ML

25/1/25

Given a dataset for performing linear regression with one variable.

Example: Used car price prediction

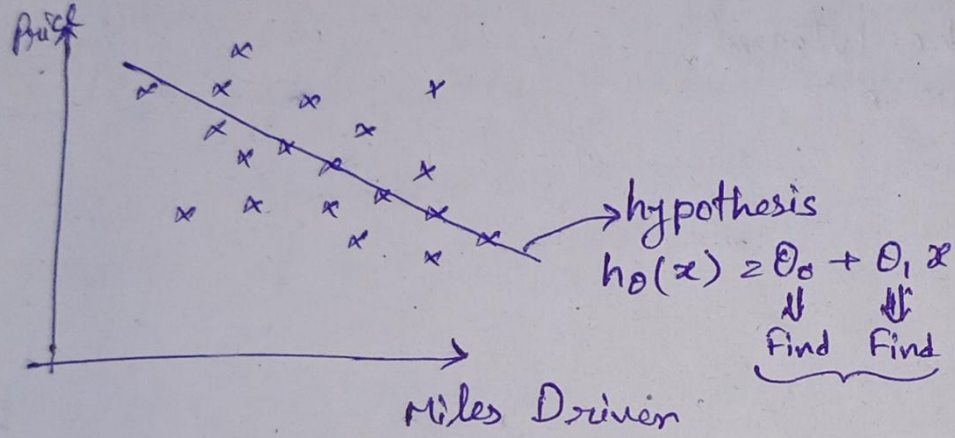
	Features		o/p
I/p →	Miles Driven	Engine Capacity	Price

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

Miles Driven	Price
1000	2.5
2520	3.6
⋮	⋮

let, 2D matrix
(1000, 2.5)
(2520, 3.6)

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

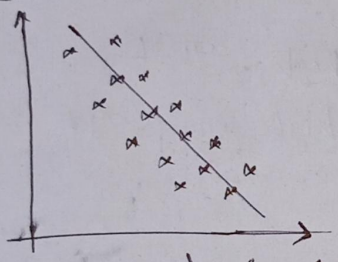


CSV (Comma Separate Vector) can store huge no. of datas in tabular form.

25/1/25

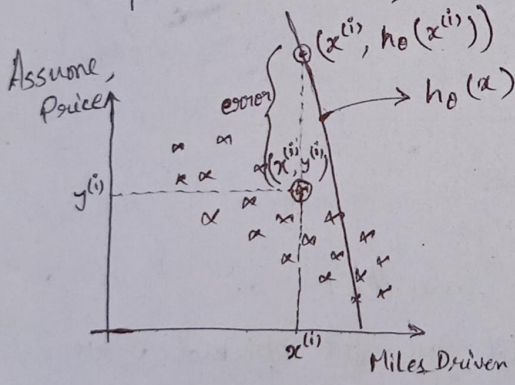
1) What is machine learning?
2) What is learning?
3) What are the different types of m/l? → Supervised learning, → Unsupervised learning, → Semisupervised learning
4) What is Supervised Learning!
5) What type of problem we can solve using Supervised learning!
   ↓ Regression ↓ Classification
6) What is Regression problem explain with a real-life example?

Linear Regression with one variable



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

Initially choose θ_0 and θ_1 randomly.



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

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

~~We only need +ve values~~

The distance can be -ve but we only need magnitude, but we can't do mod(| |). 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.

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

Minimize Error

Minimize $J(\theta_0, \theta_1)$
θ_0, θ_1

$$J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

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

$y = mx + c$ ⇒ slope

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

$$\frac{d\,J(\theta_0, \theta_1)}{d\theta_0} \qquad \frac{d\,J(\theta_0, \theta_2)}{d\theta_1}$$

Updating the value of θ_0, θ_1 :

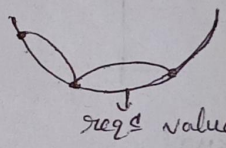
This is called Gradient Decent Algorithm for learning.

