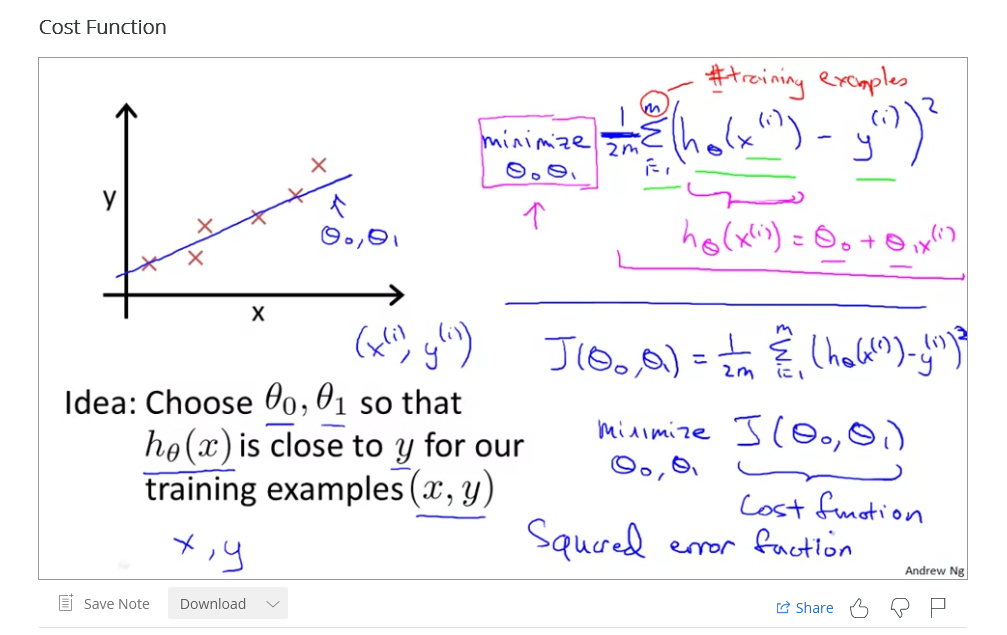
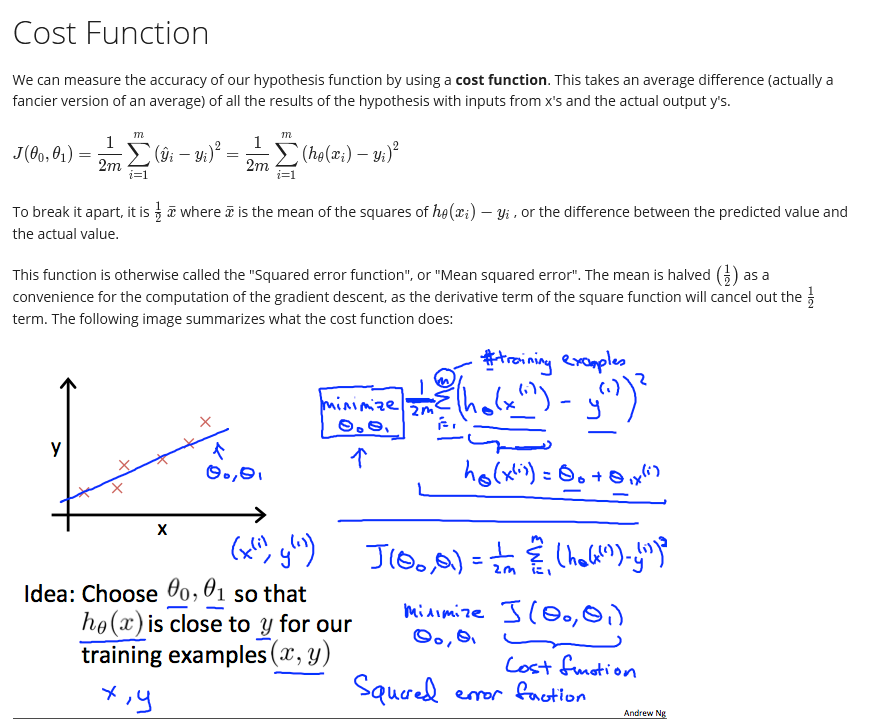
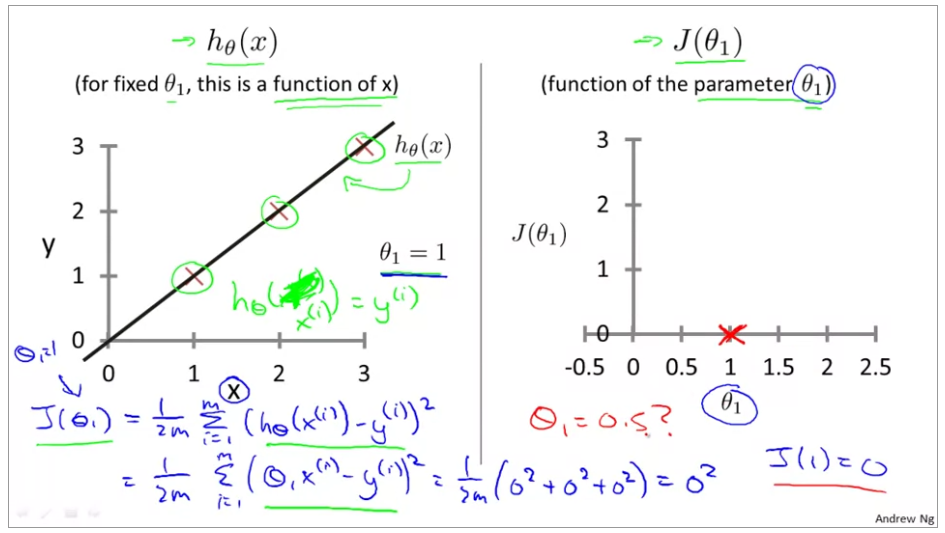
COST FUNCTION

