Machine Learning

Lecture - 2

Muhammad Affan Alim

Learning Algorithm(Function)

Regression

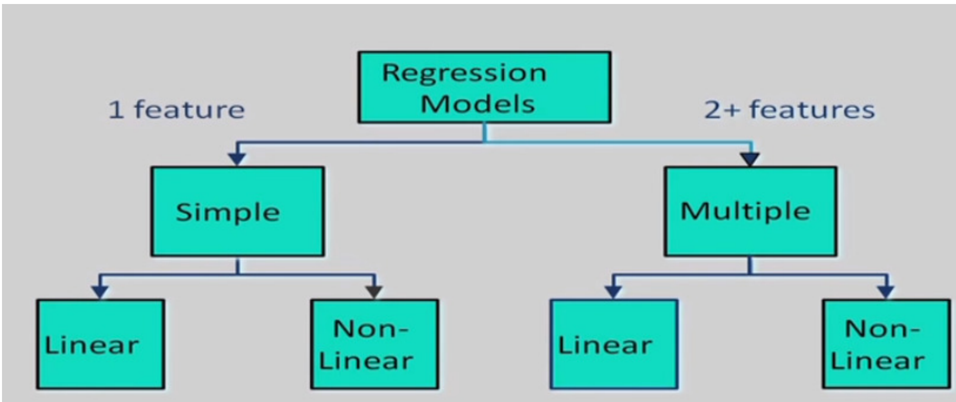
- Regression is a supervised learning problem
- At given X and Y, we need to develop a model of linear regression which predict Y on new value of X

$$f : X \longrightarrow Y$$

- For regression Y is continuous

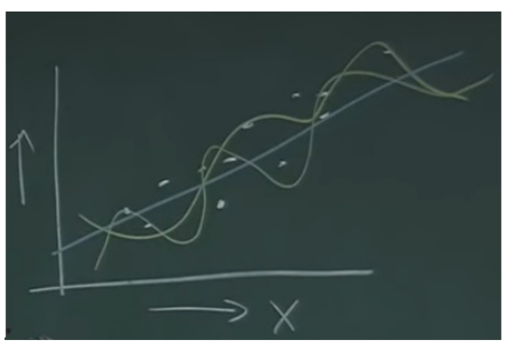
Hierarches of Regression

- Many types of regression models can be used



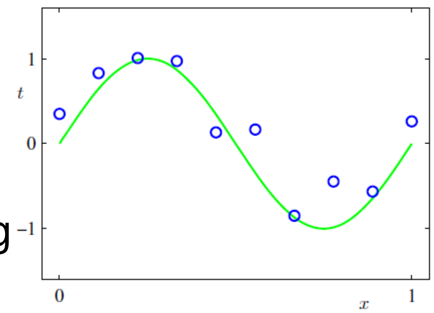
Linear Regression

- The simplest function is linear function for regression
  - Simple regression where instance *x* depends on single variable
  - Multiple regression where instance *x* depends on multiple variables



A simple Example: Fitting a Polynomial

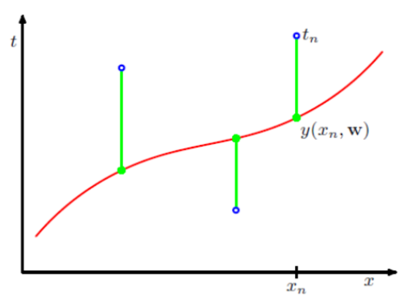
- There green curve is the true function (which is not a polynomial)
- Plot of a training data set of N=10 points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding target variable *t*.



- The green curve shows the function sin(2πx) used to generate the data

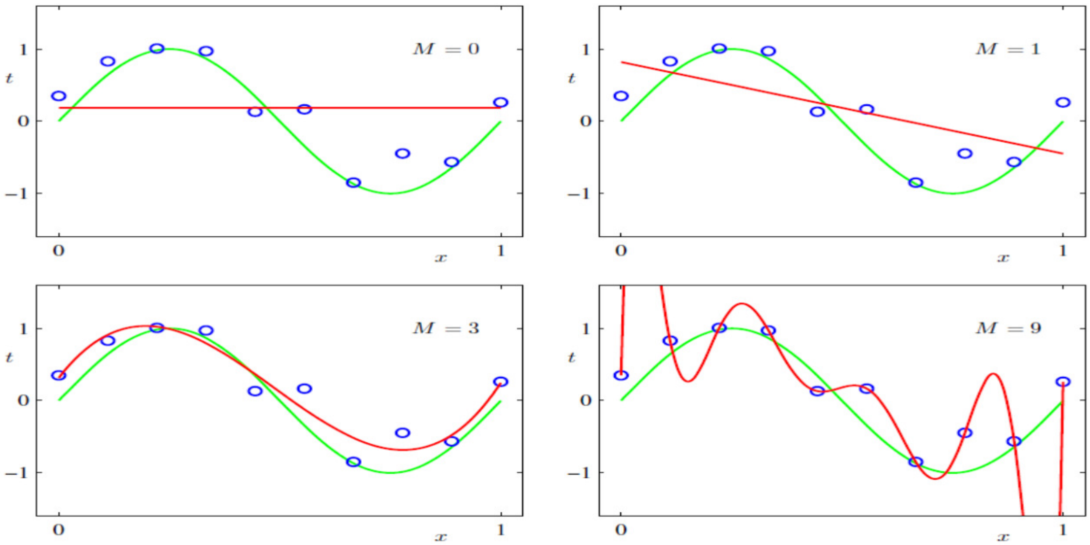
A simple Example: Fitting a Polynomial

- We may used a ***loss function*** that measures the ***squared error*** in the prediction of y(x) from x



Some fits to the data: which is best? (1/2)

•

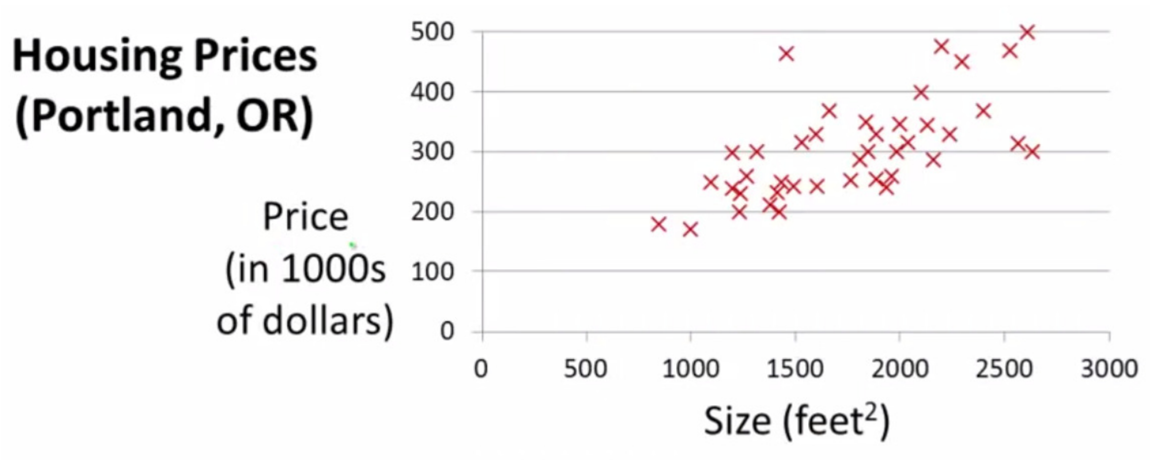


Some fits to the data: which is best? (2/2)