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

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

α = Learning rate

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



Steps :

(i) Take Random points
(ii) Calculate errors
(iii) Calculate gradients
(iv) Calculate α - value
(v) Calculate new value for θ_0 and θ_1
(vi) Repeat from step (i) until we get optimal value

Performing Partial Derivative :

For θ_0

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 \qquad h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)} \right)^2$$

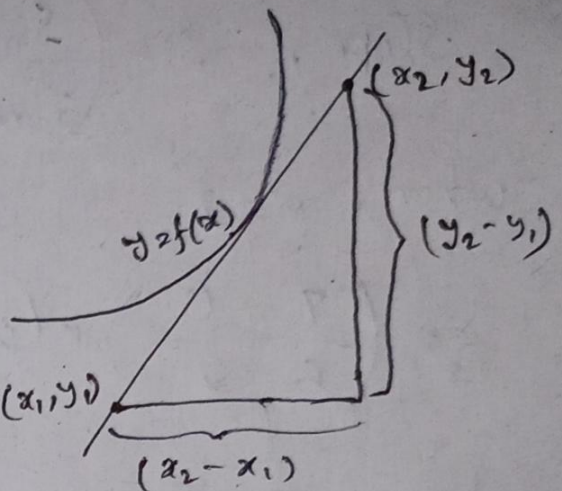
$$\frac{d}{d\theta_0} J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \frac{d}{d\theta_0} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)} \right)^2$$

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

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

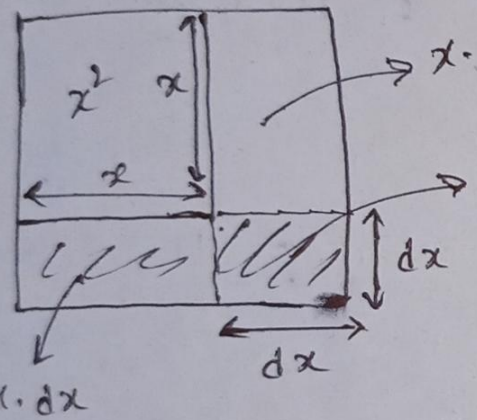
$$= \frac{1}{m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)$$

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



Now, for θ_1

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



$$(x+dx)^2$$
$$= x^2 + x\,dx + x\,dx + dx^2$$
$$\Rightarrow (x+dx)^2 - x^2 = 2x\,dx$$

dx^2 very small value

$$\Rightarrow \frac{(x+dx)^2 - x^2}{dx} = 2x$$

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

30/1/2025

ML

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

Avg. err ↓

Mean Square Error $\longrightarrow \boxed{\frac{1}{m}\sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2}$ → Loss function

(*) Gradiant Descent Algorithm

Linear Regression with multiple features:

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

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

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

Fix $x_1 = 1$

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

$= \theta^T x$

$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$ $\quad x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$

$\begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_n \end{bmatrix}_{(1\times n)} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}_{(n\times 1)}$

Repeat Control Convergent :

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

ONE HOT ENCODING :

- Non-numerical Feature → to → Vector

C	P	D	E
0	0	0	1
0	1	0	0

Fuel Type → Convert to

C	P	D	E

Normalization of Features : / Feature Scaling

[0 - 1]

min (age)
max (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 / Des. / Pet. / Bat.

	C	D	P	B
	0	1	0	0
	1	0	0	0

□ $\leq 0.5 \longrightarrow$ Class 0

□ $> 0.5 \longrightarrow$ 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 fn to determine the class?

- Cost fn and loss fn are same.

