

25/1/25

ML

Given a dataset for performing linear regression with one variable.

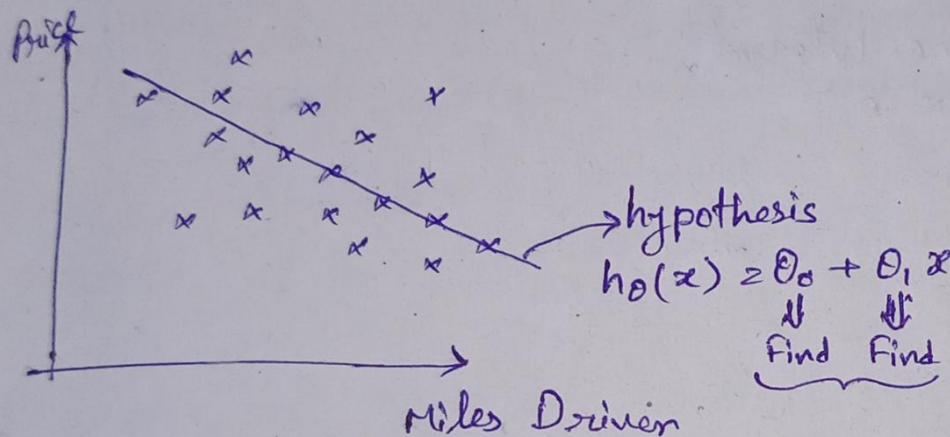
Example: Used car price prediction		Features	o/p
Q/p	Miles Driven	Engine Capacity	Price

But for simplicity purpose we are taking only one i/p.

Miles Driven	Price
1000	2.5
2520	3.6
:	:
:	:
:	:

Lk. 2D matrix  
 $(1000, 2.5)$   
 $(2520, 3.6)$

$$y = mx + c$$
$$\geq \theta_0 + \theta_1 x$$



CSV (Comma Separate Vector) can store huge no. of datas in tabular form.

1) What is machine learning?

2) What is learning?

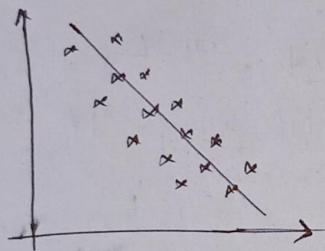
3) What are the different types of m/l?

4) What is Supervised Learning?

5) What type of problem we can solve using Supervised Learning?  
 ↓  
 Regression Classification

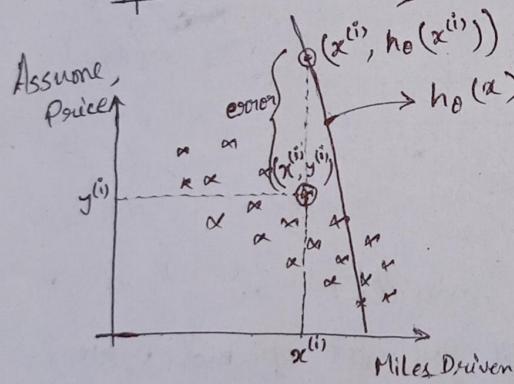
6) What is Regression problem explain with a real-life example?

7) Linear Regression with one variable



$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Initially choose  $\theta_0$  and  $\theta_1$  randomly.



$$\text{error} = (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

~~The only need is the values~~

The distance can be negative but we only need magnitude, but we can't do  $\text{mod}(1)$ . Because, for later we have to do derivative but mod makes can make problem at then. So, we can do square.

Now, assume there are  $m$  data samples.

$$\text{So, total no. of errors} = \underbrace{\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2}_{\text{mean squared error (MSE)}} \rightarrow \text{loss function}$$

Minimize Error

$$\text{Minimize } J(\theta_0, \theta_1)$$

$$\theta_0, \theta_1$$

$$J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

$y = mx + c$   
↓ slope

$$m = \frac{dy}{dx}$$

$$\frac{d J(\theta_0, \theta_1)}{d \theta_0} \quad \frac{d J(\theta_0, \theta_1)}{d \theta_1}$$

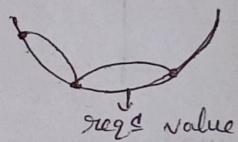
Updating the value of  $\theta_0, \theta_1$ :

$$\theta_0 = \theta_0 - \alpha \cdot \frac{d J(\theta_0, \theta_1)}{d \theta_0} \rightarrow \text{gradient}$$

$$\theta_1 = \theta_1 - \alpha \cdot \frac{d J(\theta_0, \theta_1)}{d \theta_1}$$

$\alpha$  = Learning rate

If  $\alpha = 1$  then it means  $\alpha$  is too high. That means it is seeing the slope and making a high jump. It can make problem.



Steps:

- ① Take Random points
- ② Calculate errors
- ③ Calculate gradients
- ④ Calculate  $\alpha$ -value
- ⑤ Calculate new value for  $\theta_0$  and  $\theta_1$
- ⑥ Repeat from step ① until we get optimal value

Performing Partial Derivative:

$$\text{For } \theta_0 \\ J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 \quad \hat{y}_i = \theta_0 + \theta_1 x_i^{(i)}$$

$$\therefore J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (\theta_0 + \theta_1 x_i^{(i)} - y_i)^2$$

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m \frac{d}{d \theta_0} (\theta_0 + \theta_1 x_i^{(i)} - y_i)^2$$

$$= \frac{1}{2m} \sum_{i=1}^m \frac{d(\theta_0 + \theta_1 x_i^{(i)} - y_i)^2}{d(\theta_0 + \theta_1 x_i^{(i)} - y_i)} \times \frac{d(\theta_0 + \theta_1 x_i^{(i)} - y_i)}{d \theta_0}$$

$$= \frac{1}{2m} \sum_{i=1}^m 2(\theta_0 + \theta_1 x_i^{(i)} - y_i)(1+0-0)$$

When we have more than one variable then we need to do partial derivative.

