innovate achieve lead

BITS Pilani

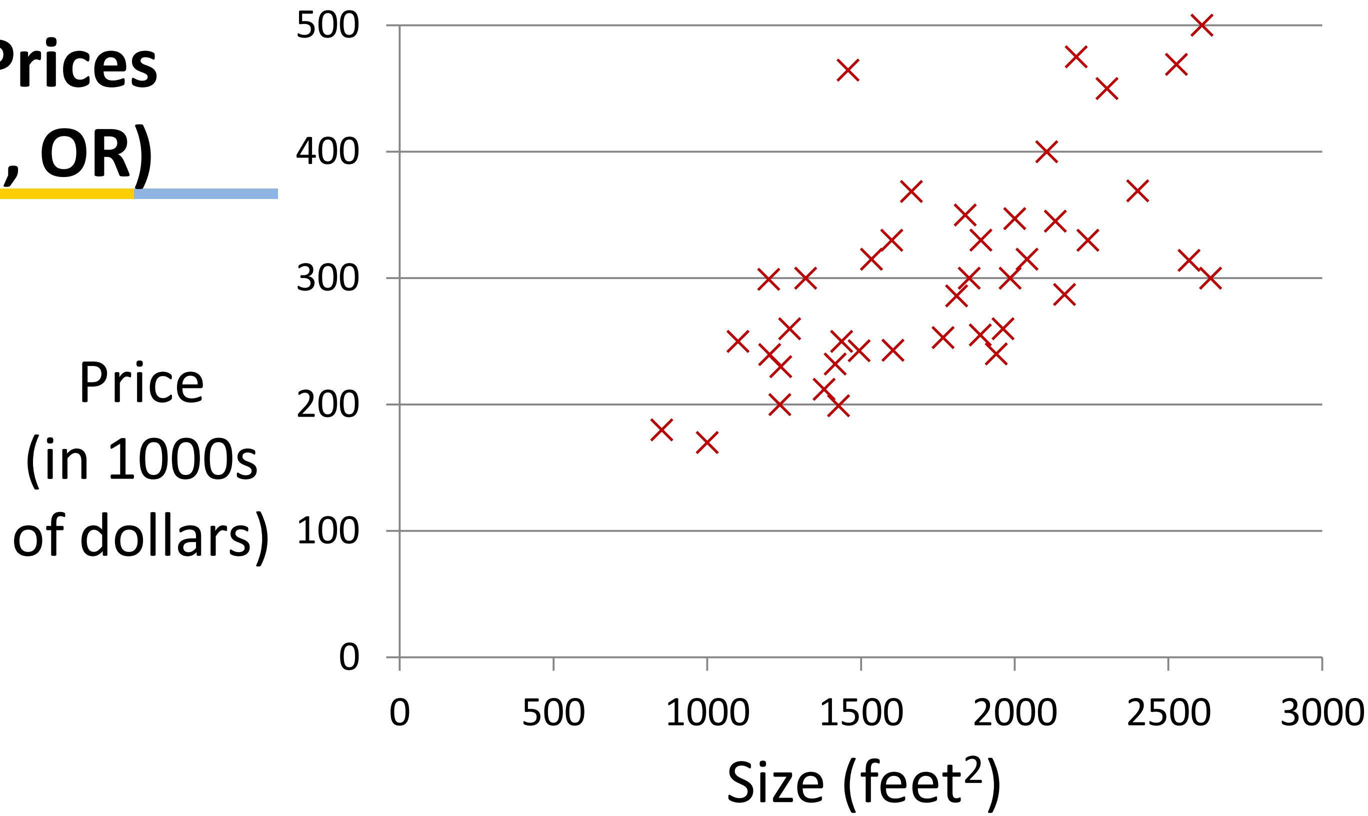
Machine Learning (IS ZC464) Session 6:

Linear models for Regression

What is Regression?

- The goal of **regression** is to predict the value of one or more continuous target variables ‘t’ given the value of a D-dimensional vector x of input variables.
- Polynomial curve fitting is an example of regression.

Housing Prices (Portland, OR)



<u>Supervised Learning</u>

Given the “right answer” for each example in the data.

<u>Regression Problem</u>

Predict real-valued output

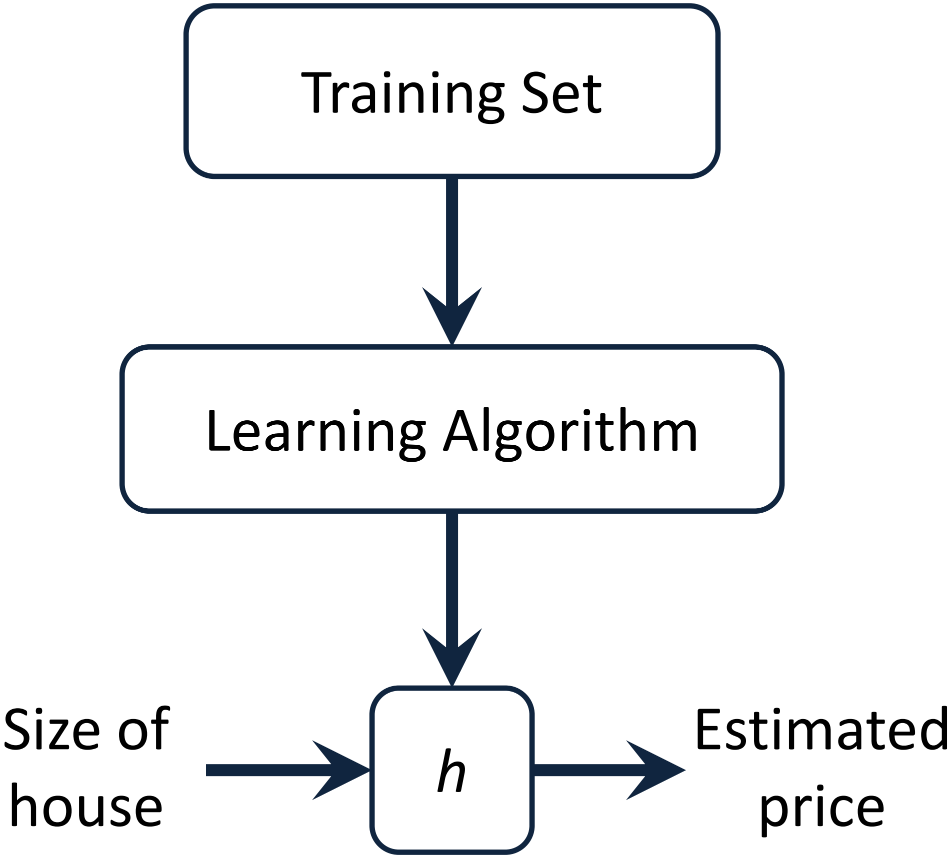
Slides adapted from Coursera Courseware on Machine Learning course offered by Prof. Andrew Ng.

Training set of housing prices

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

Notation:

m = Number of training examples
x's = "input" variable / features
y's = "output" variable / "target" variable



How do we represent *h* ?