Regularization: Overfitting Problem

Under fit, Perfect fit, Overfit

ML

8/2/25

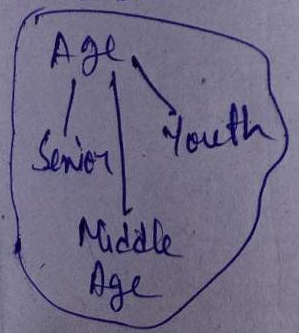
Classification using Decision-Tree

Feautes

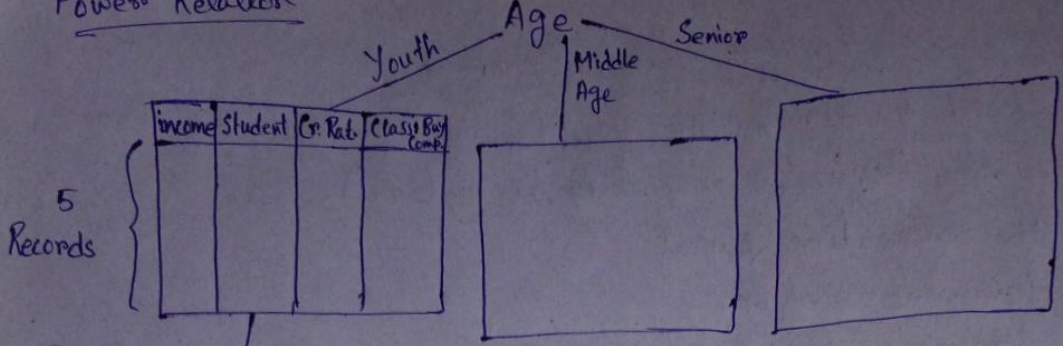
Age, Income, Student? Creadit Rating

The prediction is leasically buy computer or not.

Example :	Age	Income	Student	Credit Rating	buy By Comp.
	Youth	High	No	Fare	No



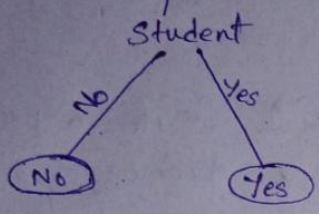
Power Relation



Biased data so, again splitting

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

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



If ~~no~~ after splitting all the 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

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

∴ m = no. of classes

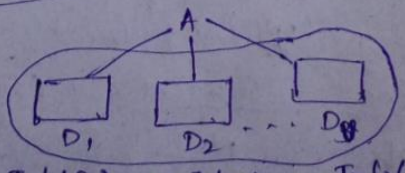
For our case,

m = 2 → Buy computer = Yes 4
p_1 → u n = No
p_2

- The unit of entropy is bits.

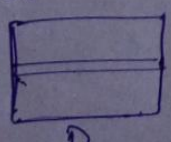
$$IC \propto \frac{1}{P(A)}$$

IC → Information Content



$Info(D_1)$ $Info(D_2)$ ... $Info(D_v)$

$\frac{|D_1|}{|D|}$ $Info_A(D)$



$Info(D)$ $|D_0|$

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

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

$$\boxed{Gain(A) = Info(D) - Info_A(D)}$$

$$Gain(Age) = Info(D) - Info_{Age}(D)$$

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

$$= -P_1 \log_2(D_1) - P_2 \log_2(D_2)$$

$$= -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

$$= -0.64 \log_2(0.64) - 0.36 \log_2(0.36)$$

$$= (-0.64) \times (-0.64) - (0.36) \times (-1.47)$$

$$= 0.288 + 0.3672$$

$$= 0.28$$

$$Info_{Age}(D) = \frac{5}{14}\left[-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right] + \frac{5}{14}\left[-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right] + \frac{4}{14}\left[0 - \frac{4}{4}\log_2\left(\frac{4}{4}\right)\right]$$

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

$$= 0.69285$$

gain(age) = 0.97 − 0.69 = 0.2465

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

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

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

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

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

$$= 0.6928$$

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

13/2/2025

ML

Entropy

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

m: no. of classes

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

$$Gain(A) = Info(D) - Info_A(D)$$

$$Split\ Info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \cdot \log_2 \frac{|D_j|}{|D|}$$

According to Income attribute

$$= -\sum_{j=1}^{3} \frac{|D_j|}{|D|} \cdot \log_2 \frac{|D_j|}{|D|}$$

$$= -\frac{|D_1|}{|D|} \cdot \log_2 \frac{|D_1|}{|D|} - \frac{|D_2|}{|D|} \cdot \log_2 \frac{|D_2|}{|D|} - \frac{|D_3|}{|D|} \cdot \log_2 \frac{|D_3|}{|D|}$$

$$= -\frac{4}{14} \log_2 \frac{4}{14} - \frac{6}{14} \log_2 \frac{6}{14} - \frac{4}{14} \log_2 \frac{4}{14}$$