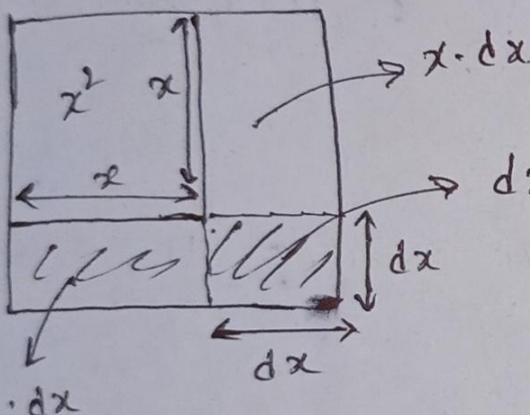
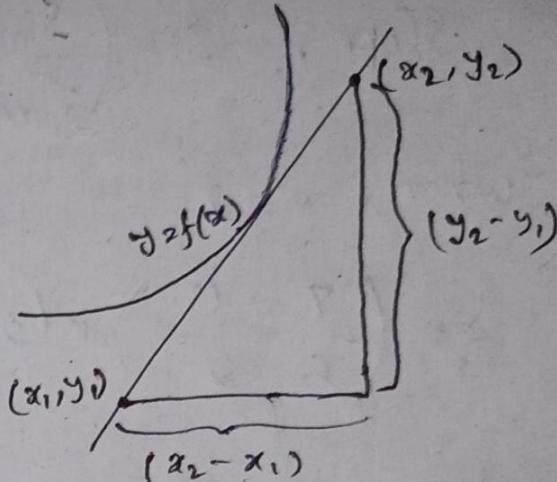
This is called  
Gradient Descent Algorithm for learning.

$$= \frac{1}{m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)})$$

$$\boxed{\frac{d}{d\theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})}$$

Now, for  $\theta_1$ ,

$$\frac{d}{d\theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$



$$\begin{aligned} & (x+dx)^2 \\ &= x^2 + 2x \cdot dx + dx^2 \\ &\approx (x+dx)^2 - x^2 \approx 2x \cdot dx \end{aligned}$$

$dx$  very small value

$$\begin{aligned} \tan \theta &= \frac{\text{height}}{\text{base}} \\ &= \frac{y_2 - y_1}{x_2 - x_1} \\ &= \frac{dy}{dx} \end{aligned}$$

ML

$$\text{Total Error} \rightarrow \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$$

Avg. err.

$$\text{Mean Square Error} \rightarrow \left[ \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 \right] \xrightarrow{\text{Loss function}}$$

### ④ Gradient Descent Algorithm

### ⑤ Linear Regression with multiple features:

Input				Output
Miles Driven ( $x_1$ )	Engine Capacity ( $x_2$ )	Fuel Type ( $x_3$ )	Price ( $y$ )	

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

Fix  
 $x_1 = 1$

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$$

$$= \theta^T x$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix}^{(1 \times n)}$$

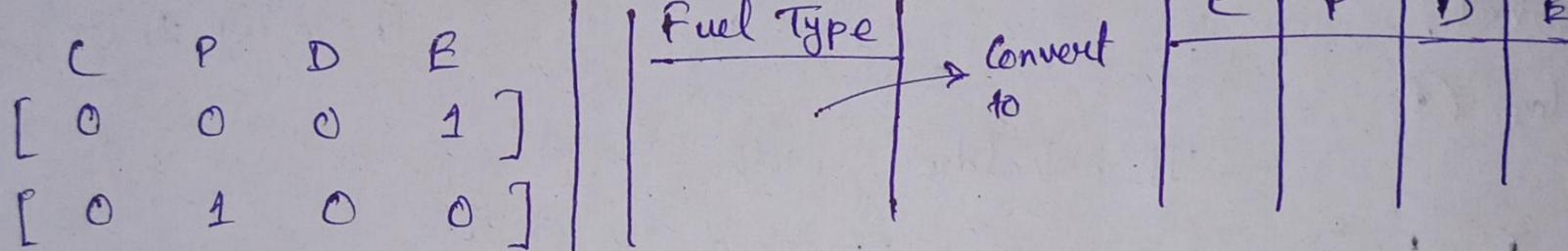
$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}^{(n \times 1)}$$

## Repeat Control Convergent:

$$\theta_j = \theta_j - \alpha \frac{d}{d\theta_j} J(\theta) \quad j = 0 \text{ to } n$$

## ONE HOT ENCODING:

Non-numerical Feature  $\xrightarrow{\text{to}}$  Vector



## Normalization of Features: / Feature Scaling

$$[0 - 1]$$

$$\min(\text{age})$$

$$\max(\text{age})$$

$$\frac{x - \min}{\max} \rightarrow [0 - 1]$$

Age	Salary
31	80000
32	80001
40	240000

ML

Miles Driven	Engine Capacity	Year	No. of Pre Owners	Fuel Types
2000	[−1 to 1]	[−1 to 1]	[−1 to 1]	CNG      C      D      P      B Dies.    0      1      0      0 Pet.     1      0      0      0 Batt.    1      0      0      0

$\leq 0.5 \rightarrow$  Class 0

$> 0.5 \rightarrow$  Class 1

0.5 is threshold value. If the probability value is 0.5 then the machine will think that there has 50% chances for both.

Qn: What is the role of the sigmoid / logistic  $f_2$  to determine the class?

• Cost  $f_2$  and loss  $f_2$  are same.

Regularization : Overfitting Problem

Underfit, Perfect fit, Overfit

ML

8/2/25

## Classification using Decision-Tree

Features

Age, Income, Student? Credit Rating

The prediction is basically buy computer or not.

Example :

