

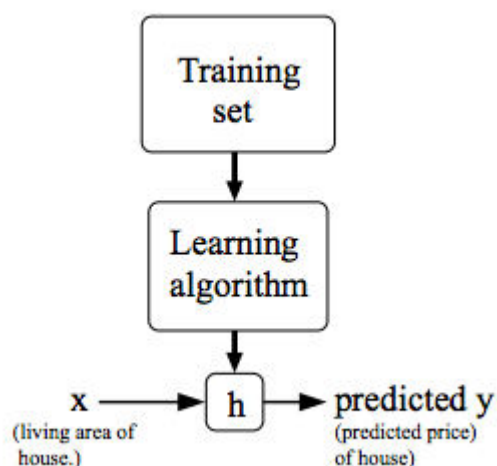
Model Representation and Cost Function

Model Representation

To establish notation for future use, we'll use $x(i)$ to denote the “input” variables (living area in this example), also called input features, and $y(i)$ to denote the “output” or target variable that we are trying to predict (price). A pair $(x(i), y(i))$ is called a training example, and the dataset that we'll be using to learn—a list of m training examples $(x(i), y(i)); i=1, \dots, m$ — is called a training set.

Note that the superscript “(i)” in the notation is simply an index into the training set, and has nothing to do with exponentiation. We will also use X to denote the space of input values, and Y to denote the space of output values. In this example, $X = Y = \mathbb{R}$.

To describe the supervised learning problem slightly more formally, our goal is, given a training set, to learn a function $h : X \rightarrow Y$ so that $h(x)$ is a “good” predictor for the corresponding value of y . For historical reasons, this function h is called a hypothesis. Seen pictorially, the process is therefore like this:

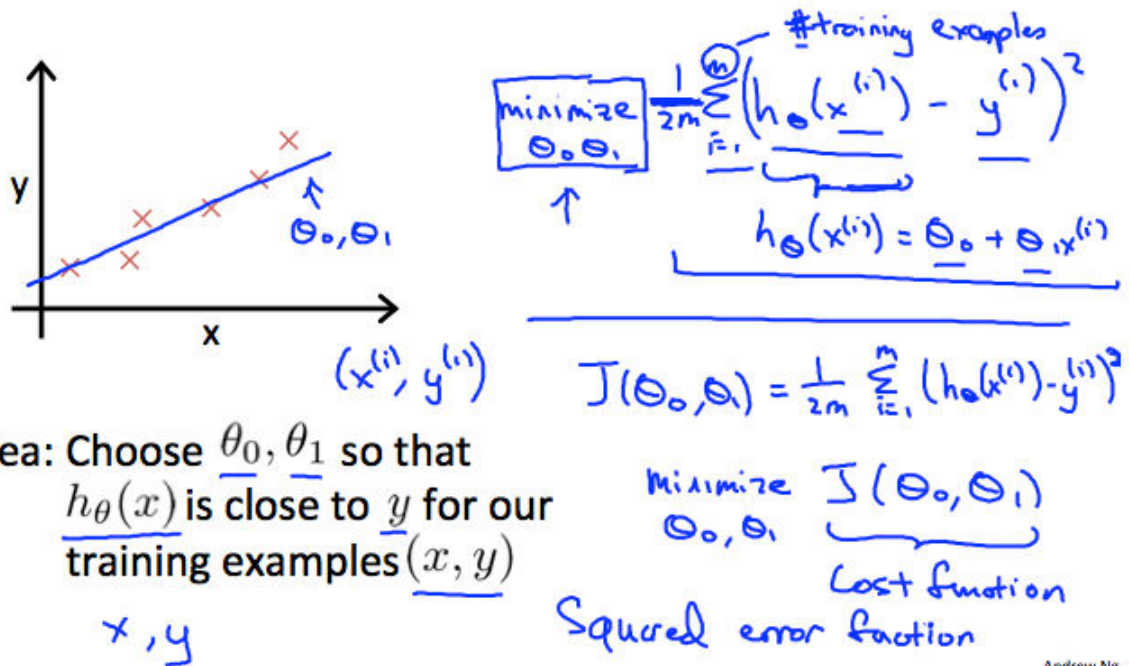


When the target variable that we're trying to predict is continuous, such as in our housing example, we call the learning problem a regression problem. When y can take on only a small number of discrete values (such as if, given the living area, we wanted to predict if a dwelling is a house or an apartment, say), we call it a classification problem.