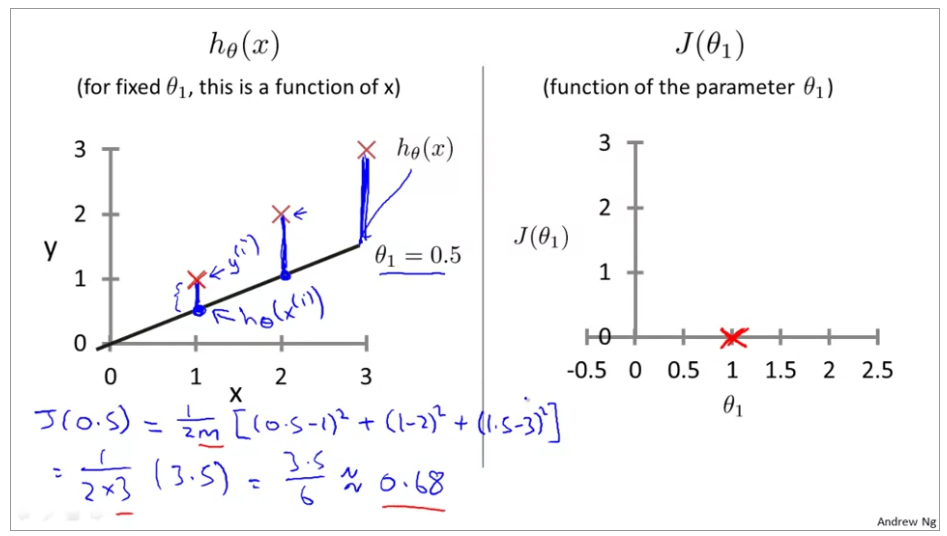


# Hypothesis vs Cost function ( When theta0 is zero)

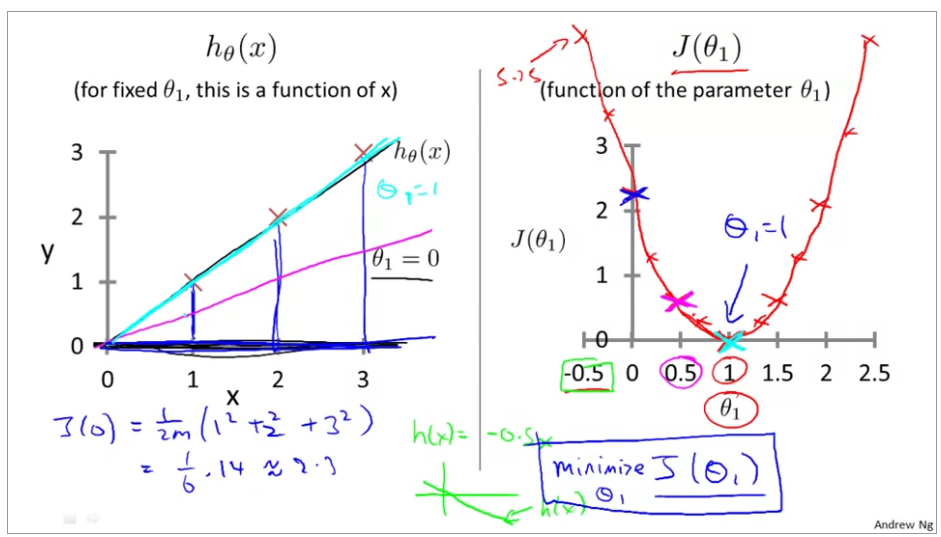
1. When the error is zero(Theta 1 = 1) i.e. the predicted model was 100% accurate



