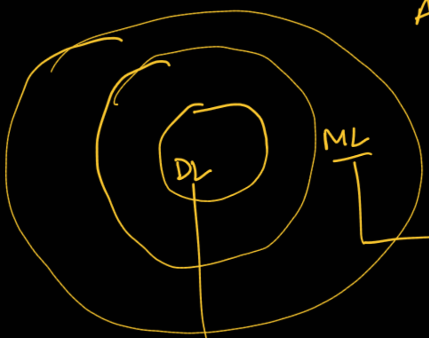
Day-1

June 5, 2024

Shashwat Bhardwaj

✓ What is ML? → Finding patterns inside data and generalize them to unseen instances.

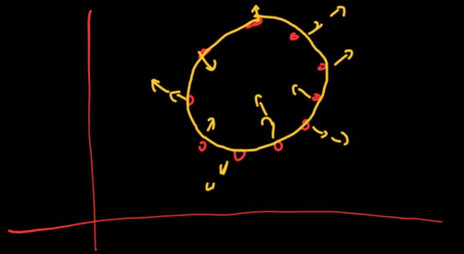
AI ML DL

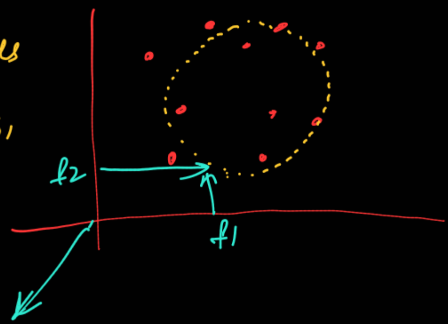


AI (superset)
↳ Robotics and many such fields

ML → algos like Lin Reg, Log Reg, SVM(s) etc.

DL → "neural networks"





Whenever I have data → I have "features"

For eg: I take housing pricing predictions.

↳ I have 10 houses. On the basis of those 10 houses, I need to generalize & find on what pattern the housing prices are decided.

	f1 (# carpet area)	f2 (# of rooms)	. . .	f6 (# floors)	Cost (₹)
h1	~~~	~~~	~~~	~~~	•
h2	~~~	~~~	~~~	~~~	•
⋮					
h10	~~~	~~~	~~~	~~~	•

Algorithms

1) Supervised ... Unsupervised ***

(T.S.) Supervised Learning | Semi-Sup Learning | Unsupervised Learning

Regression ← Supervised Learning

Classification

Supervised Learning → I explicitly have labels → (Housing Prediction)

features $f1 \ldots f6$ → Cost (labelling)

$f1 \ldots f6$ ₹1Cr

$f1 \ldots f6$ ₹1.5Cr

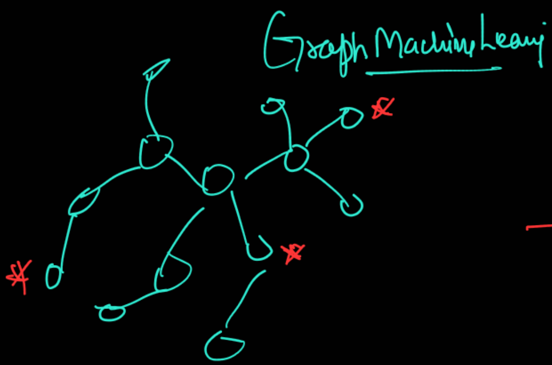
Semi-Sup Learning → I have very few labels

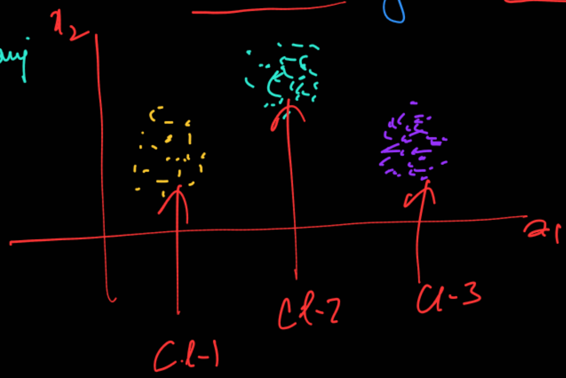
Unsupervised Learning → I do not have labells. → Labells (X)

$\boxed{f1 \ldots \ldots f6}$

(To capture underlying patterns inside data)

* Clustering → $\boxed{\text{KMeans}}$





Regression : I have to predict some continuous regressed labels

Classification : I have to predict discrete labels

✓ Regression → House Price, Score (0 - 100), Metric of happiness

✓ Classification → Disease (Yes / No), 2 labels

4 emotions

(0) V-Happy, Happ (1), Sad (2), V.Sad (3)

"discrete"

200	0.2
300	0.3
400	0.4
100	0.1

= 1

Supervised Learning Algorithm

Notation: data = $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$ → I have m examples in my dataset

features — Labels

[ ]

$f \in \mathbb{R}^n / D^n$ (e.g.) — $\in \mathbb{R}^K$ (Regression) — $\{1 \dots K\}$ (Classification)

"It may be possible we are predicting a vector" (may be possible?)