Hypothesis:

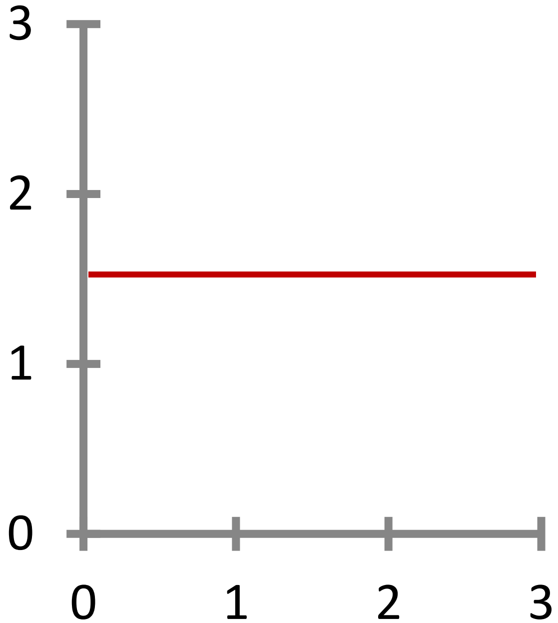
$$h_\theta(x) = \theta_0 + \theta_1 x$$

θ_i‘s: Parameters

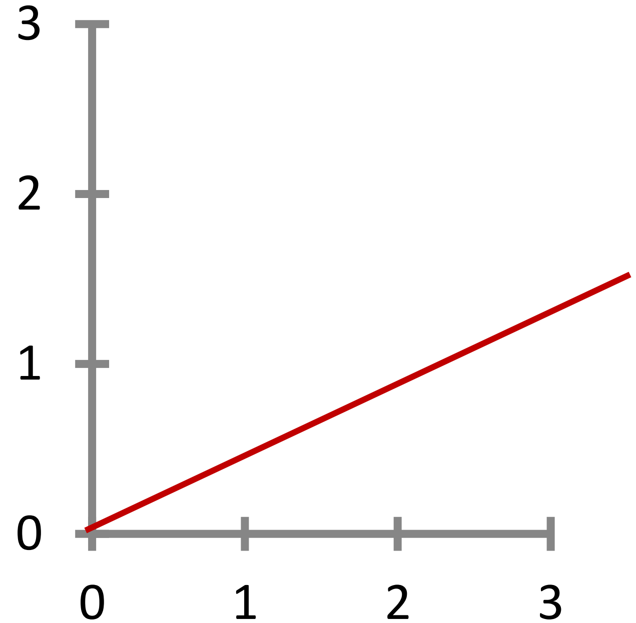
How to choose θ_i‘s ?

Linear regression with one variable.
Univariate linear regression.

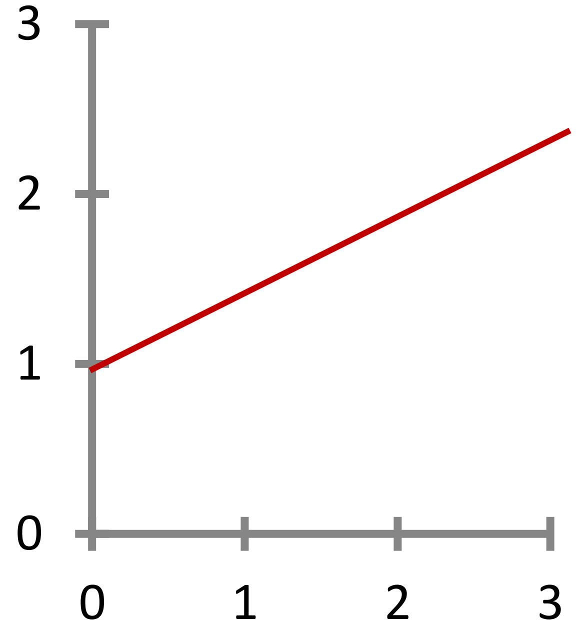
$$h_\theta(x) = \theta_0 + \theta_1 x$$



$\theta_0 = 1.5$
$\theta_1 = 0$



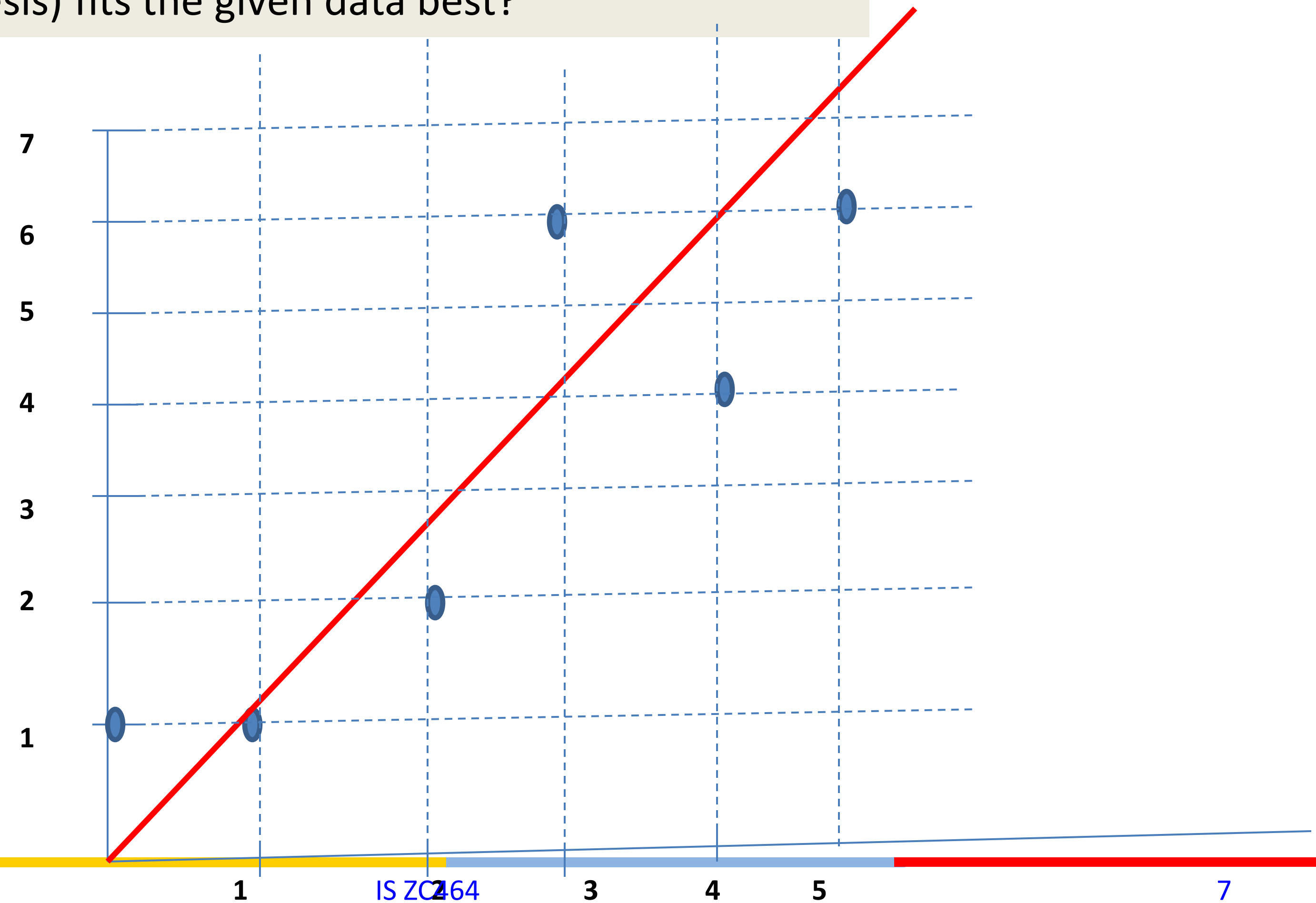
$\theta_0 = 0$
$\theta_1 = 0.5$



$\theta_0 = 1$
$\theta_1 = 0.5$

Recall: example to understand ERROR

Which line(hypothesis) fits the given data best?