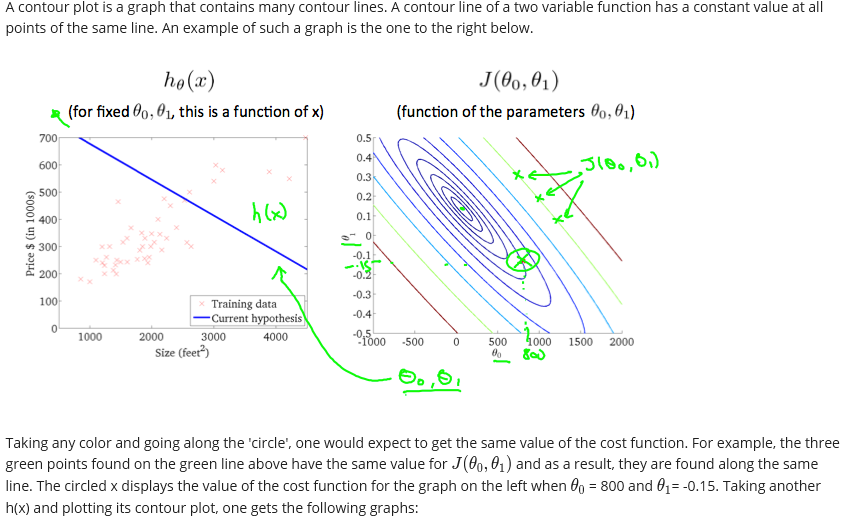
1. Theta 1 = 0.5

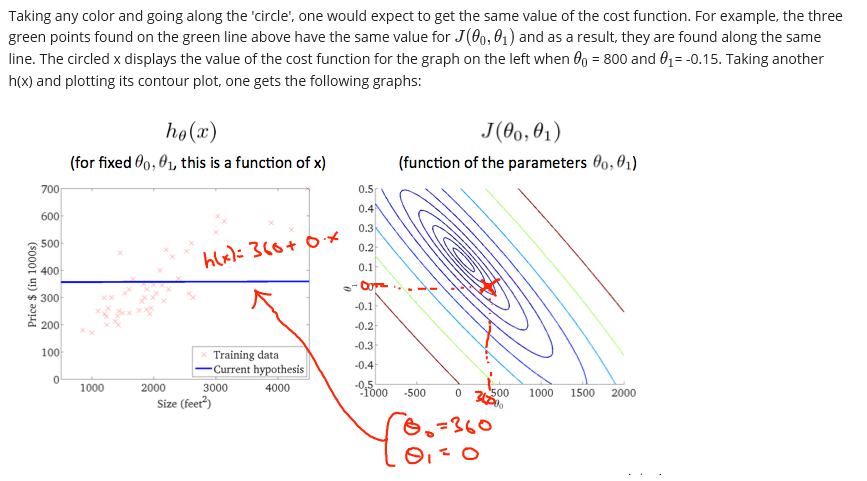


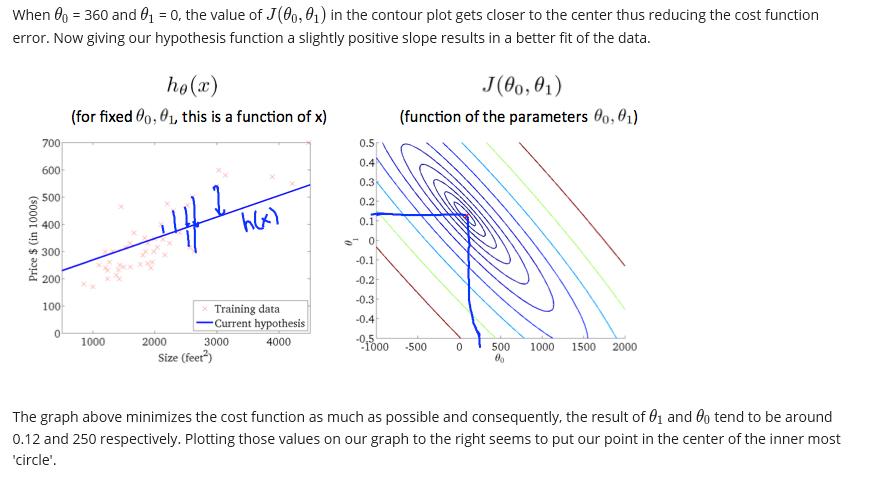
# This is how linear regression minimizes the error, when theta0 = zero:



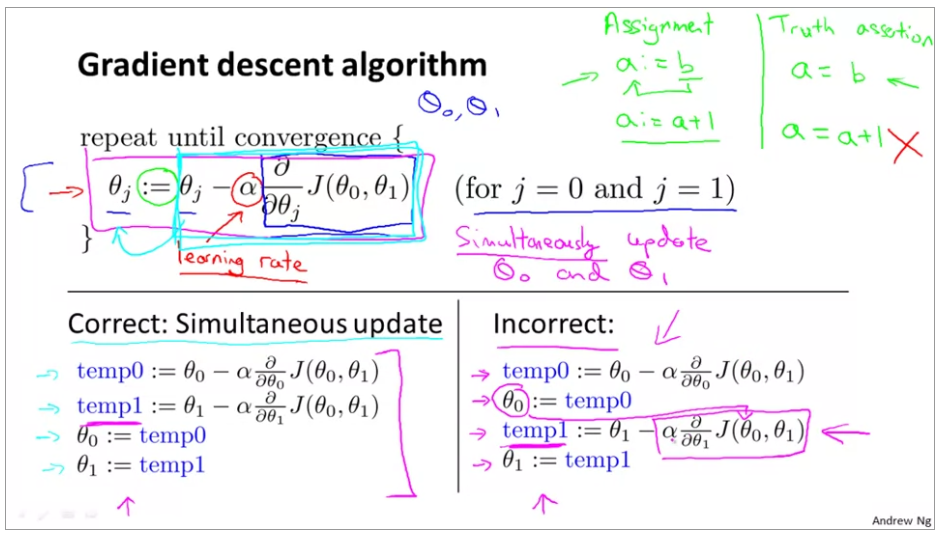
# Contour plot







# Gradient descent (GD):



So we have our hypothesis function and we have a way of measuring how well it fits into the data. Now we need to estimate the parameters in the hypothesis function. That's where gradient descent comes in.

Imagine that we graph our hypothesis function based on its fields θ0\theta\_0θ0​ and θ1\theta\_1θ1​ (actually we are graphing the cost function as a function of the parameter estimates). We are not graphing x and y itself, but the parameter range of our hypothesis function and the cost resulting from selecting a particular set of parameters.

We put θ0\theta\_0θ0​ on the x axis and θ1\theta\_1θ1​ on the y axis, with the cost function on the vertical z axis. The points on our graph will be the result of the cost function using our hypothesis with those specific theta parameters. The graph below depicts such a setup.