~~1.55644~~ $= 1.5564$

~~20.155475 + 0.157780 + 0.155475~~

$$Gain\ Ratio(A) = \frac{Gain(A)}{Split\ Info(A)}$$

$$= \frac{0.28}{1.5564} = 0.1799$$ Ans

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

$$Gain_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

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

$y^{(i)}$ ⇓ Actual class

$h_\theta(x^{(i)})$ ⇓ Predicted class

Binary cross entropy loss

$$\left(\sum_{i=1}^{m} y^{(i)} \log_2 h_\theta(x^{(i)})\right) + \sum_{i=1}^{m} (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))$$

Binary cross entropy for +ve class

Binary class entropy for -ve class

ML

Receiver Operating Characteristics (ROC) Curve

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

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

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

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

Consider T_1 as thresold value

$T_1 \longrightarrow S_1 \longrightarrow$ TPR (True Positive Rate)

$\qquad\qquad\qquad\quad \longrightarrow$ FPR (False Positive Rate)

ML

20/2/2025

SVM

class 1 ⩾ 1

class 0 ⩽ -1

Support Vector Machine :

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

(1) $\left(\sum_{i=1}^{n}\left(x_i^{(2)}-x_i^{(1)}\right)^2\right)^{1/14} \longrightarrow nD$

$\longrightarrow$ Eucledian Distance/(L2)

(2) Manhattan Distance/(L1) Distance/City Block Distance

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

nD →

$L_1 = |x_1^{(2)}-x_1^{(1)}| + |x_2^{(2)}-x_1^{(1)}| + \ldots + |x_n^{(2)}-x_n^{(1)}|$

$= \sum_{i=1}^{n} |x_i^{(2)}-x_i^{(1)}|$

KNN

Lazy Learner → no learning in historical data, but learn in prediction data.

- Solve Classification problem using KNN:
- Solve regression problem using KNN:

ML

27/2/25

- KNN
- SVM
- Importance of Bayes Theorem

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

$$P(A/B) = \frac{P(A \cap B)}{P(B)} \rightarrow \text{Hearing of that formula} \rightarrow (1)$$

↓ Here B event at first occured

(means P(A) depends on B occurance)

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

$$\Rightarrow P(B/A) = \frac{P(A \cap B)}{P(A)} \rightarrow (ii)$$

From (i) & (ii)

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

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

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

Because here we consider every single event.
(Total Probability Theorem)

Commutative Law

$$A \cup B = B \cup A$$
$$A \cap B = B \cap A$$

Diff. between Probability & Set

Distributive Law

Here we can distribute → $A \cap B$ & $A \cap C$

↳ Here we use Base Theorem.

- **Bayes Theorem** :—

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

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

ML

1/3/25

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

Sol:)

Brightness(x_2)	Saturation(y_2)	Class
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.

Euclidian Distance,

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

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

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

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

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

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

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

Choos K=3 (three) min. values.

No. of Red Samples = 2

No. of Blue Samples = 1

(20, 35) ⇒ Color Red

If K = 5,

No. of Red Samples = 3

No of Blue Samples = 2

(20, 35) ⇒ Color Red

Support Vector Machine (SVM)

(*) What is the role of support vector in SVM?

For parallel line,

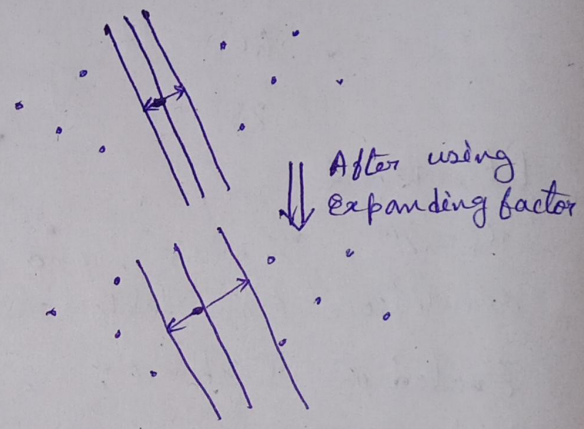
$a_1x + b_1y + c_1 = 0$

$a_2x + b_2y + c_2 = 0$

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

Expanding factor → 1

0.99



let,

Expanding factor = 0.7

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

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

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 near to 1.

