Simple Linear Regression

Supervised ML → Regression O/p → Continuous
→ Classification O/p → Binary, multiclass categories

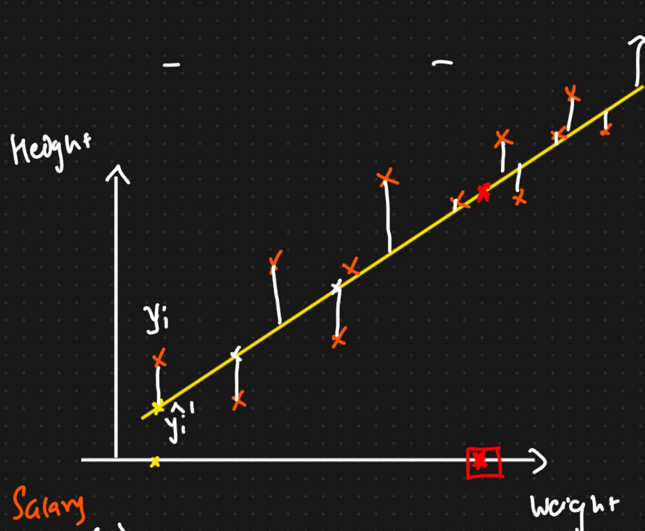
Dataset

Independent feature Dependent feature

Weight	Height
74	170
80	180cm
75	175.5cm
-	-
-	-

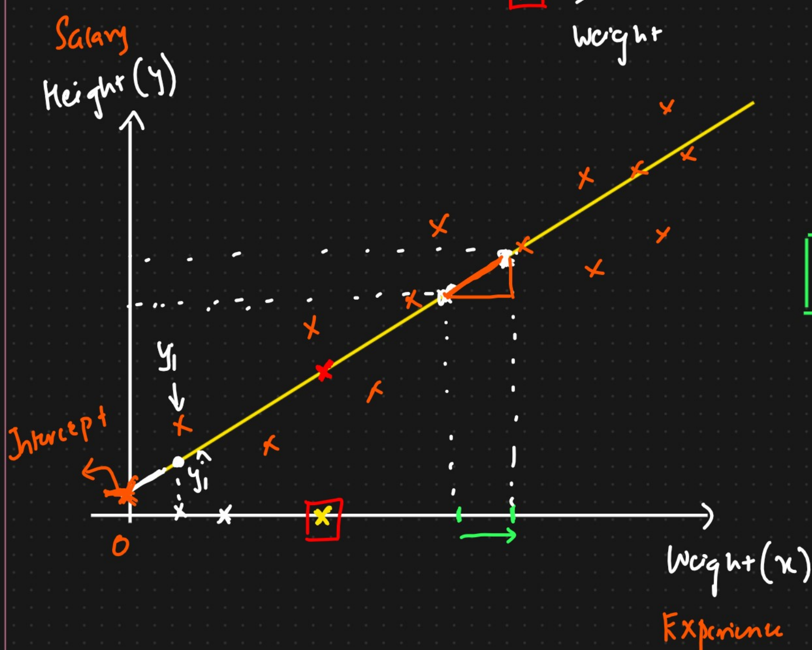
TRAIN ↓

New Weight → Model → Height



Σ Error → Minimal

Geometrical Intuition



$\hat{y} = mx + c$

$\hat{y} = \beta_0 + \beta_1 x$

we can change θ_0 and θ_1 so that we get best fit line

⇓

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

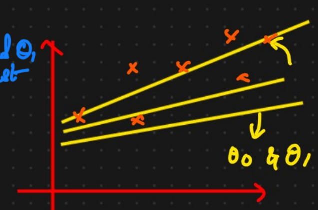
θ_0 = Intercept

θ_1 = Slope or coefficient

x_1 ⇒ data points

Predicted ↓

Error $\{y_i - \hat{y}_i\}$



Error ⇓ θ_0 & θ_1

⇓

Optimization Process

Cost function

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} \Big(\underbrace{y_i}_{\text{actual}} - \underbrace{h_\theta(x)}_{\text{prediction}} \Big)^2 \Rightarrow \boxed{\text{Mean Squared Error}}$$