$\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$

→ generally this is some real no

Example:

Pixel
Region
what is density

features

	X_1	X_2	X_3	Y
1	$[x_{11}$	x_{12}	$x_{13}]$	y_1
2	$[x_{21}$	x_{22}	$x_{23}]$	y_2
3	$[x_{31}$	x_{32}	$x_{33}]$	y_3

Labells

$[x_{11}$	x_{12}	$x_{13}]$	y_1	$[y_{11}$ y_{12} $y_{13}]$
$[x_{21}$	x_{22}	$x_{23}]$	y_2	$[y_{21}$ y_{22} $y_{23}]$
$[x_{31}$	x_{32}	$x_{33}]$	y_3	$[y_{31}$ y_{32} $y_{33}]$

$\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$ # m examples

(ith vector)

Our data is assumed to be i.i.d. → identically & independently distributed

means.

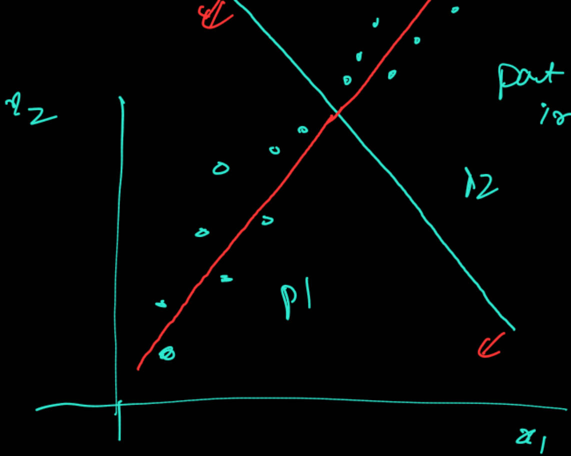
Occurrence of any data point is not dependent on occurrence of any other data point, and we assume that all data points are coming from a common distribution. (There is some distribution which is sitting and generating these data points).

Objective: I want to learn a mapping from $X \to Y$
(mat) (labels)

is a mapping from X to Y $h_\theta : X \to Y$ → here we are approximating the actual function.

I know that the

I am hoping that, a part of the line represented in in P1 is a good approximator of data distribution (in future also).



Training Set

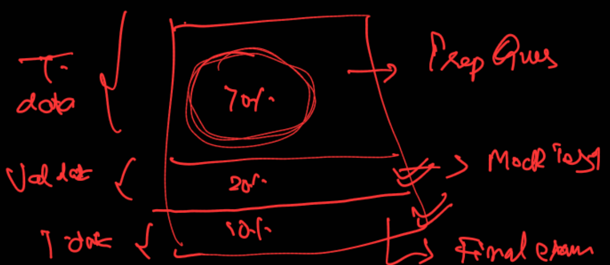
Testing Set

Actually: Train : Learning through books

(Val) Dev : Mock Test

Test : Final Exam

70-20-10 scale



$h_\theta : X \to Y$ I am approximating my actual function with h_θ.

$\nabla_\theta \left((X\theta - Y)^T (X\theta - Y) \right)$

These θ's are called as learnable parameters.

$\theta_0 x^0 + \theta_1 x \Rightarrow$ st. line (2d)

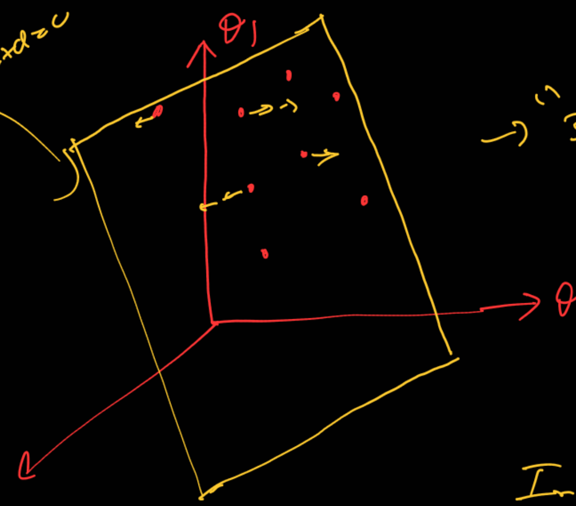
$\theta_0 x^0 + \theta_1 x_1 + \theta_2 x_2 \Rightarrow$ hyperplane

