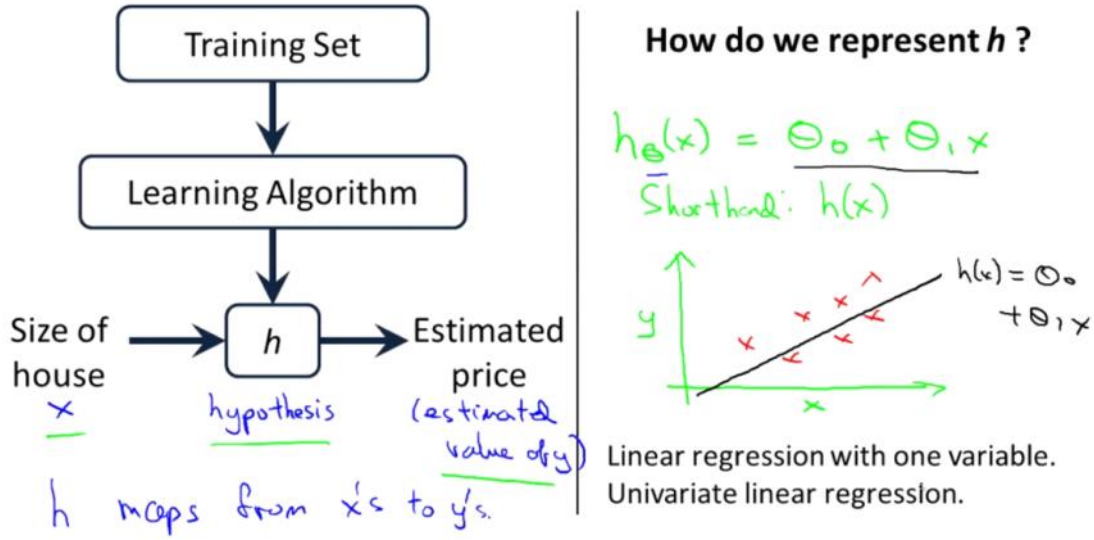
Week 1: Linear Regression

Monday, December 17, 2018 1:02 AM

Model Representation



So as discussed in learning from data we have a learning algo which outputs a hypothesis function which will then be used to estimate price of new data.

Now the hypothesis function here is a linear equation and here it is with one variable

Cost function:
This will help in fitting the best possible line

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

θ_i's: Parameters

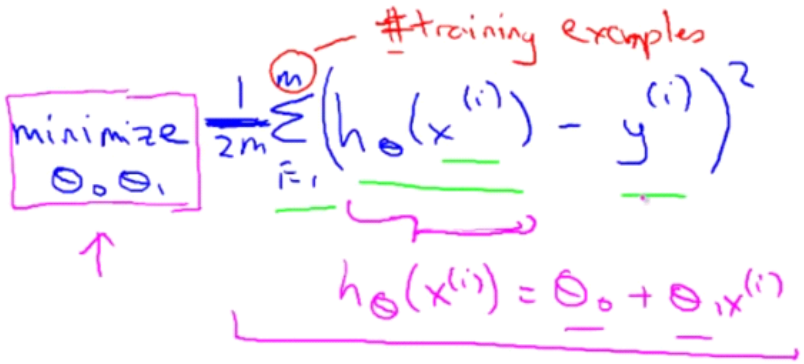
How to choose θ_i's ?

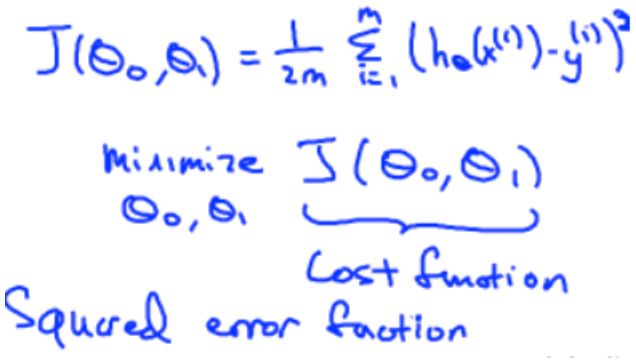
In this part we will get to know how to choose these theta values

With different theta we have different hypothesis. So this is the set of hypothesis we have.

