Gradient Descent:

May 12, 2021

Gradient descent is algorithm that minimizes the cost function.

i.e. We have $J(\theta_0, \theta_1)$, we want to find $\min J(\theta_0, \theta_1)$.

First start w/ some θ_0, θ_1 (e.g. $\theta_0 = 0, \theta_1 = 0$)

Then keep changing θ_0, θ_1 until $J(\theta_0, \theta_1)$ is a min.

i.e.

$$\boxed{\text{repeat until convergence } \left\{ \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \right\}}$$

↳ $\text{temp}0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ $\cancel{\theta_0 := \text{temp}0}$

$\text{temp}1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

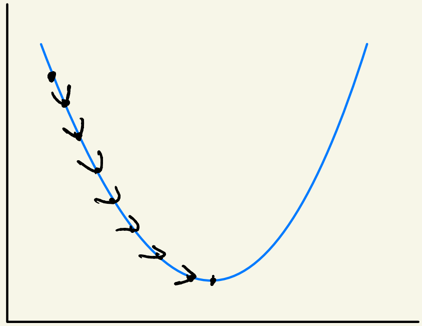
$\left.\begin{array}{l} \theta_0 := \text{temp}0 \\ \theta_1 := \text{temp}1 \end{array}\right\}$ simultaneous update or else this won't work properly

α – learning rate

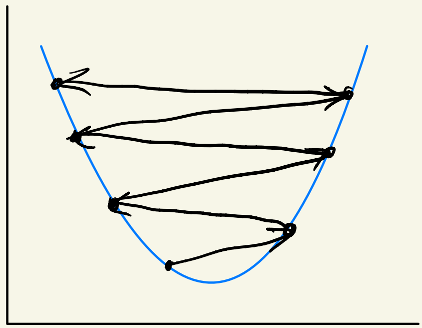
$:=$ – override value of variable

The Importance of Learning Rate:

Too small = lots of training iters needed



Too large = overshoot, fail to converge

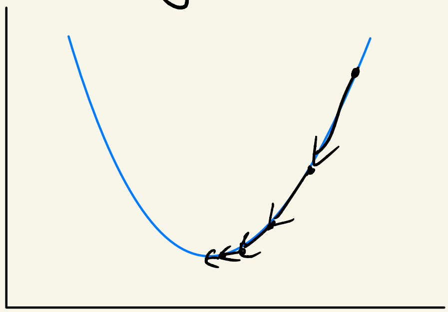


★ ESC190 exam Q4

★ If θ_j is already at a local minima, gradient descent does nothing.

As we approach a local min, the derivative term automatically gets smaller, so g.d. will take smaller steps.

↳ the learning rate can be fixed



Gradient Descent For Linear Regression:

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

$$= \frac{\partial}{\partial \theta_j} \cdot \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

$j=0$: $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$

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

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

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

}

* Updated Simultaneously

Batch gradient descent is where each step of gradient descent uses all the training examples

$$\hookrightarrow \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$