Linear Regression

→ feature - x , label - y → $h_\theta(x)$ → to be predicted as y.

→ $h_\theta(x) = \theta_0 + \theta_1 x$, where θ are parameters (hypothesis)

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

This is called the squared error function.

→ Gradient Descent:

repeat until convergence

{

$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_0 \dots \theta_1)$ [Simultaneously for all i]

}

→ here,

repeat until convergence

{

$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)}\right)$

$\theta_1 := \theta_1 - \alpha \frac{1}{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}$

} Simultaneously

< This is written after calculating derivatives manually >

→ The cost function is convex, so it does not matter where we start. So we can randomly initialize. It will converge to same point.

↳ Randomly initialize → train using grad descent (batch)

GET MODEL ↵

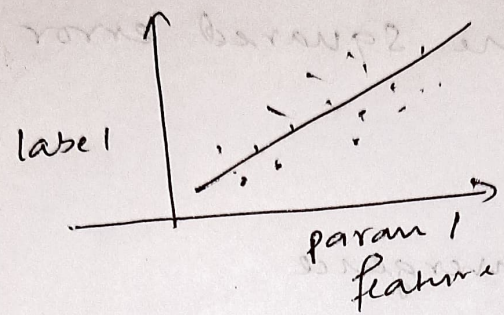
→ When more than 1 feature

$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \dots \theta_n x_n$

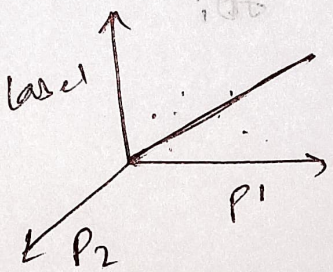
Learnable weights = n + intercept = total (n+1)

∴ n+1 steps of gradient descent together / simultaneously

1 parameter / feature



2 parameter / feature



More than 2 parameters, cannot visualize.

This is called Multiple Regression.