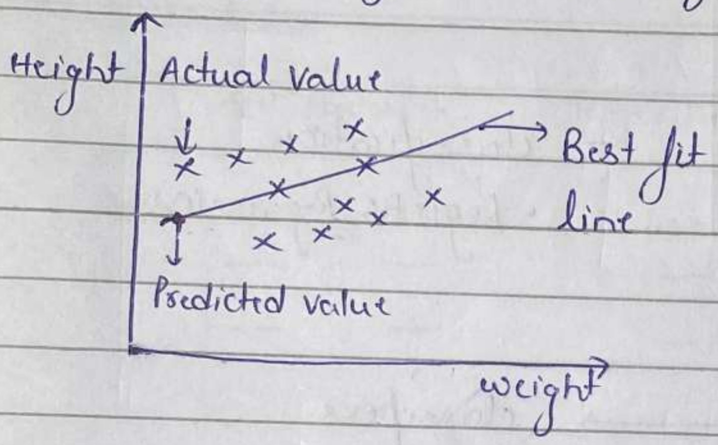
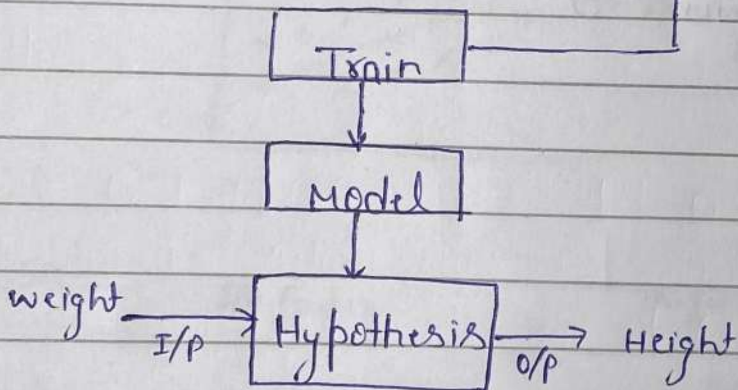
# Linear Regression

→ It is a supervised machine learning algorithm which is used to solve regression problem by finding a best fit line.



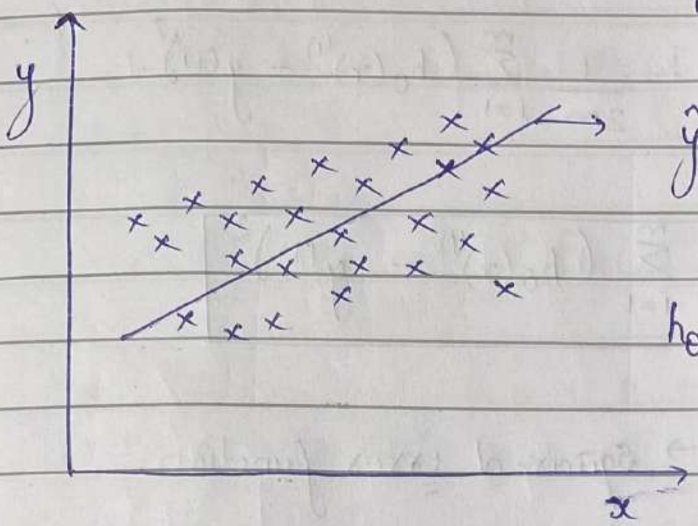

| Weight (kg) | Height (cm) |
|---|---|
| 73 | 150 |
| 63 | 155 |
| 78 | 170 |
| 82 | 165 |




Note: The distance between actual value and the predicted value is known as residual error.

The aim of linear regression is to find a best fit line so that the residual error should be as minimum as possible.

## * Mathematic Intution :



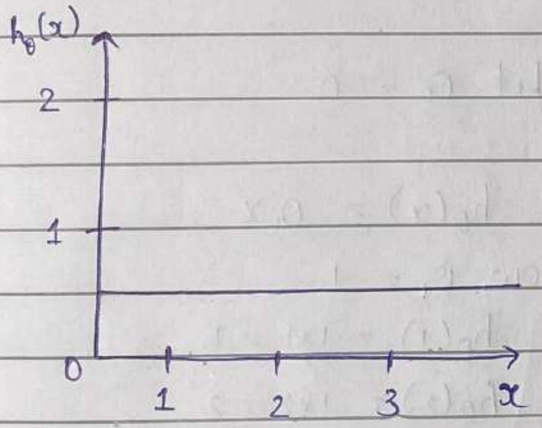

$$\hat{y} = mx + c$$

same (in hypothesis form)

