PCA Principal Component Analysis Steps

ex x_1	x_2	Target	New Feature
4	11		-4.30535
8	4		3.7361
13	5		5.6922
7	14		-5.1238
Sum = 32	34		

Step 1 mean calculation

$$\bar{x}_1 = \frac{32}{4} = 8 \quad , \quad \bar{x}_2 = \frac{34}{4} = 8.5$$

Step 2 covariance matrix

$$\begin{array}{c|c|c} & x_1 & x_2 \\ \hline x_1 & cov(x_1, x_1) & cov(x_1, x_2) \\ x_2 & cov(x_2, x_1) & cov(x_2, x_2) \end{array}$$

$$cov(x_1, x_1) = \frac{1}{n-1} \sum_{i=1}^{n} (x_1 - \bar{x}_1)^2$$

$$= \frac{(4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2}{4-1}$$

$$= \frac{16 + 0 + 25 + 1}{3} = 14$$

$$cov(x_1, x_2) = \frac{1}{n-1} \sum_{i=1}^{n} (x_1 - \bar{x}_1)(x_2 - \bar{x}_2)$$

$$= \frac{(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)}{4-1}$$

$$= -11$$

$$cov(x_2, x_1) = cov(x_1, x_2) = -11$$

$$cov(x_2, x_2) = \frac{1}{n-1} \sum_{i=1}^{n} (x_2 - \bar{x}_2)^2$$

$$= \frac{(11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2}{4-1}$$

$$= 23$$

Step 3: Eigen vector and eigen values

$$|S - \lambda I| = 0$$

$$\left| \begin{bmatrix} 14 & -11 \\ -11 & -23 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0$$

$$\begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix} = 0$$

$$(14-\lambda)(23-\lambda) - 121 = 0$$

$$\underset{\downarrow \ PC1}{\lambda_1 = 30.38} \quad , \quad \underset{\downarrow \ PC2}{\lambda_2 = 6.61}$$

Step 4 for each eigen value λ we have corresponding eigen vector $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}_{n\times 1}$

such that $(S-\lambda I)U = 0$

$$e = \frac{U}{\|U\|}$$

$$(S-\lambda I)U = \begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$$

~~for $\lambda_1 \rightarrow$~~ $\cancel{e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}}$, $e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$

for $\lambda_1 = 30.38$

$$\begin{bmatrix} 14-30.38 & -11 \\ -11 & 23-30.38 \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = 0$$

$$\cancel{\begin{bmatrix} -16.38 & -11 \\ -11 & -7.38 \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = 0}$$

$\cancel{16.38U_1 + 11U_2 = 0}$

$\cancel{11U_1 + 7.38U_2 = 0}$

$\cancel{U_1 = 0.5574, \quad U_2 = -0.8303}$

$(14-\lambda_1)U_1 = 11U_2$

$\frac{V_1}{11} = \frac{V_2}{14-\lambda_1} = t$

$V_1 = 11t, \quad V_2 = (14-\lambda_1)t$

$U_1 = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix}$ eigen vector

To find a unit eigen vector we comput length of U_1 $\|U_1\| = \sqrt{11^2+(14-\lambda)^2}$

$= \sqrt{11^2+(14-30.38)^2}$

$= 19.73079$

unit eigenvector for $\lambda_1 = 30.3849$

$$e_1 = \frac{U_1}{\|U_1\|} = \frac{1}{19.73}\begin{bmatrix} 11 \\ 14-30.38 \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$11V_1 = (23-\lambda_2)U_2 = 0$

$\frac{V_1}{23-\lambda_2} = \frac{U_2}{11} = t$

$V_1 = (23-\lambda_2)t, \quad U_2 = 11t$

$U_1 = \begin{bmatrix} 23-\lambda_2 \\ 11 \end{bmatrix}$

$\|U_2\| = \sqrt{(23-6.61)^2 + 11^2}$

$= 19.7391$

$$e_2 = \frac{U_2}{\|U_2\|} = \frac{1}{19.7391}\begin{bmatrix} 23-6.61 \\ 11 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5579 \end{bmatrix}$$

