Linear Regression (YouTube: Python Engineer)

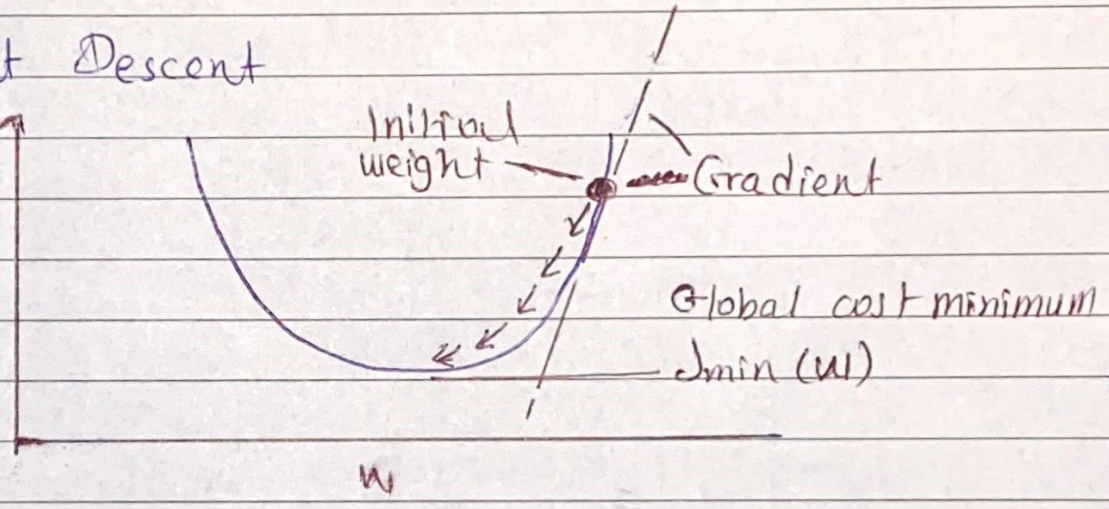
- $\hat{y} = wx + b$

- Cost function

$$MSE = J(w,b) = \frac{1}{N}\sum_{i=1}^{n}(y_i - (wx_i + b))^2$$

$$J(m,b) = \begin{bmatrix} \frac{dJ}{dw} \\ \frac{dJ}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum -2x_i(y_i - (wx_i+b)) \\ \frac{1}{N}\sum -2(y_i - (wx_i+b)) \end{bmatrix}$$

- Gradient Descent



- Update Rules

$$w = w - \alpha \cdot dw$$
$$b = b - \alpha \cdot db$$

$$\frac{dJ}{dw} = dw = \frac{1}{N}\sum_{i=1}^{n} -2x_i(y_i - (wx_i+b)) = \frac{1}{N}\sum_{i=1}^{n} -2x_i(y_i - \hat{y})$$

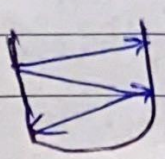
$$\therefore dw = \frac{1}{N}\sum_{i=1}^{n} 2x_i(\hat{y} - y_i)$$

$$\frac{dJ}{db} = db = \frac{1}{N}\sum_{i=1}^{n} -2(y_i - (wx_i+b)) = \frac{1}{N}\sum_{i=1}^{n} -2(y_i - \hat{y}) = \frac{1}{N}\sum_{i=1}^{n} 2(\hat{y} - y_i)$$

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

- Learning rate

i) Big learning rate



ii) Small learning rate