So now how to choose the best hypothesis,
Idea here is to choose the theta in such a wat that we have predicted value close to the actual value of the training set

Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training examples (x, y)

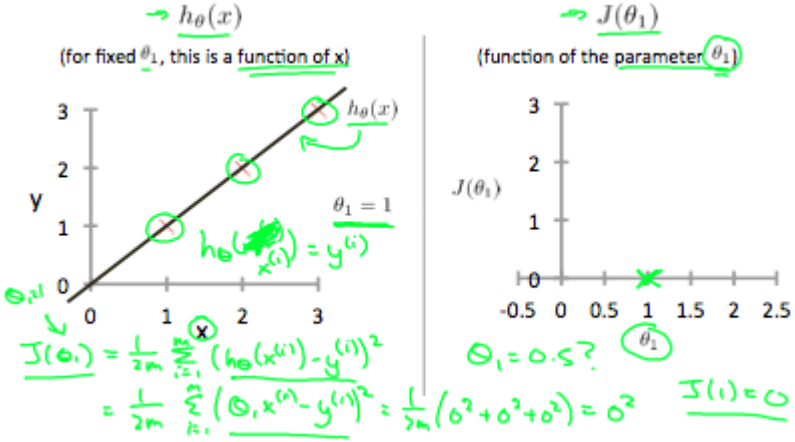




This is the cost function of the model
Squared error cost function is most commonly used

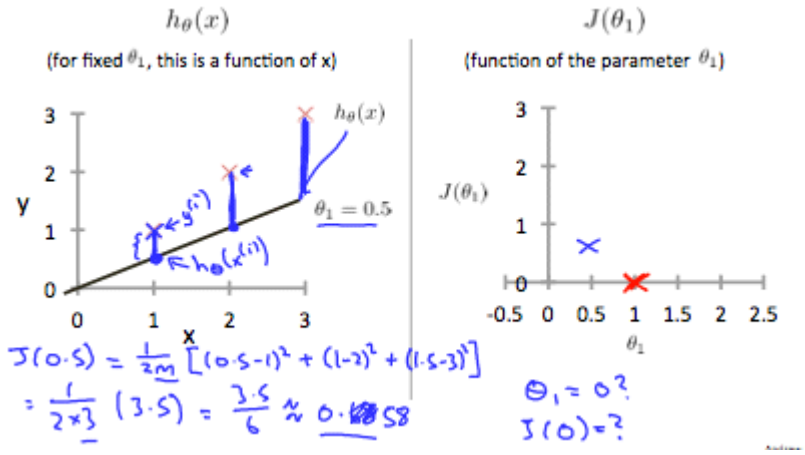
The mean is halved 1/2 as a convenience for the computation of the gradient descent, as the derivative term of the square function will cancel out the 1/2 term.

Cost function intuition:

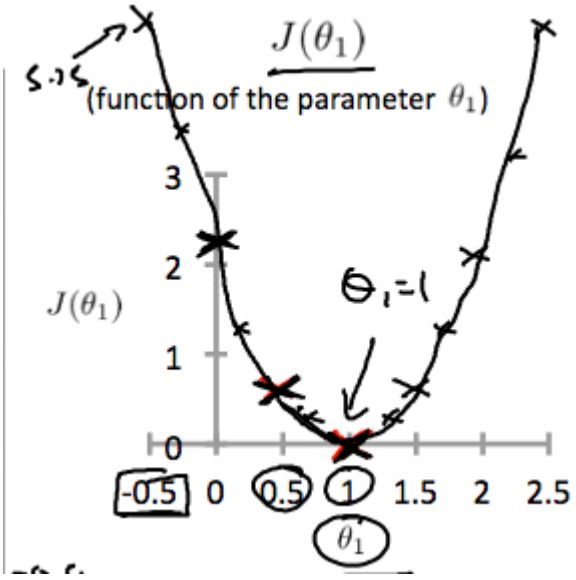


So if we select any theta, we get a line for the function h0(x).
And using that line if we check the cost function J(theta) then we get cost of the fit.
Now our goal is to minimize this J(theta)

Also if we choose different values of theta, hypothesis function varies with it and at the same time the cost function also changes as suggesting in below pic

