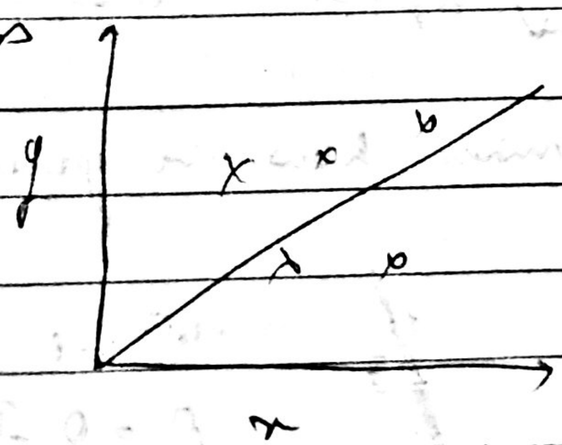
m = No. of training example
x = inputs, features
y = outputs

h → hypothesis

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

(Linear regres with X)

$h(x) = \theta_0 + \theta_1 x$



(univariate)

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

→ minimize.

Gradient descent. (linear regression)

we can use n no. of parameter

$J(\theta_0, \theta_1, \ldots, \theta_n)$.

start $\theta_0 = \theta_1 = 0$.

Keep chang. θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

Gradient descent

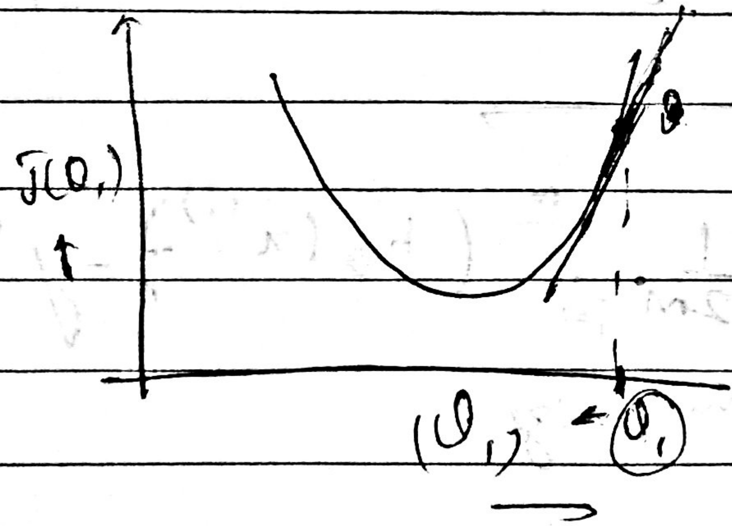
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

learning rate (huge steps downhill)

$j = 0$ & $j = 1$

Simultaneously update θ_0 & θ_1

$\alpha \rightarrow$ determines how ar your steps.



$J(\theta_1) = c$

$$\theta_1 = \theta_1 - \boxed{\alpha \frac{d}{d\theta_1} J(\theta_1)}$$

≥ 0 derivative = slope

$\therefore \theta_1 = \theta_1 - \alpha(+\text{ve no})$

therefore, θ_1 will move towards left.

→ If α is too small, it will take tiny steps

→ If α is too large, it may overshoot.