Terminology

- Use parameters θ_0 and θ_1 to represent intercept and slope of line
- Use $J(\theta_0, \theta_1)$ to represent the Error.
- Instead of root Mean Squared (RMS) error, consider Squared error.
- The number of training examples = m
- The i^{th} data is $x^{(i)}$
- The i^{th} target is $y^{(i)}$

Hypothesis

- Equation(1):

$$h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

Note: The notations used in Bishop's book are as follows

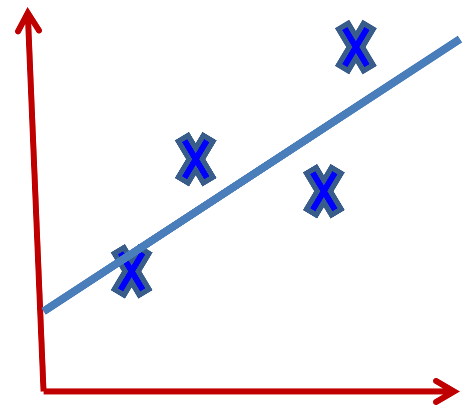
1. In place of parameters θ, The book uses the notion of w (later will be referred to as weights)
2. In place of <$x^{(i)}, x^{(2)}, x^{(3)},\ldots, x^{(m)}$>, the book uses vector x
3. In place of <$y^{(i)}, y^{(2)}, y^{(3)},\ldots, y^{(m)}$>, the book uses vector y.
4. In place of $h_\theta(x^{(i)})$, the book uses y(x,w) given by

   $\mathbf{y(x,w) = w_0 + w_1 x}$ (which is equivalent to equation (1))

Objective

- To find θ_0, θ_1 to minimize $J(\theta_0, \theta_1)$
- $J(\theta_0, \theta_1)$ is given by the expression

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



- Objective Function

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

Simplified ($\theta_0 = 0$)

Hypothesis:

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

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

Parameters:

$$\theta_0, \theta_1$$

$$\theta_1$$

Cost Function:

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

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

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

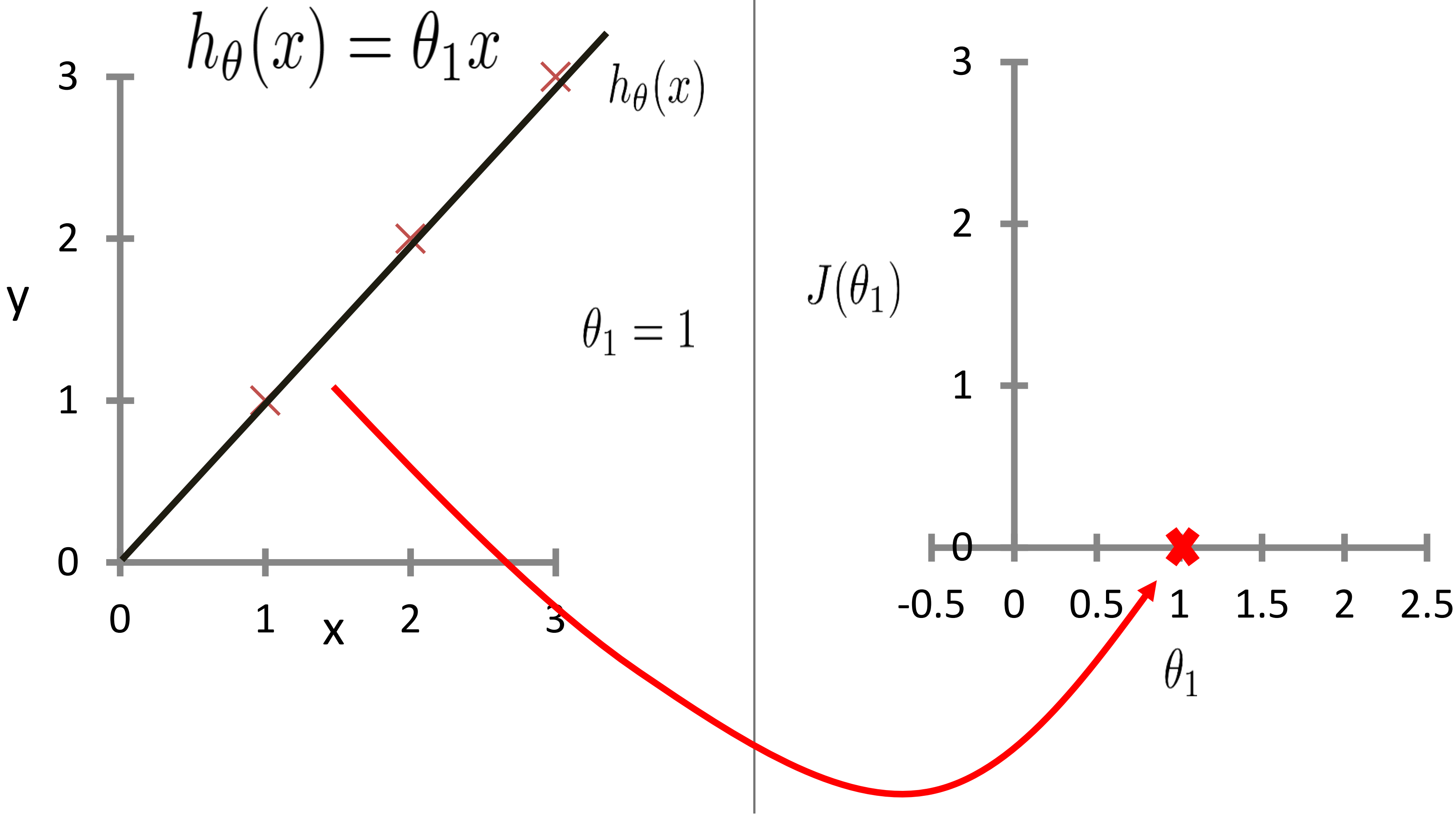
$$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$$

$h_\theta(x)$

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

$J(\theta_1)$

(function of the parameter θ_1)

