AI

Lecture # 26-March-2024

Perception Learning :- → ANN ~~for~~

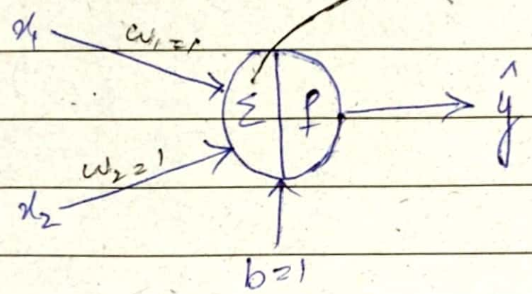
* Hebb's Rule
* Delta Rule

} → for classification

⇒ Perceptron Structure :-

- Single Layer Perceptron. (Computation only on output)
-

$(\Sigma = x_1 w_1 + x_2 w_2 + b)$



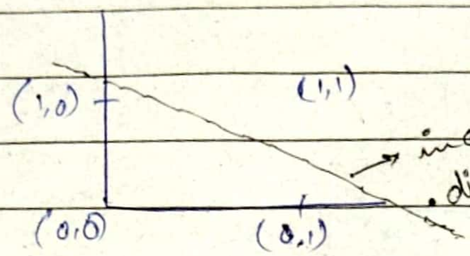
∴ Each neuron has it's own b value

x_1, x_2 → inputs
w_1, w_2 → weights
b → baise
$\hat{y}$ → output
Σ → Summation (Sigma)
f → Activation function (θ)

$\theta > 2$ bcz it divides into 2 classes as we are doing classification
↓
* one class is of 0.
* ~~one class is of 1.~~

AND

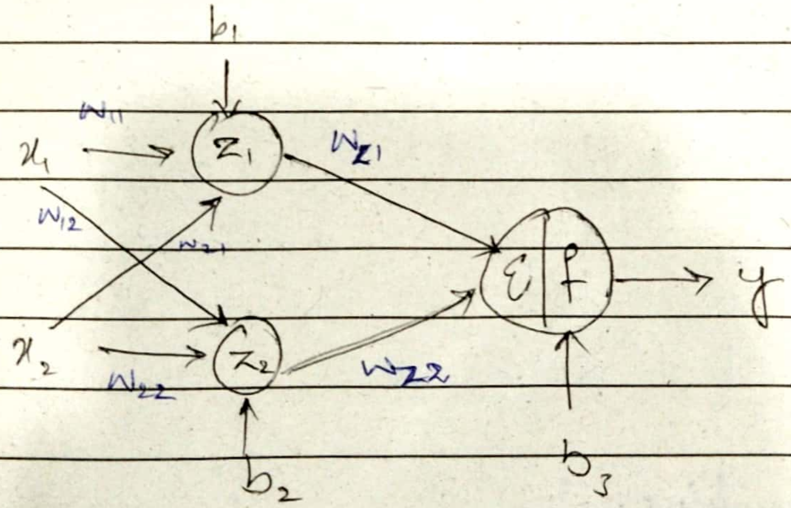
X_1	X_2	$\hat{y}$	Σ	$\hat{y} = \theta > 2$
0	0	0	1	0
0	1	0	2	0
1	0	0	2	0
1	1	1	3	1



→ in case of AND.
. divides the problem into 2 classes.

- One Perceptron fails when problem can't be classified using linear line. (e.g. in case of xor, we can't classify problem into 2 classes using one line.)
- Due to this, we have to use 2 or more perceptrons which are known as <u>multi-layer perceptron</u>.