Step 5:- Compute First principal component.

$$PC1 = e_1^T \begin{bmatrix} x_{1k} - \bar{x}_1 \\ x_{2k} - \bar{x}_2 \end{bmatrix}$$

Target feature

for e_1

$$PC_{11} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix} = -4.305$$

row 1

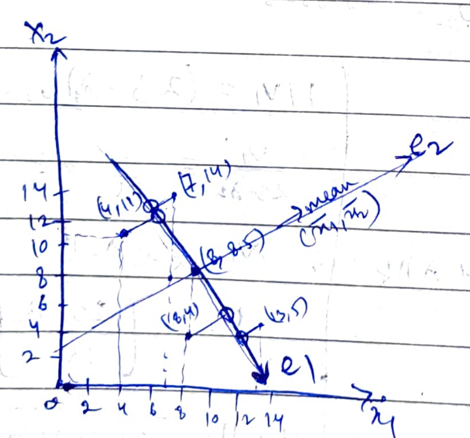
$= \cancel{-4.305}$

$$PC_{12} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 8-8 \\ 4-8.5 \end{bmatrix} = 3.73635$$

$$P_{13} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 13-8 \\ 5-8.5 \end{bmatrix} = 5.69305$$

$$P_{14} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 7-8 \\ 14-8.5 \end{bmatrix} = -5.12405$$

Similarly for e_2 $\quad PC_{21}, P_{22}, P_{23}, P_{24}$ ----



LDA - Linear Discriminant Analysis

LDA is supervised, considers class label and finds components which seperates the classes most.

LDA does not change the location of points but tries to max. the seperation ~~of~~ classes.

Fisher Discriminant Ratio:- $J(w) = \frac{(\bar{\mu}_1 - \bar{\mu}_2)^2}{\bar{S}_1^2 + \bar{S}_2^2}$ → maximize (numerator) → minimize (denominator)

S_W → (Inter-class scatter matrix) or scatter matrix within class

S_B – (Between) class scatter matrix

$J(w) = \frac{w^T S_B w}{w^T S_W w} \Rightarrow \frac{dJ(w)}{dw} = 0 \Rightarrow S_W^{-1} S_B w - Jw = 0$

Ex:- $C_1 \Rightarrow X_1 = (x_1, x_2) = \{(4,1), (2,4), (2,3), (3,6), (4,4)\}$

$C_2 \Rightarrow X_2 = (x_1, x_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$

Step 1:- Find mean:-

$\mu_1 = \left[\frac{4+2+2+3+4}{5}, \frac{1+4+3+6+4}{5}\right] = [3, 3.6]$

$\mu_2 = \left[\frac{9+6+9+8+10}{5}, \frac{10+8+5+7+8}{5}\right] = [8.4, 7.6]$

Step 2: Find scatter matrix within class

$S_1 = \sum_{x \in w_1} (x-\mu_1)(x-\mu_1)^T$

$(x-\mu_1) = \begin{bmatrix} (4-3) & (2-3) & (2-3) & (3-3) & (4-3) \\ (1-3.6) & (4-3.6) & (3-3.6) & (6-3.6) & (4-3.6) \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix}$

$S_1 = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & 0.6 & 2.4 & 0.4 \end{bmatrix} \begin{bmatrix} 1 & -2.6 \\ -1 & 0.4 \\ -1 & 0.6 \\ 0 & 0.24 \\ 1 & 0.4 \end{bmatrix}$

$= \begin{bmatrix} 1+1+1+0+1 & -2.6 - \cancel{0.4} + 0.6 + 0 + \cancel{0.4} \\ -2.6 - \cancel{0.4} + 0.6 + 0 + \cancel{0.4} & (2.6)^2 + (0.4)^2 + (0.6)^2 + (2.4)^2 + (0.4)^2 \end{bmatrix} = \begin{bmatrix} 4 & -2 \\ -2 & 13.2 \end{bmatrix}$