Age

Youth

Income

High

Student

No

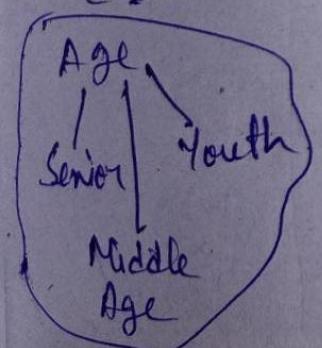
Credit Rating

Fair

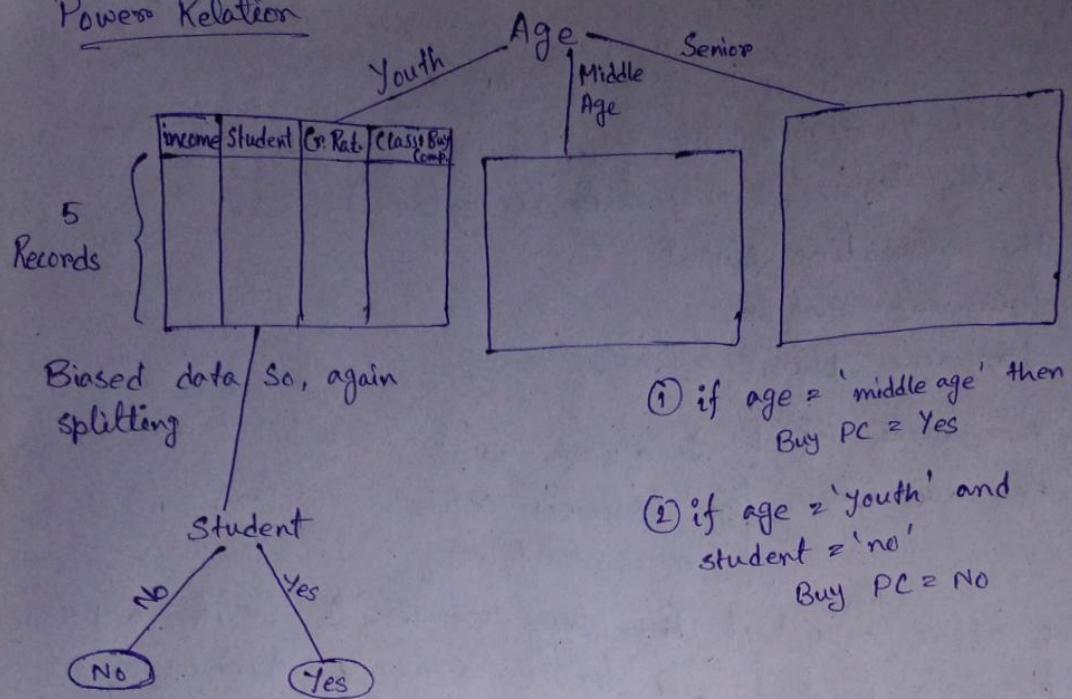
buy

By Comp.

No



## Powers Relation



① if age = 'middle age' then  
Buy PC = Yes

② if age = 'youth' and  
student = 'no'  
Buy PC = No

If no after splitting all that attributes, there left no attribute then go for majority voting. If the majority is 'yes' then all will be 'yes' and vice-versa.

## Attribute Selection for Splitting

Information gain / Entropy

$$\text{Info}(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad \therefore m = \text{no. of classes}$$

For our case,

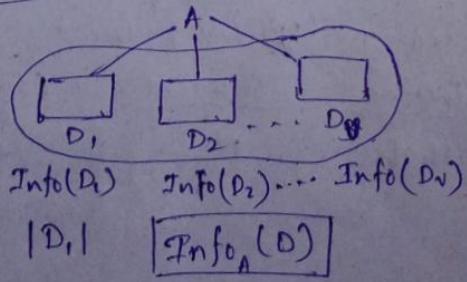
$$m=2 \rightarrow \begin{array}{l} \text{Buy computer} = \text{Yes} \\ \quad \quad \quad p_1 \\ \text{Buy computer} = \text{No} \\ \quad \quad \quad p_2 \end{array} \quad n =$$

- The unit of entropy is bits.

$$IC \propto \frac{1}{P(A)}$$

Information Content

$$A \boxed{D} \quad \text{Info}(D) | D_A |$$



$$\text{Info}_A(D) = \frac{|D_1|}{|D|} \text{Info}(D_1) + \frac{|D_2|}{|D|} \text{Info}(D_2) + \dots + \frac{|D_v|}{|D|} \text{Info}(D_v)$$

$$= \sum_{j=1}^v \frac{|D_j|}{|D|} \text{Info}(D_j)$$

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

$$\text{Gain}(Age) = \text{Info}(D) - \text{Info}_{Age}(D)$$

$$\text{Info}(D) = - \sum_{i=1}^{14} (P_i \times \log_2(P_i))$$

~~$$\text{Info}(D) = - P_1 \log_2(D_1) - P_2 \log_2(D_2)$$~~

$$= - \frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right) = 0.7400$$

$$\begin{aligned} &= -0.64 \log_2(0.64) - 0.36 \log_2(0.36) \\ &= (-0.64) \times (-0.15) - (0.36) \times (-1.02) \\ &= 0.288 + 0.3672 \\ &= 0.28 \end{aligned}$$

~~$$\text{Info}_{Age}(D) = \frac{5}{14} \left[ -\frac{2}{5} \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \log_2\left(\frac{3}{5}\right) \right] +$$

$$\frac{5}{14} \left[ -\frac{3}{5} \log_2\left(\frac{3}{5}\right) - \frac{2}{5} \log_2\left(\frac{2}{5}\right) \right] +$$

$$\frac{4}{14} \left[ 0 - \frac{4}{4} \log_2\left(\frac{4}{4}\right) \right]$$

$$= \frac{10}{14} (0.97 + 0.97)$$

$$= 0.69285 = 0.2465$$~~

$$\text{gain}(age) = 0.97 - 0.69 = 0.2465$$

$$\text{Info}_{Age}(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \text{Info}(D_j)$$

$$= \sum_{j=1}^3 \frac{|D_j|}{|D|} \text{Info}(D_j)$$

$$= \frac{|D_1|}{|D|} \text{Info}(D_1) + \frac{|D_2|}{|D|} \text{Info}(D_2) + \frac{|D_3|}{|D|} \text{Info}(D_3)$$

$$= \frac{5}{14} \left[ -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right] + \frac{4}{14} \left[ -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \right]$$

$$= \frac{5}{14} \left[ -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right] + \frac{5}{14} \left[ -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right]$$

$$= 0.6928$$

$$\text{gain}(age) = 0.97 - 0.69 = 0.28$$

Gain of Income = 0.029 bits

Gain of Student = 0.151 bits

Credit Rating = 0.048 bits

MLEntropy

$$\text{Info}(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

m: no. of classes

$$\text{Info}_A(D) = \sum_{j=1}^n \frac{|D_j|}{|D|} \cdot \text{Info}(D_j)$$

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

$$\text{Split Info}_A(D) = - \sum_{j=1}^3 \frac{|D_j|}{|D|} \cdot \log_2 \frac{|D_j|}{|D|}$$

According to  
Income  
attribute

$$\begin{aligned} &= - \sum_{j=1}^3 \frac{|D_j|}{|D|} \cdot \log_2 \frac{|D_j|}{|D|} \\ &= - \frac{|D_1|}{|D|} \cdot \log_2 \frac{|D_1|}{|D|} + \frac{|D_2|}{|D|} \cdot \log_2 \frac{|D_2|}{|D|} + \frac{|D_3|}{|D|} \cdot \log_2 \frac{|D_3|}{|D|} \\ &= - \frac{4}{14} \log_2 \frac{4}{14} - \frac{6}{14} \log_2 \frac{6}{14} - \frac{4}{14} \log_2 \frac{4}{14} \\ &\approx 1.5564 \end{aligned}$$

$$= 0.15545 + 0.15770 + 0.15545$$

$$\begin{aligned} \text{Gain Ratio}(A) &= \frac{\text{Gain}(A)}{\text{Split Info}(A)} \\ &= \frac{0.28}{1.5564} = 0.1799 \quad \text{Ans} \end{aligned}$$

$$\text{Gini}(D) = 1 - \sum_{i=1}^m p_i^2$$

$$\text{Gain}_A(D) = \frac{|D_1|}{|D|} \text{Gini}(D_1) + \frac{|D_2|}{|D|} \text{Gini}(D_2)$$

$$\Delta \text{Gini}(A) = \text{Gini}(D) - \text{Gini}_A(D)$$

$$\sum_{i=1}^m y^{(i)} \log_2 h_0(x^{(i)})$$

$\downarrow$   
 Actual class      Predicted class

Binary cross entropy loss

$$\left( \underbrace{\sum_{i=1}^m y^{(i)} \log_2 h_0(x^{(i)})}_{\text{Binary cross entropy for false class}} + \underbrace{\sum_{i=1}^m (1-y^{(i)}) \log_2 (1-h_0(x^{(i)}))}_{\text{Binary cross entropy for true class}} \right)$$