We need to minimize Cost fun by change values of $J(\theta_0, \theta_1)$

n: no. of datapoints

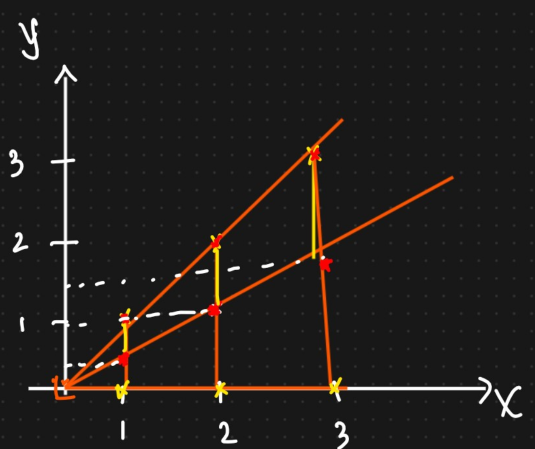
y_i = Actual value

$h_\theta(x)$ = predicted value

Final Aim {To get Best fit line}

$$\underset{\theta_0, \theta_1}{\text{Minimize}} \quad J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - h_\theta(x)_i \right)^2 \downarrow\downarrow\downarrow.$$

Optimization {Minimize the cost function}



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

$\theta_0 = 0$

$\boxed{h_\theta(x) = \theta_1 x_1}$

Dataset

X	y	$h_\theta(x)$
→ 1	1	1
2	2	2
3	3	3

	Let $\theta_1 = 1$	Let $\theta_1 = 0.5$	Let $\theta_1 = 0$
$x = 1$	$\Rightarrow h_\theta(x) = 0 + 1(1)$ $h_\theta(x) = 1$	$h_\theta(x) = 0 + 0.5(1)$ $= 0.5$	$h_\theta(x) = 0 + 0(1)$ $= 0$
$x = 2$	$h_\theta(x) = 0 + 1(2)$ $h_\theta(x) = 2$	$h_\theta(x) = 0 + 0.5(2)$ $= 1$	$h_\theta(x) = 0 + 0(2)$ $= 0$
$x = 3$	$h_\theta(x) = 3.$	$h_\theta(x) = 0 + 0.5(3)$ $= 1.5$	$h_\theta(x) = 0.$

*

Cost function

$\theta_0 = 0 \quad \Rightarrow h_\theta(x) = \theta_1 x_i \quad \left[\theta_1 = 1\right]$

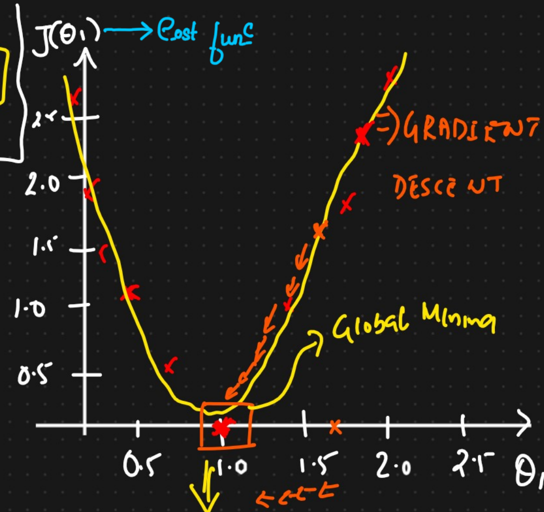
$$J(\theta_1) = \frac{1}{n} \sum_{i=1}^{n} \Big(\underbrace{y_i - h_\theta(x)}_{\text{Error}}\Big)^2$$

it can be $J(\theta_0, \theta_1)$

$n = 3$

$$= \frac{1}{3}\left[(1-1)^2 + (2-2)^2 + (3-3)^2\right]$$

$$J(\theta_1) = 0$$



Cost function will

Error will be Minimized

Cost fn

$\theta_1 = 0.5$

$$J(\theta_1) = \frac{1}{3}\left[(1-0.5)^2 + (2-1)^2 + (3-1.5)^2\right]$$

$$= \frac{1}{3}\left[0.25 + 1 + 2.25\right]$$

$$J(\theta_1) = 1.16$$

Cost fn :