(normalize) $\frac{S_1}{5} = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.64 \end{bmatrix}$

Similary $S_2 = \sum [X_2 - M_2][X_2 - M_2]^T$

$$[X_2 - M_2] = \begin{bmatrix} 9-8.4 & 6-8.4 & 9-8.4 & 8-8.4 & 10-8.4 \\ 10-7.6 & 8-7.6 & 5-7.6 & 7-7.6 & 8-7.6 \end{bmatrix} = \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 0.6 & 2.4 \\ -2.4 & 0.4 \\ 0.6 & -2.6 \\ -0.4 & -0.6 \\ 1.6 & 0.4 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 9.2 & -0.2 \\ -0.2 & 13.2 \end{bmatrix} \Rightarrow \underset{(\text{normalize})}{\frac{S_2}{5}} = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$$

$$S_W = S_1 + S_2 = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.64 \end{bmatrix} + \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$$

$$= \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix}$$

Step 3: Now calculate Between class Scatter matrix (S_B)

$$S_B = (M_1 - M_2)(M_1 - M_1)^T$$

$$= \begin{bmatrix} (3-8.4) \\ (3.6-7.6) \end{bmatrix}_{2\times1} \begin{bmatrix} (3-8.4) & (3.6-7.6) \end{bmatrix}_{1\times2}$$

$$= \begin{bmatrix} -5.4 \\ -4 \end{bmatrix} \begin{bmatrix} -5.4 & -4 \end{bmatrix} = \begin{bmatrix} (5.4)^2 & 5.4\times4 \\ 5.4\times4 & 4^2 \end{bmatrix}$$

$$S_B = \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{bmatrix}_{2\times2}$$

Step 4 :- Find LDA projection Vector

$$S_W^{-1} S_B\, w = \lambda w$$

$$\left| S_W^{-1} S_B - \lambda I \right| = 0$$

use Calculator

$$S_W^{-1} S_B = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix}^{-1} \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{bmatrix}$$

$$\lambda I = \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix}$$

$$= \begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.76 \end{bmatrix}$$

$$S_W^{-1} S_B - \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} = \begin{vmatrix} 11.89-\lambda & 8.81 \\ 5.08 & 3.76-\lambda \end{vmatrix} = 0$$

$$\Rightarrow (11.89-\lambda)(3.76-\lambda) - 8.81 \times 5.08$$

$$\Rightarrow 11.89 \times 3.76 - (11.89 + 3.76)\lambda + \lambda^2 - 8.81 \times 5.08$$

$$\Rightarrow \lambda^2 - 15.65\lambda - 0.04 = 0$$

$$\Rightarrow \boxed{\lambda = 15.65}$$

eigen vector $= S_W^{-1} [\mu_1 - \mu_2]$

$$= \underbrace{\begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix}^{-1}}_{2\times 2} \underbrace{\begin{bmatrix} -5.4 \\ -4 \end{bmatrix}}_{2\times 1}$$

$$= \begin{bmatrix} -2.2016 \\ -0.9408 \end{bmatrix}_{2\times 1}$$

eigen vector for $\lambda = \lambda_1$ (15.65)

$$\begin{bmatrix} 11.89-\lambda & 8.81 \\ 5.08 & 3.76-\lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$$(11.89-\lambda)v_1 + 8.81 v_2 = 0 \Rightarrow \frac{v_1}{-8.81} = \frac{v_2}{11.89-\lambda} = t$$

$$5.08 v_1 + (3.76-\lambda) v_2 = 0 \Rightarrow \frac{v_1}{(3.76-\lambda)} = \frac{v_2}{-5.08} = t$$

Date
DELTA Pg No.

