$X \rightarrow$ Matrix

x_1 cgpa	x_2 iq	Y Package
3.8 X_{11}	10 X_{12}	20 Y_1
4.8 X_{21}	20 X_{22}	50 Y_2

$$\hat{Y}_1 = \beta_0 + \beta_1 X_{11} + \beta_2 X_{12}$$

$$\hat{Y}_2 = \beta_0 + \beta_1 X_{21} + \beta_2 X_{22}$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \cdots \\ Y_n \end{bmatrix}$$

Now imagine n rows and m columns

$$\hat{Y}_1 = \beta_0 + \beta_1 X_{11} + \beta_2 X_{12} + \cdots + \beta_m X_{1m}$$

$$\hat{Y}_2 = \beta_0 + \beta_1 X_{21} + \beta_2 X_{22} + \cdots + \beta_m X_{2m}$$

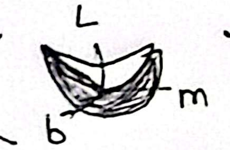
$$\cdots$$

$$\hat{Y}_n = \beta_0 + \beta_1 X_{n1} + \beta_2 X_{n2} + \cdots + \beta_m X_{nm}$$

$$\Rightarrow \hat{Y} = \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \cdots \\ \hat{Y}_n \end{bmatrix}_{(n\times 1)} = \beta_0 + \underbrace{\begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ X_{n1} & X_{n2} & \cdots & X_{nm} \end{bmatrix}_{(n\times m)}}_{X \text{ Matrix}} \underbrace{\begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdots \\ \beta_m \end{bmatrix}_{(m\times 1)}}_{\beta \text{ matrix}}$$

$$\boxed{\therefore \hat{Y} = \underset{\text{scalar value}}{\beta_0} + X\beta} \rightarrow (n\times 1) \text{ shape.}$$

In ~~For~~ "Gradient Descent for Simple Linear Regression", we saw, to move a point inside a 3D parabola

(

) we need 2 directions, m and b. To find those directions we calculated slope $\left(\frac{\partial L}{\partial b}, \frac{\partial L}{\partial m}\right)$ and then subtracted it.

Same here, for coefficients: $\frac{\partial B_1}{\partial L}, \frac{\partial B_2}{\partial L} \dots \frac{\partial B_m}{\partial L}$.

And for intercept: $\frac{\partial B_0}{\partial L}$.

$$\therefore B_0 = B_0 - \eta \cdot \text{slope-B0}$$

$$B_1 = B_1 - \eta \cdot \text{slope-B1}$$

$$B_2 = B_2 - \eta \cdot \text{slope-B2}$$

$$\dots \dots \dots$$

$$B_m = B_m - \eta \cdot \text{slope-Bm}$$

→ In Multimple Linear Regression, to move a point we need the Directions of B_0(b) and $B_1, B_2 \dots B_m$ (m).

Lets ~~calcute~~ calculate slope-B0, slope-B1 .....

Loss Function,

$$L = MSE = \frac{1}{n} \cdot \sum_{i=1}^{n} (Y_i - \underline{\hat{Y}_i})^2$$

(n = number of rows)

$$\Rightarrow \therefore L = \frac{1}{n} \cdot \sum_{i=1}^{n} (Y_i - \underbrace{\beta_0 - \beta_1 X_{i1} - \beta_2 X_{i2} - \ldots - \beta_m X_{im}}_{\hat{Y}_i})^2$$

$$\frac{\partial L}{\partial \beta_0} = \frac{1}{n} \sum_{i=1}^{n} 2(Y_i - \hat{Y}_i)(0-1-0-0-\ldots-0)$$

$$\therefore \frac{\partial L}{\partial \beta_0} = \boxed{\frac{-2}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i) \longrightarrow \text{slope\_B0}}$$

For slope-B1:

$$\frac{\partial L}{\partial \beta_1} = \frac{1}{n} \sum_{i=1}^{n} 2(Y_i - \hat{Y}_i)(0-0-x_{i1}-0-\ldots-0)$$

$$\therefore \frac{\partial L}{\partial \beta_1} = \boxed{\frac{-2}{n} \sum_{i=1}^{n} \underbrace{(Y_i - \hat{Y}_i)}_{n\times 1} \underbrace{x_{i1}}_{n\times 1}}$$

$$= \boxed{\frac{-2}{n} \sum_{i=1}^{n} \underbrace{(Y_i - \hat{Y}_i)}_{1\times n} \underbrace{x_{i1}}_{1\times 1 = 1\times 1}} \quad \text{slope\_}\beta 1$$

$\sum_{i=1}^{n} x_{i1} = \text{sum}(x_1 \text{ column})$

if $i=1$, x_{11} = 1st value of x_1 column.

$i=2$, x_{21} = 2nd value of X_1 column.