$\cancel{X_1}$	$\cancel{X_2}$	$\cancel{X_3}$	Actual (y)	Predicted ($\theta^T x$)
1. x_{11} (02)	x_{12} (0p)	x_{13} (04)	y_1^a	$y_1^p = \theta_0 + \theta_1 x_{11}^{(1)} + \theta_2 x_{12}^{(1)}$
2. x_{21}	x_{22}	x_{23}	y_2^a	$y_2^p = \theta_0 + \theta_1 x_{21}^{(2)} + \theta_2 x_{22}^{(2)}$

$\theta_0 + \theta_1 x$

$c + mx$

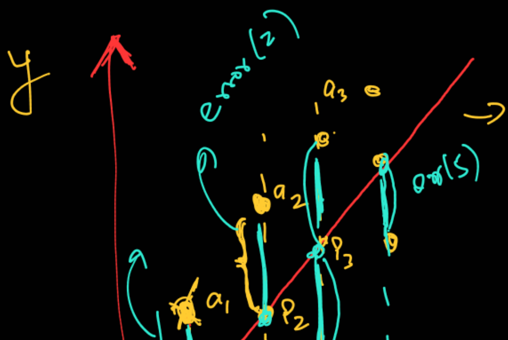
$ax + by + cz + d = 0$



→ "3d"

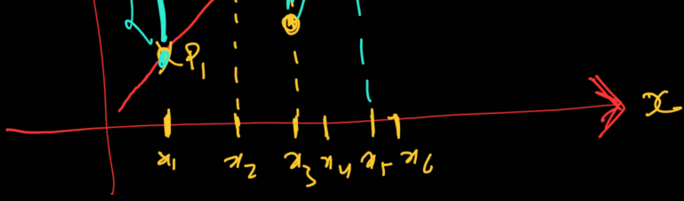
$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

In other words they denote how important one feature is to decide our final label. (weights) (coefficient)



→ "best line fit"

... corresponding too



→ θs → corresponding line must be best for all data points.

How would we signify it

↳ If I am able to averagely compute all θ's then my task is done.

→ $\theta_0 + \theta_1 x_1$ ✓

If I try to reduce average error of all points, then the task would be done.

→ We take some error function → (Loss)

"USER CHOICE"

e_1: $(A_1 - P_1)^2 = (y_1 - (\theta_0 + \theta_1 x_1))^2$

e_2: $(A_2 - P_2)^2 \quad (y_2 - (\theta_0 + \theta_1 x_2))^2$

⋮

e_m: $(A_m - P_m)^2 \quad (y_m - (\theta_0 + \theta_1 x_m))^2$

MSE - Loss

$$\text{MSE-loss} = \frac{e_1^2 + e_2^2 + e_3^2 + \cdots e_m^2}{m}$$

$$\text{Avg. error} = \frac{(A_1 - P_1)^2 + (A_2 - P_2)^2 + \cdots (A_m - P_m)^2}{m}$$

$$\text{Avg error} = \frac{(y_1 - (\theta_0 + \theta_1 x_1))^2 + (y_2 - (\theta_0 + \theta_1 x_2))^2 + \cdots (y_m - (\theta_0 + \theta_1 x_m))^2}{m}$$

"one"

$\theta_0 1 + \theta_1 x_1$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} \leftarrow \begin{bmatrix} \theta_0 & \theta_1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \end{bmatrix}$$

$\theta^T x_1$ ✓ $\qquad \theta^T x_2 \qquad \theta^T x_m$

m examples $\qquad$ # n features.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right)^2 \quad \text{where } h_\theta(x^{(i)}) = \theta^T x^{(i)}$$

↳ Hypothesis function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - (\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \theta_3 x_3^{(i)} + \dots + \theta_m x_m^{(i)})\right)^2$$ ← Regression

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Loss(y^a, y^p)$$

I have to decrease my average Loss.

Minimize

$$\rightarrow \min : J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

$$\min : \frac{J(\theta)}{2} = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

$J(\theta)$

Differentiate w.r.to θ :

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

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - (\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \dots + \theta_n x_n^{(i)})\right)^2$$

$(\theta_0 \dots \theta_n)$

I have to diff w.r.to every parameter.

Because in every dim I will get an optimal θ_j

vec X

mat X

Loss - thetas

$n=3$

3 thetas.

$$\sum_{j=0}^{n}\theta_j x_j$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - (\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \theta_j x_j^{(i)} + \theta_n x_n^{(i)})\right)^2$$

Chain

If I am taking derivative w.r.to some sp. θ_j

$$\Rightarrow \frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{2m}\sum_{i=1}^{m} 2\left(y^{(i)} - (\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} \cdot \theta_n x_n^{(i)})\right) \cdot \left(0 - x_j^{(i)}\right)$$

