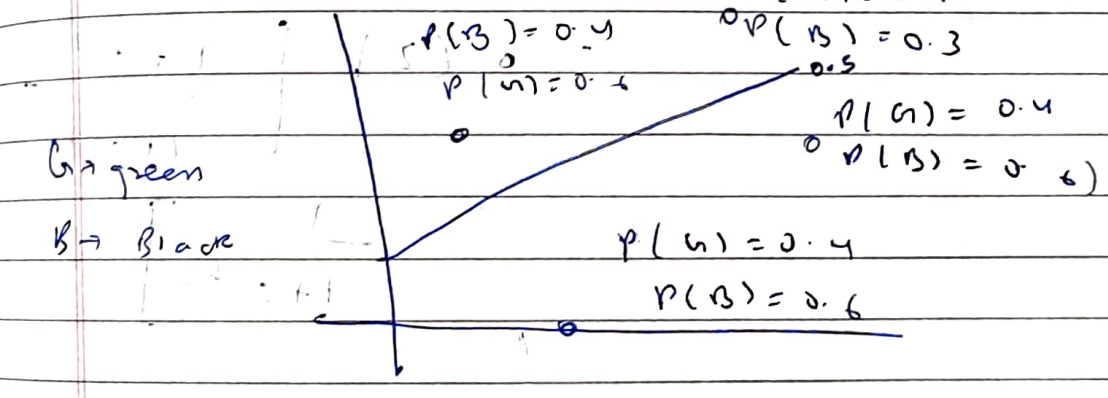
$$\text{NL} = -y_i \log(\hat{y_i}) - (1-y_i) \log(1-\hat{y_i})$$

$y_i = 1 \quad = -\log(0.7)$

for $y_i = 0 \quad = -\log(1-0.6)$

$= -\log 0.4$

$\log(ab) = \log(a) + \log(b)$



3 mod 4

Date :
Page :

NS

1. RSA
2. Knapsack
3. ECC & Diffie Hellman
4. Rabin Cryptography
5. Elgamal
6. Message Digest (MD5)

ML

$$\text{Likelihood} = L = \sum_{i=1}^{n} [-y_i (\log(\hat{y}_i)) - (1-y_i) \log(1-\hat{y}_i)]$$

$$L = \frac{1}{N} \sum_{i=1}^{N} [-y_i \log(y_i) - (1-y_i) \log(1-\hat{y}_i)]$$

(N = number of)

$$\sigma'(z) = \sigma(z) \left[\frac{1+e^{-z}-1}{1+e^{-z}} \right]$$

$$= \sigma(z) \left[\frac{1+e^{-z}}{1+e^{-z}} - \frac{1}{1+e^{-z}} \right]$$

ML

$$L = \frac{-1}{n} \sum_{i=1}^{n} y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i)$$

Loss function

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma'(z) = \sigma(z)[1-(\sigma z$$

Cost function → is a total loss

$$y_{predicted} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ \vdots & & & & \\ x_{m1} & x_{m2} & x_{m3} & & x_{mn} \end{bmatrix}$$

$$\times \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix}$$

$$\hat{y}_1 = \sigma(x_{11}w_1 + x_{12}w_2 + x_{13}w_n \cdots x_{1n}w_n)$$

$$\hat{y}_2 = \sigma(x_{21}w_1 + x_{22}w_2 + x_{23}w_3 \cdots x_{2n}w_n)$$

$$\hat{Y} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

σ_1

$z =$

Taking

Date : __/__/____
Page : _____

$$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_n \end{bmatrix} =$$

$$\sigma_1 \left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & & x_{2n} \\ \vdots & & & & \\ x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} \right)$$

$$z = \sum_{i=1}^{n} \sigma(w_i x_i)$$

$$\hat{Y} = \sigma(X w)$$

Taking $\sum_{i=1}^{n} y_i \log \hat{y}_i = y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + y_3 \log \hat{y}_3 + \cdots y_n \log \hat{y}_n$

$$= \begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_n \end{bmatrix} \begin{bmatrix} \log \hat{y}_1 \\ \log \hat{y}_2 \\ \vdots \\ \log \hat{y}_n \end{bmatrix}$$

$$= \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix} \log \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix}$$

$$= Y \log \hat{Y}$$

Taking $\sum_{i=1}^{n} (1-y_i) \log(1-\hat{y}_i)$

$$= (1-Y) \log(1-\sigma(X\omega))$$

$$L = \frac{1}{m} \left[Y \log \hat{Y} + (1-Y) \log(1-\sigma(X\omega)) \right]$$

differentiating

$$\frac{\partial L}{\partial \omega} = \frac{\partial L}{\partial}$$

$\omega_{new} =$

$\omega_1' =$

$\frac{\partial L}{\partial \omega} =$

$=$

Differentiating wrt "w"