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

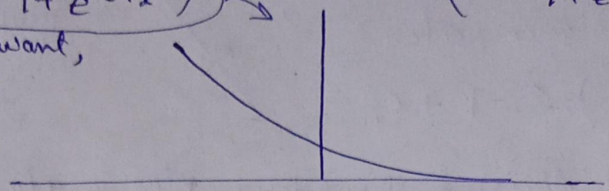
$h_\theta(x)$ < 0.5 $y = 0$

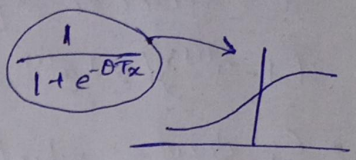
≥ 0.5 $y = 1$ } Depending on probability

$\theta^T x \geq 0 \quad y=1$
$\theta^T x < 0 \quad y=0$
} Depending on Data Boundary

$$-y \log\left(\frac{1}{1+e^{-\theta^T x}}\right) - (1-y)\log\left(1-\frac{1}{1+e^{-\theta^T x}}\right)$$

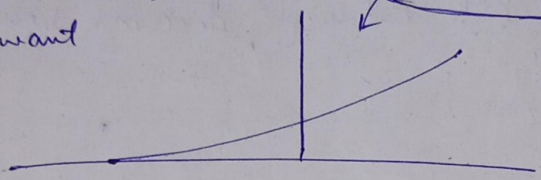
If $y=0$ we want, $\theta^T x \ll 0$





$$-y \log\left(\frac{1}{1+e^{-\theta^T x}}\right) - (1-y)\left(-\log\left(1-\frac{1}{1+e^{-\theta^T x}}\right)\right)$$

If $y=0$ we want $\theta^T x \gg 0$



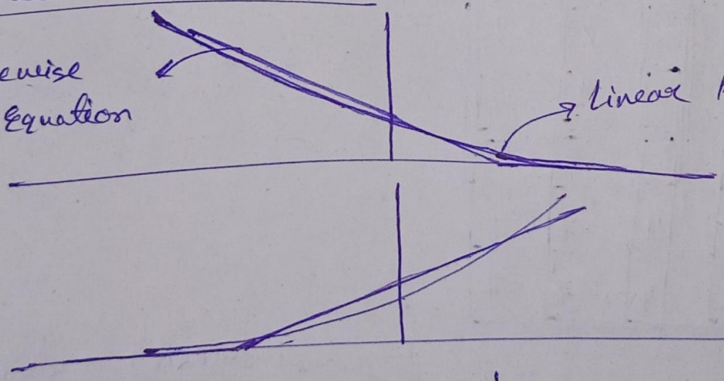
Logistic Regression

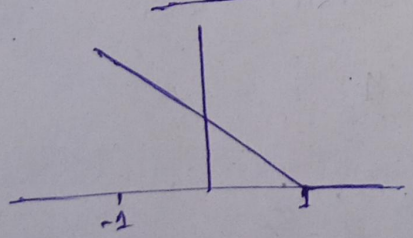
$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}(-\log h_\theta(x)) + (1-y^{(i)})(-\log(1-h_\theta(x)))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Support Vector Machine

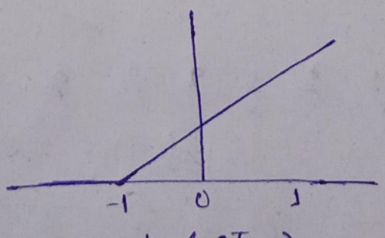
$$\min_\theta C\left[\sum_{i=1}^{m} y^{(i)}(\text{cost}_1(\theta^T x^{(i)})) + (1-y^{(i)})(\text{cost}_0(\theta^T x^{(i)}))\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

Piecewise Linear Equation

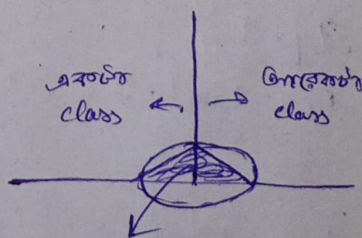




$\text{cost}_1(\theta^T x)$



$\text{cost}_0(\theta^T x)$



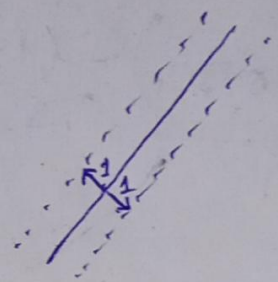
No Man Land
No Data Item can present there.