ML

## Receiver Operating Characteristics (ROC) Curve

if  $g(\theta^T x) \geq 0.5 \Rightarrow$  Class 1

$$h_\theta(x) = g(\theta^T x)$$

$g(\theta^T x) < 0.5 \Rightarrow$  Class 0

$$0 \leq h_\theta(x) \leq 1$$

Consider  $T_1$  as threshold value

$T_1 \rightarrow s,$   $\begin{cases} \xrightarrow{\quad} \text{TPR (True Positive Rate)} \\ \xrightarrow{\quad} \text{FPR (False Positive Rate)} \end{cases}$

SVM

Support Vector Machine:

Class 1  $\geq 1$

Class 0  $< -1$

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \rightarrow 2D$$

$$\textcircled{1} \quad \left( \sum_{i=1}^n (x_i^{(2)} - x_i^{(1)})^2 \right)^{1/2} \rightarrow nD \quad \text{Euclidean Distance (L2)}$$

$$\textcircled{2} \quad \text{Manhattan Distance (L1) Distance / City Block Distance}$$

$$L_1 = |x_2 - x_1| + |y_2 - y_1| \rightarrow 2D$$

$$\begin{aligned} \textcircled{2} &\rightarrow \\ L_1 &= |x_1^{(2)} - x_1^{(1)}| + |x_2^{(2)} - x_1^{(1)}| + \dots + |x_n^{(2)} - x_n^{(1)}| \\ &= \sum_{i=1}^n |x_i^{(2)} - x_i^{(1)}| \end{aligned}$$

KNN

 Lazy Learner  $\rightarrow$  no learning in historical data, but learn in prediction data.



Solve Classification problem using KNN:



Solve regression problem using KNN:

- KNN
- Event (outcome of the experiment)
- SVM
- Experience (Base on some information)
- Importance of Bayes Theorem
- Conditional Probability  
A given B (B event already occurs over here)

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

(means  $P(A)$  depends on B occurrence)

Here B event at first occurred

From (i) & (ii)

$$P(A|B) \cdot P(B) = P(A \cap B)$$

$$\Rightarrow P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

$$\therefore P(B|A) \cdot P(A) = P(A \cap B)$$

Because here we consider every single event.

Commutative Law

$$\begin{bmatrix} A \cup B = B \cup A \\ A \cap B = B \cap A \end{bmatrix}$$

Diff. between Probability &amp; Set

Here we can distribute

$$\rightarrow A \cap B$$

$$\& A \cap C$$

Here we use Base Theorem.

Bayes Theorem :-

$$P(B_i|A) = \frac{P(B_i \cap A)}{P(A)}$$

$$P(B_i|A) = \frac{P(A|B_i) \cdot P(B_i)}{\sum_{i=1}^n P(A|B_i) \cdot P(B_i)}$$

ML

Qn: Using KNN classify the data  $x = (\text{brightness} = 20, \text{saturation} = 35)$  when two class labels red & blue are given and the dataset is as follows.

Consider,  $K=3$  and  $K=5$

If  $K$  even no. then there may be a tie

<u>Sol:</u>	<u>Brightness(<math>x_1</math>)</u>	<u>Saturation(<math>y_1</math>)</u>	<u>Class</u>
40	20	Red	
50	50	Blue	
60	90	Blue	
10	25	Red	
70	70	Blue	
60	10	Red	
25	80	Blue	

Let,  $K=3$

$x_1 = 20, y_1 = 35$  and  $x_2$  and  $y_2$  are brightness and saturation from table respectively.

Eucleadian Distance,

$$d_1 = \sqrt{(40-20)^2 + (20-35)^2} = \sqrt{400+225} = \sqrt{625} = 25$$

$$d_2 = \sqrt{(50-20)^2 + (50-35)^2} = \sqrt{900+225} = 33.54$$

$$d_3 = \sqrt{(60-20)^2 + (90-35)^2} = 68.07$$

$$d_4 = \sqrt{(10-20)^2 + (25-35)^2} = 14.14$$

$$d_5 = \sqrt{(70-20)^2 + (70-35)^2} = 61.03$$

$$d_6 = \sqrt{(60-20)^2 + (10-35)^2} = 47.16$$

$$d_7 = \sqrt{(25-20)^2 + (80-35)^2} = 45.27$$

Choose  $K=3$  (three) min. values.

No. of Red Samples = 2

No. of Blue Samples = 1

(20, 35)  $\Rightarrow$  Color Red

If  $K=5$ ,

No. of Red Samples = 3

No. of Blue Samples = 2

(20, 35)  $\Rightarrow$  Color Red

### Support Vector Machine (SVM)

What is the role of support vector in SVM?

For parallel line,

$$a_1x + b_1y + c_1 = 0$$

$$a_2x + b_2y + c_2 = 0$$

$$\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}$$

Expanding factor  $\rightarrow \frac{1}{0.99}$

Let,

Expanding factor = 0.7

$$0.7(2x + 3y + (-6)) = 1.4x + 2.1y + (-4.2) = \pm 1$$

$$0.7(2x + 3y + (-6)) = 1.4x + 2.1y + (-4.2) = -1$$

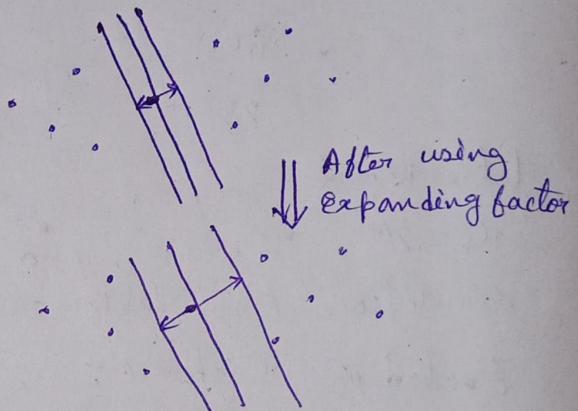
Expanding factor is needed to increase the gap between the support vectors.