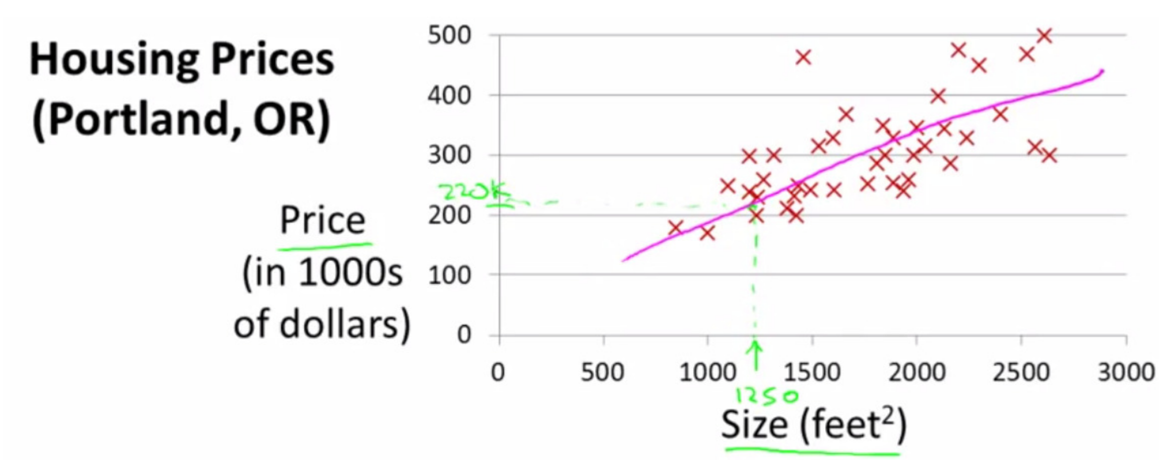
Explanation

- We notice that the constant (M=0) and first order (M=1) polynomials give rather poor fits to the data and consequently rather poor representations of the function sin(2 x). The third order (M=3) polynomial seems to give the best fit to the function sing (2πx) of the examples shown in figure.
- When we go to a much higher order polynomial (M=9), we obtain an excellent fit to the training data.
- In fact, the polynomial passes exactly through each data point and E(w*) = 0.

Linear Regression – Example (1/7)



Linear Regression – Example (2/7)



This an example of supervised learning

Linear Regression – Example (3/7)

- d

Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

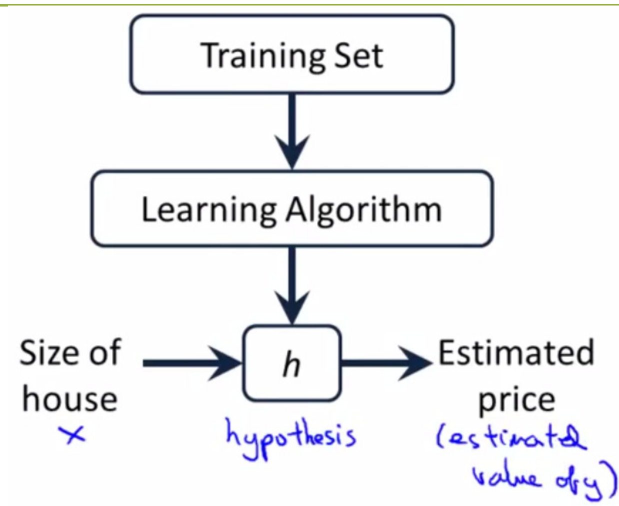
$m = 47$

Linear Regression – Example (4/7)

- Training set (this is your data set)
- Notation (used throughout the course)

m = number of training examples
x's = input variable "target" variables
y's = output variable target example
(x, y) - single training example
(x^i, y^j) - specific example (i^{th} training example)
i is index to training set

Linear Regression – Example (5/7)

- 

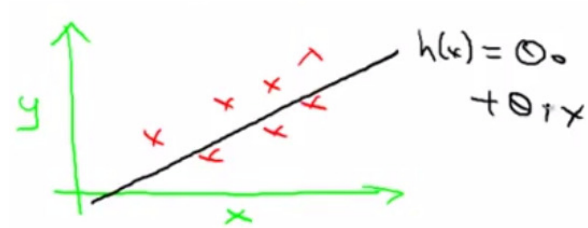
- h is a function which maps x's to y's

Linear Regression – Example (6/7)

- With our training set defined - how do we used it?
  - Take training set
  - Pass into a learning algorithm
  - Algorithm outputs a function (denoted h) (*h* = hypothesis)
    - This function takes an input (e.g. size of new house)
    - Tries to output the estimated value of Y

Linear Regression – Example (7/7)

- How do we represent hypothesis *h* ?
  - Going to present h as;
  - $h_\theta(x) = \theta_0 + \theta_1 x$
    - *h*(x) (shorthand)



- What does this mean?
- Means Y is a linear function of x!
- θ_i are parameters
  - θ_0 is zero condition
  - θ_1 is gradient
- Linear regression with one variable or univariate linear regression

Linear Regression – Cost Function (1/7)

- $h_\theta(x) = \theta_0 + \theta_1 x$

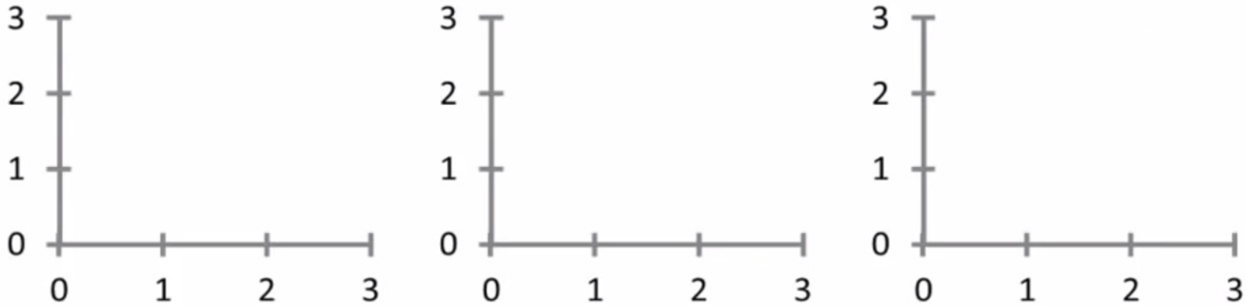
Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

m = 47

- A cost function lets us figure out how to fit the best straight line to our data
- Choosing values for θ_i (parameters)