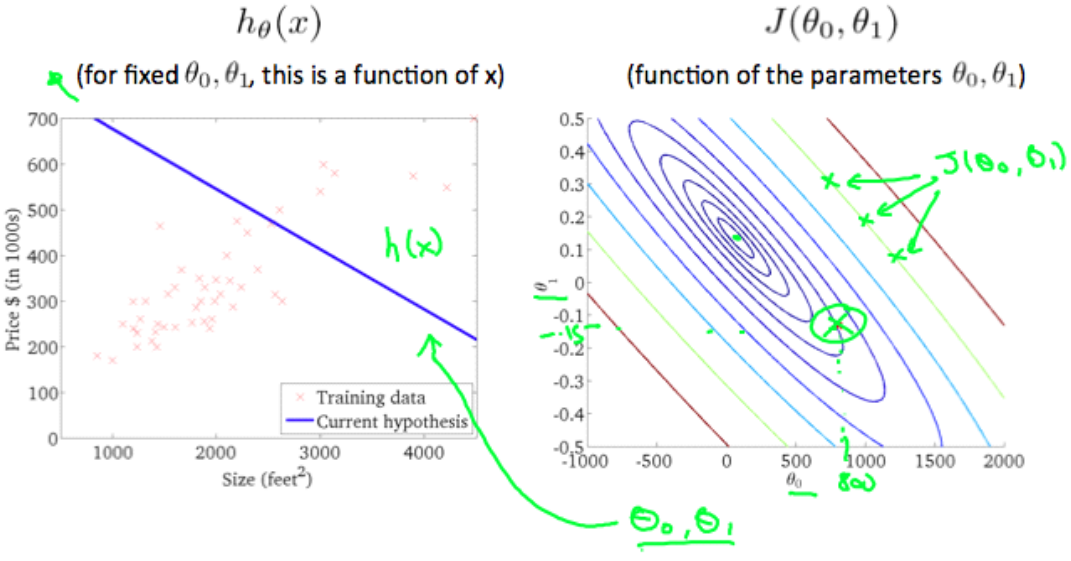


If we plot the cost function for many theta, we get below graph



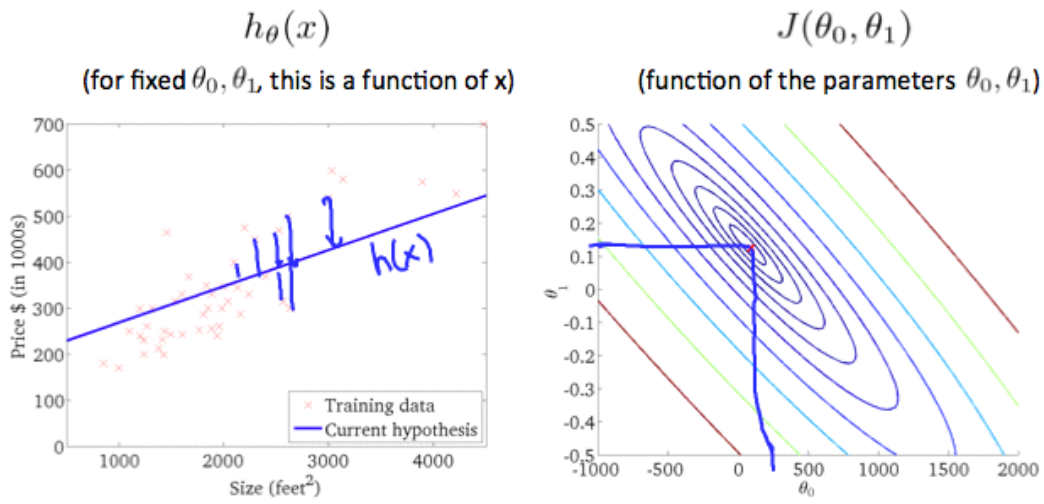
As our goal is to minimize cost function, here it can be achieved by using theta =1

Contour plot is used below::



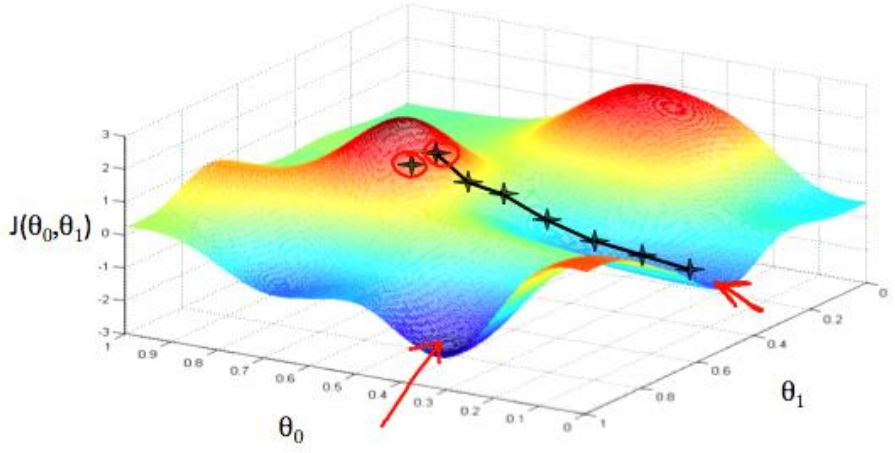
Now the above graph is when we have two theta in picture, here the cost function graph can be

represented as contour graph on the right.
So the lines in the contour graph will have same cost function value. As the theta changes the cost function might get towards the center which is local minima



