Cost function

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (\overbrace{y_i - h_\theta(x)}^{\text{Error}})^2$$

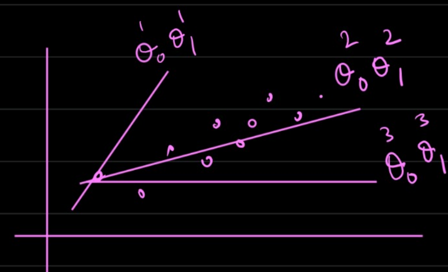
y_i → y actual, $h_\theta(x)$ → y predicted

mean-squared error.

n – no. of data points
y = Actual value
$h_\theta(x)$ = predicted value.

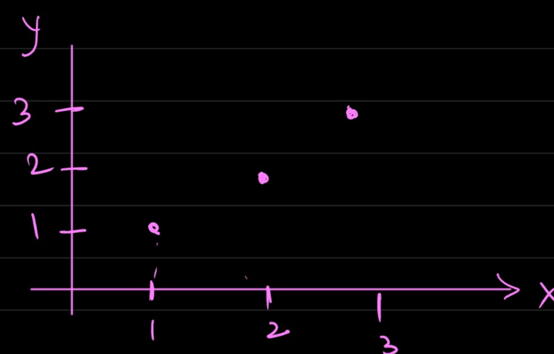
$$\min J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2$$

$$y_{pred} = h_\theta(x) = \theta_0 + \theta_1 x$$
$$= \beta_0 + \beta_1 x$$
$$= c + m x$$



→ Aim → minimize the cost fn to get optimal coefficient.

X	y
1	1
2	2
3	3



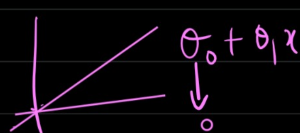
$h_\theta(x) = \theta_0 + \theta_1 x$

θ_1 ↳ slope

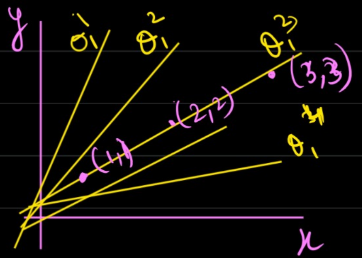
θ_0 ↳ intercept

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

→ Assuming best fit line passes through origin.



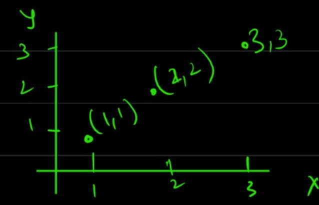
$\theta_0 = 0, \theta_1$



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

$\theta_0 = 0$

x	y
→1	1
→2	2
3	3



Scen 1 $\theta_1 = 1$

$x = 1 \Rightarrow h_\theta(x) = \theta_1 x + 0$
$= 1 \times 1$
$h_\theta(x) = 1$
$y_{pred} = 1$

$x = 2 \Rightarrow h_\theta(x) = \theta_0 + \theta_1 x$
$= 0 + 1 \times 2 = 2$

$x = 3 \Rightarrow h_\theta(x) = \theta_0 + \theta_1 x$
$= 0 + 1 \times 3$
$h_\theta(x) = 3$

$\theta_1 = 1$
$h_\theta(x) = 0 + \theta_1 x$

x	y	y_{pred} ($h_\theta(x)$)
1	1 ✓	1 ✓
2	2 ✓	2 ✓
3	3 -	3 -

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

min

Scen 2 $\theta_1 = 0.5$

$x = 1 \Rightarrow h_\theta(x) = \theta_0 + \theta_1 x$
$= 0 + 0.5 \times 1$
$= 0.5$

$x = 2 \Rightarrow 0 + 0.5 \times 2$
$h_\theta(x) = 1$

$x = 3 \Rightarrow 0 + 0.5 \times 3$
$h_\theta(x) = 1.5$

$\theta_0 = 0$
$\theta_1 = 0.5$
$h_\theta(x) = 0 + 0.5 \times x$

x	y	y_{pred}
1	1	0.5
2	2	1
3	3	1.5

$$J(\theta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_\theta(x))^2$$
$= \frac{1}{3}\left[(1-.5)^2 + (2-1)^2 + (3-1.5)^2\right]$
$= \frac{1}{3}\left([0.25 + 1 + 2.25\right)$
$= 1.16$