$\therefore x_{i1}$ is the value of x_1 column $\begin{bmatrix} x_{11} \\ x_{21} \\ x_{n1} \end{bmatrix}$

For slope_β2:

$$\frac{\partial L}{\partial \beta_2} = \frac{1}{n} \sum_{i=1}^{n} 2(Y_i - \hat{Y}_i)(0-0-0-X_{i2}-0)$$

$$\therefore \frac{\partial L}{\partial \beta_2} = \frac{-2}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^T x_{i2}$$

X_{i2} = value of X_2 column. $\begin{bmatrix} X_{12} \\ X_{22} \\ X_{n2} \end{bmatrix}$

So we've got,

Inter cept ← $\text{slope\_B0} = \frac{-2}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)$

$\text{slope\_B1} = \frac{-2}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^T X_{i1}$ → For 1st column.

$\text{slope\_B2} = \frac{-2}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^T X_{i2}$ → " 2nd "

← slopes

∴ For the Last(m) column, x_m

$$\text{slope\_Bm} = \frac{-2}{n} \sum_{i=1}^{n} \underbrace{\underbrace{(Y_i - \hat{Y}_i)^T}_{1\times n} \underbrace{X_{im}}_{n\times 1}}_{1\times 1}$$

1X1 scalar value in 2D shape.

$$X_m = \begin{bmatrix} X_{1m} \\ X_{2m} \\ \vdots \\ X_{nm} \end{bmatrix}$$

Calculations:

1) so in each epoch/loop, we need to calcutate all those slope_B1, ...2, ...3...m. This doesn't look cool. How about we just calculate slope_Bs,

$$slope\_Bs = \begin{bmatrix} slope\text{-}B1 \\ slope\text{-}B2 \\ \cdots \\ slope\text{-}Bm \end{bmatrix}$$

slope-B1, slope-B2....slope-Bm, they all have $\frac{-2}{n}\sum_{i=1}^{n}(Y_i-\hat{Y}_i)^T$ in common. Further, in slope-B we're doing multiplication i.e. dot matrix ~~of~~ $(Y_i-\hat{Y}_i)^T$ with X_{i1}. In slope-B2, $(Y_i-\hat{Y}_i)^T$ with X_{i2}.

It looks like:

2nd dot/multiplication

1st dot

$$\underbrace{(Y_i-\hat{Y}_i)^T}_{(1,n)\text{ shape matrix}} \qquad \overbrace{\begin{array}{cc} \underline{X_1} & \underline{X_2} \\ x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{n1} & x_{n2} \end{array}}^{X}$$

e.g. [1, 2, 3]

$$= \Big[\underbrace{\overbrace{x_{11}+2x_{21}+3x_{n1}}^{\sum_{i=1}^{n}(Y_i-\hat{Y}_i)X_{i1}}}_{slope\text{-}B1} \quad \underbrace{\overbrace{x_{12}+2x_{22}+3x_{n2}}^{\sum_{i=1}^{n}(Y_i-\hat{Y}_i)X_{i2}}}_{slope\text{-}B2}\Big]$$

$$= np.dot\left((Y_i-\hat{Y}_i).T,\ X\right)$$

<u>Multiply $\frac{-2}{n}$:</u> $\frac{-2}{n}\cdot np.dot\left((Y_i-\hat{Y}).T,\ X\right) = slope\_Bs$

FINAL CALCULATIONS :

To calculate $\beta_0 = \beta_0 - \eta . slope\text{-}\beta 0$, $\beta_1 = \beta_1 - \eta . slope\text{-}\beta 1$ ___

at first we need to initialize them to a random or specific value since we are using Gradient Descent. $m(\beta_1, \beta_2 \ldots \beta_m) = 1$ and $b(\beta_0) = 0$.

$$\therefore \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \ldots \\ \beta_m \end{bmatrix} = \begin{bmatrix} \beta_1 - \eta . slope\text{-}\beta 1 \\ \beta_2 - \eta . slope\text{-}\beta 2 \\ \ldots \\ \beta_m - \eta . slope\text{-}\beta_m \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \ldots \\ 1 \end{bmatrix}$$

and $\beta_0 = \beta_0 - \eta . slope\text{-}\beta 0 = 0$

for _ in range (epochs):

→ Inside this loop we've to keep updating $\beta_1, \beta_2, ... \beta_m$ i.e. β and β_0.

→ First we need to update slope_B1, slope_B2....

i.e. slope_Bs and slope_B0.

→ Before doing this we need to update $\hat{Y}$ inside slope_Bs and slope_B0.

→ $\hat{Y} = \beta_0 +$ np.dot(X, B)

→ slope_B0 $= \frac{-2}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)$

$= -\frac{2}{n} \cdot$ np.sum$(Y - \hat{Y})$

→ slope_Bs $= \frac{-2}{n} \cdot (Y_i - \hat{Y})^T \cdot X$

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

→ B0 = B0 − η · slope_B0

→ $\beta = \beta - \eta$. slope_Bs.

$$B = \begin{bmatrix} B1 \\ B2 \\ Bm \end{bmatrix} \quad \text{slope\_Bs} = \begin{bmatrix} \text{slope\_B1} \\ \text{slope\_B2} \\ \text{slope\_Bm} \end{bmatrix}$$

After the end of the loop, the final B0 is intercept and B is coefficients.