Linear Regression – Cost Function (2/7)

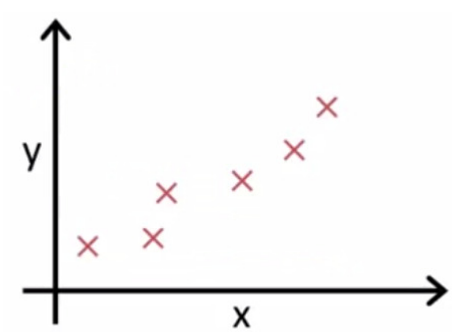
$h_\theta(x) = \theta_0 + \theta_1 x$

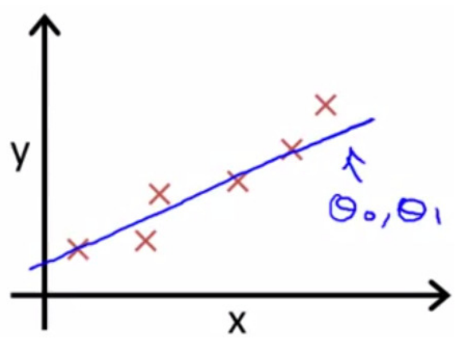


$\theta_0 = 1.5$	$\theta_0 = 0$	$\theta_0 = 1$
$\theta_1 = 0$	$\theta_1 = 0.5$	$\theta_1 = 0.5$

Linear Regression – Cost Function (3/7)

- $h_\theta(x) = \theta_0 + \theta_1 x$





Linear Regression – Cost Function (4/7)

- ***Idea***: Choose θ_1, θ_2 so that *h*(x) is close to y for our training examples (x, y)
- To formalize this;
  - We want to want to solve a minimization problem
  - Minimize $(h_\theta(x) - y)^2$
    - i.e. minimize the difference between h(x) and y for each/any/every example
- Sum this over the training set

Linear Regression – Cost Function (5/7)

- Sum this over the training set

#training examples

$$\underset{\theta_0 \theta_1}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

- Minimize squared different between predicted house price and actual house price

Linear Regression – Cost Function (6/7)

- ***Note:***
- (1/2)m
- 1/m - means we determine the average
- (1/2)m the 2 makes the math a bit easier, and doesn't change the constants we determine at all (i.e. half the smallest value is still the smallest value!)

Linear Regression – Cost Function (7/7)

- More cleanly, this is a cost function

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

$$\underset{\theta_0, \theta_1}{\text{minimize}} \; \underbrace{J(\theta_0, \theta_1)}_{\text{Cost function}}$$

- Some time it is called *squared error function*

Linear Regression – Cost Function Summary

- Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

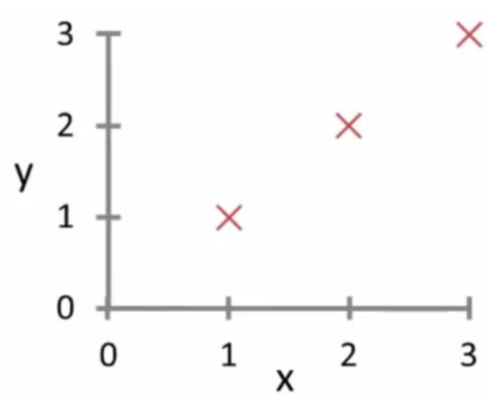
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Linear Regression – Cost Function Example (1/4)

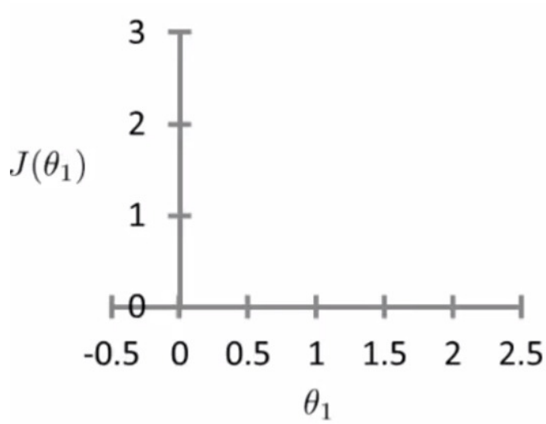
$h_\theta(x)$

(for fixed θ_1 this is a function of x)



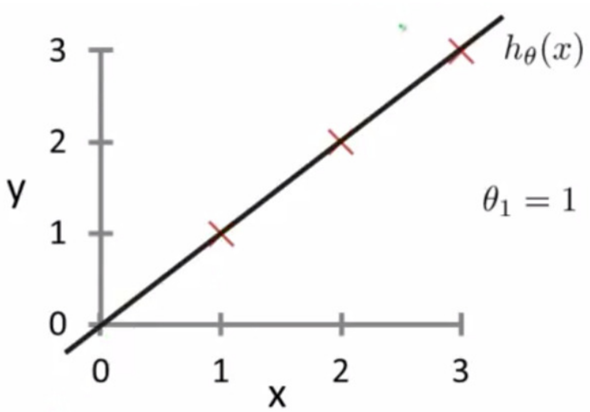
$J(\theta_1)$

Function of the parameter θ_1

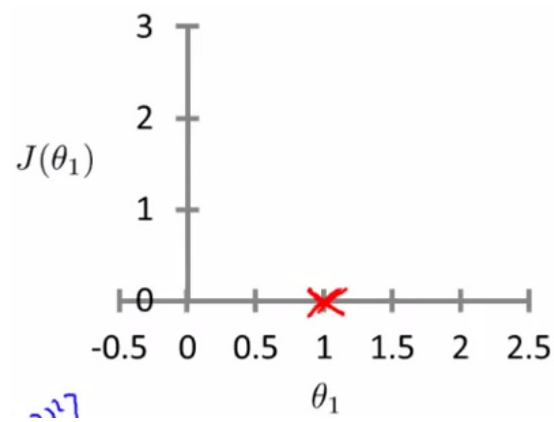


Linear Regression – Cost Function Example (2/4)

$h_\theta(x)$
(for fixed θ_1 this is a function of x)



$J(\theta_1)$
Function of the parameter θ_1



Linear Regression – Cost Function Example (3/4)