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

$$\min_{\gamma, w, b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \epsilon_i \quad \text{s.t.} \quad y^{(i)}(w^T x^{(i)} \geq 1 - \epsilon_i \quad i = 1, \ldots, m$$
$$\epsilon_i \geq 0, \quad i = 1, \ldots, m$$

ϵ_i = Error

w = vector norm / distance vector



$\frac{2}{\|w\|}$

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

- ϵ_i decides whether decision boundary ~~plac~~ is placed in proper location or not.
- $\|w\|$ ↓ increases the gap between the decision boundary and the support vector

ML

20/3/2025

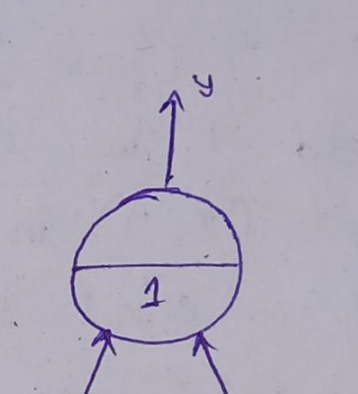
Artificial Neural Network — MP Neuron

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

→

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

$\theta = 1$



$f(g(x)) = 1$ if $g(x) \geq 1$

$= 0$ if $g(x) < 1$

Qn: How can we degin AND Gate.

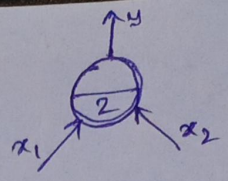
x_1	x_2	x_1 and x_2	$g(x) = \sum x_i$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	2

$\theta = 2$

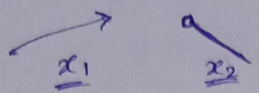
~~$f(g(x)) = 1$ if $g(x)$~~

$f(g(x)) = 1$ if $g(x) \geq 2$

$= 0$ if $g(x) < 2$



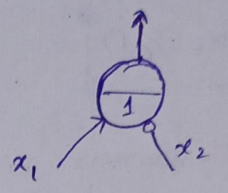
$x_1 \cdot \bar{x}_2$



x_1	x_2	$\bar{x}_2$	$x_1 \cdot \bar{x}_2$	$\sum_{i=1}^{2} x_i$
0	0	1	0	0
0	1	0	0	1
1	0	1	1	1
1	1	0	0	2

$y \in \{0, 1\}$

$\theta = 1$

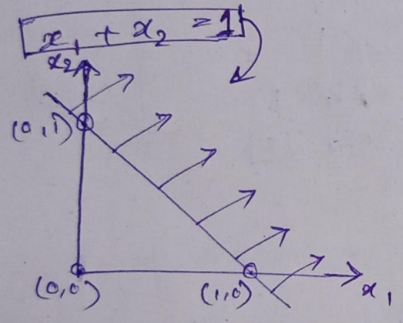


x_1 AND $!x_2$

For two i/p OR Gate

$y = f(g(x)) = 1$ if $x_1 + x_2 \geq 1$

$= 0$ if $x_1 + x_2 < 1$



Perceptrons

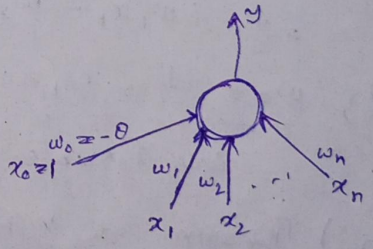
$g(x) = x_1 \cdot w_1 + x_2 \cdot w_2 + \ldots + x_n \cdot w_n$

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

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

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

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



Perceptrons

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

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

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

$$w_1 x_1 + w_2 x_2 + \ldots\ldots + w_n x_n \geq \theta$$

$$w_1 x_1 + w_2 x_2 + \ldots\ldots + w_n x_n - \theta \cdot 1 \geq 0$$

(θ → w_0, 1 → x_0)

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

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

For MP neuron no learning is present, its a static model and we handpicked θ values. But in case of perceptrons we have to basically train the θ values.

