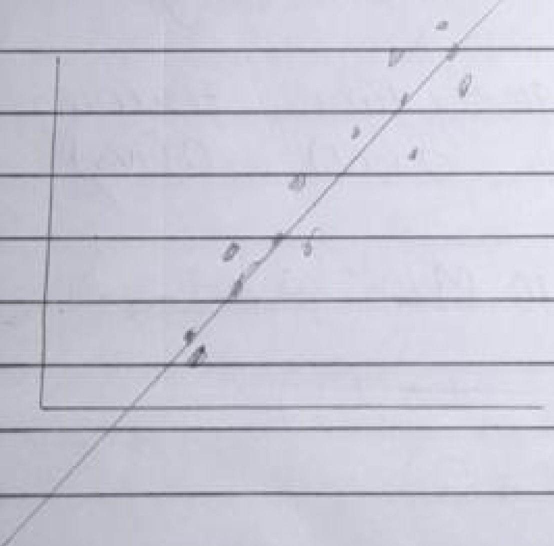
Linear Regression

→ input and output are continuous
→ Type of supervised learning
→ We supply labelled X and y as training data
we observe patterns in training data and try
to predict outcome on test data using this.
like in this example :-

Training data		Testing data	
X_train	Y_train	X_test	Prediction
Time spent (hr)	(marks)	Time spent)	?
2	4	5	?
3	7	9	?
10	19		
12	25		
8	15		



$f(x) \rightarrow y$

$y = f(x)$ → linear function

equation of line $y = \underset{\text{slope}}{m}x + \underset{\text{intercept}}{c}$

$y_{Test} = m x_{Test} + C$

y_{Test} ↓ prediction

main goal - learn best value of m & c

notation :- $y = \underset{\text{slope}}{\theta_1} x + \underset{\text{intercept}}{\theta_0}$

Date :

Training Data

↓

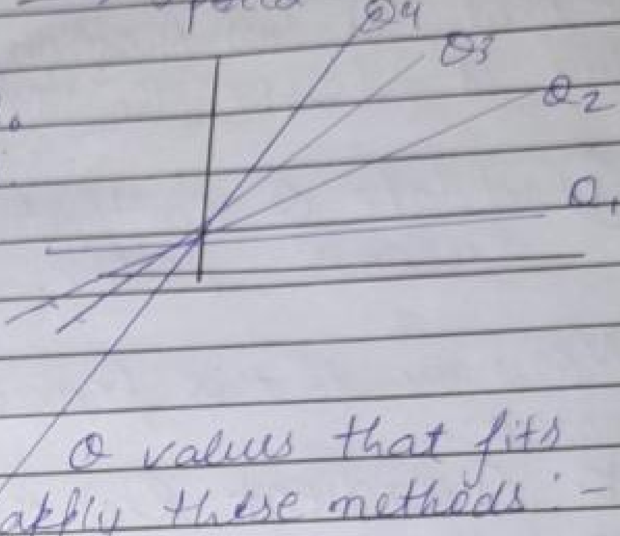
Algorithm

↓

X Test ⟶ Hypothesis $f(x)$ ⟶ Y_{pred}

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

Goal $\theta = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}$ learn this



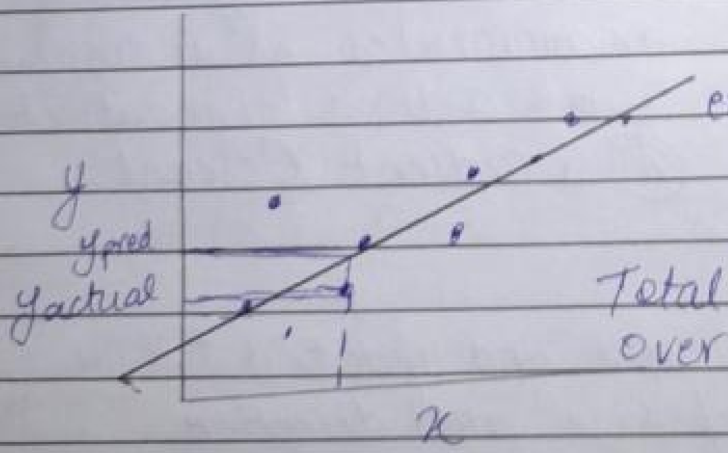
We have 4 lines which have different values of $\theta = [\theta_1, \theta_0]$ so our goal is to find the best θ values that fits the model best. We apply these methods:-

1. θ = initialize θ with random values
2. Measure how good is our θ

Best θ is the one for which error is the least

How do we find error?