Gradient Descent:
Used All over machine learning. Used to minimize many cost function

It can be used to minimize many parameter(theta) but here we will just take two theta



We will succeed when the cost function reaches the blue area
We do this by taking the derivative of cost function. Derivative is the tangential line to a 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.
So we check which directing will give us the best result using the derivative and then move towards it.
The size of each step is defined by a parameter alpha called learning rate.
Smaller alpha is smaller step, and vice versa
The direction in which the step is taken is determined by the partial derivative of cost function

Also it depends on the starting point where the gradient descent will end. For eg in the above graph we have two red circle and there

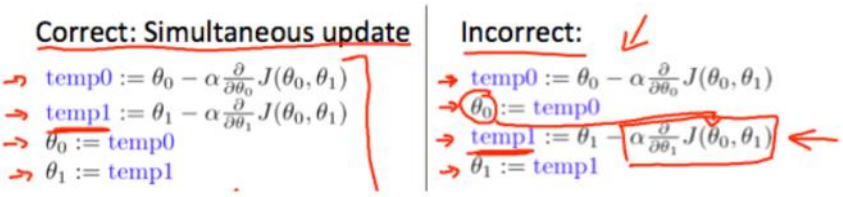
The gradient descent algorithm is:

repeat until convergence:

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

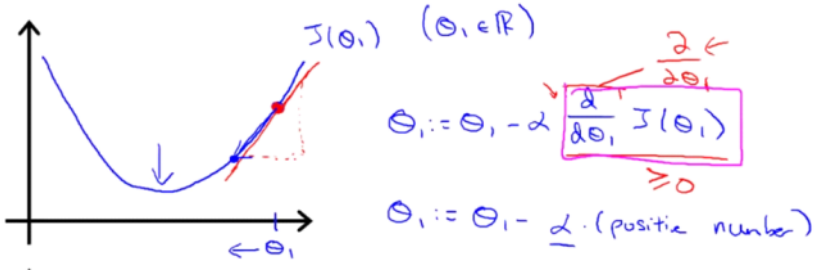
where

j=0,1 represents the feature index number.