$$U_1 = \begin{bmatrix} -8.81 \\ 11.89 - 15.65 \end{bmatrix} \qquad U_2 = \begin{bmatrix} 3.16 - 15.65 \\ -5.08 \end{bmatrix}$$

$$U_1 = \begin{bmatrix} -8.81 \\ -3.76 \end{bmatrix} \qquad U_2 = \begin{bmatrix} -12.49 \\ -5.08 \end{bmatrix}$$

$$||U_1|| = \sqrt{(-8.81)^2 + (-3.76)^2} \qquad ||U_2|| = \sqrt{(-12.49)^2 + (-5.08)^2}$$

$$= 9.5788 \qquad = 13.4835$$

$$e_1 = \frac{U_1}{||U_1||} = \begin{bmatrix} -8.81/9.5788 \\ -3.76/9.5788 \end{bmatrix}, \quad e_2 = \frac{U_2}{||U_2||} = \begin{bmatrix} -12.49/13.4835 \\ -5.08/13.4835 \end{bmatrix}$$

$$e_1 = \begin{bmatrix} 0.91 \\ 0.39 \end{bmatrix} \qquad e_2 = \begin{bmatrix} 0.9263 \\ 0.3767 \end{bmatrix}$$

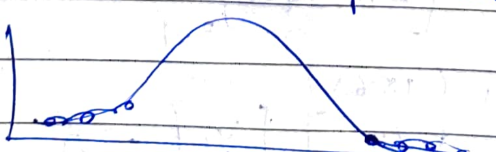
Here in LDA, No need to calculate PC unlike in PCA

[t-Distributed Stoichastic Neighbour Embedding]

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

$\frac{\text{Score}}{\text{Sum of all Score}}$ $\Rightarrow$ 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 trail



Step 1 Check whether x_j point is neighbour of x_i or not for this we use:-

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

, σ_i is the variance of the Gaussian that is centered on x_i datapoint

conditional probability

Step2: For low-dimensional counterparts y_i and y_j and of high-dimensional datapoints x_i and x_j it is possible to compute a similar conditional probability, which we denote by $q_{j/i}$

$$q_{j/i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}$$

(KL divergence)

according to Kullback Leibler divergence method for discrete probability distribution P and Q defined on ~~sample~~ same space X relative entropy from Q to P is defined as

$$(\text{loss}) \rightarrow D_{KL}(P\|Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

$P(x) \rightarrow P_{j/i}$, $Q(x) \rightarrow q_{j/i}$

eg:-

x	0	1	2
distribution $P(x)$	$\frac{9}{25}$	$\frac{12}{25}$	$\frac{4}{25}$
" $Q(x)$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

$$D_{KL}(P\|Q) = \sum_{x \in X} P(x) \ln\left(\frac{P(x)}{Q(x)}\right)$$

$$= \frac{9}{25} \times \log\left(\frac{9/25}{1/3}\right) + \frac{12}{25} \log\left(\frac{12/25}{1/3}\right) + \frac{4}{25} \ln\left(\frac{4/25}{1/3}\right)$$

$= 0.08529$

if $P_{j/i}$ is more which means P_j is the neighbour of P_i loss needs to be less for that $Q_{j/i}$ should be maximum.

$P_{j/i} \rightarrow$ minimum $\Rightarrow$ Therefore log should be minimum

Softmax Regression:-

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

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

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

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

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

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

Softmax Regression:- is when for K no of classes

Logistic Regression:- for binary class classification

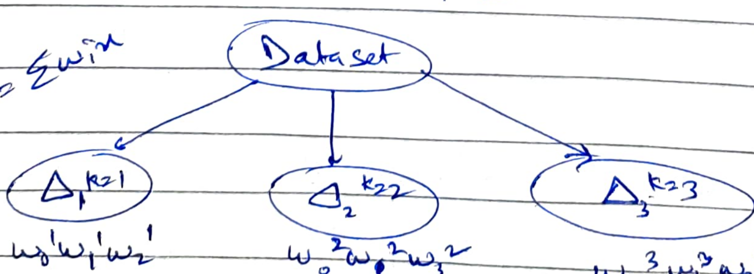
$$\text{Softmax function } \sigma(\vec{Z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

