Notation: May 12, 2021

m: # of training examples
x: input variable/feature
y: output variable/feature
(x, y): one training example
$(x^{(i)}, y^{(i)})$: i^{th} training example

<u>Supervised Learning:</u>

Training Set

↓

Learning Algorithm

↓

input → h → output

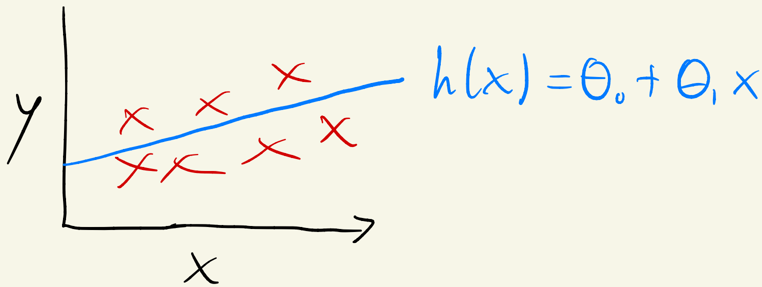
x hypothesis y

h is a function that maps x's to y's.
If y is continous, regression.
If y has discrete categories, classification.

So how do we represent h?

$h_\theta(x) = \theta_0 + \theta_1 x \rightarrow$ it's a linear function for now

or $h(x)$



↳ Linear regression w/ one variable, a.k.a. univariate linear regression

Cost Function:

$h_\theta(x) = \theta_0 + \theta_1 x \rightarrow \theta_0$ and θ_1 are trained parameters

The idea is to choose θ_0, θ_1 s.t. $h(x)$ is close to y for our training examples (x, y).

How?

e.g. Minimize square difference of prediction and actual output.

$\hookrightarrow \frac{1}{2m}\sum_{i=1}^{m}\left[\underbrace{h_\theta(x^{(i)})}_{=\theta_0+\theta_1 x} - y^{(i)}\right]^2$, minimize

Cost Function:

The $\frac{1}{2}$ is just for cancelling out the 2 from the derivative - convenient

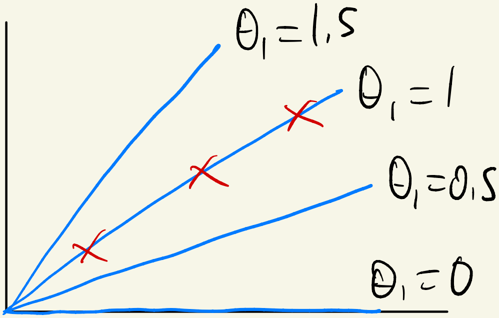
$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$ ← Mean Squared Error

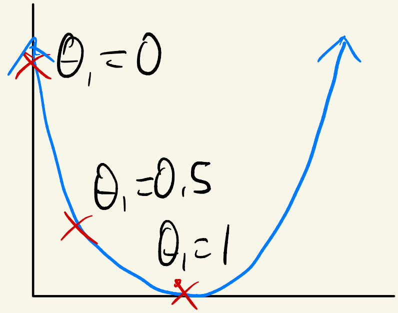
$\hookrightarrow$ minimize this

takes the average difference of all the results of the hypothesis

Let $h_\theta(x) = \theta_1 x$ for now

$h_\theta(x)$ $\quad\quad\quad$ $J(\theta_1)$

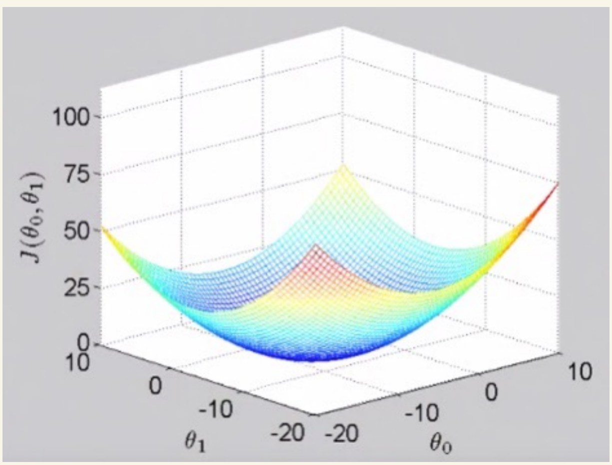




Parabola cuz

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left[\theta_1 x^{(i)} - y^{(i)}\right]^2$$

For $J(\theta_0, \theta_1)$:



So then what is the algorithm we use to minimize our cost function?