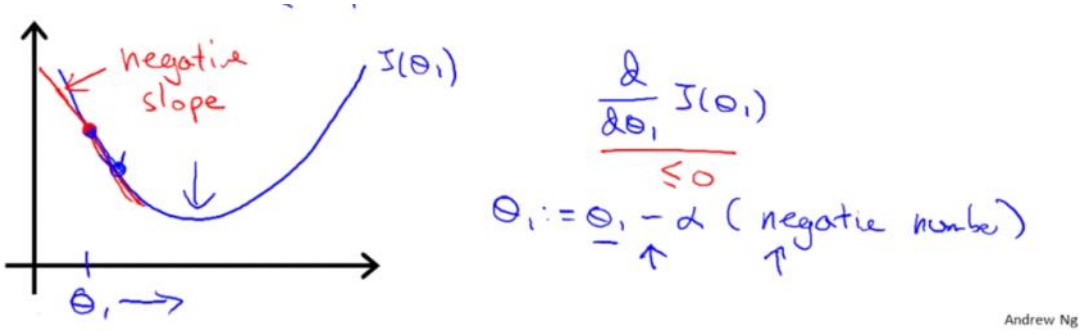


Now we will look at the intuition behind gradient descent

Lets again take only one theta for better understanding

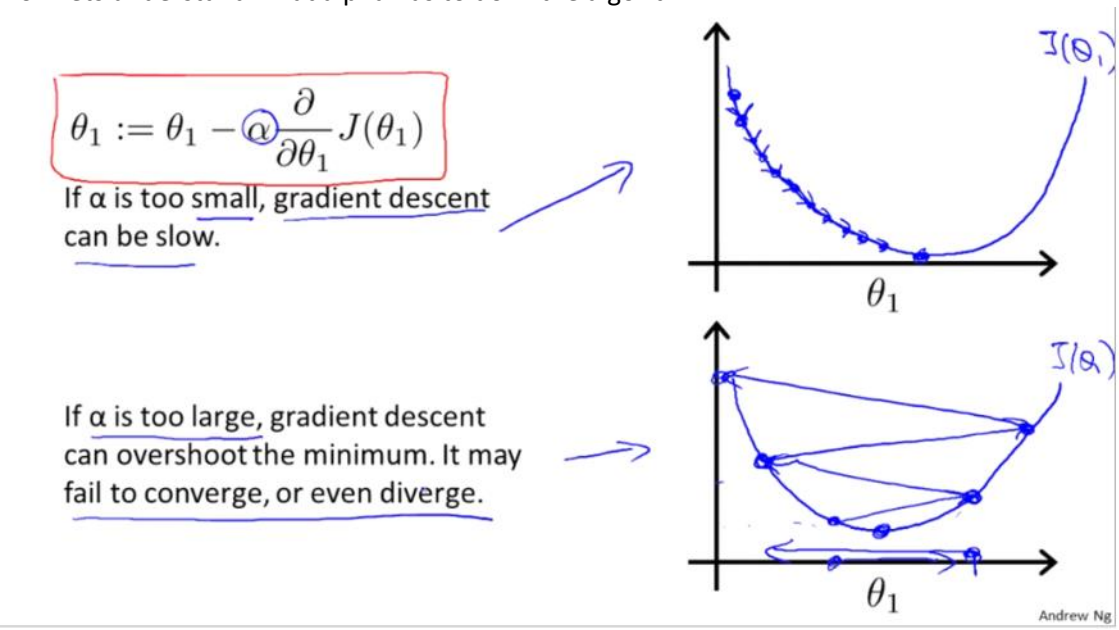


So we have the cost function graph. Lets say we started from theta 1.
Now if we take the derivative of the cost function at theta 1 then we will get slope of the tangent at that point. Now as can be seen the red line in the graph has a positive slope so we will have positive number as derivative. Now we will subtract (alpha multiplied by that positive number we got from the derivative) to the theta 1. And assign this new value to theta 1
Now what will happen is the theta 1 will move towards right in the graph which is towards local minima

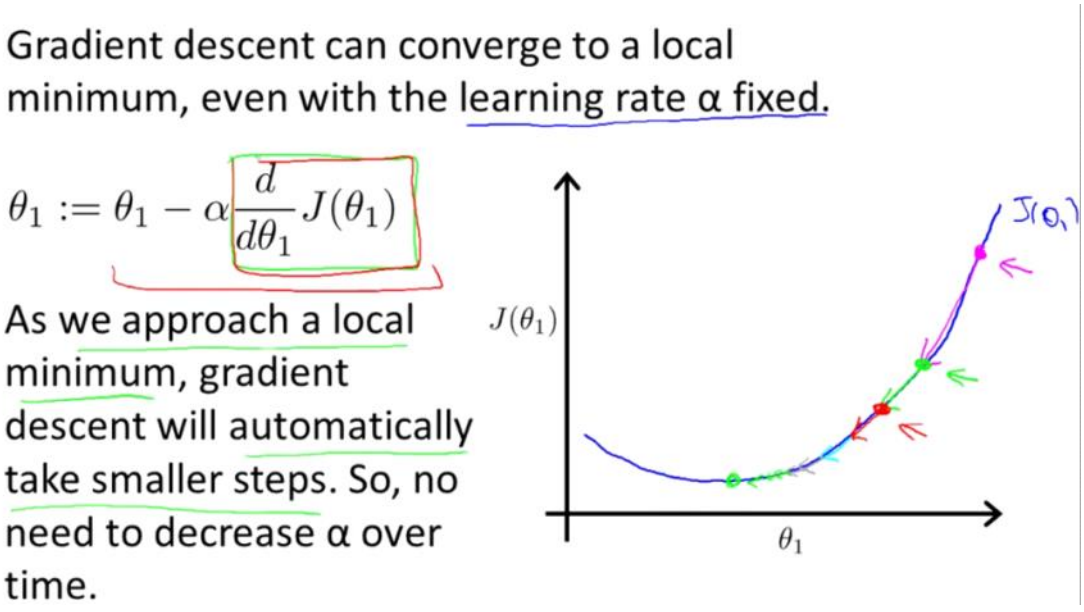


Now consider another scenario, here we will have negative derivative. Now what will happen is the negative will cancel each other and we will add the product of alpha and derivative to the theta

Now lets understand what alpha has to do in the algorithm



If theta is initialized at local minimum then we gradient descent will keep theta unchanged



Gradient descent can converge even if α is kept fixed. (But α cannot be too large, or else it may fail to converge.)
For the specific choice of cost function used in linear regression, there are no local optima (other than the global optimum).

Below is the formula for the gradient descent

repeat until convergence: {

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

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

}

"Batch" Gradient Descent

"Batch": Each step of gradient descent uses all the training examples.

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

So to calculate the gradient we are using all the point