Scen 3 $\theta_1 = 0$

$x_1 = 1$, $h_\theta(x) = \theta_0 + \theta_1 x$
$h_\theta(x) = 0 + 0 \times 1$
$= 0$

$x = 2$ $h_\theta(x) = 0 + 0 \times 2$
$= 0$

$x = 3$, $h_\theta(x) = 0$

$\theta_0 = 0$
$\theta_1 = 0$

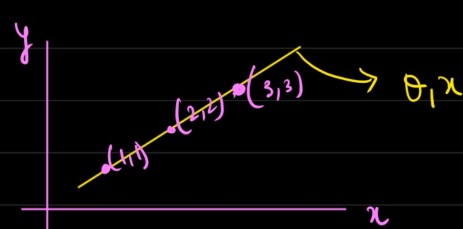
x	y	y_{pred}
1	1	0
2	2	0
3	3	0

$$J(\theta_1) = \frac{1}{3}\left[(1-0)^2 + (2-0)^2 + (3-0)^2\right]$$
$= \frac{1}{3}\left(1 + 4 + 9\right)$
$J(\theta_1) = \frac{14}{3} = 4.66$

$\theta_1 = 1$, $J(\theta_1) = 0$
$(1, 0)$

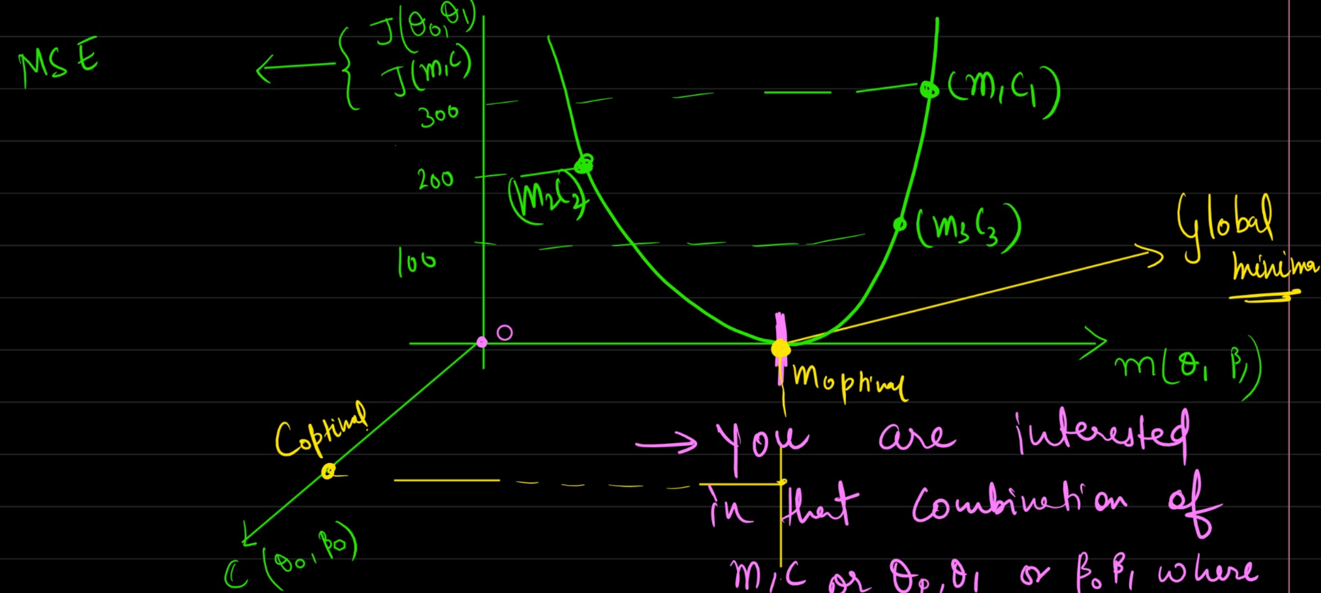
$\theta_1 = 0.5$, $J(\theta_1) = 1.16$
$(0.5, 1.16)$

