

## Welcome

## Introduction

## Review

## Model and Cost Function

## Parameter Learning

- ✓ 视频: Gradient Descent  
11 min
- ✓ 阅读材料: Gradient Descent  
3 min
- ✓ 视频: Gradient Descent Intuition  
11 min
- ✓ 阅读材料: Gradient Descent Intuition  
3 min

# Gradient Descent For Linear Regression

## Gradient Descent For Linear Regression

**Note:** [At 6:15 "h(x) = -900 - 0.1x" should be "h(x) = 900 - 0.1x"]

When specifically applied to the case of linear regression, a new form of the gradient descent equation can be derived. We can substitute our actual cost function and our actual hypothesis function and modify the equation to:

repeat until convergence: {

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

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m ((h_{\theta}(x_i) - y_i)x_i)$$

}

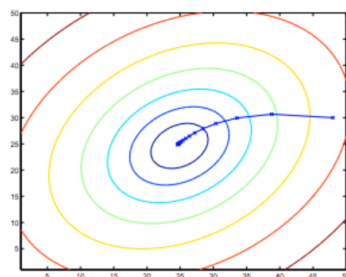
where m is the size of the training set,  $\theta_0$  a constant that will be changing simultaneously with  $\theta_1$  and  $x_i, y_i$  are values of the given training set (data).

Note that we have separated out the two cases for  $\theta_j$  into separate equations for  $\theta_0$  and  $\theta_1$ ; and that for  $\theta_1$  we are multiplying  $x_i$  at the end due to the derivative. The following is a derivation of  $\frac{\partial}{\partial \theta_j} J(\theta)$  for a single example:

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_{\theta}(x) - y)^2 \\ &= 2 \cdot \frac{1}{2} (h_{\theta}(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_{\theta}(x) - y) \\ &= (h_{\theta}(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left( \sum_{i=0}^n \theta_i x_i - y \right) \\ &= (h_{\theta}(x) - y) x_j \end{aligned}$$

The point of all this is that if we start with a guess for our hypothesis and then repeatedly apply these gradient descent equations, our hypothesis will become more and more accurate.

So, this is simply gradient descent on the original cost function J. This method looks at every example in the entire training set on every step, and is called **batch gradient descent**. Note that, while gradient descent can be susceptible to local minima in general, the optimization problem we have posed here for linear regression has only one global, and no other local, optima; thus gradient descent always converges (assuming the learning rate  $\alpha$  is not too large) to the global minimum. Indeed, J is a convex quadratic function. Here is an example of gradient descent as it is run to minimize a quadratic function.



The ellipses shown above are the contours of a quadratic function. Also shown is the trajectory taken by gradient descent, which was initialized at (48,30). The x's in the figure (joined by straight lines) mark the successive values of  $\theta$  that gradient descent went through as it converged to its minimum.

✓ 已完成

转到下一个课程内容

 Like  Dislike  报告问题