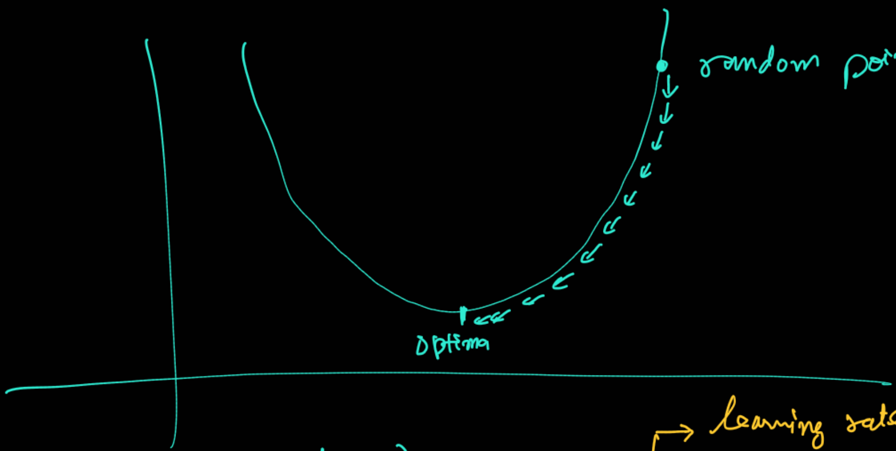
$$F = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

$$\frac{dF}{d\theta_2} = x_2$$

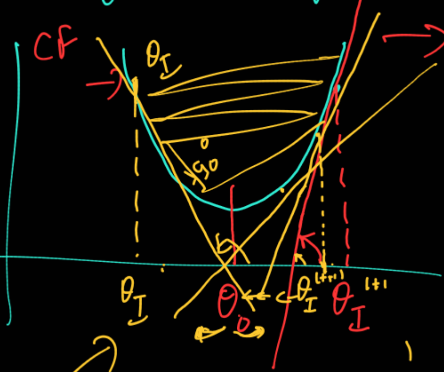
$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \Big(y^{(i)} - \underbrace{(\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \dots + \theta_n x_n^{(i)})}_{h_\theta(x^{(i)})}\Big) \cdot x_j^{(i)}$$

$$\boxed{\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}}$$

Gradient Descent



Parameter: θ $\longrightarrow$ $\theta_j^{(t+1)} := \theta_j^{(t)} - \eta \cdot \left(\nabla_j J(\theta)\right)$ ← learning rate = 0.001 0.002



→ derivative tangent $(\nabla J(\theta))$

$\theta_I^{(random)} > \theta_0^{(optima)}$

I should substract something from θ_I to go to θ_0.

$\theta^{(t+1)} = (\theta_I^{(t)} - +ve)$

tangent $< 90^\circ$

$\tan\theta \rightarrow +ve$

$\theta_I < \theta_0$

$\theta_I - (-ve)$

$\tan(\theta)$ $-ve$

$\theta^{(t+1)} > \theta^{(t)}$

θ_I

$$\theta_j^{(t+1)} = \theta_j^{(t)} - \eta \cdot \left(\nabla_{\theta_j} J(\theta)\right)$$

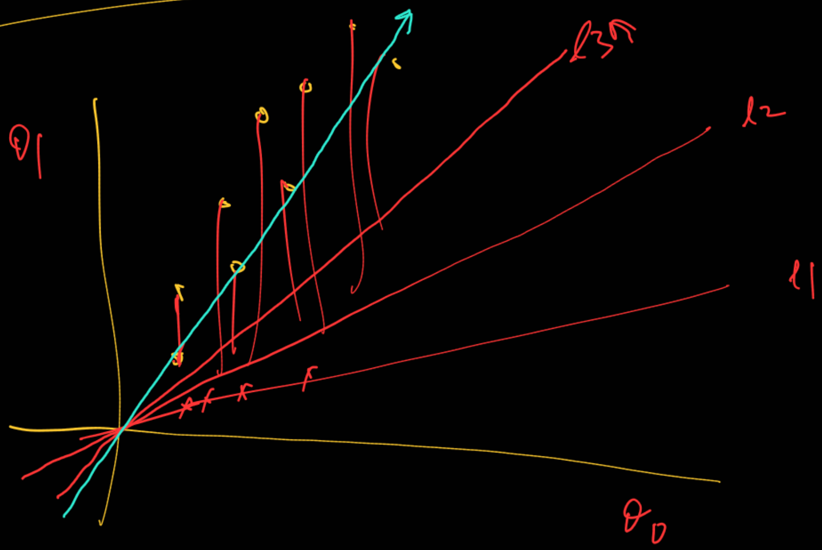
η: Hyperparameter here.

$$\boxed{\theta_j^{(t+1)} = \theta_j^{(t)} - \eta \cdot \left(\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}\right)}$$

After some steps I will have Convergence!!!

Linear Regression + G.D

Gradient descent.



Optimality

At optimal pt, my gen error is minimal.