$\theta_1 = 0$, $J(\theta) = 4.66$
$(0, 4.66)$

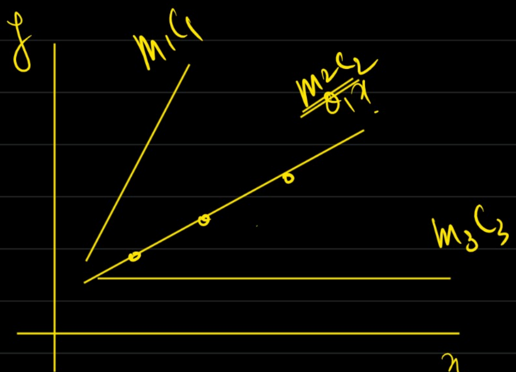


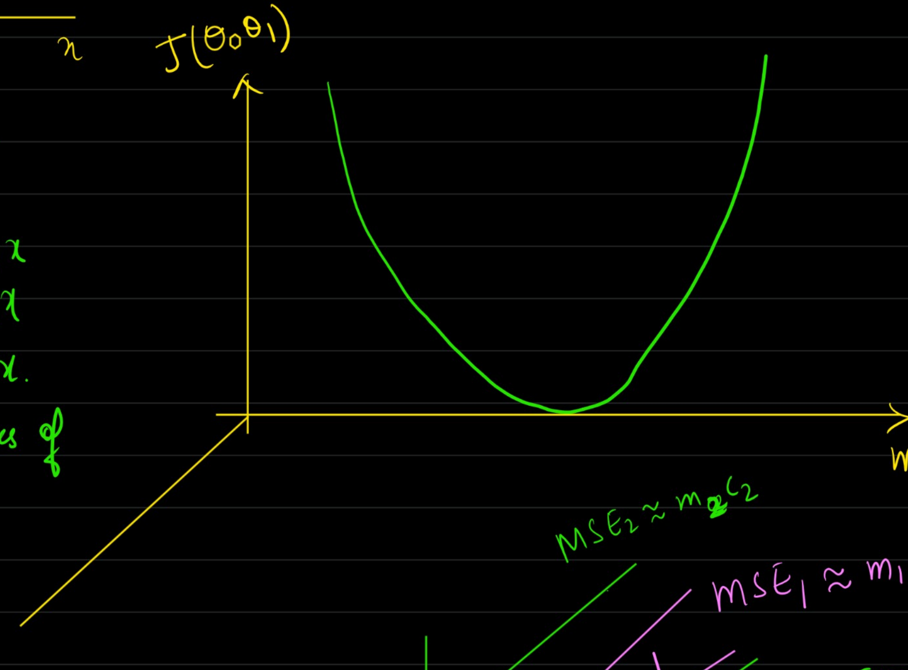
$$J(\theta_0, \theta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_\theta(x_i))^2$$
(min)

$$\underbrace{y = x^2}_{\text{Parabola}}$$



→ You are interested in that combination of m, c or θ_0, θ_1 or $\beta_0 \beta_1$ where error is minimum.





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

$$\Downarrow$$

$$\theta_0 + \theta_1 x$$

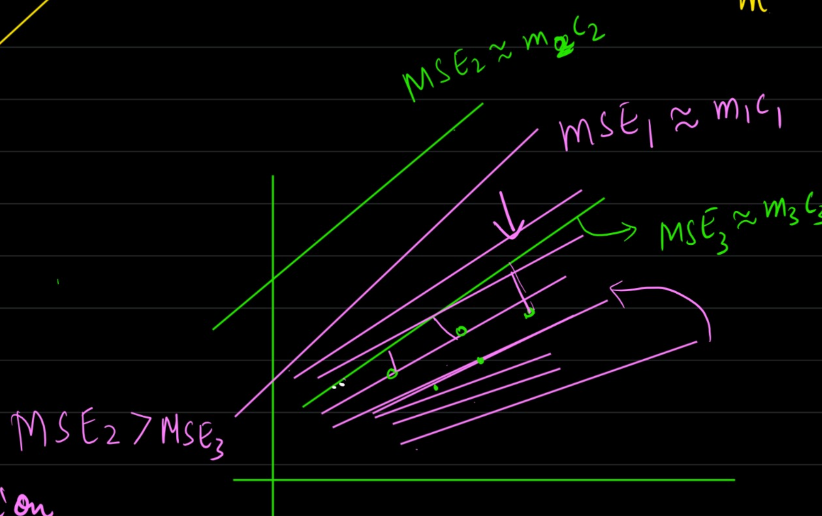
$$\beta_0 + \beta_1 x$$

$$c + mx.$$

→ infinite possibilities of m, c.

* Convergence Algorithm

→ Keep making new best fit line in the direction of the data points, until error (MSE|CF) is reduced as compared to previous lines.