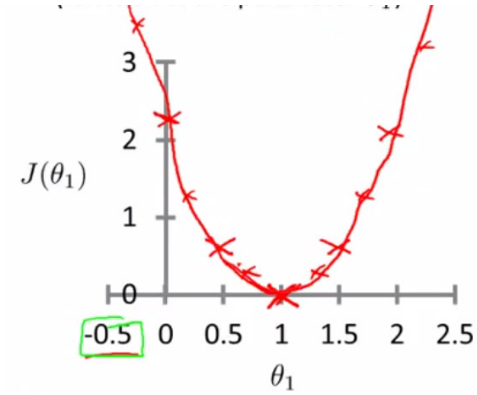
If θ_1=1 (page number 13)
J(1) = 0

If θ_1=0.5
J(0.5) = 0.58

If θ_1=0
J(0) = 2.3

We can do for other values of θ_1's

Linear Regression – Cost Function Example (4/4)



- The optimization objective for the learning algorithm is find the value of θ_1 which minimizes $J(\theta_1)$
- So, here θ_1 = 1 is the best value for θ_1

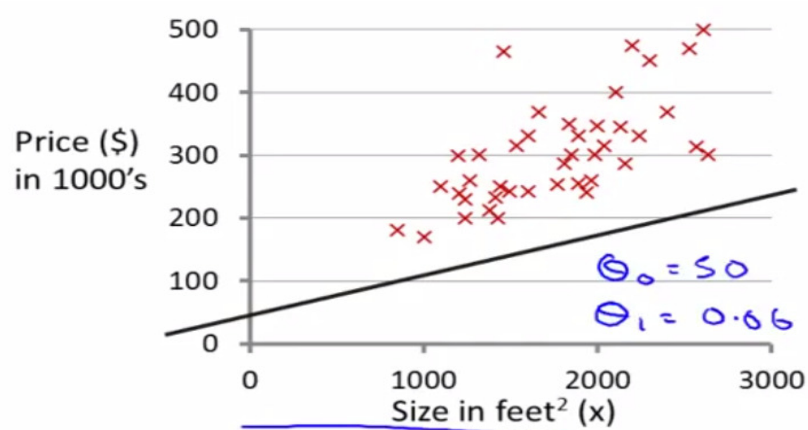
Linear Regression – Cost Function
Two or more constants (1/10)

- Assume you're familiar with contour plots or contour figures
  - Using same cost function, hypothesis and goal as previously
  - It's OK to skip parts of this section if you don't understand contour plots
- Using our original complex hypothesis with two variables,
  - So cost function is
    $J(\theta_0, \theta_1)$

Linear Regression – Cost Function
Two or more constants - Example (2/10)

- Suppose
  - $\theta_0 = 50$ and $\theta_1 = 0.06$
- Previously we plotted our cost function by plotting θ_1 vs $J(\theta_1)$
- Now we have two parameters
  - Plot becomes a bit more complicated
  - Generates a 3D surface plot where axis are
    - $X = \theta_1$
    - $Z = \theta_0$
    - $Y = J(\theta_0, \theta_1)$

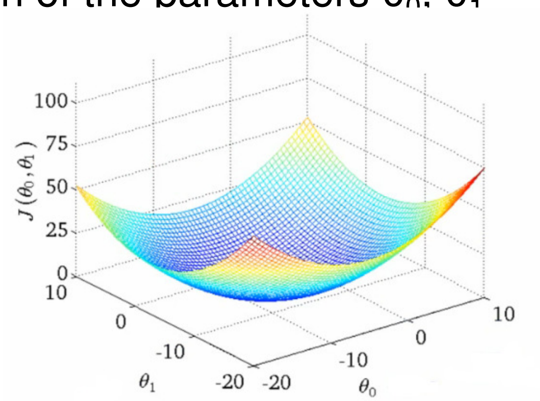
Linear Regression – Cost Function
Two or more constants – Example (3/10)



$$\boxed{h_\theta(x) = 50 + 0.06x}$$

Linear Regression – Cost Function
Two or more constants - Example (4/10)

- Function of the parameters θ_0, θ_1



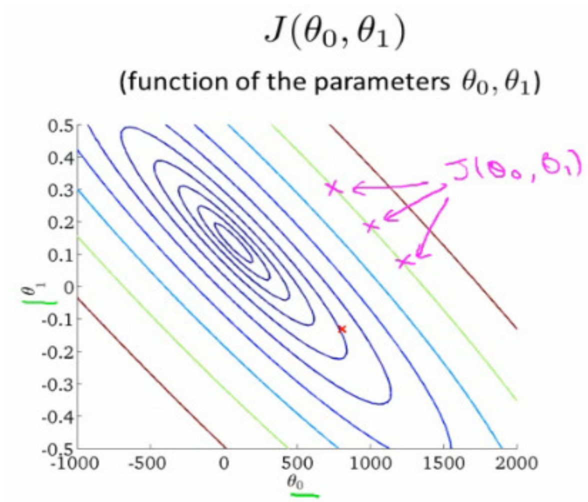
- We can see that the height (y) indicates the value of the cost function, so find where y is at a minimum

Linear Regression – Cost Function
Two or more constants - Example (5/10)

- Instead of a surface plot we can use a **contour figures/plots**
  - Set of ellipses in different colors
  - Each color is the same value of $J(\theta_0, \theta_1)$, but obviously plot to different locations because θ_1 and θ_0 will vary
  - Imagine a bowl shape function coming out of the screen so the middle is the concentric circles

Linear Regression – Cost Function
Two or more constants - Example (6/10)

- x



Linear Regression – Cost Function
Two or more constants – Example (7/10)

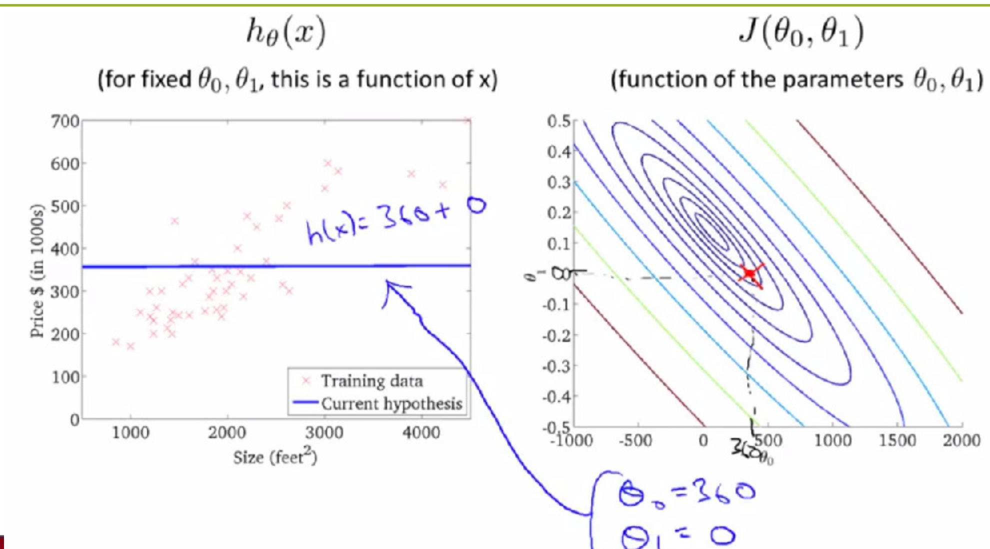
- Each point (like the red one above) represents a pair of parameter values for Θ_0 and Θ_1
  - Our example here put the values at
    - θ_0 = ~800
    - θ_1 = ~-0.15
- Not a good fit
- i.e. these parameters give a value on our contour plot far from the center