Qn:) Draw OR gate using perceptron.

x_1	x_2	$x_1 + x_2$	$g(x)$
0	0	0	$w_0 + w_1 \cdot 0 + w_2 \cdot 0 = \boxed{w_0 < 0}$
0	1	1	$w_0 + w_1 \cdot 0 + w_2 \cdot 1 = \boxed{w_0 + w_2 \geq 0}$
1	0	1	$w_0 + w_1 \cdot 1 + w_2 \cdot 0 = \boxed{w_0 + w_1 \geq 0}$
1	1	1	$w_0 + w_1 \cdot 1 + w_2 \cdot 1 = \boxed{w_0 + w_1 + w_2 \geq 0}$

$$g(x) = w_0 x_0 + w_1 x_1 + w_2 x_2$$
$$= w_0 + w_1 x_1 + w_2 x_2$$

Let, $w_0 = -1$, $w_1 = 1.1$, $w_2 = 1.1$

One Possible Solution

Qn:) Design a XOR gate using perceptron.

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

ML

22/3/2025

Implement XOR using a single perceptron —

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

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

XOR is not linearly separable 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 perceptron.

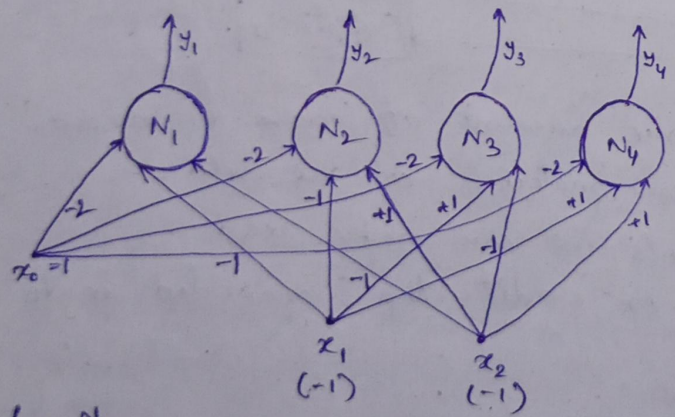
XOR using Network of Perceptrons —

For Input

Binary $0 \Rightarrow -1$

$1 \Rightarrow +1$

	x_1	x_2	XOR	AIM 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



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_1,

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

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

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

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

for N_2,

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

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

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

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

for N_3,

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

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

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

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

for N_4,

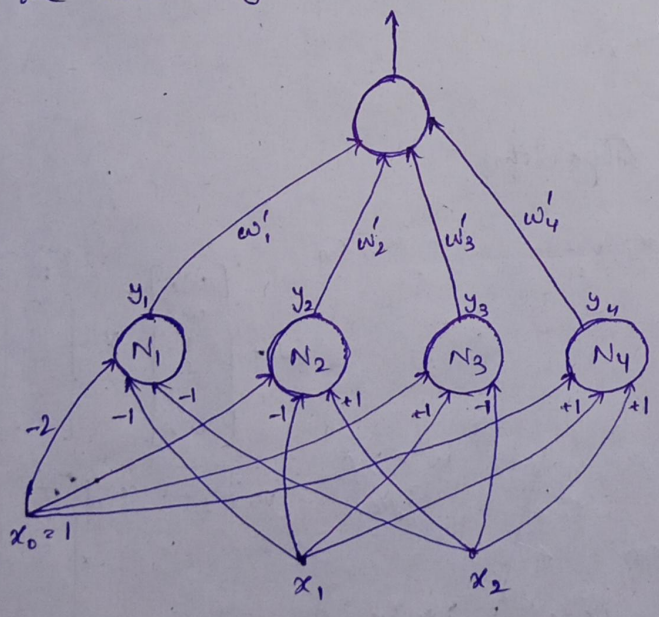
$y_1 = w_0 + w_1 x_1 + w_2 x_2 = -2 + (-1)\cdot(+1) + (-1)\cdot(+1) = -4 < 0 = 0$

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

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

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

But we need only one output.