One-hot encoding:- → whenever ~~Y3~~ $Y=1$ put 1 in $Y_{K=1}$ and other Y_K's are 0

x_1	x_2	Y	$Y_{K=1}$	$Y_{K=2}$	$Y_{K=3}$
8	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

where $Y_{K=2}=1$ when $Y=2$ where $Y_{K=3}=1$ when $Y=3$

$Z = \sum w_i x_i$



$$\sigma(Z)_{K=1} = \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}} = 0.40 \text{ for } K=3 \qquad \because Z = \sum_{j=1} w_j x_i$$

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

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

$$L = \frac{-1}{m}\sum_{i=1}^{\overset{\nearrow \text{rows}}{n}}\sum_{k=1}^{\overset{\nearrow\, y \text{ columns}}{k}} y_k^{(i)} \log \hat{y}_k^{(i)}$$

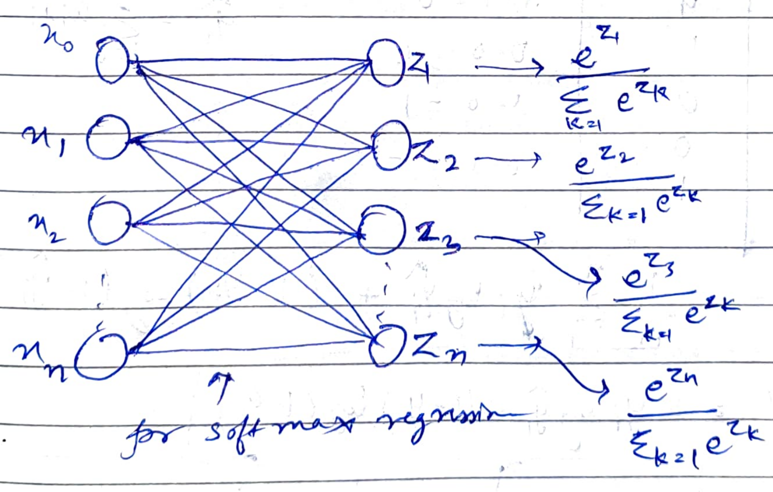
X_1	X_2	$\cancel{y}$	y_1	y_2	y_3
X_{11}	X_{12}	1	1	0	0
X_{21}	X_{22}	2	0	1	0
X_{31}	X_{32}	3	0	0	1

$$= \frac{-1}{m}\Big[\Big\{y_1^{(1)}\log \hat{y}_1^{(1)} + \underbrace{y_2^{(1)}\log \hat{y}_2^{(1)}}_{0} + \underbrace{y_3^{(1)}\log \hat{y}_3^{(1)}}_{0}\Big\}$$

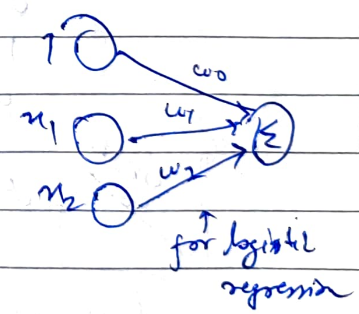
$$+\Big\{\underbrace{y_1^{(2)}\log \hat{y}_1^{(2)}}_{0} + y_2^{(2)}\log \hat{y}_2^{(2)} + \underbrace{y_3^{(2)}\log \hat{y}_3^{(2)}}_{0}\Big\} + \Big\{\underbrace{y_1^{(3)}\log \hat{y}_1^{(3)}}_{0} + \underbrace{y_2^{(3)}\log \hat{y}_2^{(3)}}_{0} + y_3^{(3)}\log \hat{y}_3^{(3)}\Big\}\Big]$$

$$L = \frac{-1}{m}\left(y_1^{(1)}\log \hat{y}_1^{(1)} + y_2^{(2)}\log \hat{y}_1^{(2)} + y_3^{(3)}\log \hat{y}_3^{(3)}\right)$$

$y_k^{(i)}$ means → i^{th} row, y_k column

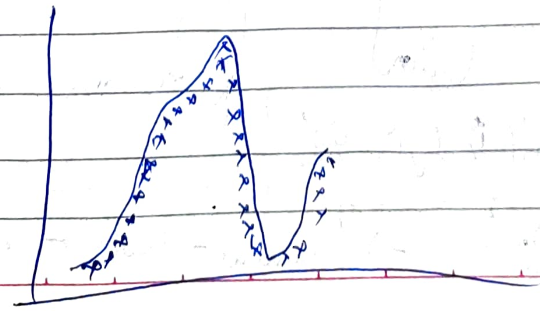


for softmax regression



for logistic regression

⇒ for non-linear data polynomial logistic reg is used, but it does not give best results and efficiency is less.