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial w_1} + \frac{\partial L}{\partial w_2} + \frac{\partial L}{\partial w_3} + \cdots + \frac{\partial L}{\partial w_n} + \cdots$$

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w_{old}}$$

$$w_1' = w_1 - \eta \frac{\partial L}{\partial w_1}$$

$$\frac{\partial L}{\partial w} = \frac{\partial (2 \log \hat{y})}{\partial w} = \frac{\partial \log \hat{y}}{\partial w} \lambda$$

$$= \frac{\lambda}{\hat{y}} \frac{\partial (\sigma(wx))}{\partial w}$$

$$= \frac{\lambda}{\hat{y}} \sigma(wx)[1 - \sigma(wx)] \frac{\partial wx}{\partial w}$$

$$= \frac{\lambda}{\hat{y}} \hat{y}(1-\hat{y}) x = \lambda(1-\hat{y}) x$$

$$\frac{\partial L}{\partial w} = y - (1-\hat{y})x$$

This is the differentiation part of $y \log \hat{y}$

$$\frac{\partial (1-y)\log(1-\sigma(xw))}{\partial w}$$

$$= \frac{\partial (1-y)\log(1-\hat{y})}{\partial w}$$

$$= \frac{(1-y)}{(1-\hat{y})} \times \frac{\partial (1-\hat{y})}{\partial w}$$

$$= \frac{(1-y)}{(1-\hat{y})} \times -1 \left(\frac{\partial \sigma(xw)}{\partial w}\right)$$

$$= -\frac{(1-y)}{(1-\hat{y})} \times \sigma(xw)[1-\sigma(xw)] \times \frac{dxw}{dw}$$

$$= -\frac{(1-y)}{(1-\hat{y})} \times \hat{y}[1-\hat{y}] \times x$$

$$= -\hat{y}(1-y)x$$

$$\frac{\partial L}{\partial w} = \frac{-1}{m}$$

$$= -\frac{1}{}$$

$$= -$$

w_{new}

$\therefore w_{new}$

Softma

repres

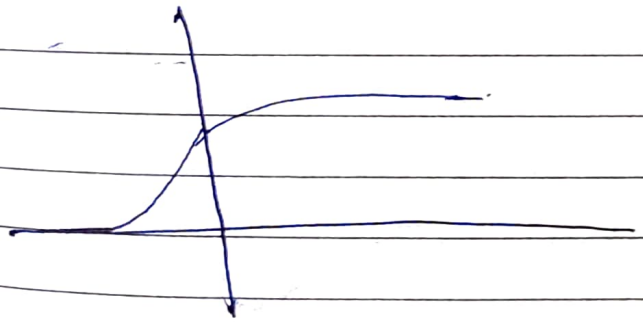
$$\frac{\partial L}{\partial w} = \frac{-1}{m}\left[y(1-\hat{y})x - \hat{y}(1-y)x \right]$$

$$= -\frac{1}{m}\left[y - y\hat{y} - \hat{y} + \hat{y}y \right]x$$

$$= -\frac{1}{m}\left[y - \hat{y} \right]x$$

$$w_{new} = w_{old} - \eta\left(\frac{y-\hat{y}}{m}\right)x$$

$$\therefore w_{new} = w_{old} - \eta\frac{\partial L}{\partial w}$$



Softmax Regression / Multinomial Logistic regression.

M L

Date : __/__/____
Page : _____

Softmax Regression :-

$$L = \frac{1}{n} \sum_{i=1}^{m} y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)$$

$$\frac{\partial L}{\partial w} = -\frac{1}{m} (Y - \hat{Y}) x$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \end{bmatrix} \qquad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$$w_{new} = w_{old} + \eta (y_i - \hat{y}_i) x_i$$

$$w_{new} = w_{old} - \eta (Y - \hat{y}) x$$

Sigmoid function :- $\sigma(z) = \frac{1}{1 + e^{-z}}$

Softmax Reg. → is when for R no of classes

Logistic for binary class classification

Softmax function $\frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$

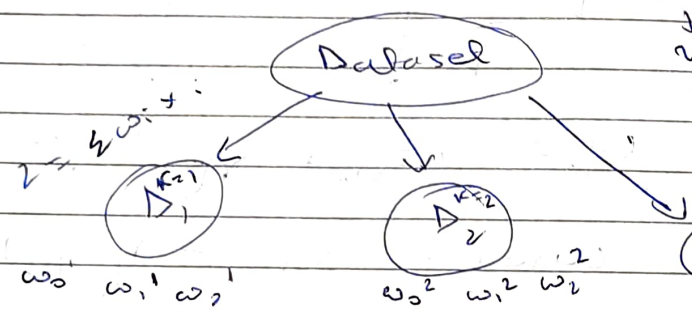
One Hot encoding :-

x_1	x_2	y	$y_{K=1}$	$y_{K=2}$	$y_{K=3}$
6	7	1	1	0	0
5	3	1	1	0	0
3	4	2	0	1	0
1	2	1	1	0	0
6	8	3	0	0	1