⟹ Number of neurons = no. of baises



∴ Multi-Layer Perceptron

$z_1 = x_1 w_{11} + x_2 w_{21} + b_1$

$z_2 = \cancel{x_2} w + x_2 w_{22} + b_2$

$y = z_1 w_{z1} + z_2 w_{z2} + b_3$

$\theta \geq 3$

x_1	x_2	y	Σ	$\hat{y}$
-1	-1	-1	-5	0
-1	+1	-1	-1	0
+1	-1	-1	-1	0
+1	+1	1	3	1

threshold

* **Iteration # 1 :-**

$w_1 = 0 + 1 = 1$

$w_2 = 0 + 1 = 1$

$b = 1 + (-1) = 0$

* **Iteration # 2 :-**

$w_1 = 1 + 1 = 2$

$w_2 = 1 - 1 = 0$

$b = 0 + (-1) = -1$

* **Iteration # 3 :-**

$w_1 = 2 + (-1) = 1$

$w_2 = 0 + 1 = 1$

$b = -1 - 1 = -2$

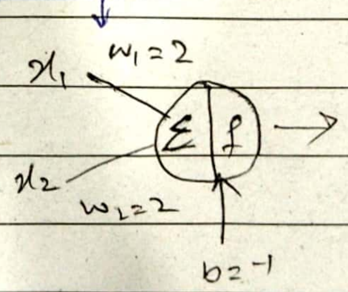
* **Iteration # 4 :-**

$w_1 = 1 + 1 = 2$

$w_2 = 1 + 1 = 2$

$b = -2 + 1 = -1$

after iteration #4



imp.

⇒ **Equations :-**

$$W_{i\,new} = W_{i\,old} + \Delta W_i$$

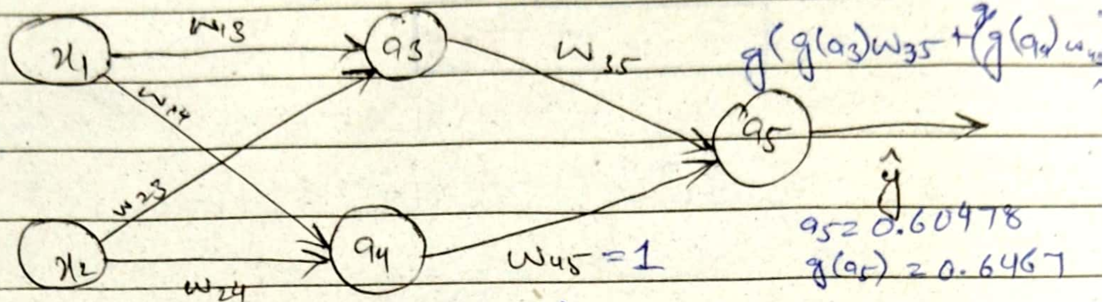
$$b_{new} = b_{old} + \Delta b$$

28-March-20

AI Lecture

- Delta Rule :-

$$g(\underbrace{x_1 w_{13} + x_2 w_{23}}_{a_3 = 0.2}) = g(a_3) = 0.5498$$



$$g(\underbrace{x_1 w_{14} + x_2 w_{24}}_{a_4 = 0.2}) = g(a_4) = 0.5498$$

imp.

* When data is not linearly classifiable, Hebb's rule can't be applied. That's why Delta Rule is there.

* Assuming baise is zero
* Inputs & weights are :
  $x_1 = 1$, $x_2 = 1$, $y = 1$
  $w = 0.1$, except $w_{45} = 1$
  $\gamma = 0.2$
* Activation function = Sigmoid = $g(x) = \frac{1}{1+e^{-x}}$ $\quad g'(x) = x(1-x)$

Abs. Error = $|y - \hat{y}|$ — ①
Squared Error = $\frac{1}{2}(y - \hat{y})^2$ — ②

- Forward propogation
- Backward propogation.

• $\alpha / \gamma / N$ = Learning Rate

→ for finding Error, we'll use eq. 1 (for now),

Abs. Error = | 1 − 0.6467 |

$\boxed{E = 0.3533}$

Now, finding delta,

$\Delta 5 = E(g'(a_5))$ — ③ = 0.3533 × 0.239 = 0.084.

$\Delta 3 = \Delta 5 \cdot w_{35} \cdot g'(a_3)$ — ④ = 0.00135

$\Delta 4 = \Delta 5 \cdot w_{45} \cdot g'(a_4)$ — ⑤ = 0.0135

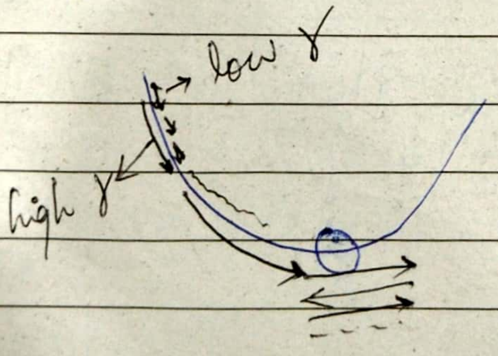
• $\gamma \Rightarrow$ learning rate

imp.