Cost Function

We can measure the accuracy of our hypothesis function by using a **cost function**. This takes an average difference (actually a fancier version of an average) of all the results of the hypothesis with inputs from x 's and the actual output y 's.

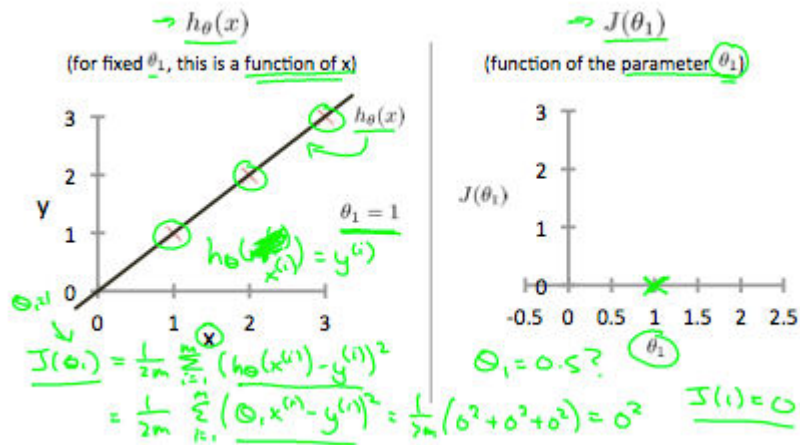
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$



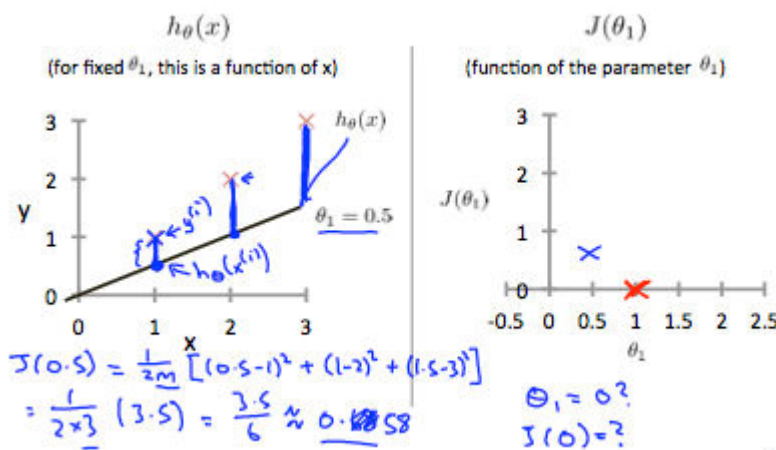
Andrew Ng

If we try to think of it in visual terms, our training data set is scattered on the x - y plane. We are trying to make a straight line $h_{\theta}(x)$ which passes through these scattered data points.

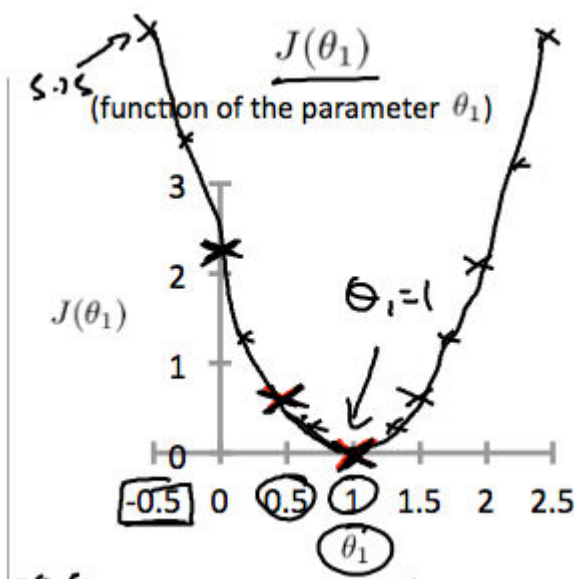
Our objective is to get the best possible line. The best possible line will be such so that the average squared vertical distances of the scattered points from the line will be the least. Ideally, the line should pass through all the points of our training data set. In such a case, the value of $J(\theta_0, \theta_1)$ will be 0. The following example shows the ideal situation where we have a cost function of 0.



When $\theta_1=1$, we get a slope of 1 which goes through every single data point in our model. Conversely, when $\theta_1=0.5$, we see the vertical distance from our fit to the data points increase.



This increases our cost function to 0.58. Plotting several other points yields to the following graph:



Thus as a goal, we should try to minimize the cost function. In this case, $\theta_1=1$ is our global minimum.