↑ wherever 1 → put $y_{K=1}$ in y for others it's 0

↓ where $y_{K=1}$ where $y=2$

↓ put $y_{K=3}$ where $y=3$ for others 0



$$\sigma(z)_{K=1} = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} = 0.40 \text{ for } K=3$$

$$\therefore z = \sum_{i=1} w_i x_i$$

$$\sigma(z)_{K=2} = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} = 0.30$$

$$\sigma(z_1) + \sigma(z_2) + \sigma(z_3) = 1$$

$$L = -\frac{1}{m} \sum_{i=1}^{m}$$

x_1

x_{11}

x_{21}

x_{31}

$= y$

$y_2^{(1)}$

$y_3^{(1)}$

$y_1^{(2)}$

$y_3^{(2)}$

$$L = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_k^{(i)} \log \hat{y}_k^{(i)}$$

x_1	x_2	Y	Y_1	Y_2	Y_3
x_{11}	x_{12}	1	1	0	0
x_{21}	x_{22}	2	0	1	0
x_{31}	x_{32}	3	0	0	1

put $Y_{k=3}$ theorem $Y=3$ for other 0

$$= y_1^{(1)} \log \hat{y}_1^{(1)} + y_2^{(1)} \log \hat{y}_2^{(1)} + y_3^{(1)} \log \hat{y}_3^{(1)} +$$

$$y_1^{(2)} \log \hat{y}_1^{(2)} + y_2^{(2)} \log \hat{y}_2^{(2)} + y_3^{(2)} \log \hat{y}_3^{(2)} +$$

$$y_1^{(3)} \log \hat{y}_1^{(3)} + y_2^{(3)} \log \hat{y}_2^{(3)} + y_3^{(3)} \log \hat{y}_3^{(3)}$$

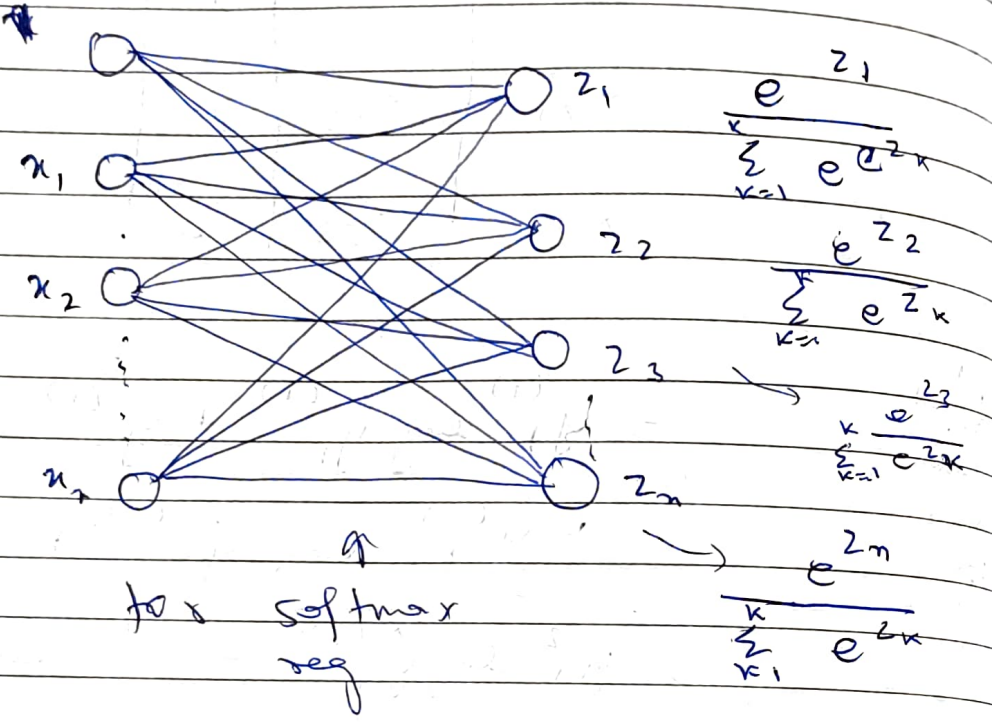
$y_2^{(1)}$ → means 1st row Y_2 col^m = 0