Linear Regression – Cost Function
Two or more constants – Example (8/10)

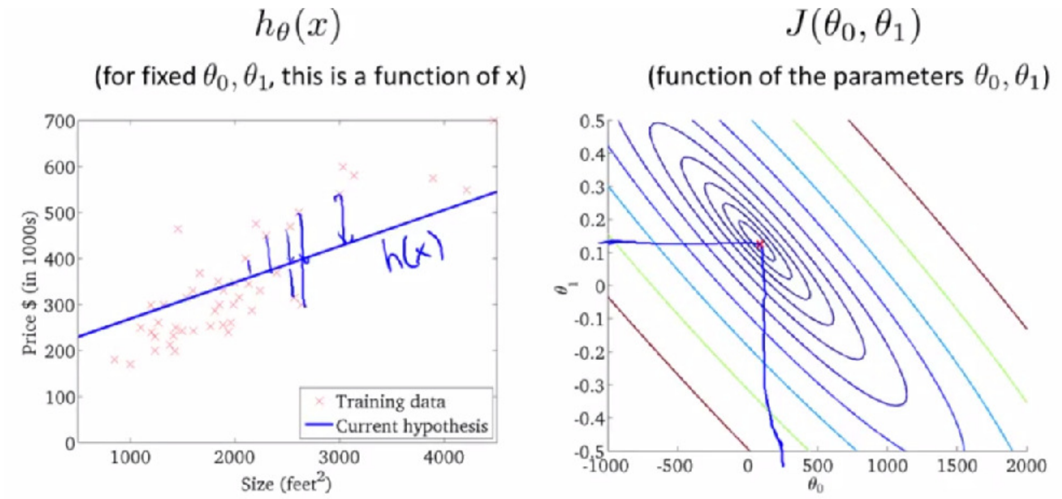
- If we have
  - θ_0 = ~360
  - $\theta_1 = 0$
  - This gives a better hypothesis, but still not great - not in the center of the contour plot
- Finally we find the minimum, which gives the best hypothesis
- Doing this by eye/hand is a pain
- What we really want is an efficient algorithm fro finding the minimum for θ_0 and θ_1

Linear Regression – Cost Function
Two or more constants – Example (9/10)

-



Linear Regression – Cost Function
Two or more constants – Example (10/10)

- 

Gradient Decent Algorithm

- It uses all over machine learning domain for finding the minimum cost function
- <u>IMPORTANCE</u>
  - After finding the first values of parameters of θ then it would be difficult to find the next best value of θ which converge the minimum of cost function.
- Basically, gradient descent algorithm depends on first order newton Rapson's method

Gradient Decent Algorithm

- The more generalized gradient decent algorithm is

  repeat until convergence {

  $$\Theta_j = \Theta_j - \alpha \frac{\partial}{\partial \Theta_j} J(\Theta_0, \Theta_1)$$

  (simultaneously updated j= 0 and j=1)

  }

here, α = learning rate

Gradient Decent Algorithm - Example

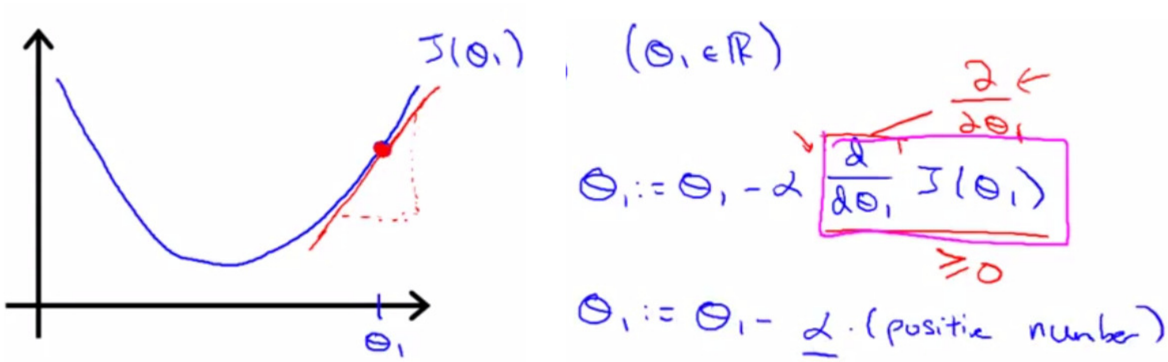
- Consider the previous example, where we have selected single parameter θ_1 then the cost function is

  min $J(\theta_1)$

- The derivative of any curve at specific point produces the slope of tangent at same point

Gradient Decent Algorithm - Example

- Suppose our cost function is



Gradient Decent Algorithm - Example

- In given diagram, at point θ_1, the slope of tangent produces the positive value because of the acute inclination

$$\frac{d}{d\theta_1} J(\theta_1) \leq 0$$

then

$$\frac{d}{d\theta_1} J(\theta_1) \geq 0$$ positive value of

now gradient descent function

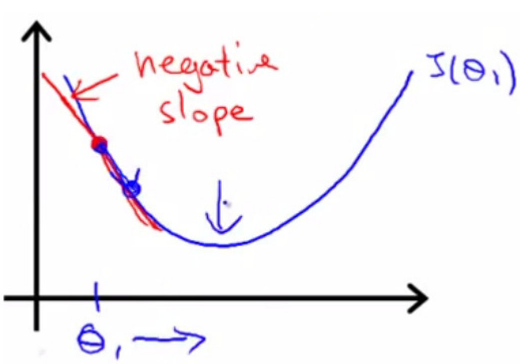
$$\theta_1 = \theta_1 - \alpha(positive\ value)$$

Gradient Decent Algorithm - Example

- In this case the value of new θ_1 will be smaller then the old θ_1
- It will move (converge) to left side and approach to min $J(\theta_1)$

Gradient Decent Algorithm - Example

- Consider the second diagram



$$\underbrace{\frac{d}{d\theta_1}J(\theta_1)}_{\leq 0}$$

$$\theta_1 := \theta_1 - \alpha\,(\text{negatie number})$$

Gradient Decent Algorithm - Example

- Here at the point θ1, the derivative value (slope of the tangent) will be negative because of obtuse angle of tangent

$\theta_1 = \theta_1 - \alpha(negative\ value)$

$$\frac{d}{d\theta_1} J(\theta_1) \leq 0$$ negative value

- now gradient descent function

$$\theta_1 = \theta_1 - \alpha(negative\ value)$$

Gradient Decent Algorithm - Example

- In this case the value of new θ_1 will be greater then the old θ_1
- It will move (converge) to right-side and approach to min $J(\theta_1)$