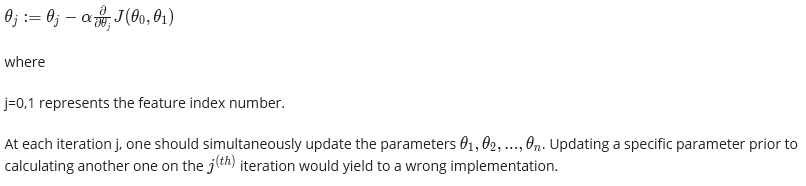


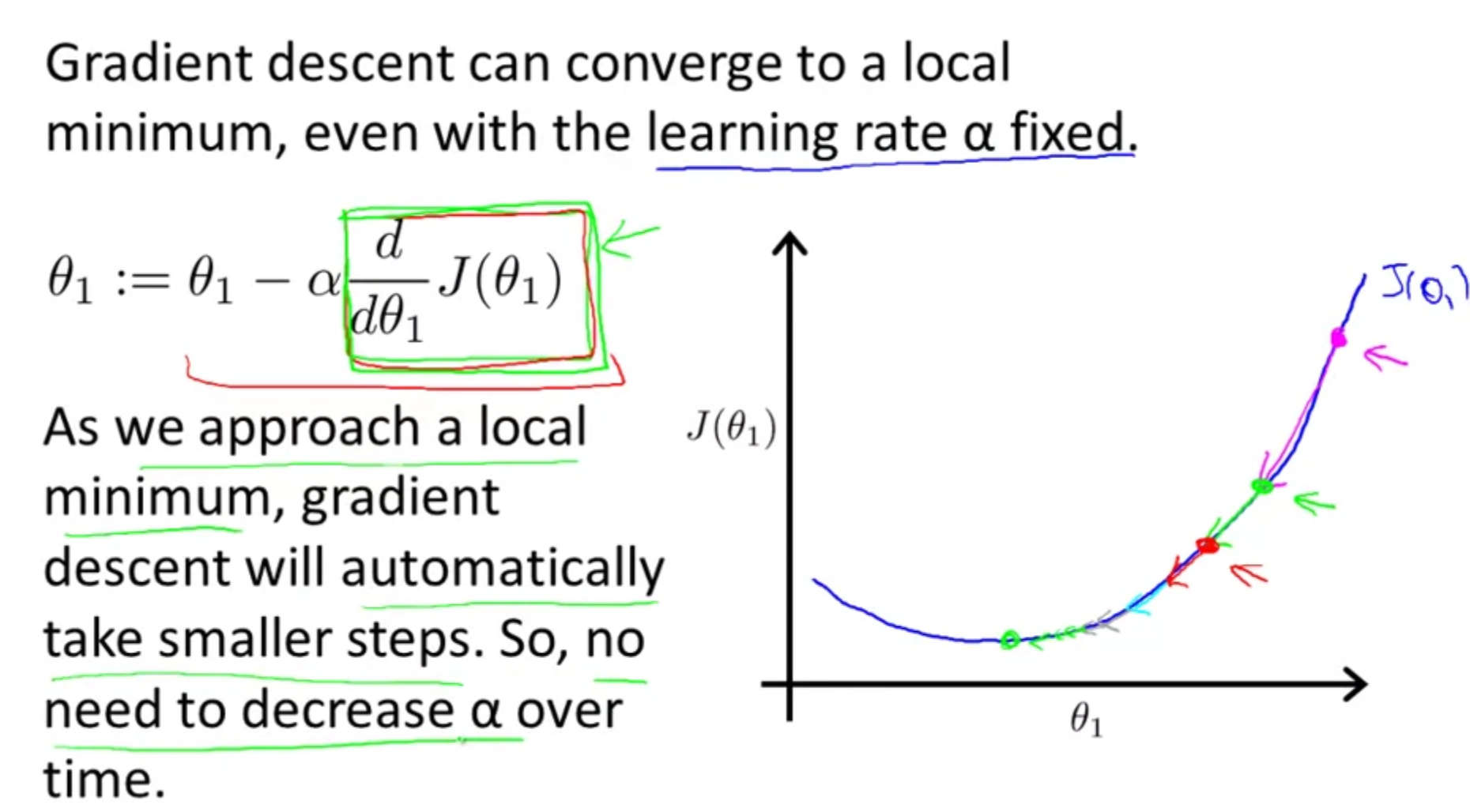
We will know that we have succeeded when our cost function is at the very bottom of the pits in our graph, i.e. when its value is the minimum. The red arrows show the minimum points in the graph.