• High Learning Rate(γ) will cause that we will keep oscillating ~~bet~~ in between local minimas/maximas.

imp.

• Low Learning Rate(γ) will cause that it can take many many steps & time to reach optimal.



→ Now, for calculating weights,

$$W_{35} = W_{35} + \gamma \times \Delta 5 \times (a_3) = 0.1034$$
$$W_{45} = W_{45} + \gamma \times \Delta 5 \times (a_4) = 0.1034$$
$$W_{13} = W_{13} + \gamma \times \Delta 3 \times x_1 = 0.10027$$
$$W_{14} = W_{14} + \gamma \times \Delta 4 \times x_1 = 0.1027$$
$$W_{23} = W_{23} + \gamma \times \Delta 3 \times x_2 = 0.10027$$
$$W_{24} = W_{24} + \gamma \times \Delta 4 \times x_2 = 0.1027$$

also, from previous eq. (3), (4) & (5), we get

$$\Delta 5 = 0.084$$
$$\Delta 3 = 0.00135$$
$$\Delta 4 = 0.0135$$

→ Above is first iteration, after first iteration, find error & compare it with previous one, the change will be very very minute.

→ So, you will have to do hundreds of iterations.

→ iteration will start from finding sigmoid fn() values & everything ....

• If activation function is not explicitly defined, then, you'll use sigmoid fn(), else if given in question that activation fn() is linear, then (σ) sigma is the output of linear fn().

e.g. $h_1 = x_1 w_{11} + x_2 w_{12} + b_1 = g(h_1) = h_1$

-: Formula for updation of biase :-

$$b_{new} = b_{old} + \gamma \times \Delta \text{ (of same whose biase we taking)}$$

old

e.g.

$$b_3 = b_3 + \gamma \times \Delta O_1$$

16-04-2021

Linear Regression

Goal: Minimize Error, x: driving experience (years)
y: insurance ($)

x (indep. var)	y (dep. var)	x^2	y^2	xy	$\hat{y}$	$\lvert y-\hat{y}\rvert$
5	64	25	4096	320	68.9	
2	87	4	7569	174		
12	50	144	2500	600		
9	71	81	5041	639		
15	44	225	1936	660		
6	56	36	3136	336		
25	42	625	1764	1050		
16	60	256	3600	960		
Σ = 90	Σ = 474	Σ = 1396	Σ = 29,642	Σ = 4739		

* $y = mx + c$
  - m ↓ slope
  - c ↓ intercept

* in LR, we have,
  $y = bx + a$ ⟹ (Single LR) (A)
  - y ↓ (dependent var)
  - x ↓ (independent var.)

* If more than 1 independent variables, then
  $$y = \underbrace{b_1x_1 + b_2x_2 + b_3x_3 + a}_{\text{(Multiple LR)}}$$

$$SS_{xy} = \Sigma xy - \frac{\Sigma x \Sigma y}{n} \quad \text{--- } ①$$

$$SS_{xx} = \Sigma x^2 - \frac{(\Sigma x)^2}{n} \quad \text{--- } ②$$

$$SS_{yy} = \Sigma y^2 - \frac{(\Sigma y)^2}{n} \quad \text{--- } ③$$

$$b = \frac{SS_{xy}}{SS_{xx}} \quad \text{--- } ④$$

$$a = \bar{y} - b\bar{x} \quad \text{--- } ⑤$$

($\bar{y}$ = mean of y)
($\bar{x}$ = mean of x)

* Formula 1:-

$$SS_{xy} = 4739 - \frac{(90)(474)}{8}$$

$$= 4739 - 5332.5$$

$$\boxed{SS_{xy} = -593.5}$$

* Formula 2:-

$$SS_{xx} = 1396 - \frac{8100}{8}$$

$$= 1396 - 1012.5$$

$$\boxed{SS_{xx} = 383.5}$$

CamScanner

* ~~SS~~ **Formula 3:-**

$$SS_{yy} = 29642 - \frac{224676}{8}$$

$$= 29642 - 28084.5$$

$$\boxed{SS_{yy} = 1557.5}$$

* **Formula 4:-**

$$b = \frac{-593.5}{383.5}$$

$$\boxed{b = -1.55}$$

this shows that x & y are inversely proportional.