Gradient Decent for Linear Regression

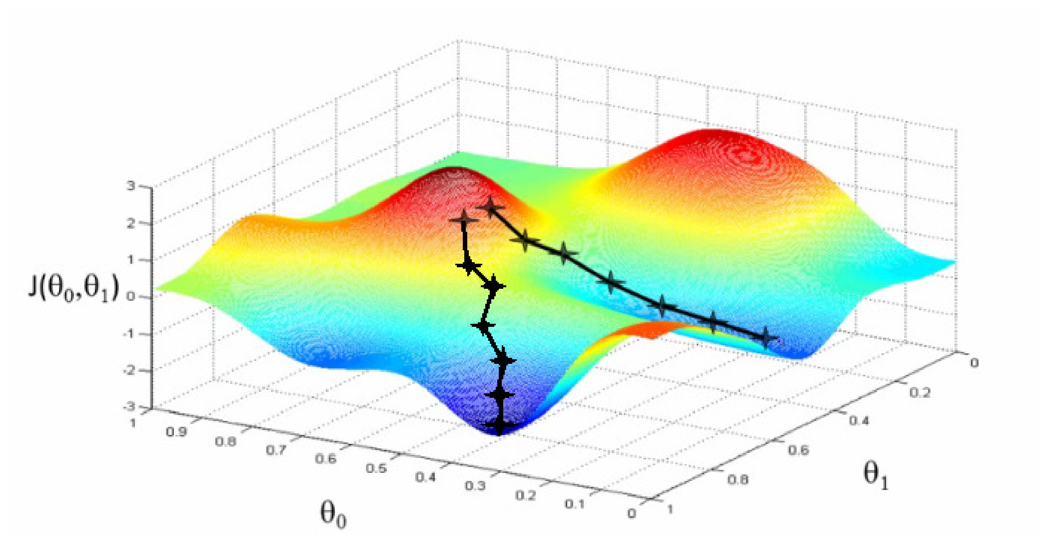
- Gradient descent algorithm
  repeat until convergence {

$$\Theta_j = \Theta_j - \alpha \frac{\partial}{\partial \Theta_j} J(\Theta_0, \Theta_1) \quad \ldots\text{(i)}$$

  }

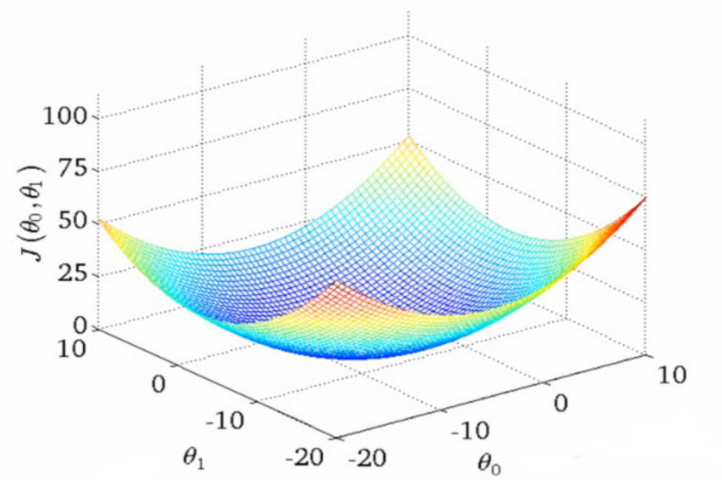
  see copy for further prove

Gradient Decent Algorithm



Cost Function with two parameters (θ_0 , θ_1)

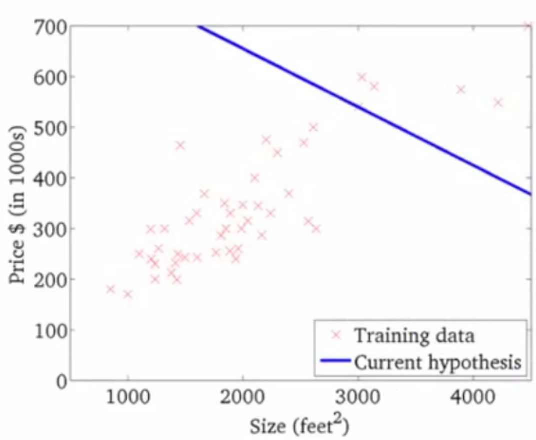
- Function of the parameters θ_0, θ_1



Gradient Decent Algorithm

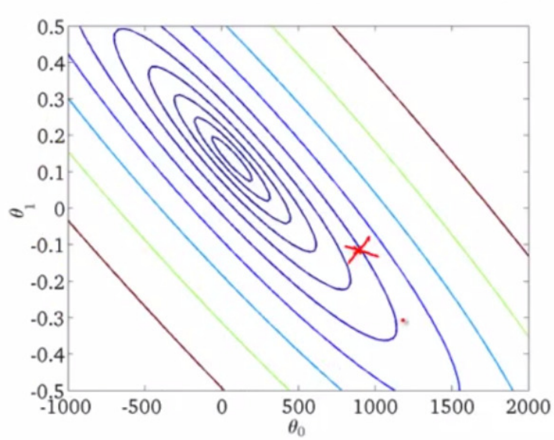
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

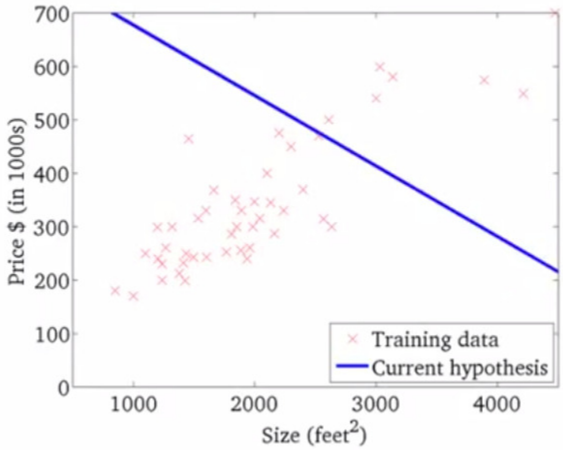
(function of the parameters θ_0, θ_1)



Gradient Decent Algorithm

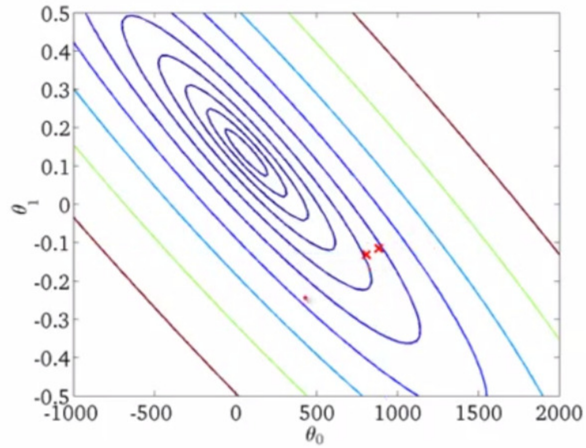
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