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 α, which is called the learning rate.

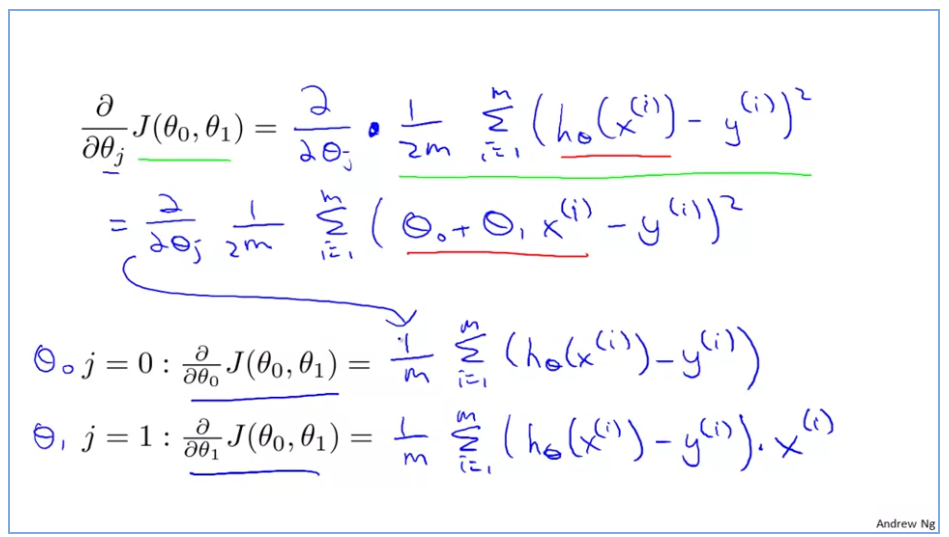
For example, the distance between each 'star' in the graph above represents a step determined by our parameter α. A smaller α would result in a smaller step and a larger α results in a larger step. The direction in which the step is taken is determined by the partial derivative of J(θ0,θ1)J(\theta\_0,\theta\_1)J(θ0​,θ1​). 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.

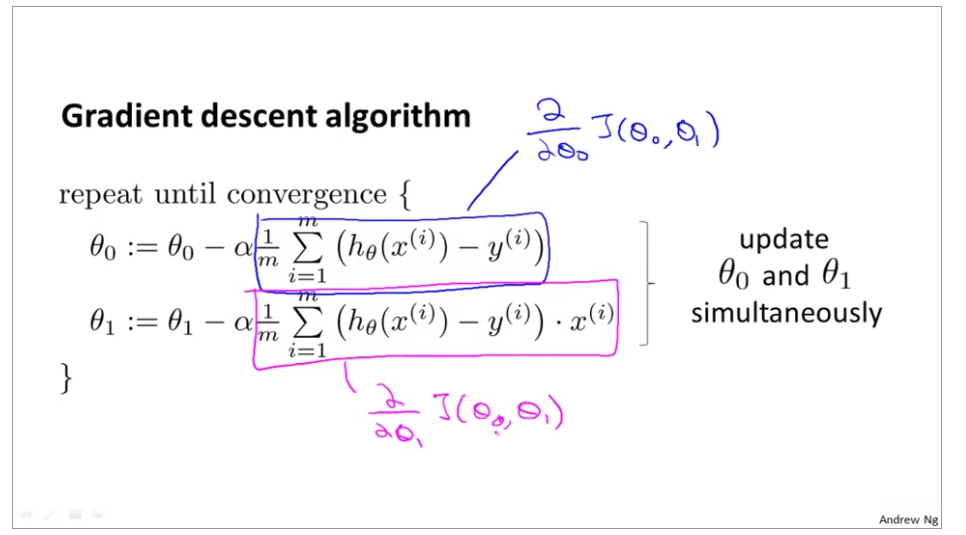
The gradient descent algorithm is:





# Partial Derivative of Cost function for GD:





# Lots of hypotheses but matrix multiplication saves the day

Here every column of the resultant matrix is the result of applying the hypotheses to the training data.  