Logistic Regression :-

$Ax + By + c = 0$

$By = -Ax - c$

$y = \left(\frac{-A}{B}\right)x + \left(\frac{-C}{B}\right)$

for multiple inputs

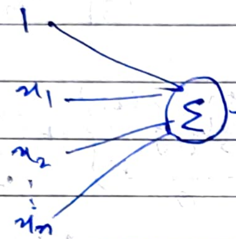
$Ax_1 + Bx_2 + Cx_3 + D = 0$

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

$w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_0 = 0$

$y = w_0 + w_1x_1 + w_2x_2$] for 2 points

perceptron



$\rightarrow y = \sum_{i=0}^{n} w_i x_i$ for $i=0$ $w_0 = b$, $x_0 = 1$

$y = b + w_1x_1 + w_2x_2$

$\boxed{w_{new} = w_{old} + \eta(y_i - \hat{y}_i)x_i}$, $\boxed{w_{new} = w_{old} + \eta\frac{dL}{dw}}$

actual y	predicted $\hat{y}$	$y_i - \hat{y}_i$
1	1	1-1 = 0
0	0	0-0 = 0
1	0	1-0 = 1
0	1	0-1 = -1

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

~~Loss~~ Cost fun^c $= \frac{-1}{n}\sum_{i=1}^{n} y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)$

$\boxed{\text{Loss fun}^c = y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)}$

Sigmoid fun, $\boxed{\sigma(z) = \frac{1}{1+e^{-z}}}$ $\frac{d\sigma(z)}{dz} = \boxed{\sigma'(z) = \sigma(z)[1-\sigma(z)]}$

$\hat{y} = \sigma(xw) \Rightarrow \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \sigma\left(\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & & & \\ \vdots & & & \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{m\times n} \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}_{n\times 1} \right)$

$\hat{y} = \sigma(z)$

$\boxed{L = \frac{-1}{m}\left(y\log\hat{y} + (1-\hat{y})\log(1-\hat{y})\right)}$

$\boxed{\frac{dL}{dw} = \frac{-1}{m}(y-\hat{y})x} \Rightarrow \boxed{w_{new} = w_{old} - \eta\frac{(y-\hat{y})x}{m}}$

Classification

		predicted 0	predicted 1
Actual	0	T_N	F_P
	1	F_N	T_P

0 → Negative
1 → positive
for predicted
T if actual = predi
F if actual ≠ predi

(A) Accuracy $= \dfrac{T_P + T_N}{T_P + T_N + F_P + F_N}$ = $\dfrac{\text{no. of correctly classified}}{\text{Total no. of prediction}}$

(P) Precision $= \dfrac{T_P}{T_P + F_P}$ = what proportion of predicted positive is True true

(R) Recall $= \dfrac{T_P}{T_P + F_N}$ = what proportion of ~~predicted~~ Actual positive is True positive

F1 score $= \dfrac{2PR}{P+R}$,

High P, Low R ——— Low P, High R

Recall = Precision when $F_P = F_N$

$\dfrac{T_P}{T_P + F_P} = \dfrac{T_P}{T_P + F_N}$

$F_P = F_N$

Q

Actual		predicted 0	predicted 1
Not Spam	0	700 T_N	30 F_P
Spam	1	170 F_N	100 T_P

		0	1
Not Spam	0	700	10
Spam	1	190	100

A

Accuracy $= \dfrac{700+100}{700+30+170+100} = 80\%$ | $= \dfrac{700+100}{700+10+190+100} = 80\%$

