

The way we do this is by taking the derivative (the tangential line to a function) of our cost function. The slope of the tangent is the derivative at that point and it will give us a direction to move towards. We make steps down the cost function in the direction with the steepest descent. The size of each step is determined by the parameter  $\alpha$ , which is called the learning rate.

For example, the distance between each 'star' in the graph above represents a step determined by our parameter  $\alpha$ . A smaller  $\alpha$  would result in a smaller step and a larger  $\alpha$  results in a larger step. The direction in which the step is taken is determined by the partial derivative of  $J(\theta_0, \theta_1)$ . Depending on where one starts on the graph, one could end up at different points. The image above shows us two different starting points that end up in two different places.

The gradient descent algorithm is:

repeat until convergence:

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

where

$j=0,1$  represents the feature index number.

At each iteration  $j$ , one should simultaneously update the parameters  $\theta_1, \theta_2, \dots, \theta_n$ .

Updating a specific parameter prior to calculating another one on the  $j^{(th)}$  iteration would yield to a wrong implementation.

<u>Correct: Simultaneous update</u>	<u>Incorrect:</u>
$\rightarrow \text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ $\rightarrow \text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$ $\rightarrow \theta_0 := \text{temp0}$ $\rightarrow \theta_1 := \text{temp1}$	$\rightarrow \text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ $\rightarrow \theta_0 := \text{temp0}$ $\rightarrow \text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$ $\rightarrow \theta_1 := \text{temp1}$

Mark as completed