This shows that 1 unit increase in x, will cause 1.55 units decrease in y. ($\because$ decrease bcz it's -ve)

* **Formula 5:-**

$$a = \bar{y} - b\bar{x}$$

$$\boxed{a = 76.65}$$

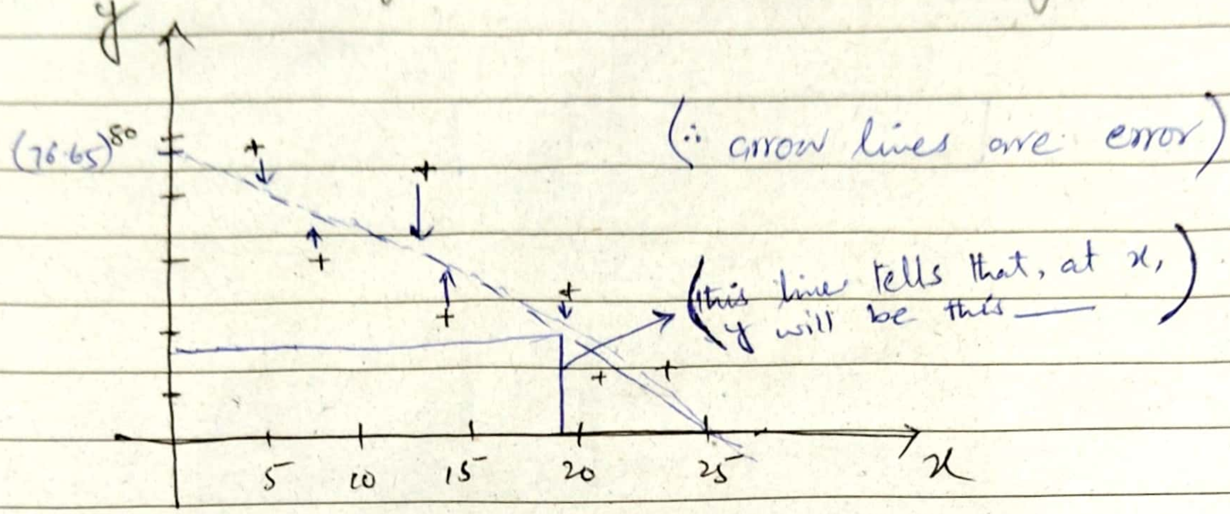
Now, eq. A becomes,

$$\boxed{y = -1.55x + 76.65} \quad \text{——(i)}$$

* crosses are marked using given values of x & y.

<u>Now, draw graph</u>—

* Below graph is called scatter graph.



∴ put $x = 0$, $y = \cancel{8}\ 76.65$

∴ take max. x, i.e. 25

* The more, the pluses are closer to line, the less error it will be.

Mean Absolute Error = MAE = $\dfrac{\Sigma |y - \hat{y}|}{n} = \pm 8.46$

margin of error

R — corelation Coefficient

$$R = \frac{SS_{xy}}{\sqrt{SS_{xx} \cdot SS_{yy}}} = -0.76 \Rightarrow$$

(range: −1 −ve strong corelation coefficient, 0 weak cc, 1 +ve strong cc.)

(−ve Moderate cc)

for $\hat{y}$, put each value of x in eq (i), e.g.

$y = -1.55(5) + 76.65 = 68.9$

$\therefore$ Error is interpreted on basis of data.

$\therefore$ Error = 200,000 → model can be ideal model.

$\therefore$ Error = 0.5 → model can be worst model.

$(MSE) =$

Mean Squared Error $= \frac{1}{n} \sum (y - \hat{y})^2$

Root Mean Squared Error = RMSE = $\sqrt{MSE}$

($\therefore$ we always prefer MAE & RMSE but for RMSE, we'll have to find MSE).

Now, $R^2 = (-0.76)^2 = 0.58972$

↓

coefficient of Determination

range : 0 (bad) 1 (good).

$\therefore$ it determines variance in data set, if it's 0.58, this means that we need more features for better computations.

