For, 3D dataset,

dataset:-

	x_1	x_2	y
P_1	x_{11}	x_{12}	$\hat{y}_1$
P_2	x_{21}	x_{22}	$\hat{y}_2$

$$\hat{y}_1 = \beta_0 + \beta_1 x_{11} + \beta_2 x_{12}$$

$$\hat{y}_2 = \beta_0 + \beta_1 x_{21} + \beta_2 x_{22}$$

For this,

b = intercept = $\beta_0 = \beta_0 - \eta\,(\beta_0\_slope)$

$\uparrow$ learning rate (η)

m_1 = slope/weight = $\beta_1 = \beta_1 - \eta\,(\beta_1\_slope)$

m_2 = weight = $\beta_2 = \beta_2 - \eta\,(\beta_2\_slope)$

our work is to find β_0_slope, β_1_slope, β_2_slope from loss function.

$$L = \sum_{i=1}^{2} (y_i - \hat{y}_i)^2 \times \frac{1}{2} \quad (\because 2, \text{ no of person} = 2)$$

$$L = \left((y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2\right) \frac{1}{2}$$

$\Rightarrow$ we want to find $\frac{dL}{d\beta_0}$ to find ~~to~~ intercept,

$$L = \left((y_1 - \beta_0 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_0 - \beta_1 x_{21} - \beta_2 x_{22})^2 \right) / 2 \quad \text{--- (1)}$$

$$\frac{dL}{d\beta_0} = \left(2(y_1 - \beta_0 - \beta_1 x_{11} - \beta_2 x_{12})(-1) + 2(y_2 - \beta_0 - \beta_1 x_{21} - \beta_2 x_{22})(-1) \right) / 2$$

$$= -\frac{\cancel{2}}{\cancel{2}} \left((y_1 - \hat{y}_1) + (y_2 - \hat{y}_2) \right) \quad \times \text{ wrong.}$$

We cannot cut 2 & 2 as 2 in denominator is the total no of person.

$$= \frac{-2}{n} \left((y_1 - \hat{y}_1) + (y_2 - \hat{y}_2) \right)$$

In general for n-person data,

$$= \frac{-2}{n} \left((y_1 - \hat{y}_1) + (y_2 - \hat{y}_2) + \dots + (y_5 - \hat{y}_5) + \dots + (y_n - \hat{y}_n) \right)$$

$$= \frac{-2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)$$

Now for finding β_1 - slope, $\frac{\partial L}{\partial \beta_1}$

$$L = \left((y_1 - \beta_0 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_0 - \beta_1 x_{21} - \beta_2 x_{22})^2 \right) 1/2 \quad (\because \text{from } 1)$$

$$\frac{\partial L}{\partial \beta_1} = \left(2(y_1 - \beta_0 - \beta_1 x_{11} - \beta_2 x_{12})(-x_{11}) + 2(y_2 - \beta_0 - \beta_1 x_{21} - \beta_2 x_{22})(-x_{21}) \right) 1/2$$

$$= \cancel{\frac{-2}{2}\left((y_1 - \hat{y}_1)x_{11}(y_2 - \hat{y}_2)x_{21}\right)}$$

$$= \frac{-2}{2}\left((y_1 - \hat{y}_1) x_{11} + (y_2 - \hat{y}_2) x_{21} \right)$$

In general,

$$\boxed{\beta_{1\text{-slope}} = \frac{-2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) x_{i1}}$$

to find $\beta_{2\text{-slope}}$, $\frac{\partial L}{\partial \beta_2}$

$$L = \left((y_1 - \beta_0 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_0 - \beta_1 x_{21} - \beta_2 x_{22})^2 \right) 1/2$$

$$\frac{\partial L}{\partial \beta_2} = \left(2(y_1 - \beta_0 - \beta_1 x_{11} - \beta_2 x_{12})(-x_{12}) + 2(y_2 - \beta_0 - \beta_1 x_{21} - \beta_2 x_{22})(-x_{22}) \right) 1/2$$

$$\frac{\partial L}{\partial \beta_2} = \frac{-2}{2}\left((y_1-\hat{y_1})X_{12} + (y_2-\hat{y_2})X_{22}\right)$$

In general,

$$\frac{\partial L}{\partial \beta_2} = -\frac{2}{n}\left(\sum_{i=1}^{n}(y_i-\hat{y_i})X_{i2}\right)$$

For m^{th} column

$$\frac{\partial L}{\partial \beta_m} = \frac{-2}{n}\left(\sum_{i=1}^{n}(y_i-\hat{y_i})X_{im}\right)$$

For n^{th} column

$$\frac{\partial L}{\partial \beta_n} = \frac{-2}{n}\left(\sum_{i=1}^{n}(y_i-\hat{y_i})X_{in}\right)$$

example, For 5^{th} column

$$\frac{\partial L}{\partial \beta_5} = \frac{-2}{5}\left(\sum_{i=1}^{n}(y_i-\hat{y_i})X_{i5}\right)$$

Make it General,

According to Multiple linear Regression, we know that,

In $\sum_{i=1}^{n}(y_i - \hat{y}_i)$ If we take Y is the matrix then it would be,

$$(Y-\hat{Y})_{n\times 1}$$

n = No of person

X would be the metrix then,

$$(X)_{n\times m}$$

n = No of person/Row
m = No of Feature/column
[illegible]

as $(y-\hat{y})$ is symmetric matrix then,

$$(y-\hat{y}) = (y-\hat{y})^T$$

∴ In general it can be written as

$$\text{coff slope} = \left((Y - \hat{Y})^T_{1\times n} \cdot (X)_{n\times m} \right) \left(\frac{-2}{n} \right)$$

In numpy it can be written as,

coff_slope $= \frac{-2}{n}$(np.dot((Y - $\hat{y}$), X))

∵ $((y-\hat{y})^T$ taken care by numpy itself.

Disadvantages:- (1) Large memory required, If our dataset is too large then It can through memory Exhausted Exception.

(2) More ~~accurate~~ time required when Big dataset comes into the picture.

Advantages:- (1) More accurate & focused compared any other Gradient descent.

:- To overcome This disadvanteges, Stochastic Gradient Descent comes into the picture.