$y_3^{(1)}$ → mean 1st row of $Y_3 = 0$

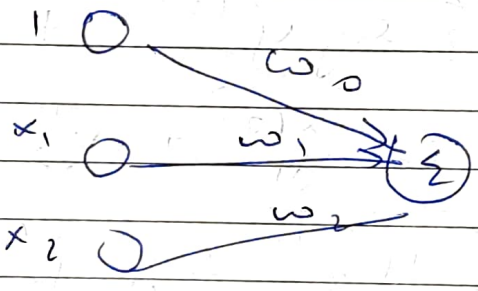
$y_1^{(2)}$ → 0

$y_3^{(2)} = 0$; $y_1^{(3)} = 0$, $y_2^{(3)} = 0$

$$l = y_1^{(1)} \log \hat{y}_1^{(1)} + y_2^{(2)} \log \hat{y}_2^{(2)} + y_3^{(3)} \log \hat{y}_3^{(3)}$$



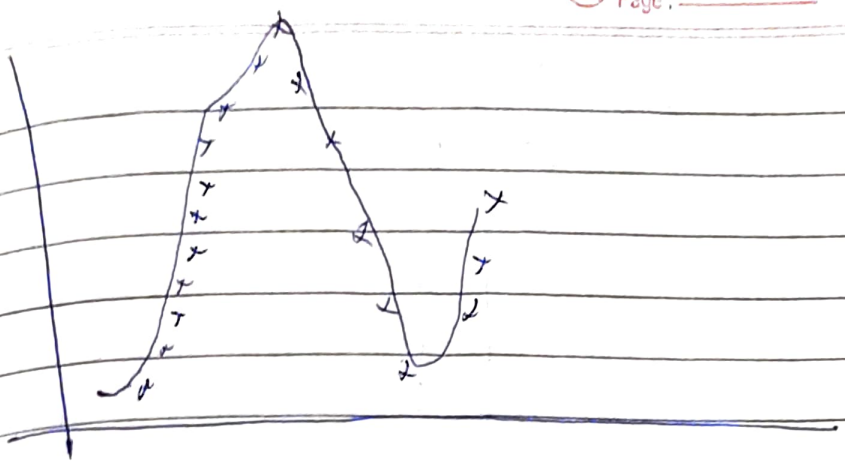
for softmax reg



for logistic reg.

for non-linear data → polynomial logistic reg. used, but its not giving best results → efficiency is less.

Date : __/__/__
Page : _____



Sessional-II ⇒ everything after sessional I

Accuracy = $\frac{\text{No. of Correctly classified}}{\text{Total no of predictions}}$

$= \frac{6}{10}$

$= \frac{TP + TN}{TP + TN + FN + FP}$ $= \frac{12 + 14}{14 + 12 + 2 + 3}$

Confusion matrix :- Predicted

Actual		0		1	
	0	TN	12	FP	3
	1	FN	2	TP	14

$35 \times \pi = 1$

$\pi = 70$

$\Rightarrow 140$

$95 + 1$

Y	$\hat{Y}$	
0	0	✓
0	0	✓
1	0	✗
0	1	
1	0	
0	1	
1	1	
0	0	
1	1	
0	0	

		Sent to spam	Not sent to spam
Not Spam	0	100	170
Spam	1	30	700

Model A

1	100	190
0	10	700

Model B

Model A ज्यादा गलती कर रहा

Precision :-

predicted

Model A =

Model B =

	Has cancer 1	No cancer 0

Precision

Recall

Model

Model

Date : __/__/____
Page : _____

Precision :- What portion of predicted the is true true. =
TP / TP + FP

Model A = $\frac{TP}{TP+FP}$ = ~~100 / 100~~ / ~~100 + 30 = 870~~

= $\frac{100}{100+30}$ = $\frac{100}{130}$

Model B : $\frac{100}{100+10}$ = $\frac{100}{110}$

	Detected cancer 1		Not detected 0	
Has cancer 1	1000	TP	200	FN
No cancer 0	800	FP	8000	TN

Model A

1000	500
500	8000

Model B

Precision of model B > model A

Recall :- $\frac{TP}{TP+FN}$

Model A = 1000 / 1200

Model B = 100 / 1500

F1 - score → lies that side where the value is low

$$= 2 \times \left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

If a model predicts that a cat is dog or a dog is cat, for that F1 score works.