25-April-2021

Lecture

CLUSTERING

(unlabeled data)

⇒ K-Means

Choose min value from each row

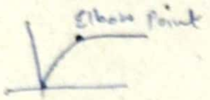
K-Means	x	y	d(C1) A1	d(C2) A4	d(C3) A7
$C_1 \to A_1$	2	10	0 ✓	X	X
A_2	2	5	5	4.24	3.16 ✓
A_3	8	4	8.43	5 ✓	7.28
$C_2 \to A_4$	5	8	X	0 ✓	X
A_5	7	5	7.07	3.6 ✓	6.7
A_6	6	4	7.21	4.12 ✓	5.38
$C_3 \to A_7$	1	2	X	X	0
A_8	4	9	~~X~~ 2.24	~~X~~ 1.41 ✓	7.6

* x & y are both independent vars.
* if we get 3 features, we'll need 3D graph to plot it for clustering.
* if we get 4 features, i.e. 4 independent variables, we'll need 4D graph to plot which becomes impossible.
* We cluster data on basis of their differences & similarities.

⇒ Euclidean Distance $= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

2 issues in K-Method:- (1) "K" is issue.

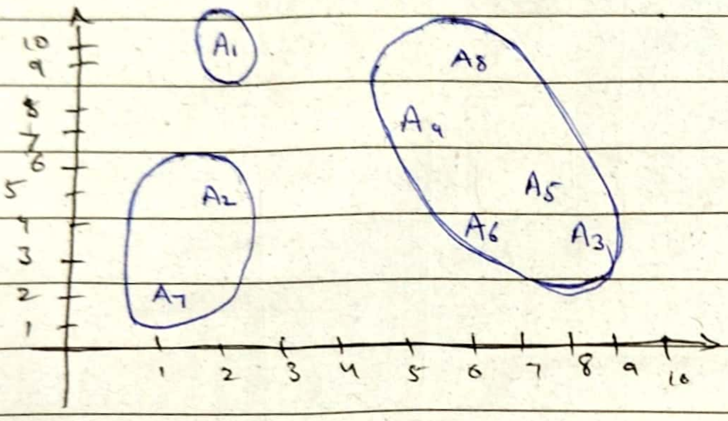
(2) -Initial centroids



Cluster 1 = { A_1 }

Cluster 2 = { A_3, A_4, A_5, A_6, A_8 }

Cluster 3 = { A_2, A_7 }



$C_1 = (2, 10)$

$C_2 = \frac{8+5+7+6+4}{5}$,

(

Again Calculate Below Table

K-Means	x	y	d(C1)	d(C2)	d(C3)
$C_1 \rightarrow A_1$	2	10	0 ✓	5.66 *	6.5 *
A_2	2	5	5	4.1 *	1.58 * ✓
A_3	8	4	8.43	2.8 * ✓	6.5 *
$C_2 \rightarrow A_4$	5	8	3.6 *	2.2 ✓ *	5.7 *
A_5	7	5	7.07	1.41 ✓ *	5.7 *
A_6	6	4	7.21	2 ✓ *	4.5 *
$C_3 \rightarrow A_7$	1	2	8.06	6.4 *	1.5 * ✓
A_8	4	9	2.24	3.61 ✓ *	6.7 *

* Storic values are all new calculated values.

this methods solve issue no.(2), i.e. initial centroids

K-Means ++ (Partition-based)

$K=3$, $C_1 = A_1$, $C_2 = A_3$, $C_3 = A_7$ } after doing calculation

↳ (3 clusters will be made)

	X		X				X	
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
x	2	2	8	5	7	6	1	4
y	10	5	4	8	5	4	2	9
$d(C_1)$	0	5	8.43 ✓*	3.6	7.07	7.21	8.06	2.24
$d(C_2)$	8.48 ✓* X	6.08	0 X	5	1.41	2	7.28	6.4
	X	5	X	3.6	1.41	2	7.28 ✓*	2.24

imp.

* we select first centroid randomly.
* second centroid on basis of first centroid & so on.

* use old values of $d(C_1)$ to ~~fille~~ fill row no. 3 -
* we'll chose maximum from $d(C_1)$ row bcz we want maximum distance b/w clusters.

*

2 methods to chose C_3:-

(i) - Avg Dist: add $C_1 + C_2$ & divide by 2. C_1 & C_2

(ii) - Max. of Min. distance: Find min from all ~~cols~~, then select max. from them.