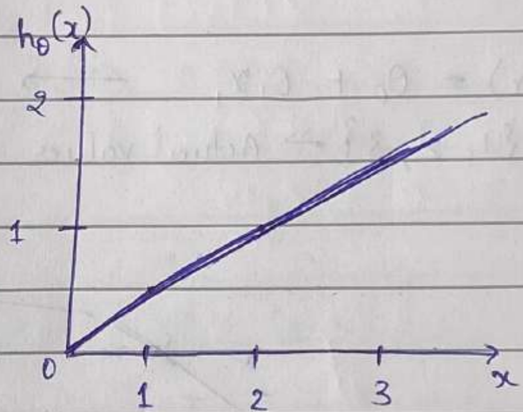
$$h_\theta(x) = \theta_0 + \theta_1 x$$

→ Hypothesis : {(y is a linear function of x)}

$$h_\theta(x) = \theta_0 + \theta_1 x_1$$




If $\theta_0 = 0.5$
$\theta_1 = 0$




If $\theta_0 = 0$
$\theta_1 = 0.5$

Note : we randomly initialize a linear line, then adjust the line to make it best fit, by finding the values of coefficients and intercept $(\theta_0, \theta_1)$ in such a way that the residual error should be as minimum as possible.

* **Cost Function :**

$$\frac{1}{2m}\sum_{i=1}^{m}(\hat{y}-y)^2 \xrightarrow{\text{same}} \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)}-y^{(i)}\right)^2$$

$$\boxed{J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)}-y^{(i)}\right)^2}$$

→ Squared error function.

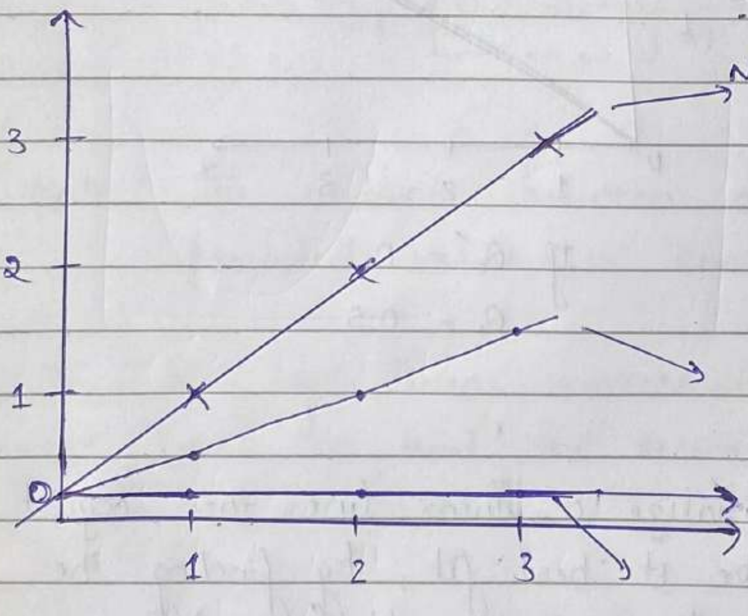
The main aim is to minimize this cost function.

→ **Hypothesis :**

$h_\theta(x) = \theta_0 + \theta_1 x_1 \longrightarrow$ Let $\theta_0 = 0$

$x \in \{1, 2, 3\} \rightarrow$ Actual values

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




Now, $\theta_1 = 1$

$h_\theta(1) = 1\times1 = 1$

$h_\theta(2) = 1\times2 = 2$

$h_\theta(3) = 1\times3 = 3$

Now, $\theta_1 = 0.5$

$h_\theta(1) = 0.5\times1 = 0.5$

$h_\theta(2) = 0.5\times2 = 1.0$

$h_\theta(3) = 0.5\times3 = 1.5$

Now, $\theta_1 = 0$

$h_\theta(1) = 0\times1 = 0$

$h_\theta(2) = 0\times2 = 0$, $h_\theta(3) = 0\times3 = 0$.