Convergence Algorithm

learning rate (α)

Repeat until Convergence

$$\begin{cases} m_{new} = m_{old} - \eta \dfrac{\partial(CF)}{\partial m_{old}} \\ C_{new} = C_{old} - \eta \dfrac{\partial(CF)}{\partial C_{old}} \end{cases}$$

Repeat unit convergence (Simultaneously)

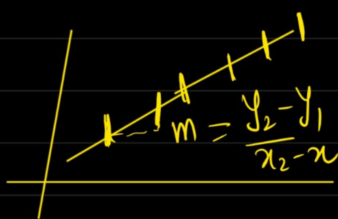
$j = 0, 1$

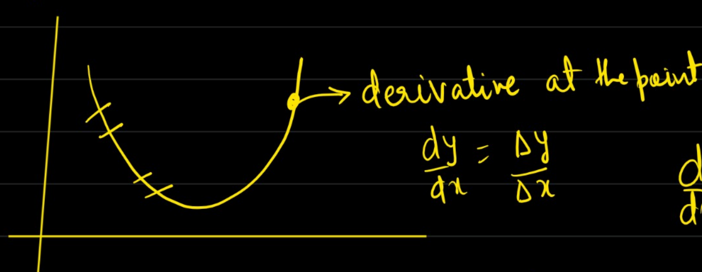
$$\left\{ \theta_j : \theta_j - \eta \frac{\partial J(\theta_j)}{\partial \theta_j} \right\}$$

slope at a point

$\underset{C_{new}}{\theta_0} = \underset{C_{old}}{\theta_0} - \eta \dfrac{\partial CF}{\partial \theta_0} \sim C_{old}$

$\underset{m_{new}}{\theta_1} = \underset{m_{old}}{\theta_1} - \eta \dfrac{\partial(CF)}{\partial \theta_1} \, m_{old}$





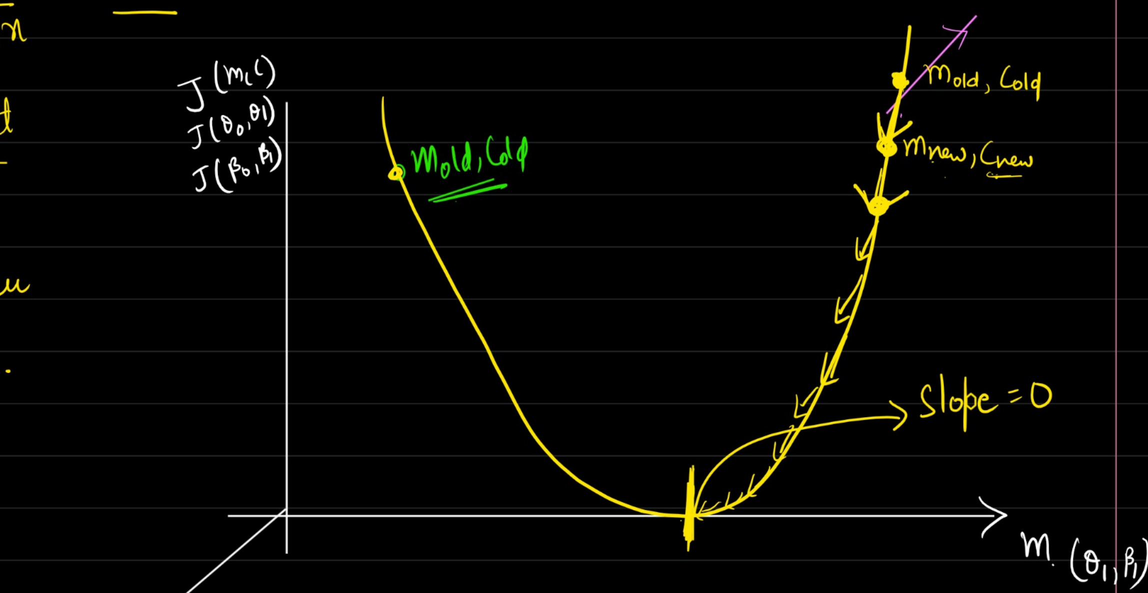
$\dfrac{dy}{dx} = \dfrac{\Delta y}{\Delta x}$ $\qquad \dfrac{dy}{dx} \approx \dfrac{\partial y}{\partial x}$

$$\frac{dx^n}{dx} = \underline{n x^{n-1}}$$

Gradient descent

$\Downarrow$

Slope guides you for convergence.