Table 1

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

* we'll choose min from upper ~~Tra~~ right triangle.
* Distance from, A to B & B to A is same & so on.

d	# K	clusters
—	4	A, B, C, D
1	3	{AB}, C, D
2	2	{{AB}, C}, D
3	1	{{{AB}, C}, D}

Table 2 :-

	AB	C	D
AB	0	2	5
C		0	3
D			0

distance from AB to C & D can be found using 3 methods :

* you have to follow 1 method throughout solution.

(1)- Single Link (min)
(2)- Complete Link (max)
(3)- Avg Link (avg).

i.e. $\min[\overset{4}{d(A,C)}\ \overset{2}{d(B,C)}]$

$= AB \rightarrow C = 2$

(Now update table by choosing min from ~~new~~ table 2.)

Now,

Table 3:-

	ABC	D
ABC	0	3
D		0

Now, 3 is written after calculating using any of above 3 formulae i.e. single, complete or avg link

Lecture

-: Performance Measures :-

→ Classification
- Accuracy
- Precision
- Recall
- F1

+ve 900

Actual

Type 1 error

		+ve	-ve
Predicted	+ve	TP 50	FP 0
	-ve	FN 50	TN 900

Type 2 error

* Accuracy: $\frac{TP + TN}{100} = \frac{50 + 900}{100} = 0.95$ i.e. 95%

* Precision :-
  * We always talk about predictions made by model.
  * The rate of correct predictions made by Model.
  * Out of prediction, how many actually had tumour.

Precision = $TP / (TP + FP) = \frac{50}{50 + 0} = 1$ (very high precision)

* We try to increase prediction and recall but practically, there's always a trade off

* Recall:-

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{50}{100} = 0.5$$

* Out of actual tumours, how many predictions were correct.

* F1:-

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{recall}}$$

* Harmonic mean of precision & recall.

range:-

$0 \longrightarrow 1$

bad precision & recall — good precision & recall

		Actual	
		+ve	-ve
Predictions	+ve	28000	9000
	-ve	1000	100

Accuracy = A = 73.7 %
Prediction = P = 0.757
Recall = R = 0.965
F_1 = 0.469

Actual sick (T/P)	Predicted sick (+ve/-ve)	Outcome FP/FN/TP/TN
1	1	TP
0	0	TN
0	0	TN
1	1	TP
0	0	TN
0	0	TN
1	0	FN
0	1	FP
0	0	TN
1	0	FN

		A	
		+ve	-ve
P	+ve	TP_2	FP_1
	-ve	FN_2	TN_5

* Diagnols are correctly predicted values.

A =

P =

R =

F_1 =

* If 2 or more than :-

		Actual Covid	Actual TB	Actual Healthy
Predict	Covid	TP 200	30 FP	30 FP
	TB	50 FP	TP 250	20 FP
	Healthy	50 FP	20 FP	TP 350
		300	300	400

Total = 1000
Covid = 300
TB = 300
Healthy = 400

$$A = \frac{200 + 250 + 350}{1000} = 80\%$$

$$P_c = 200 / 200+30+30 = 0.76$$

$$P_{TB} = 250 / 250+20+50 = 0.78$$

$$P_H = 350 / 350+20+50 = 0.83$$

means, better predictions of Healthy people that they are healthy.
& struggling to predict covid & TB.