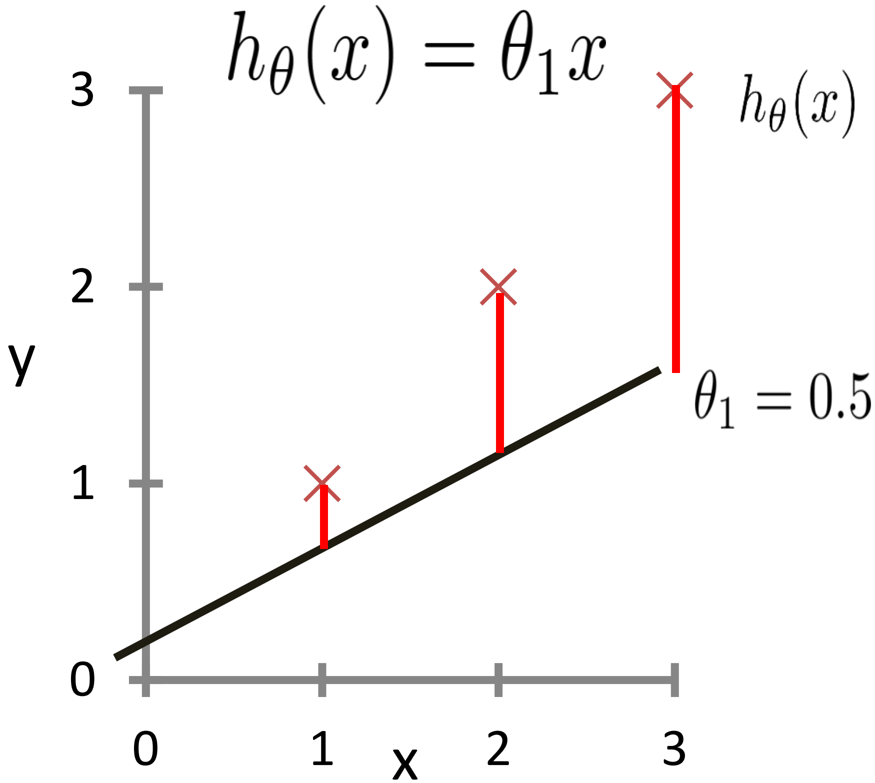


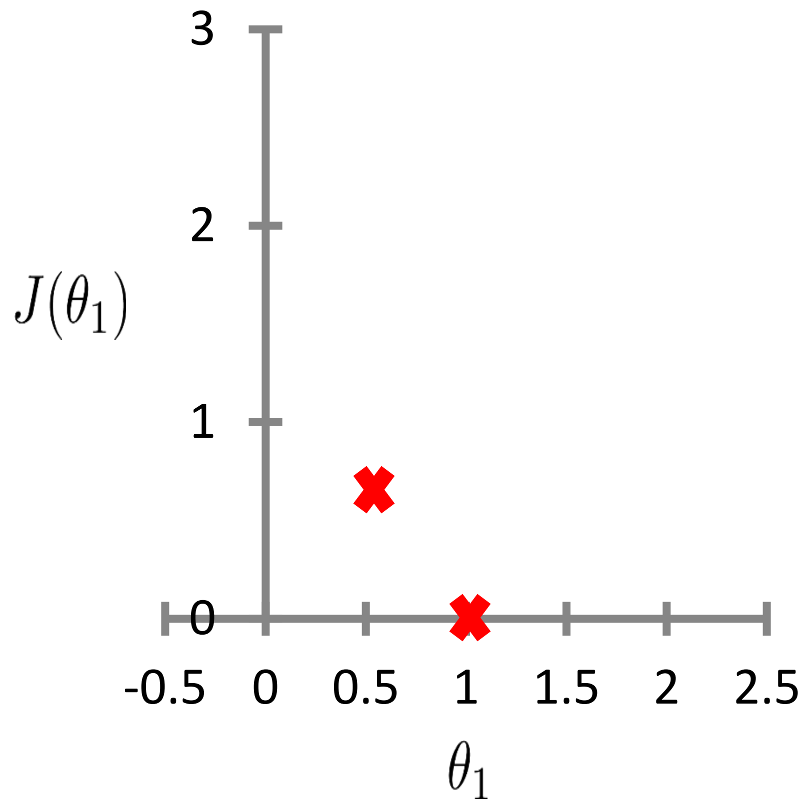
$h_\theta(x)$

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

$J(\theta_1)$

(function of the parameter θ_1)



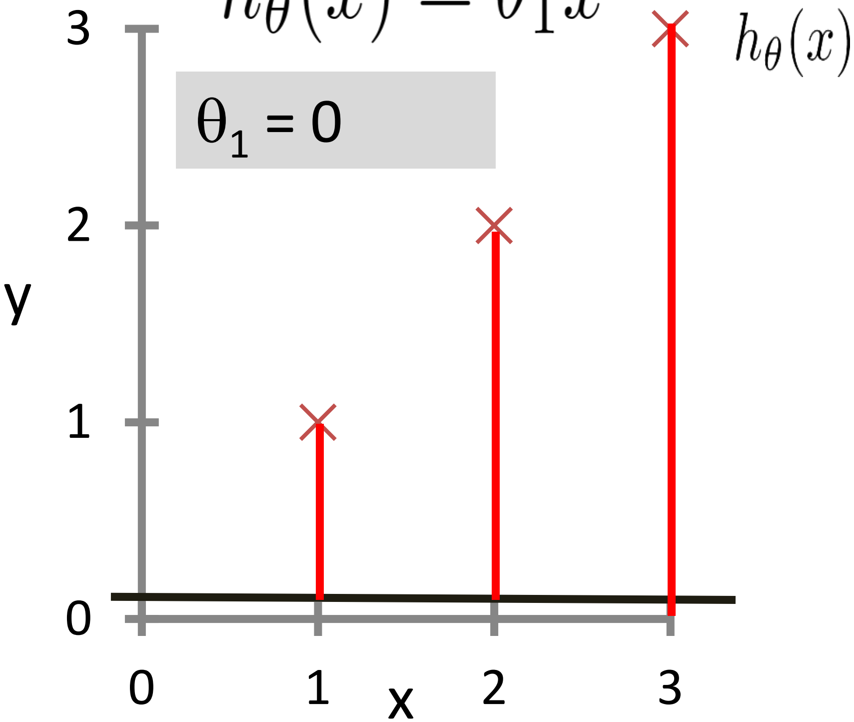


Compute $J(\theta_1)$ = (1/2*3)*{$(0.5-1)^2$+ $(1-2)^2$ + $(1.5-3)^2$ }
= (1/6)*(0.25+1+2.25)
=(1/6)* 3.5 = 0.58