There will be no data value between decision boundary and support vectors.

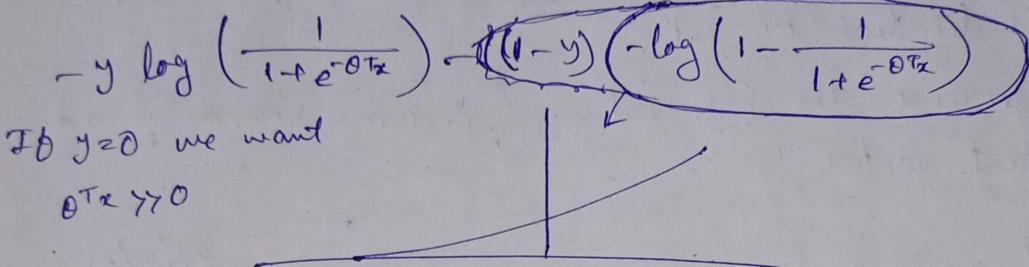
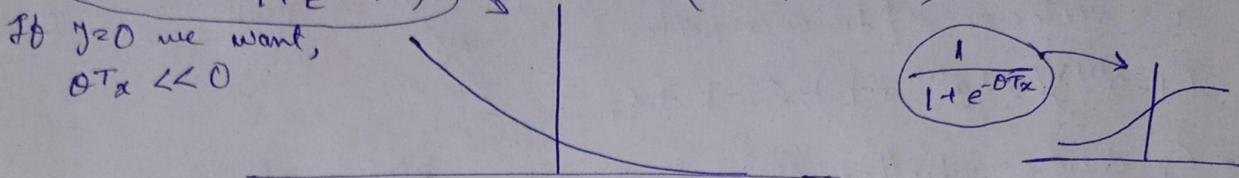
Less increase when expanding factor is ~~tends~~ near to 1.

Binary Cross Entropy Loss Function:  $\log\left(\frac{1}{1+e^{-0.7x}}\right)$

$h_0(x) < 0.5 \quad y=0$       }  
 $\geq 0.5 \quad y=1$       } Depending on probability



$$\left. \begin{array}{ll} \theta^T x > 0 & y = 1 \\ \theta^T x \leq 0 & y = 0 \end{array} \right\} \text{Depending on Data Boundary}$$



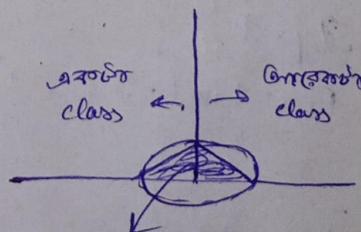
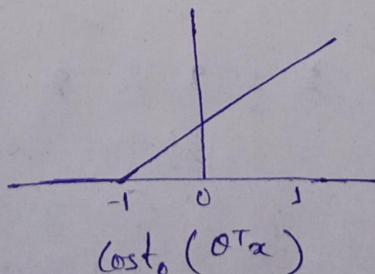
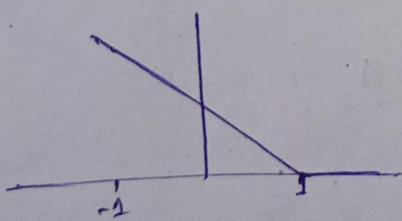
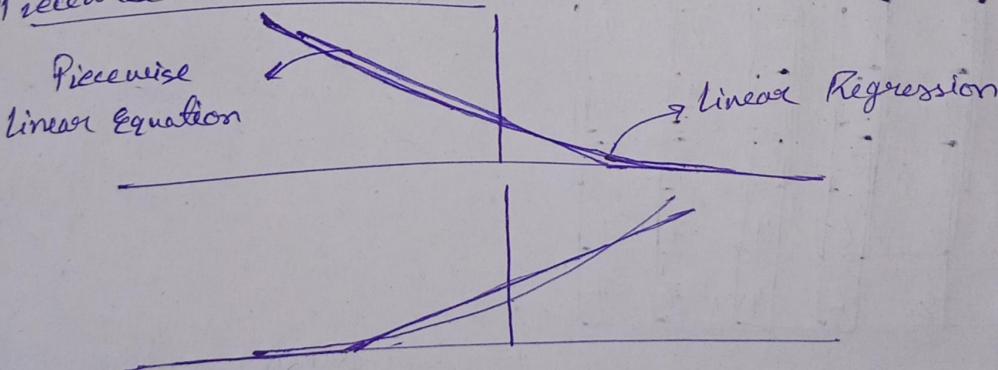
### Logistic Regression

$$\min_{\theta} \frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} (-\log h_{\theta}(x)) + (1-y^{(i)}) (-\log (1-h_{\theta}(x))) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

### Support Vector Machine

$$\min_{\theta} C \left[ \sum_{i=1}^m y^{(i)} (\text{Cost}_1(\theta^T x^{(i)})) + (1-y^{(i)}) (\text{Cost}_0(\theta^T x^{(i)})) \right] + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

### Piecewise Linear Equation



No Man Land  
No Data Item  
can present  
there.

If  $y=1$  then  $\theta^T x \geq 1$

If  $y=0$  then  $\theta^T x \leq -1$

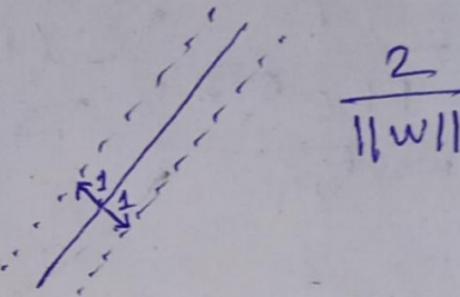
$$\min_{w, b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \epsilon_i \quad \text{s.t. } y^{(i)} (w^T x^{(i)}) \geq 1 - \epsilon_i \quad i=1, \dots, m$$

$$\epsilon_i \geq 0, \quad i=1, \dots, m$$

$\epsilon_i$  = Error

w = vector norm / distance vector

$$(1 - y^{(i)}) (w^T x^{(i)} + b) \leq -1 + \epsilon_i$$



- $\epsilon_i$  decides whether decision boundary ~~place~~ is placed in proper location or not.