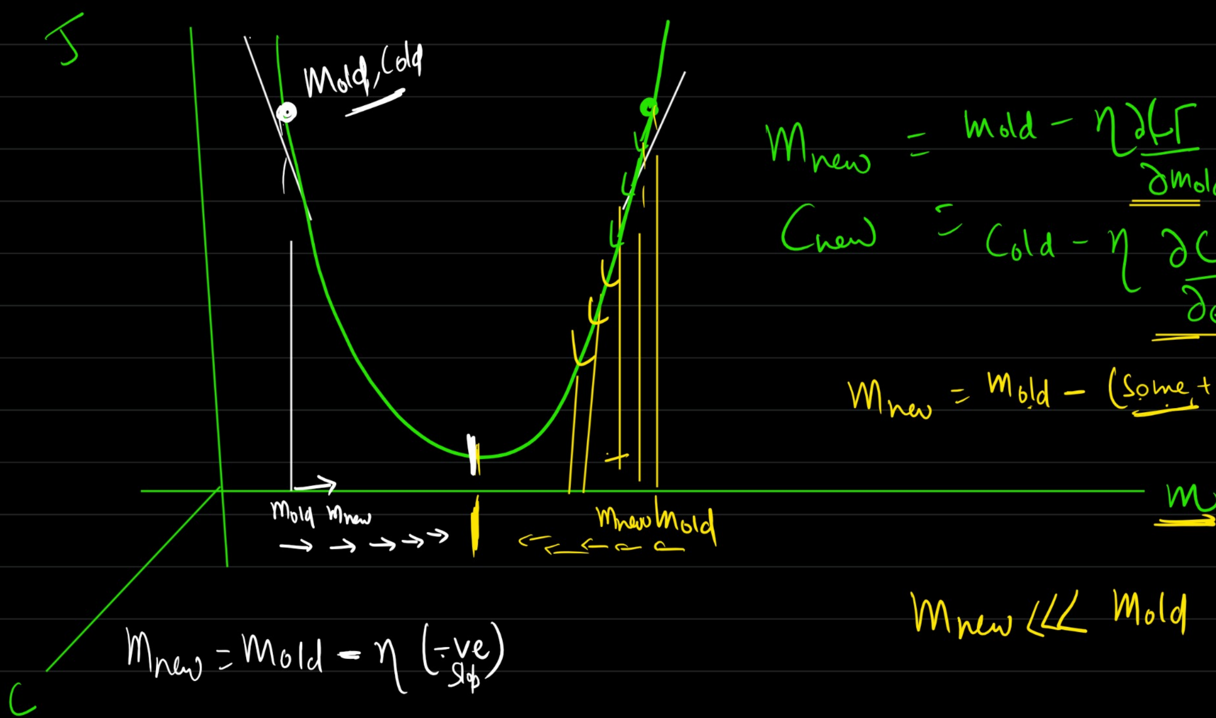


$$\left\{ m_{new} = m_{old} - \underset{\text{learning}}{\eta} \underbrace{\frac{\partial CF}{\partial m_{old}}}_{\text{slope}} \Rightarrow m_{new} = m_{old} - \eta \underbrace{(+ve)}_{m_{old} - +ve} \right.$$

$$\left\{ C_{new} = C_{old} - \overset{\text{rate}}{\eta} \frac{\partial CF}{\partial (C_{old})} \right.$$