$$E^{(i)} = |y^{i}_{actual} - y^{i}_{predict}|$$



error at i^{th} pt $E^{i} = |y^{(i)} - \hat{y}^{(i)}|$ error for 1 ex

Total Error over all ex : $\sum_{i=1}^{m} |\hat{y}^{i} - y^{i}|$

m = total no of ex in training data

Avg error $= \frac{1}{m} \sum_{i=1}^{m} |\hat{y}^{i} - y^{i}|$

we do not use absolute error as this function is not differentiable

Mean Square Error :- $\frac{1}{m} \sum_{i=1}^{m} [\hat{y}^i - y^i]^2$
(loss function)
$\downarrow$
$J(\theta)$

$\hat{y}^i (pred) = h_\theta (x^i)$
$= \theta_1 x^i + \theta_0$

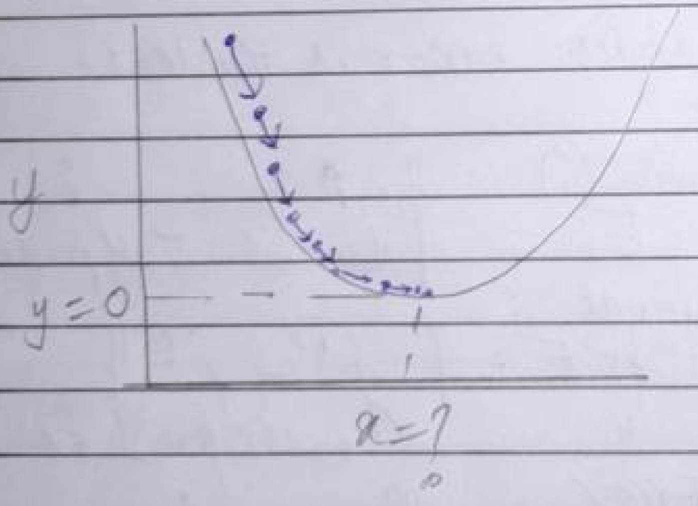
we need this loss function to measure how good our line is. lesser the $J(\theta)$ better the θ values.

③ update θ so that $J(\theta)$ reduces and θ becomes better.

Now we do this pracedure using an algorithm called :-

Gradient Descent :-

→ helps to find a local minima in any given function



Find x where $y = f(x)$ is minimum

① we can directly differentiate it but this not ~~differentiable~~ recommended as it doesn't work well on large dataset

② Gradient Descent

we choose a random value of x and update it till we reach minima depends on step size and direction

$$x = x - \underset{\text{step size}}{\overset{\text{const}}{\eta}} \overbrace{\underbrace{\frac{\partial f(x)}{\partial x}}_{\text{direction}}}^{\text{gradient}}$$

if ∂x is very small that means you're close to minima whatever be the starting point you'll always end up at local minima in a convex function.

For concave

$$x = x + \eta \frac{\partial f x}{\partial \partial x}$$

Gradient Descent Update for Linear Regression (Univariate)

(1) θ = initialize θ randomly

(2) $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} [\hat{y}^i - y^i]^2$

(3) update θ using gradient descent algorithm

$\theta = [\theta_0 \ \theta_1]$

$\hat{y}^i = h_\theta(x^i) = \theta_0 + \theta_1 x^{(i)}$ (prediction)

Update $\theta = \theta - \eta \ \nabla_\theta J(\theta)$

[$m = m -$ [learning rate $\times \partial/\partial m$] $c = c - l.r \times \partial f/\partial c$

$\theta = [\theta_0 \ \theta_1]$

$$\theta_0 = \theta_0 - \eta \frac{\partial J(\theta)}{\partial \theta} \quad \text{—(i)}$$

$$\theta_1 = \theta_1 - \eta \frac{J\theta}{\partial\theta} \quad - (2)$$

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

(1) $$\frac{\partial J(\theta)}{\partial\theta_0} = \frac{2}{m} \sum_{i=1}^{m} [\theta_0 + \theta_1 x^i] - y^i] \times 1$$

$$= \frac{2}{m} \sum_{i=1}^{m} [\hat{y}^i - y^i]$$

(2) $$\frac{\partial J\theta}{\partial\theta_1} = \frac{2}{m} \sum_{i=1}^{m} [\hat{y} - y^i] \times x^i$$

$$= \frac{2}{m} \sum_{i=1}^{m} (\hat{y} - y^i) x$$

Finally updation

$$\theta_0 = \theta_0 - \eta \frac{1}{m} \sum_{i=1}^{m} [\hat{y}^i - y^i]$$

$$\theta_1 = \theta_1 - \eta \frac{1}{m} \sum_{i=1}^{m} [\hat{y}^i - y^i] x^i$$