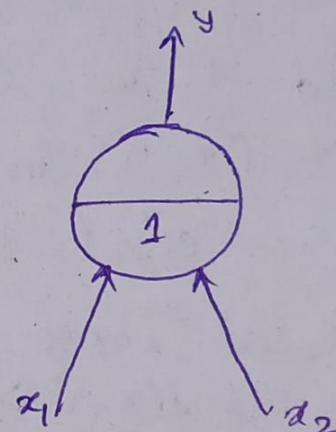
- $\|w\|$  increases the gap between <sup>the</sup> decision boundary and the support vector

# Artificial Neural Network — MP Neuron

Qn: Design a two input OR Gate using MP neuron.

$x_1$	$x_2$	$x_1 + x_2$	$g(x) = \sum x_i$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	2

$$\theta = 1$$



$$f(g(x)) = 1 \text{ if } g(x) \geq 1$$

$$= 0 \text{ if } g(x) < 1$$

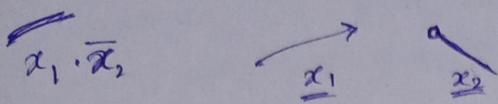
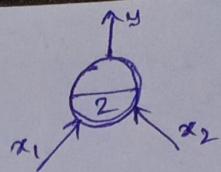
Qn: How can we design AND gate.

$x_1$	$x_2$	$x_1 \text{ and } x_2$	$g(x) = \sum x_i$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	2

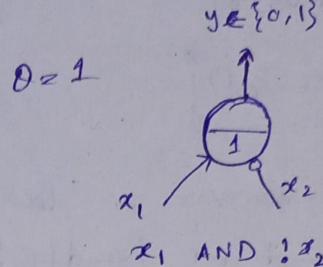
$$\theta = 2$$

$$f(g(x)) = 1 \text{ if } g(x) \geq 1$$

$$f(g(x)) = 1 \text{ if } g(x) \geq 2 \\ = 0 \text{ if } g(x) < 2$$

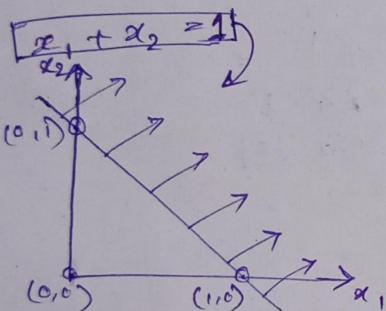


$x_1$	$x_2$	$\bar{x}_2$	$x_1, \bar{x}_2$	$\frac{2}{\sum_i} x_i$
0	0	1	0	0
0	1	0	0	1
1	0	1	1	1
1	1	0	0	2



For two input OR gate

$$y = f(g(x)) = 1 \text{ if } x_1 + x_2 \geq 1 \\ = 0 \text{ if } x_1 + x_2 < 1$$



### Perceptrons

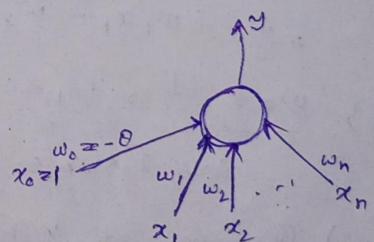
$$g(x) = x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + x_n \cdot w_n$$

$$g(x) = \sum_{i=1}^n x_i$$

$$\boxed{g(x) = \sum_{i=1}^n w_i x_i}$$

$$y = 1 \text{ if } \sum_{i=1}^n w_i x_i \geq \theta$$

$$= 0 \text{ if } \sum_{i=1}^n w_i x_i < \theta$$



### Perceptions

$$y = 1 \text{ if } \sum_{i=0}^n w_i x_i \geq 0$$

$$= 0 \text{ if } \sum_{i=0}^n w_i x_i < 0$$

$$\sum_{i=1}^n w_i x_i \geq 0$$

$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n \geq 0$$

$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n - \theta \geq 0$$

$\downarrow$   
 $w_0 \quad x_0$

$$w_0 x_0 + w_1 x_1 + \dots + w_n x_n \geq 0$$

$$\boxed{\sum_{i=0}^n w_i x_i \geq 0}$$

For MP neuron no learning is present, it's a static model and we handpicked  $\theta$  values. But in case of perceptrons we have to basically train the  $\theta$  values.

Qn:) Draw ~~AND~~ OR gate using perceptron.

$x_1$	$x_2$	$x_1 + x_2$	$g(x)$
0	0	0	$w_0 + w_1 \cdot 0 + w_2 \cdot 0 = w_0 \leq 0$
0	1	1	$w_0 + w_1 \cdot 0 + w_2 \cdot 1 = w_0 + w_2 \geq 0$
1	0	1	$w_0 + w_1 \cdot 1 + w_2 \cdot 0 = w_0 + w_1 \geq 0$
1	1	1	$w_0 + w_1 \cdot 1 + w_2 \cdot 1 = w_0 + w_1 + w_2 \geq 0$

$$\begin{aligned} g(x) &= w_0 x_0 + w_1 x_1 + w_2 x_2 \\ &= w_0 + w_1 x_1 + w_2 x_2 \end{aligned}$$

$$\text{Let, } w_0 = -1, w_1 = 1, w_2 = 1$$

One Possible Solution

Qn:) Design a XOR gate using perceptron.

$x_1$	$x_2$	$x_1 \oplus x_2$	$g(x)$
0	0	0	$w_0 + w_1 \cdot 0 + w_2 \cdot 0 = w_0 \leq 0$
0	1	1	$w_0 + w_1 \cdot 0 + w_2 \cdot 1 = w_0 + w_2 \geq 0 \Rightarrow w_2 \geq -w_0$
1	0	1	$w_0 + w_1 \cdot 1 + w_2 \cdot 0 = w_0 + w_1 \geq 0 \Rightarrow w_1 \geq -w_0$
1	1	0	$w_0 + w_1 \cdot 1 + w_2 \cdot 1 = w_0 + w_1 + w_2 \leq 0 \Rightarrow w_1 + w_2 \leq -w_0$

Implement XOR using a single perceptron —

$$y = 1 \text{ if } \sum_{i=0}^m w_i x_i \geq 0$$

$$y = 0 \text{ if } \sum_{i=0}^n w_i x_i < 0$$

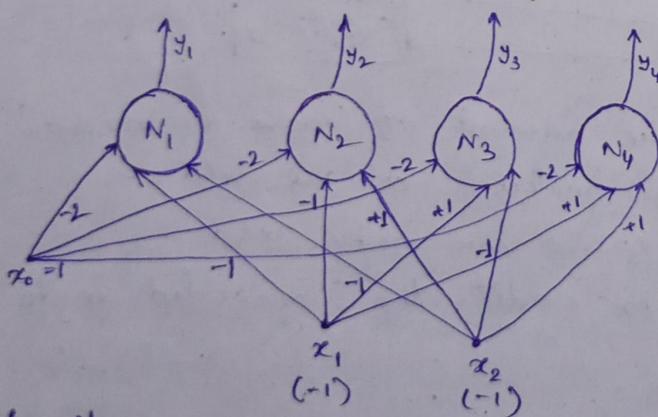
XOR is not linearly separable  $\Rightarrow$  and a single perceptron can only handle linearly separable data. So, using a single perceptron we can not implement XOR gate. We need a network of perceptrons.