$\theta_1 = 0.$

$$J(\theta_1) = \frac{1}{3}\left[(1-0)^2 + (2-0)^2 + (3-0)^2\right]$$

$$= \frac{1}{3}\left[1 + 4 + 9\right]$$

$$J(\theta_1) = \frac{14}{3} = 4.66$$

Convergence Algorithm

{optimize the change of θ_1}

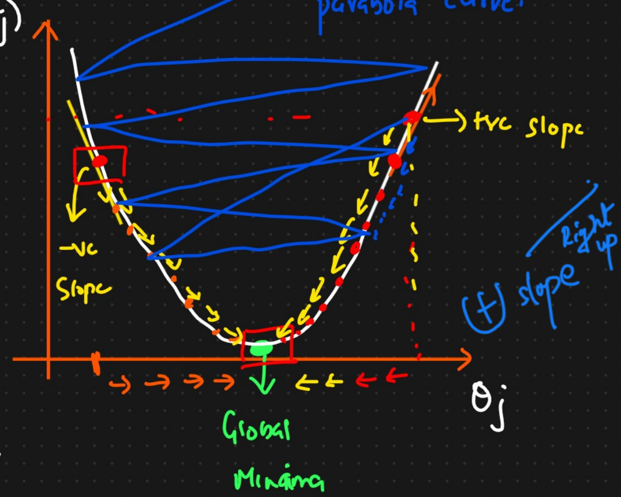
we initialized a θ_j and we ~~got a~~ cost func $J(\theta)$

$J(\theta_0, \theta_1)$

$\theta_0 \quad \theta_1$

$j = 0,1$

parabola curve.



Repeat until Convergence

{

$$\theta_j : \theta_j - \alpha \boxed{\frac{\partial J(\theta_j)}{\partial \theta_j}}$$

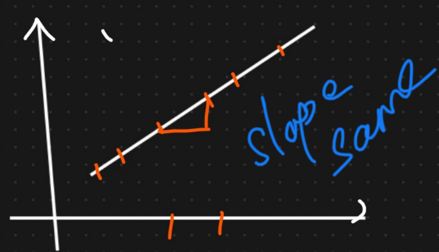
learning Rate

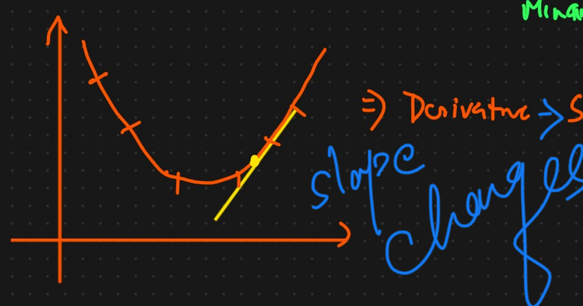
→ slope at a point

→ Derivative

}

$\boxed{\alpha = 0.01}$





⇒ Derivative → Slope of a point

$\theta_1 = \theta_1 - \alpha\,(+ve)$ ← slope

$\Rightarrow \theta_{1new} = \theta_{1old} - \alpha\,(+ve)$

$\boxed{\theta_{1new} << \theta_{1old}}$ ⇒ θ_1 is getting Reduced

$\theta_{1new} = \theta_{1old} - \alpha\,(-ve)$

Learning Rate decides the convergence speed

$\theta_{1new} = \theta_{1old} + (+ve)$

$\boxed{\theta_{1new} >> \theta_{1old}}$

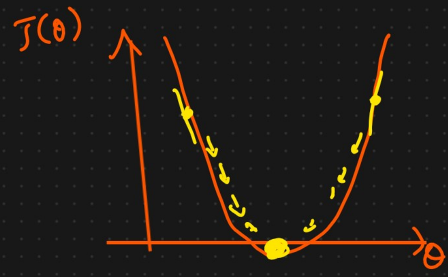
α (learning rate)

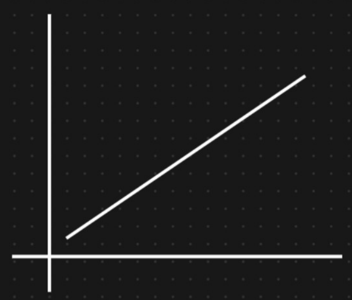
The rate at which it will converge

if learning rate large

Conclusion

GRADIENT DESCENT





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

Convergence Algorithm

Repeat until convergence

{

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

$$\boxed{J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2}$$