→ Cost Function : {For same hypothesis}

$$\theta_0 = 0.$$

when, $\theta_1 = 1$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2$$

$$\Rightarrow J(1) = \frac{1}{2 \times 3} \left[ 0 + 0 + 0 \right] = 0$$

when, $\theta_1 = 0.5$

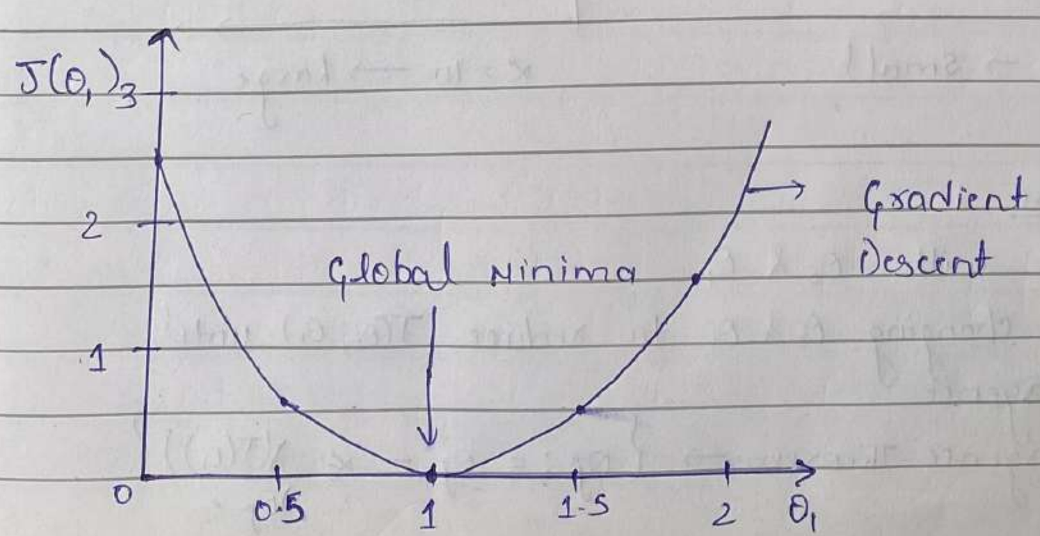
$$J(0.5) = \frac{1}{6} \left[ (0.5-1)^2 + (1-2)^2 + (1.5-3)^2 \right]$$

$$= \frac{1}{6} \times (3.5) \approx 0.58$$

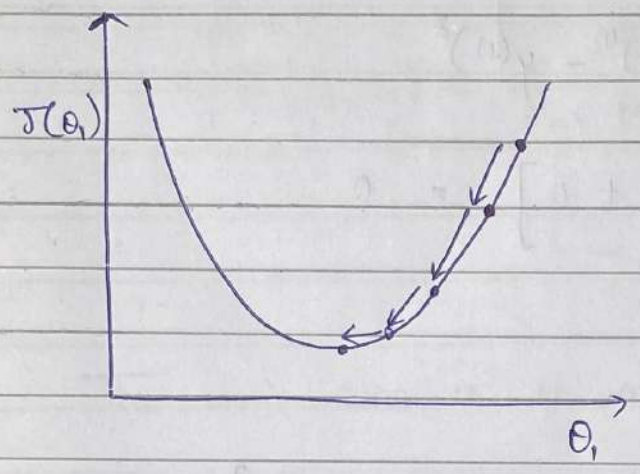
when, $\theta_1 = 0$

$$J(0) = \frac{1}{6} \left[ (0-1)^2 + (0-2)^2 + (0-3)^2 \right]$$

$$= \frac{14}{6} \approx 2.5$$



Note: But we don't know how to find the global minima. To do so, we use gradient descent which is a step-by-step process to find global minima.




Repeat Convergence Theorem

$$\left\{ \theta_j := \theta_j - \alpha \frac{\partial (J(\theta_1))}{\partial \theta_j} \right\}$$

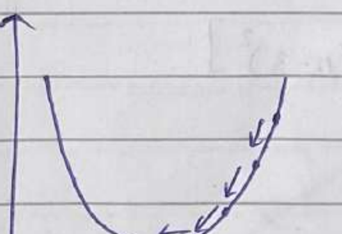
$\alpha$ → Learning Rate

$\frac{\partial (J(\theta_1))}{\partial \theta_j}$ → slope

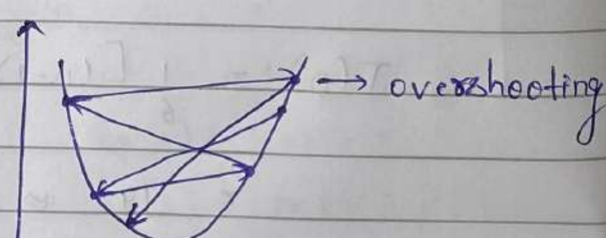
$\theta_1 := \theta_1 - \alpha(+ve) \} \longrightarrow$ Decrement

$\theta_1 := \theta_1 - \alpha(-ve) \} \longrightarrow$ Increment

$= \theta_1 + \alpha \frac{dy}{dx}$



$\alpha = 0.5 \rightarrow$ small




$\alpha = 10 \longrightarrow$ large

→ Outline:

(i) Start with $\theta_0$ & $\theta_1$

(ii) keep changing $\theta_0$ & $\theta_1$ to reduce $J(\theta_0, \theta_1)$ until convergence

(iii) convergence Theorem → $\left\{ \theta_j := \theta_j - \alpha \frac{\partial (J(\theta))}{\partial \theta_j} \right\}$