$h_\theta(x)$

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

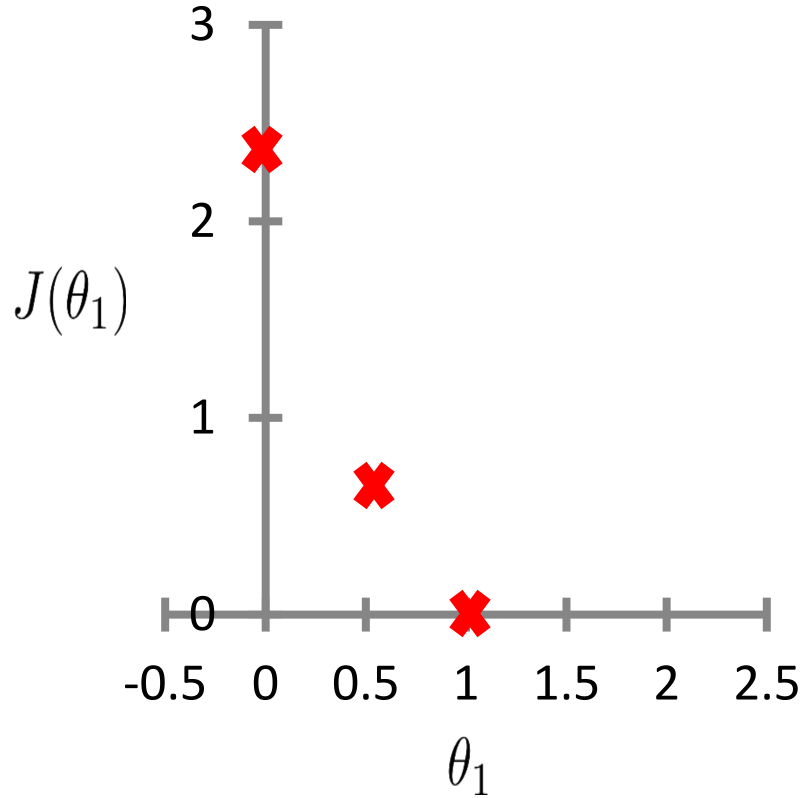
$$h_\theta(x) = \theta_1 x$$



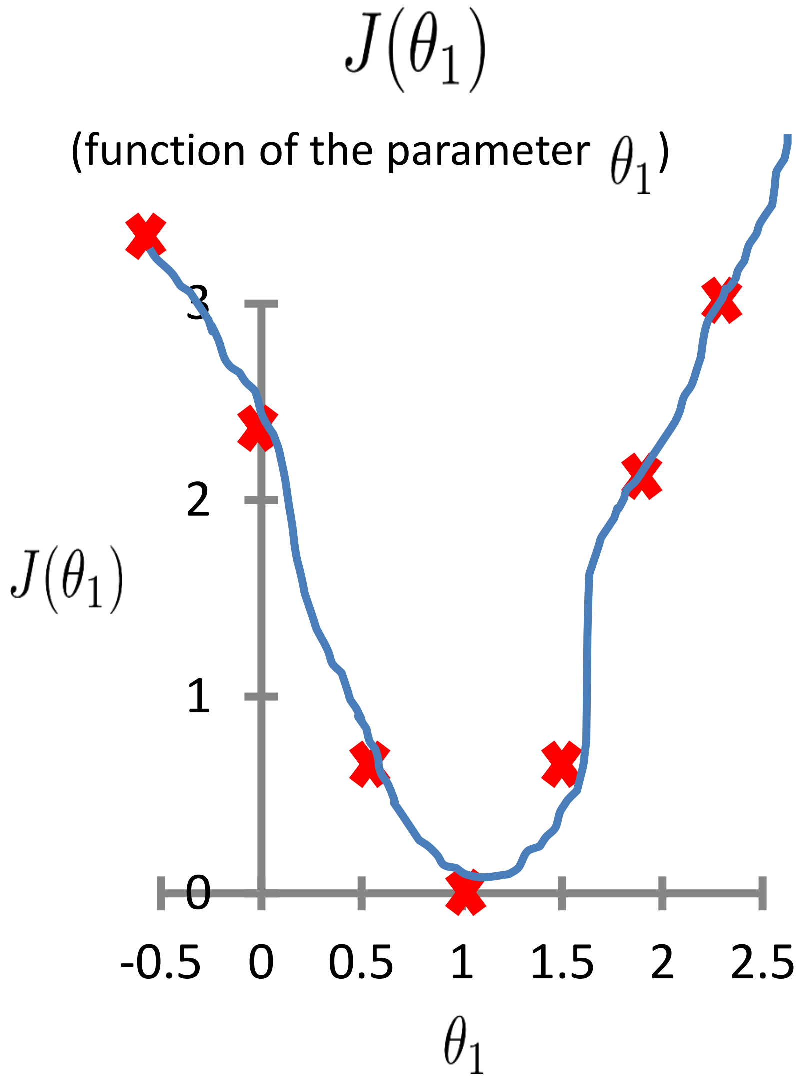
Compute J(θ_1) = (1/2*3)*{(0-1)^2+ (0-2)^2 + (0-3)^2 }
= (1/6)*(1+4+9)
=(1/6)* 14 = 2.3

$J(\theta_1)$

(function of the parameter θ_1)



The error curve $J(\theta_1)$ is plotted for varying values of the parameter θ_1



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

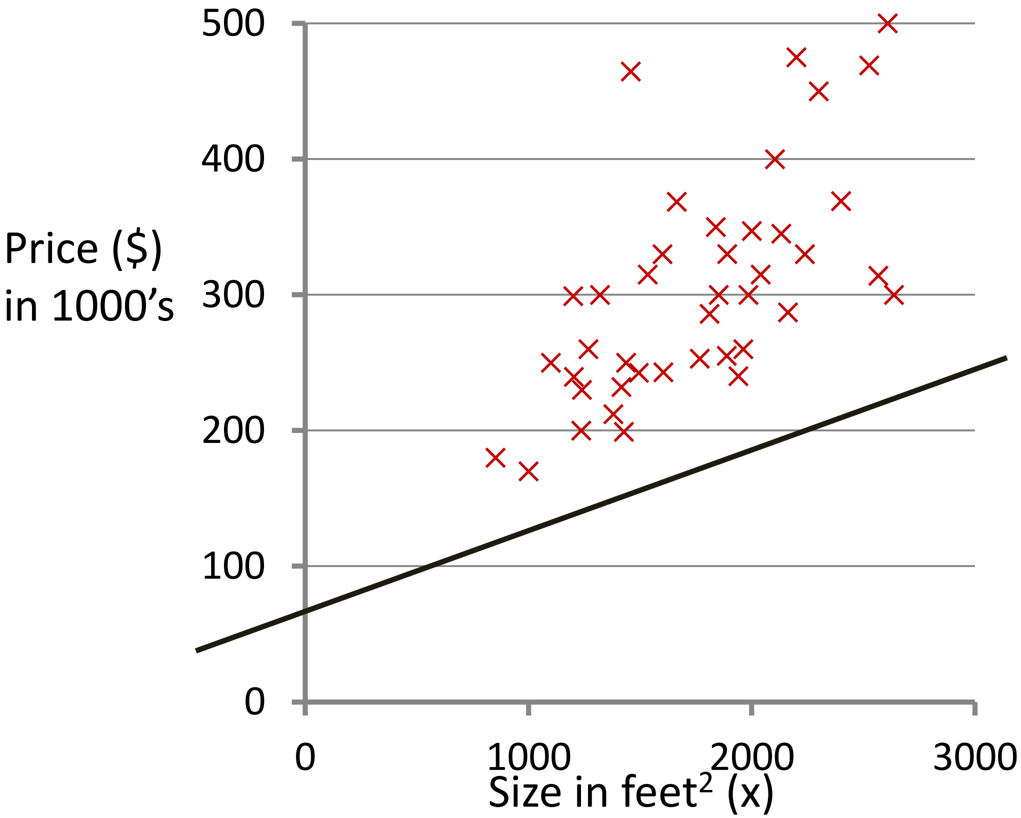
Parameters: θ_0, θ_1

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

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

$$h_\theta(x)$$

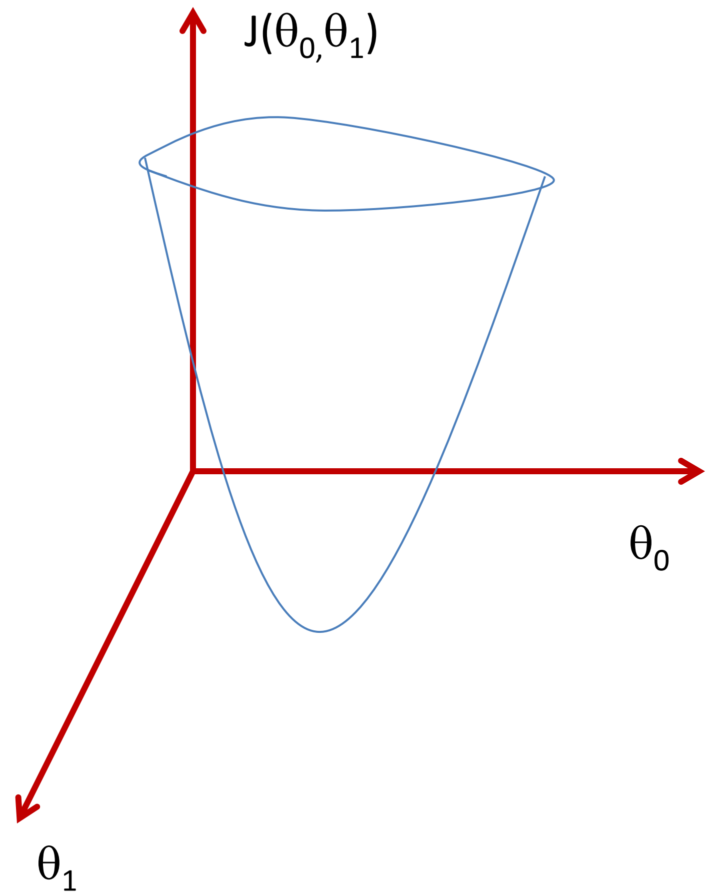
(for fixed θ_0, θ_1, this is a function of x)