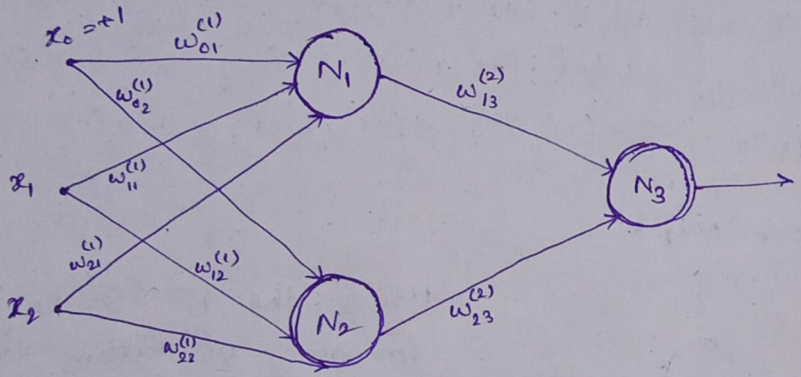
Using the perceptron learning algorithm we can get the weights

Design XOR gate using Network of Perceptrons

(Approach 2) →

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

$x_1 \oplus x_2 = \underbrace{x_1\overline{x_2}}_{N_1} + \underbrace{\overline{x_1}\cdot x_2}_{N_2}$



Perceptron Learning Algorithm ←

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

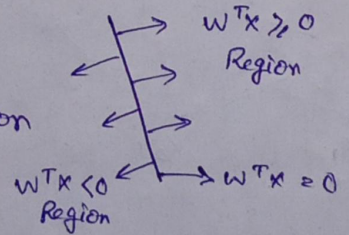
$$= W^T X$$

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

$= 0$ if $W^T X < 0$

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

$$[w_0\ w_1\ \dots\ w_n] \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$



- $W^T X = 0$ is a Decision Boundary

Dot Product

$A \cdot B = A^T B$

$W^T X = W \cdot X$

As, $W^T X = 0 \Rightarrow \boxed{W \cdot X = 0}$ → W and X are orthogonal (angle is 90°)

Dot product

$\boxed{W \cdot X = WX \cos\theta}$

$w^T x = 0 \Rightarrow \alpha = 90°$

$w^T x \geq 0 \Rightarrow \alpha \geq 90°$

$w^T x < 0 \Rightarrow \alpha < 90°$

Angle between two vectors,

$$\cos\alpha = \frac{w \cdot x}{\|w\| \cdot \|x\|} = 0$$

α = angle b/w w and x

if $w^T x = 0 \Rightarrow \cos\alpha = 0, \quad \alpha = 90°$

$w^T x \geq 0 \Rightarrow \cos\alpha \geq 0, \quad \alpha \leq 90°$

$y = 1,$

if $w^T x \geq 0 \Rightarrow \cos\alpha \geq 0 \Rightarrow \alpha \leq 90°$

$y = 0,$

if $w^T x < 0 \Rightarrow \cos\alpha < 0 \Rightarrow \alpha > 90°$

where α = angle b/w x and current w.

Actual class label is 1

Predicted class label is 0

when, $w^T x < 0$

$\Rightarrow \cos\alpha < 0$

$\Rightarrow \alpha \geq 90$ (current angle)

→ to make a correct prediction we have to reduce α

$\alpha \geq 90°$

$\alpha \leq 90$

$w_{new} = w + x$

let,

w_{new}

α_{new}

$$\cos\alpha_{new} = \frac{w_{new}^T x}{\|w_{new}\| \|x\|}$$

$$\cos\alpha = \frac{w^T x}{\|w\| \cdot \|x\|}$$

$$\cos\alpha \propto w^T x$$

$\cos\alpha_{new} \propto w_{new}^T x$

$\Rightarrow \cos\alpha_{new} \propto (w + x)^T x$

$\Rightarrow \cos\alpha_{new} \propto w^T x + x^T x$

$$\cos\alpha_{new} \propto \cos\alpha + X^T X$$

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

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

ML

27/3/2025

Feed Forward Neural Network

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

The pre-activation at layer i:

$a_i(x) = b_i + w_i h_{i-1}(x)$ → This is still linear eqn

But our 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.