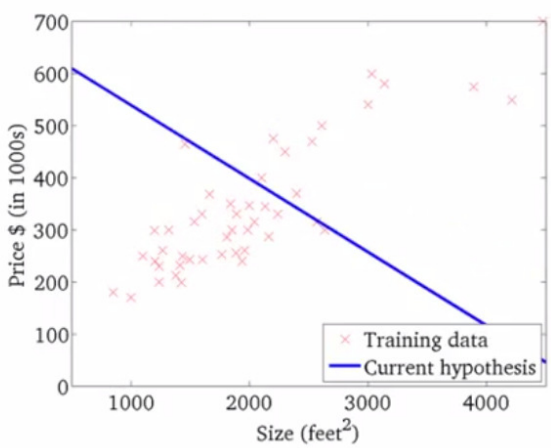
(function of the parameters θ_0, θ_1)



Gradient Decent Algorithm

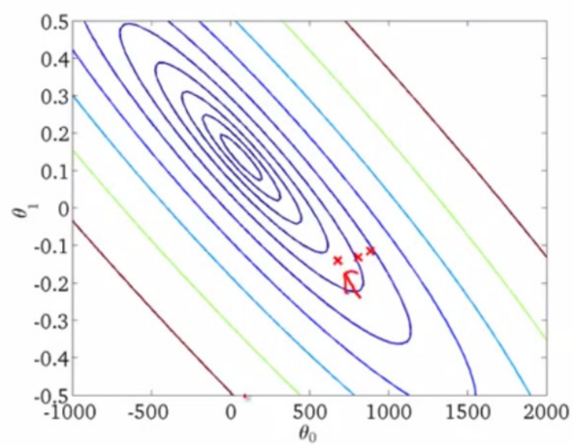
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

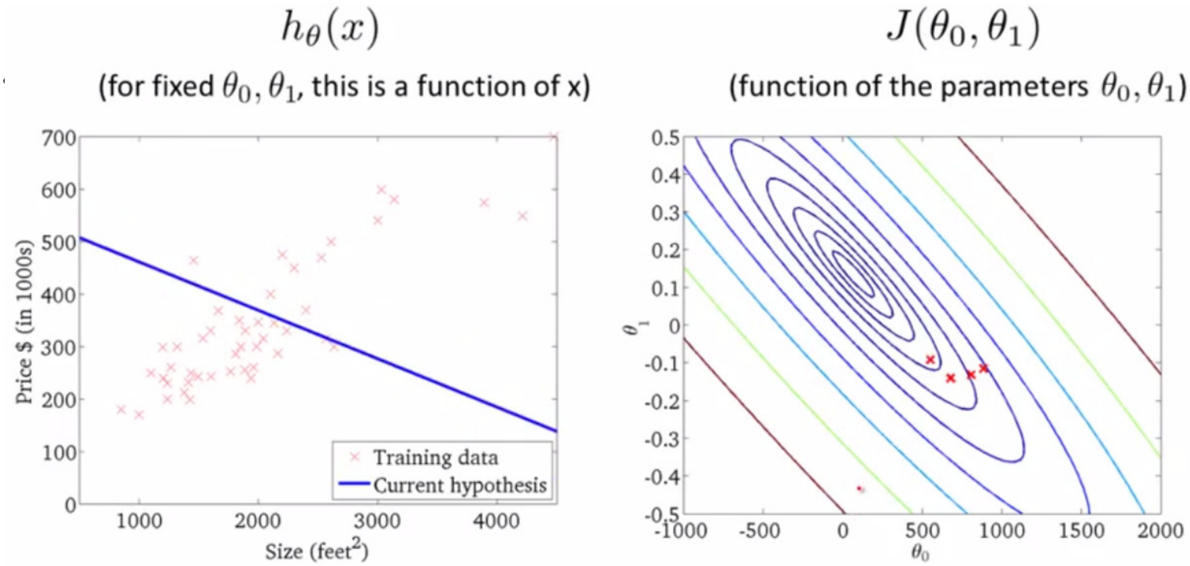


$J(\theta_0, \theta_1)$

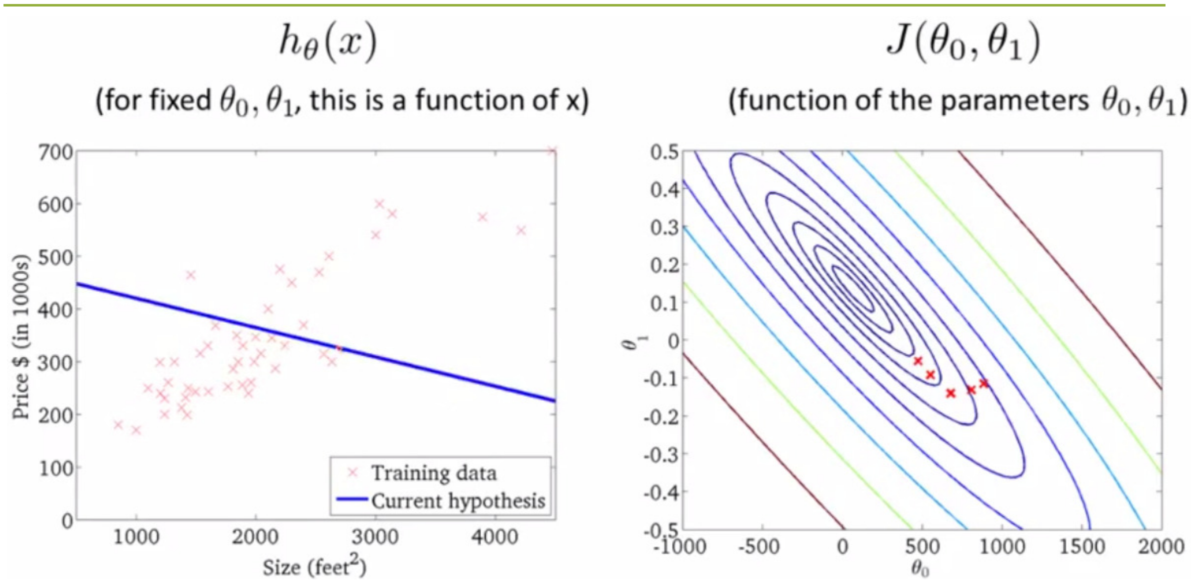
(function of the parameters θ_0, θ_1)