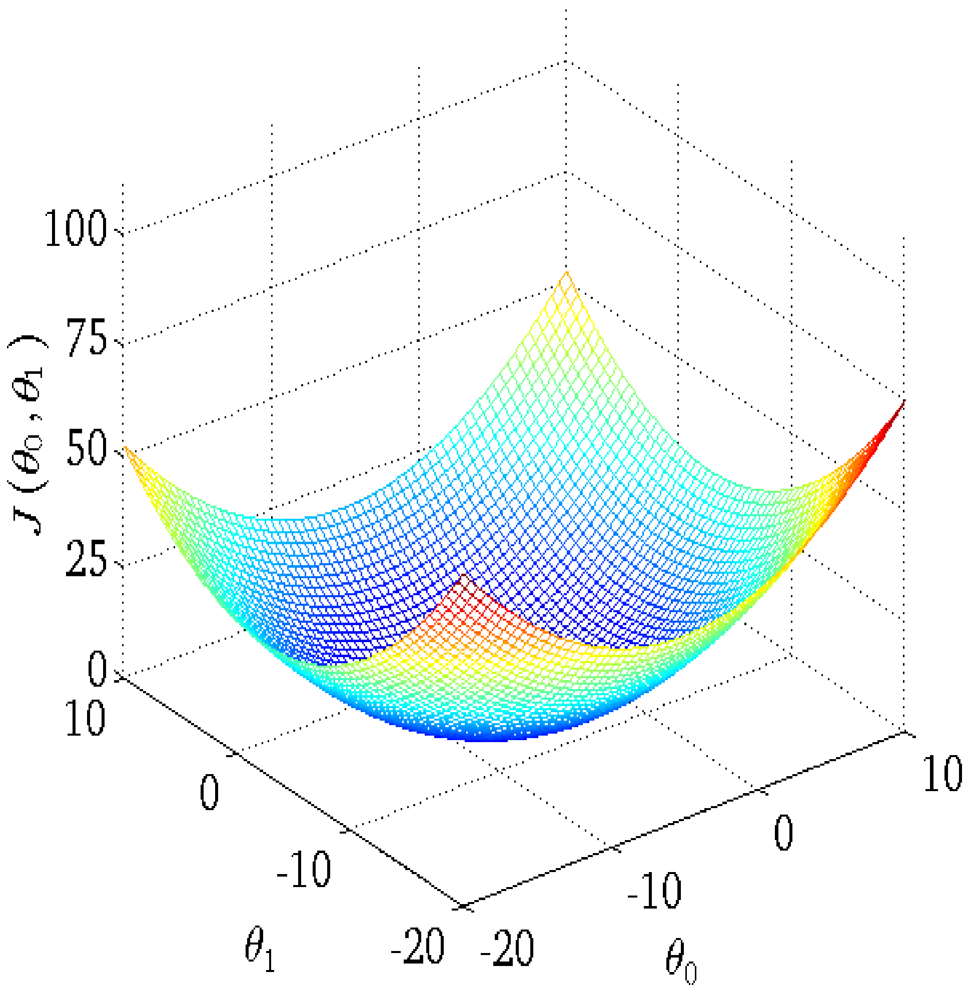
$$h_\theta(x) = 50 + 0.06x$$

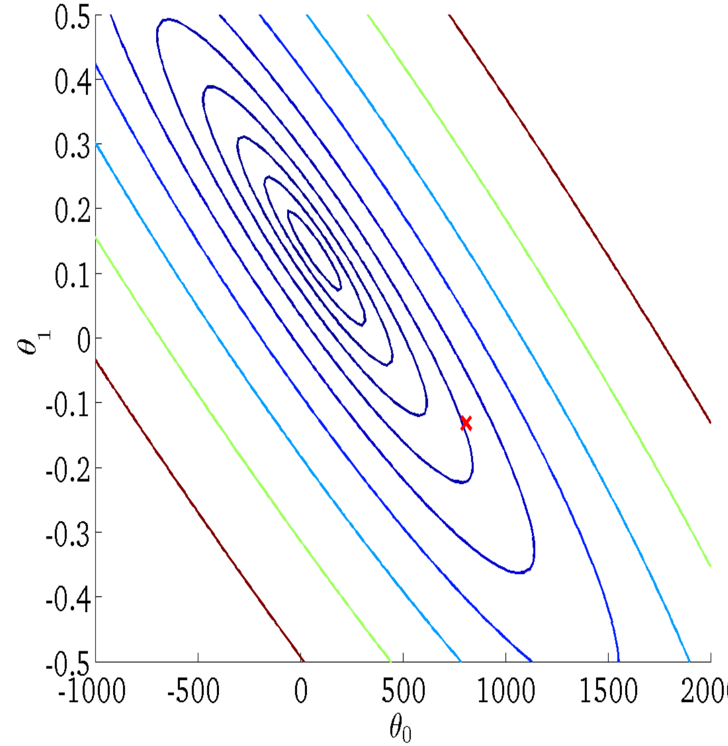
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