Precision $= \dfrac{100}{100+30} = 76.9\%$ | $P = \dfrac{100}{10+100} = 90.9\%$

Recall $= \dfrac{100}{100+170} = 37.03\%$ | $= \dfrac{100}{100+190} = 34.48\%$

Recall is more imp than precision

R > P For rare cancer data modelling

P > R For Youtube recommendation, false-negative is loss of a concern

Actual \ Predicted	Dog	Cat	Rabbit	Σ predicted
Dog	25	5	10	40
Cat	0	30	4	34
Rabbit	4	10	20	34
Σ actual	29	45	34	108

Accuracy $= \dfrac{25+30+20}{25+5+10+0+30+4+4+10+20} = \dfrac{75}{108} = 0.694$

Precision$_{Dog} = \dfrac{25}{29} = 0.862$ $P_{cat} = \dfrac{30}{45} = 0.66$, $P_{Rabbit} = \dfrac{20}{34} = 0.588$

Macro Precision $= \dfrac{0.862 + 0.66 + 0.588}{3} = 0.7$

weighted Precision $= \dfrac{40}{108} \times 0.862 + \dfrac{34}{108} \times 0.66 + \dfrac{34}{108} \times 0.588 = 0.7114$

Recall$_{Dog} = \dfrac{25}{40} = 0.625$, $R_{cat} = \dfrac{30}{34} = 0.883$, $R_{Rabbit} = \dfrac{20}{34} = 0.588$

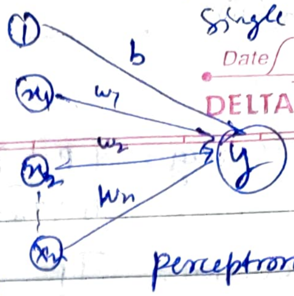
Macro Recall $= \dfrac{0.625 + 0.883 + 0.588}{3} = 0.698$

weighted Recall $=$ ~~$\dfrac{40}{108} \times \dfrac{25}{40}$~~ $\dfrac{29}{108} \times 0.625 + \dfrac{45}{108} \times 0.883 + \dfrac{34}{108} \times 0.588 = 0.7204$

Architecture
Single layer perceptron

Date / Pg No.

Mc culloch Pitts model of a neuron



perceptron is built around a single neuron is limited to performing pattern classification with only two classes.
Also classes should be linearly seperable for the perceptron to work

training
Algo:-

Step1: initialize weights and bias (initally can be 0)
Set learning rate α (0 to 1)

Step2: While stopping condition is false do steps 3 to 7

Step3: For each training pair s:t do steps 4-6

step 4 Set activations of input units
$x_i = s_i$ for i = 1 to n

step 5: Compute the output unit response
$$y_{in} = b + \sum_i x_i w_i$$
The activation func used is,
$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > \theta \\ 0 & \text{if } -\theta \le y_{in} \le \theta \\ -1 & \text{if } y_{in} < -\theta \end{cases}$$

step 6: The weights and bias are updated if the target is not equal to the output response
if $t \neq y$ and value of x_i is not 0
$$w_{i\,new} = w_{i\,old} + \alpha t x_i$$
$$b_{new} = b_{old} + \alpha t$$
else
$$w_{i\,new} = w_{i\,old}, \quad b_{new} = b_{old}$$

Step 7: Test for stopping condition :- stopping condition may be weight changes

Note: 1) only weights connecting active input units ($x_i \neq 0$) are updated
2) weights are update only for patterns that donot produce the correcty value

testing Algo:-
step 1: The weights to be used here are taken from the training algo
step 2 For each input vector x to be classified do steps 3-4
step 3: Input units activations are set
step 4: Calculate response of output unit
$$y\text{-}in = \sum_i x_i w_i, \quad y = f(y\text{-}in) = \begin{cases} 1 & \text{if } y_{in} > \theta \\ 0 & \text{if } -\theta \le y_{in} \le \theta \\ -1 & \text{if } y\text{-}in < \theta \end{cases}$$