XOR using Network of Perceptrons —

For Input

Binary  $0 \Rightarrow -1$

		1 $\Rightarrow$ +1	AIM				
$x_1$	$x_2$	XOR		$y_1$	$y_2$	$y_3$	$y_4$
$N_1$	-1	-1	0	1	0	0	0
$N_2$	-1	+1	1	0	1	0	0
$N_3$	+1	-1	1	0	0	1	0
$N_4$	+1	+1	0	0	0	0	1



for  $N_1$ ,

$$y_1 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1) \cdot (-1) + (-1) \cdot (-1) = 0 \geq 0 = 1$$

$$y_2 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1) \cdot (-1) + (+1) \cdot (-1) = -2 < 0 = 0$$

$$y_3 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (-1) + (-1) \cdot (-1) = -2 < 0 = 0$$

$$y_4 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (-1) + (+1) \cdot (-1) = -4 < 0 = 0$$

$$N_1, \\ x_1 = -1, x_2 = -1$$

$$N_2, \\ x_1 = -1, x_2 = +1$$

$$N_3, \\ x_1 = +1, x_2 = -1$$

$$N_4, \\ x_1 = +1, x_2 = +1$$

for  $N_2$ ,

$$y_1 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1) \cdot (-1) + (-1) \cdot (+1) = -2 < 0 \Rightarrow 0$$

$$y_2 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1) \cdot (-1) + (+1) \cdot (+1) = 0 \geq 0 \Rightarrow 1$$

$$y_3 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (-1) + (-1) \cdot (+1) = -4 < 0 \Rightarrow 0$$

$$y_4 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (-1) + (+1) \cdot (+1) = -2 < 0 \Rightarrow 0$$

for  $N_3$ ,

$$y_1 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1) \cdot (+1) + (-1) \cdot (-1) = -2 < 0 \Rightarrow 0$$

$$y_2 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1) \cdot (+1) + (+1) \cdot (-1) = -4 < 0 \Rightarrow 0$$

$$y_3 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (+1) + (-1) \cdot (-1) = 0 \geq 0 \Rightarrow 1$$

$$y_4 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (+1) + (+1) \cdot (-1) = -2 < 0 \Rightarrow 0$$

for  $N_4$ ,

$$y_1 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1) \cdot (+1) + (-1) \cdot (+1) = -4 < 0 \Rightarrow 0$$

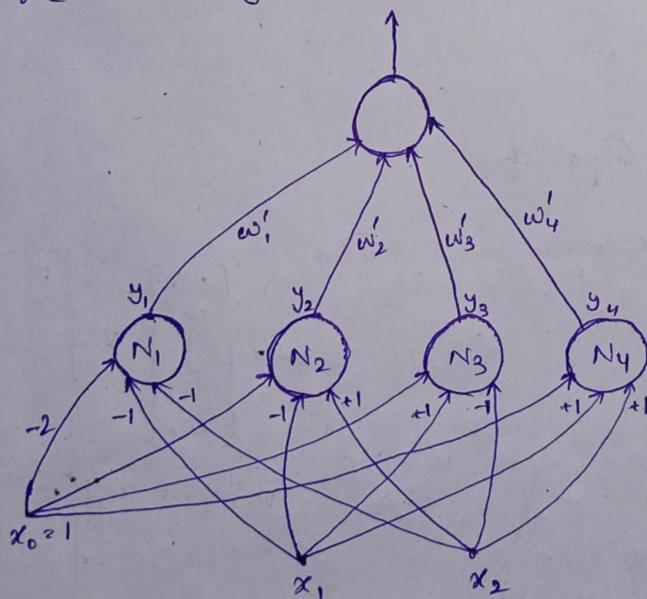
$$y_2 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1) \cdot (+1) + (+1) \cdot (+1) = -2 < 0 \Rightarrow 0$$

$$y_3 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (+1) + (-1) \cdot (+1) = -2 < 0 \Rightarrow 0$$

$$y_4 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (+1) + (+1) \cdot (+1) = 0 \geq 0 \Rightarrow 1$$

$$y_M = w_0 + w_1 x_1 + w_2 x_2 = -2 + (+1) \cdot (+1) + (+1) \cdot (+1) = 0 \geq 0 \Rightarrow 1$$

But we need only one output.



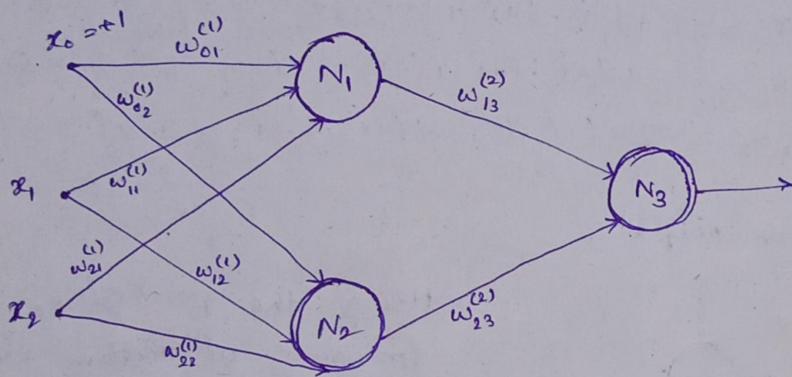
Using the perceptron learning algorithm we can get the weights

# Design NOR gate using Network of Perceptrons

(Approach 2) ←

$x_1$	$x_2$	NOR
0	0	0
0	1	1
1	0	1
1	1	0

$$x_1 \oplus x_2 = (\overline{x_1} \cdot \overline{x_2}) + (\overline{x_1} \cdot x_2)$$



Perception Learning Algorithm ← →

$$\sum_{i=0}^n w_i x_i = w_0 x_0 + w_1 x_1 + \dots + w_n x_n \\ = W^T X$$

$$y=1 \text{ if } W^T X \geq 0$$

$$= 0 \text{ if } W^T X < 0$$

- $W^T X \geq 0$  is a Decision

Boundary

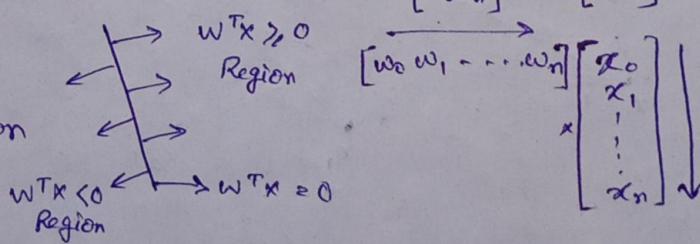
Dot Product

$$A \cdot B = A^T B$$

$$W^T X = W \cdot X$$

$$\text{As, } W^T X = 0 \Rightarrow W \cdot X = 0$$

$$W = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \quad X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$