$$m_{new} = m_{old} - \eta(\text{slope})$$

$$\boxed{m_{new} = m_{old}}$$



$$m_{new} = m_{old} - \eta \frac{\partial CF}{\partial m_{old}}$$

$$C_{new} = C_{old} - \eta \frac{\partial CF}{\partial C_{old}}$$

$$m_{new} = m_{old} - (\text{some } +ve)$$

$$m_{new} \lll m_{old}$$

$$m_{new} = m_{old} - \eta\,(-ve \text{ slope})$$

$$m_{new} = m_{old} + \eta\,(\text{slope})$$

When slope is +ve

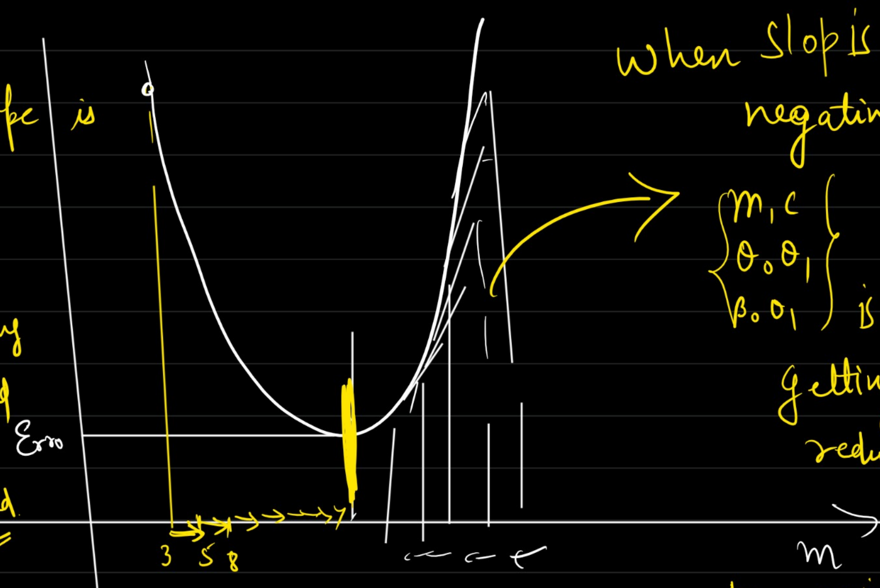
m, c
$\theta_0 \theta_1$ is
$\beta_0 \beta_1$
increasing

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

$3+2$

$\beta_{new} > \beta_{old}$

$m_{new} > m_{old}$



When slop is negative

$\left\{ \begin{matrix} m, c \\ \theta_0 \theta_1 \\ \beta_0 \theta_1 \end{matrix} \right\}$ is getting reduced