Surface and Corresponding contour plot



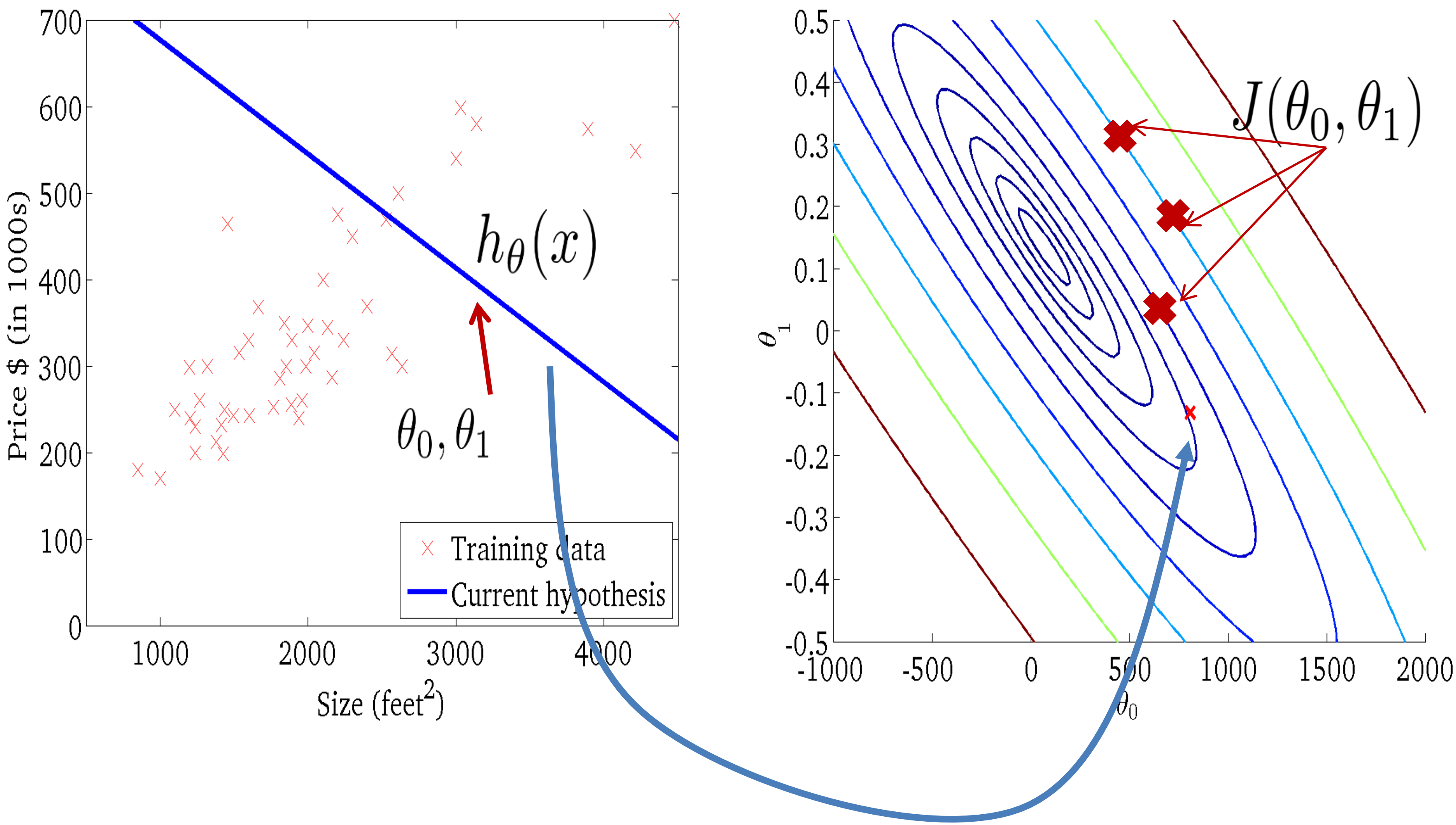


$h_\theta(x)$

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

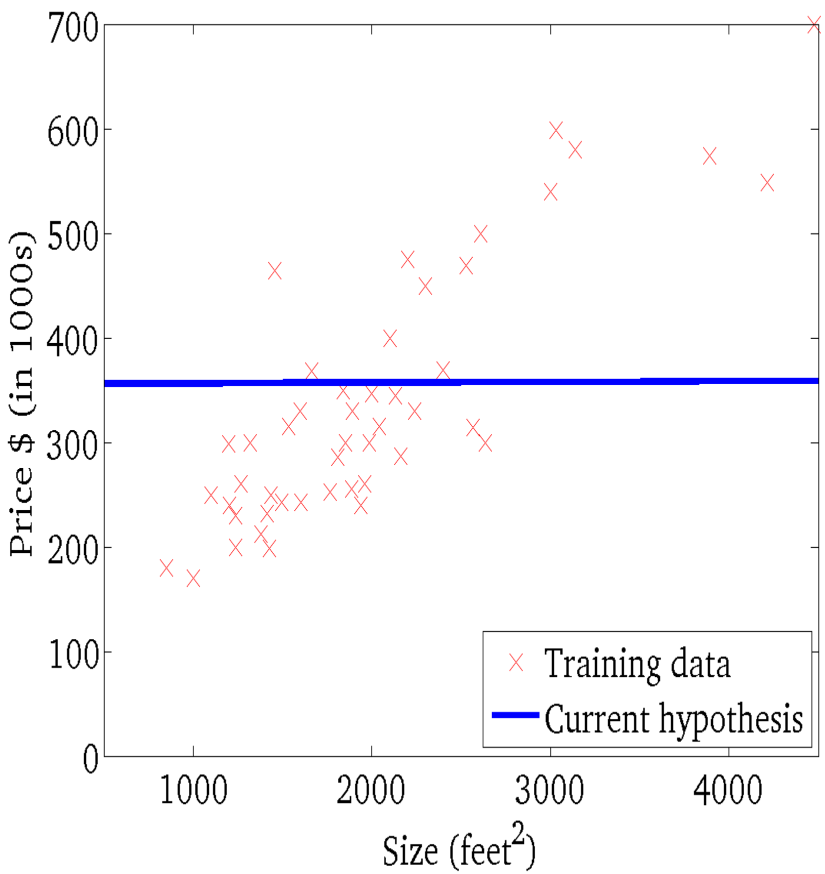
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