Gradient Decent Algorithm



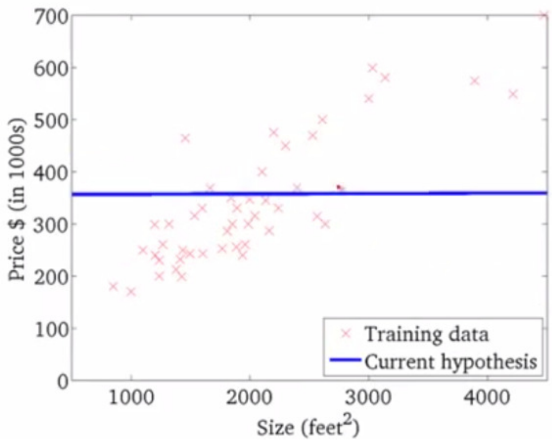
Gradient Decent Algorithm



Gradient Decent Algorithm

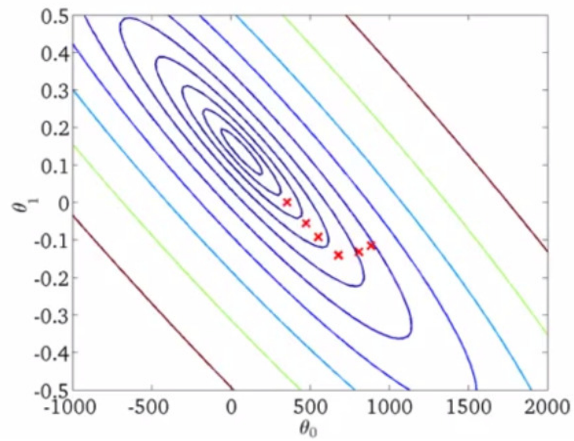
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

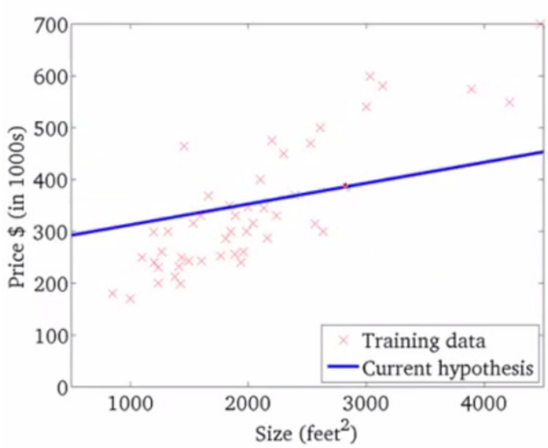
(function of the parameters θ_0, θ_1)



Gradient Decent Algorithm

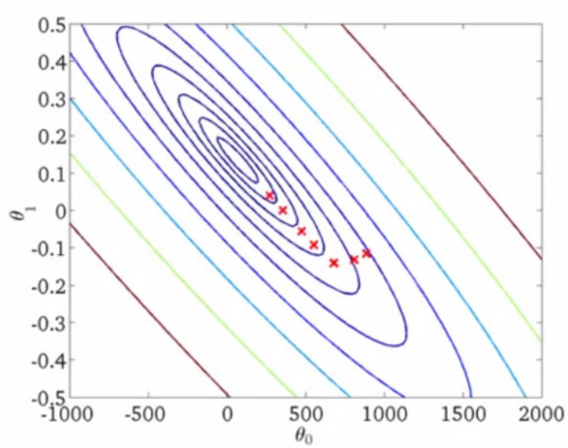
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

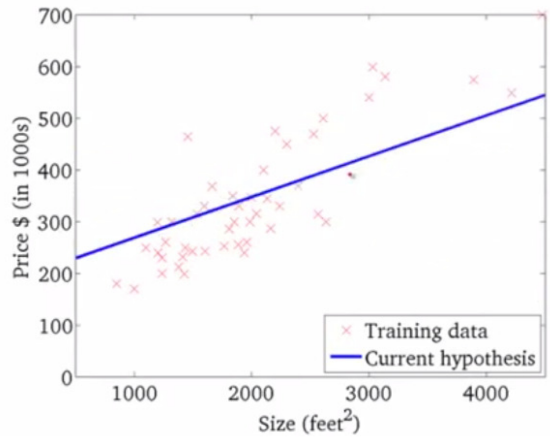
(function of the parameters θ_0, θ_1)



Gradient Decent Algorithm

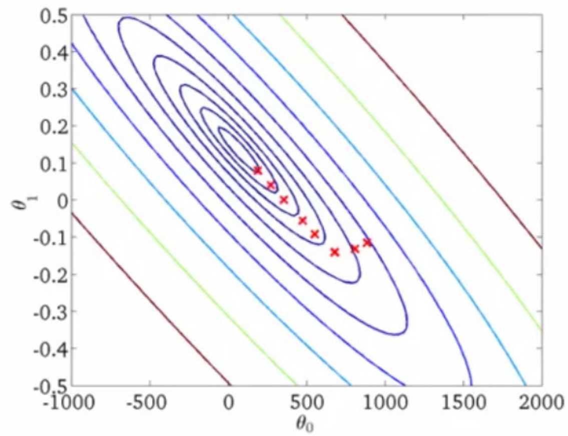
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

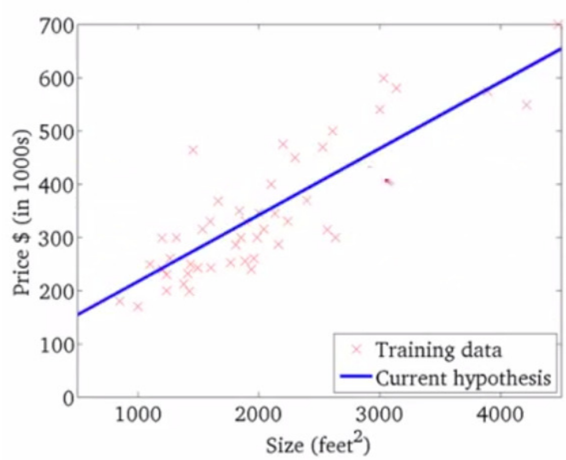
(function of the parameters θ_0, θ_1)



Gradient Decent Algorithm

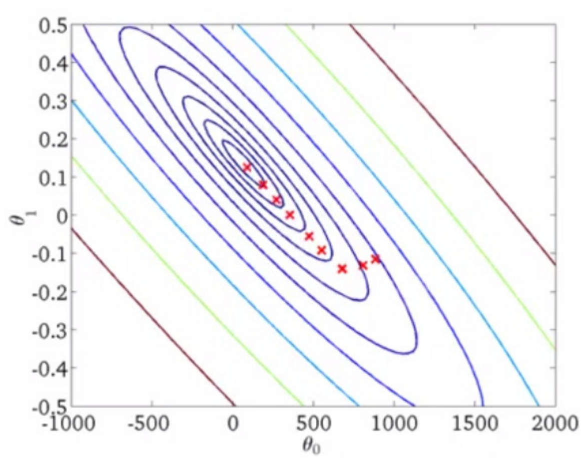
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



Gradient Decent Algorithm