Dot product  
 $W \cdot X = W^T X \cos \theta$

W and X are orthogonal (angle is 90°)

$$W^T X = 0 \Rightarrow \alpha = 90^\circ$$

$$W^T X > 0 \Rightarrow \alpha > 90^\circ$$

$$W^T X < 0 \Rightarrow \alpha < 90^\circ$$

Angle between two vectors,

$$\cos \alpha = \frac{W \cdot X}{\|W\| \cdot \|X\|} = 0$$

$\alpha$  = angle b/w  $W$  and  $X$

if  $W^T X = 0 \Rightarrow \cos \alpha = 0, \alpha = 90^\circ$

$W^T X > 0 \Rightarrow \cos \alpha > 0, \alpha \leq 90^\circ$

$y = 1$ ,

if  $W^T X > 0 \Rightarrow \cos \alpha > 0 \Rightarrow \alpha \leq 90^\circ$

$y = 0$ ,

if  $W^T X < 0 \Rightarrow \cos \alpha < 0 \Rightarrow \alpha > 90^\circ$

where  $\alpha$  = angle b/w  $X$  and vector  $W$ .

Actual class label is 1

Predicted class label is 0

when,  $W^T X < 0$

$\Rightarrow \cos \alpha < 0$

$\Rightarrow \alpha > 90^\circ$

current angle

to make this a correct prediction we have to reduce  $\alpha$

$\alpha > 90^\circ$

$\downarrow$

$\alpha \leq 90^\circ$

$$W_{\text{new}} = W + X$$

let,

$$W_{\text{new}}$$

$$\alpha_{\text{new}}$$

$$\cos \alpha_{\text{new}} = \frac{W_{\text{new}}^T X}{\|W_{\text{new}}\| \|X\|}$$

$$\cos \alpha_{\text{new}} \propto W_{\text{new}}^T X$$

$$\Rightarrow \cos \alpha_{\text{new}} \propto (W + X)^T X$$

$$\Rightarrow \cos \alpha_{\text{new}} \propto W^T X + X^T X$$

$$\cos \alpha = \frac{W^T X}{\|W\| \|X\|}$$

$$\cos \alpha \propto W^T X$$

$$\cos \alpha_{\text{new}} \propto \cos \alpha + \mathbf{x}^T \mathbf{x}$$

$$\cos \alpha_{\text{new}} > \cos \alpha$$

$$\alpha_{\text{new}} < \alpha$$

## FeedForward Neural Network

$$a_2(x) = b_2 + w_2 \cdot h_1(x)$$

The pre-activation at layer i :

$$a_i(x) = b_i + w_i h_{i-1}(x) \rightarrow \text{This is still linear eqs}$$

But we aim to perform non-linear operation.

The activation at layer i :

$$h_i(x) = g(a_i(x))$$

here, we consider  $g$  as the non-linear operation, to perform non-linearity.

The activation output layer :

$$h_L(x) = f(x) = \hat{y} = o(a_L(x))$$

We consider  $o$  to generate certain output.  
 $g$  and  $o$  may be same.

## ML Gradient Descent Algorithm

3/4/25

Forward Propagation contains all the pre-activation, activation and as well as the final output

$\theta$  contains weights and bias

$$\theta_{t+1} = \theta_t - n\bar{\nabla}\theta_t \rightarrow \text{gradient descent}$$

↳ learning rate

## Forward Propagation Algorithm

## Back Propagation Algorithm

## Unsupervised Learning:

### K-Means Clustering

Record Number			Euclidean Distance		Cluster
	A	B	dist(C <sub>1</sub> , R <sub>i</sub> )	dist(C <sub>2</sub> , R <sub>i</sub> )	
R <sub>1</sub>	1.0	1.0	3.6	5.3	Cluster 1
R <sub>2</sub>	1.5	2.0	2.5	4.2	"
R <sub>3</sub>	3.0	4.0	0	1.80	"
R <sub>4</sub>	5.0	7.0	3.6	2.06	Cluster 2
R <sub>5</sub>	3.5	5.0	1.11	1	"
R <sub>6</sub>	4.5	5.0	1.803	0	"
R <sub>7</sub>	3.5	4.5	0.70	1.11	Cluster 1

K=2



cluster 1



cluster 2

$$C_1 = (3.0, 4.0)$$

$$C_2 = (4.5, 5.0)$$

$$\Rightarrow \{R_1, R_2, R_3, R_7\} \quad \Rightarrow \{R_4, R_5, R_6\}$$

$C_1$  and  $C_2$  are chosen arbitrarily. So, we need to calculate  $C_1$  and  $C_2$  again using K-mean.

$$\text{new } C_1 = \left( \frac{1.0 + 1.5 + 3.0 + 3.5}{4}, \frac{1 + 2 + 4 + 5}{4} \right)$$

$$\Rightarrow (2.25, 2.875)$$

$$\text{new } C_2 = \left( \frac{5 + 3.5 + 4.5}{3}, \frac{7 + 5 + 5}{3} \right)$$

$$\Rightarrow (4.33, 5.66)$$

Again calculate distances,

Record Number	A	B	dist(C <sub>1</sub> , R <sub>i</sub> )	dist(C <sub>2</sub> , R <sub>i</sub> )	
R <sub>1</sub>	1.0	1.0	2.25	5.72	Class. 1
R <sub>2</sub>	1.5	2.0	1.18	4.62	1
R <sub>3</sub>	3.0	4.0	1.35	2.12	1
R <sub>4</sub>	5.0	7.0	4.93	1.49	2
R <sub>5</sub>	3.5	5.0	2.46	1.06	2
R <sub>6</sub>	4.5	5.0	3.09	0.68	2
R <sub>7</sub>	3.5	4.5	2.05	1.42	2

We will stop calculating C<sub>1</sub> and C<sub>2</sub> when the value of both C<sub>1</sub> and C<sub>2</sub> will be equal.

$$\text{Cluster 1} = \{R_1, R_2, R_3\}$$

$$\text{Cluster 2} = \{R_4, R_5, R_6, R_7\}$$

Single Linkage → Closest Distance