$R_c = 0.667$
$R_{TB} = 0.83$
$R_H = 0.87$

predicted

	A	B	C	sum
Actual A	16	0	0	16
B	0	17	1	18
C	0	0	11	11
sum	16	17	12	

$P_A = 16/16+0+0 = 1$ $\quad F_{1A} = 2 \times 1/2 = 1$

$R_A = 16/16+0+0 = 1$

$P_B = 17/0+17+0 = \cancel{0.94}\ 1$

$R_B\ 17/18 = 0.94$

$P_c = 11/12+0+0 = 0.91$

$R_c = 11/11+0+0 = 1$

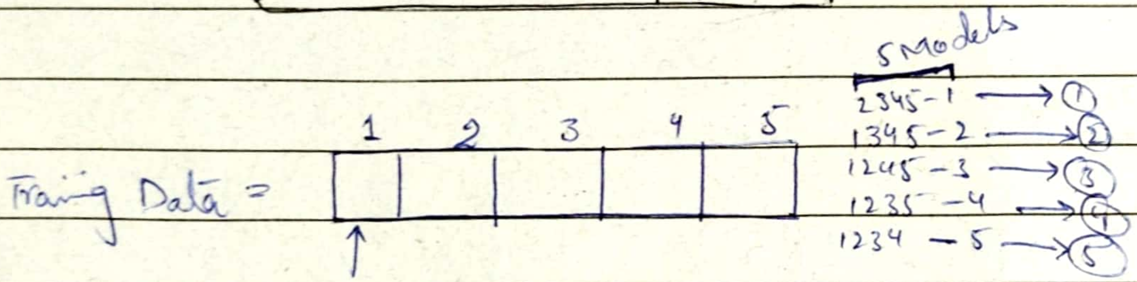
* for precision & ~~diagonal~~ recall, choose diagnol value & divide by respective total accordingly.

Model Verification using Cross Validation:-

We split our data into 3 parts.

- Training
- Testing
- Validation

Training = 80	Testing 20



* **K-fold cross validation**: gives the number of ~~splits~~ sections, training data will be splitted. In above case, K = 5. (training data will be splitted into 5 parts)

$\alpha = [A, B, C]$
Activation = [A, B]
Loss = [A, B]
Batch_size = [A, B]

total models = $3 \times 2 \times 2 \times 2$
= 24

Now, max combinations with variations = 0

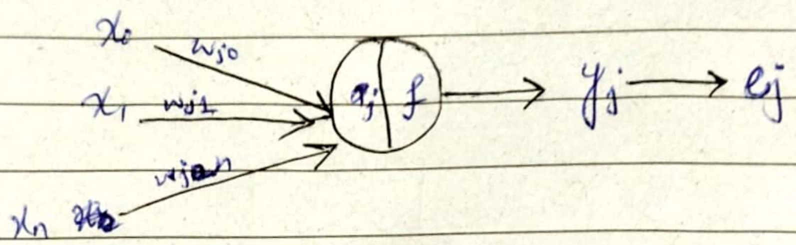
max-combinations {{A, A, A, A}, {A, B, B, B}, {B, A, A, A}, {—, —}}

9-May-2024

Lecture

- Derivation of Delta Rule
- CNN.

* Delta Rule Derivation :- (this derivation is for single perceptron).



$$a_j = \sum_{i=0}^{n} w_{ji} X_i^n \quad \text{——— (i)}$$

$$\hat{y}_j^n = f(a_j^n) \quad \text{——— (ii)}$$

$$e_j = y_j - t_j \quad \text{——— (iii)} \quad (\because y_j = \text{predicted}) \quad (\because t_j = \text{target})$$

$$E = \frac{1}{2} \Sigma_j (e_j^n)^2 \quad \text{——— (iv)}$$

* Chain Rule :-

concept :- $$\frac{dE^n}{dw_{ij}} = \frac{\partial E}{\partial e} \times \frac{\partial e}{dy} \times \frac{\partial y}{\partial a} \times \frac{\partial a}{\partial w} = \frac{\partial E}{\partial w} \quad \text{——— (A)}$$

$$* \quad \frac{\partial E}{\partial e} = \frac{\partial}{\partial e}\left[\frac{1}{2} \Sigma (e_j^n)^2\right] = 2 \times \frac{1}{2} (e_j^n) = e_j^n$$

* $\frac{\partial e}{\partial y} = \frac{\partial}{\partial y}[y_j - t_j] = 1$ (∵ t_j is constant)

* $\frac{\partial y}{\partial a} = f(a_j^n) = f'(a_j^n)$

* $\frac{\partial a}{\partial w} = \frac{\partial}{\partial w}\left[\Sigma w_{ji} X_i^n\right] = X_i$

⇒ put all above derivatives in eq. (A), we g

$$\frac{\partial E}{\partial w_{ji}} = \underbrace{e_j^n \cdot f'(a_j^n)}_{\delta_j^n} X_i$$ (∵ δ = delta)

$$\boxed{\frac{\partial E}{\partial w_{ji}} = \delta_j^n X_i}$$