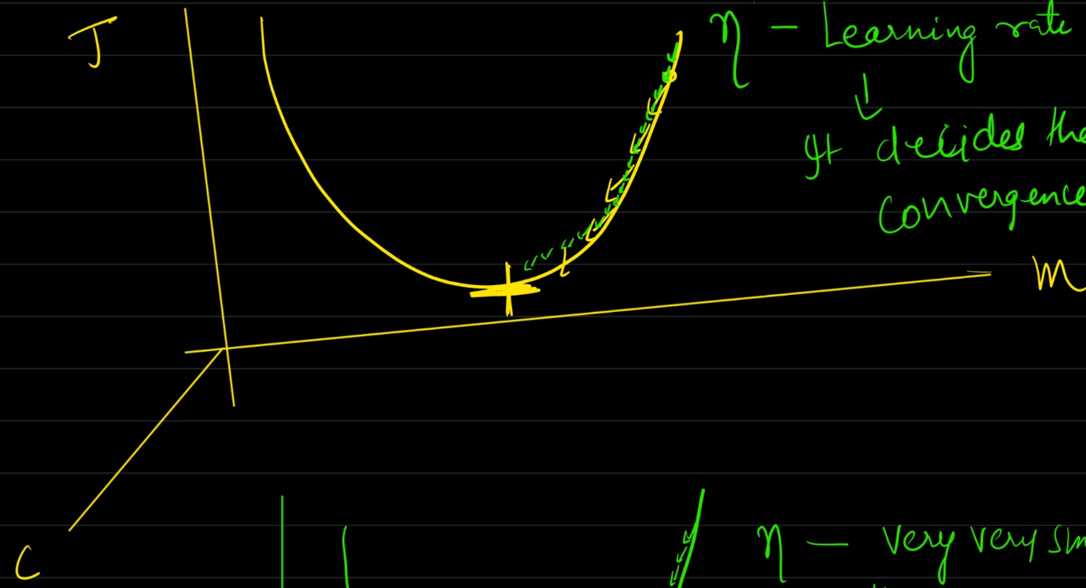
$m_{new} \ll m_{old}$

$\theta_{new} \ll \theta_{old}$

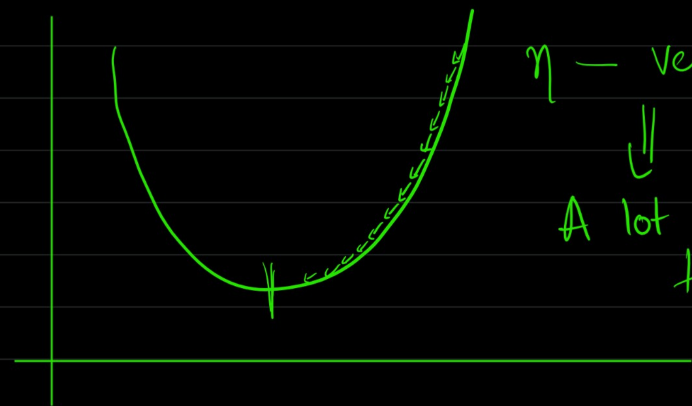
$\beta_{new} \ll \beta_{old}$

$$m_{new} = m_{old} - \eta \frac{\partial CF}{\partial m_{old}}$$

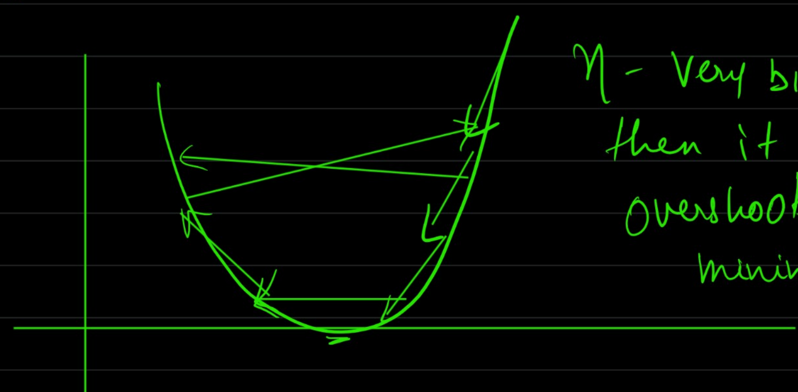
$$C_{new} = C_{old} - \eta \frac{\partial CF}{\partial C_{old}}$$



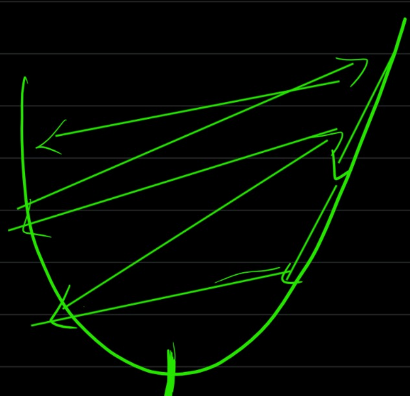
η – Learning rate
⇓
It decides the convergence speed

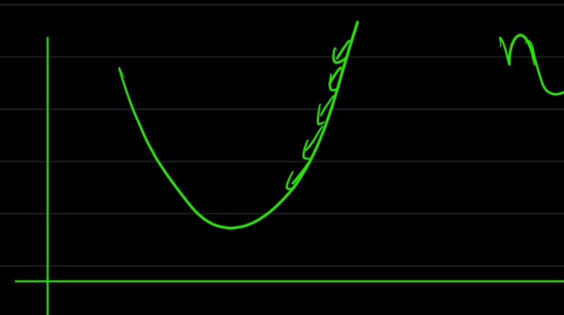


η — very very small
⇓
A lot of time for convergence



η – very big no then it overshoots minima

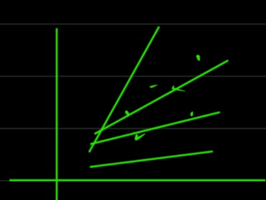




η

$\eta \approx 0$ to 1

Conclusion

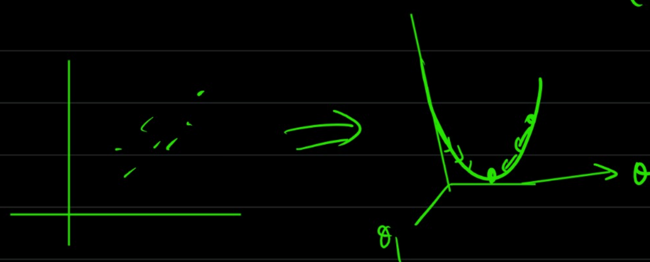


⟶ by optimising the CF
↓
MSE

Repeat until convergence

$$\left\{ \theta_j : \theta_j - \eta \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} \right\}$$

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



$$\Rightarrow \; y_{pred} = \theta_{0\,optimal} + \theta_{1\,optimal} X$$

$$y_{pred} = \beta_{0\,optimal} + \beta_{1\,optimal} X$$