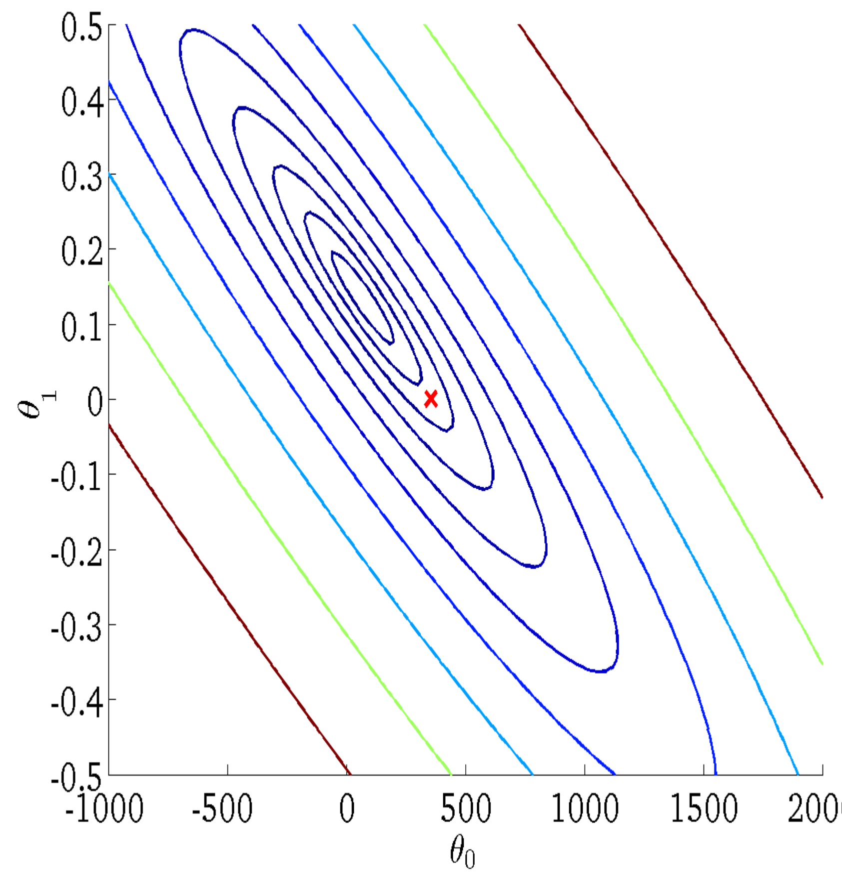
$$h_\theta(x)$$

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



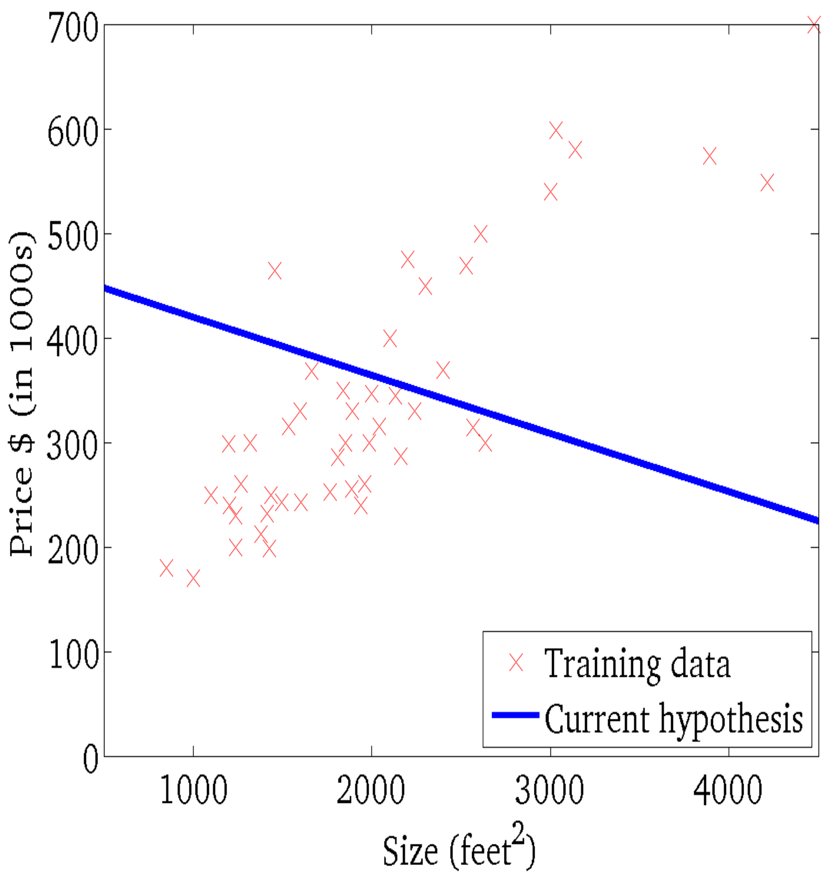
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



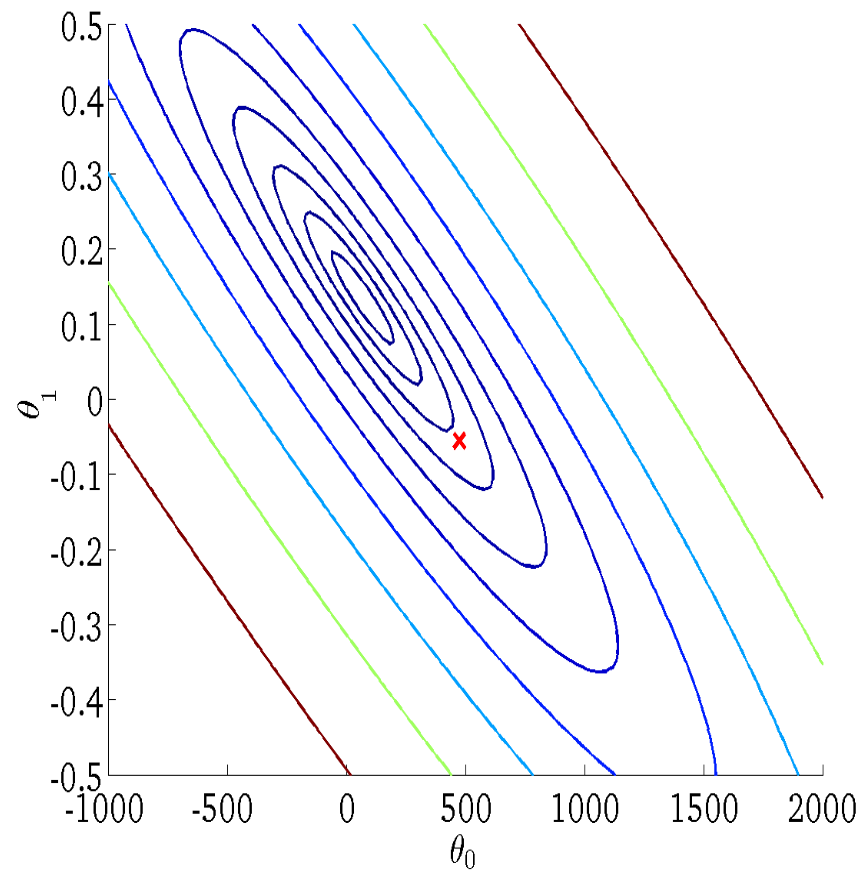
$$h_\theta(x)$$

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



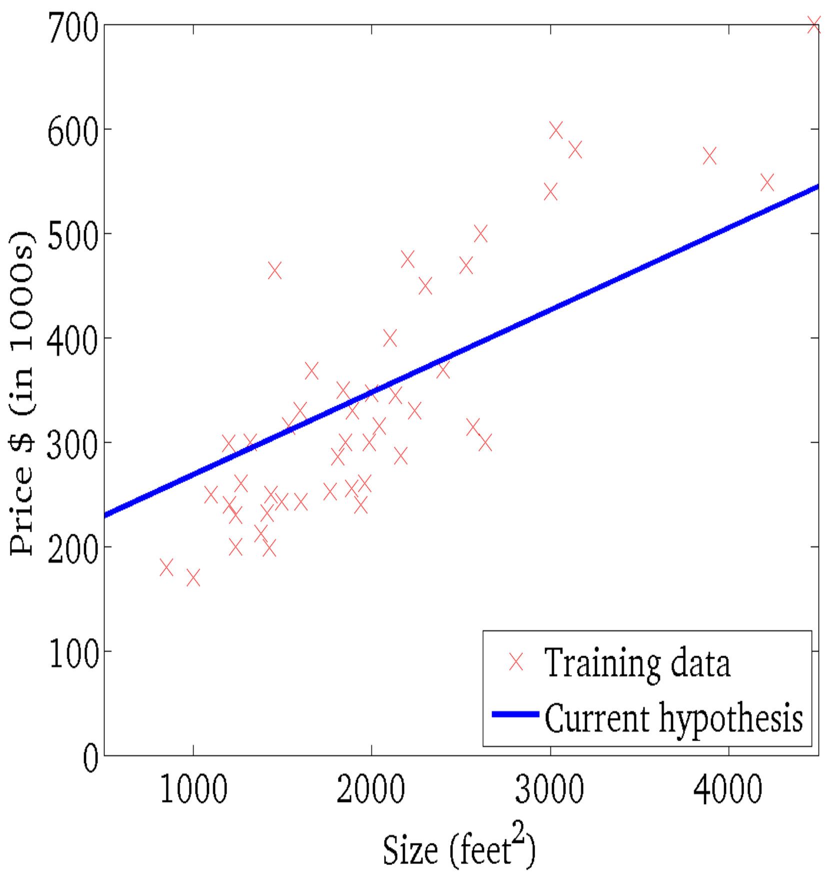
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



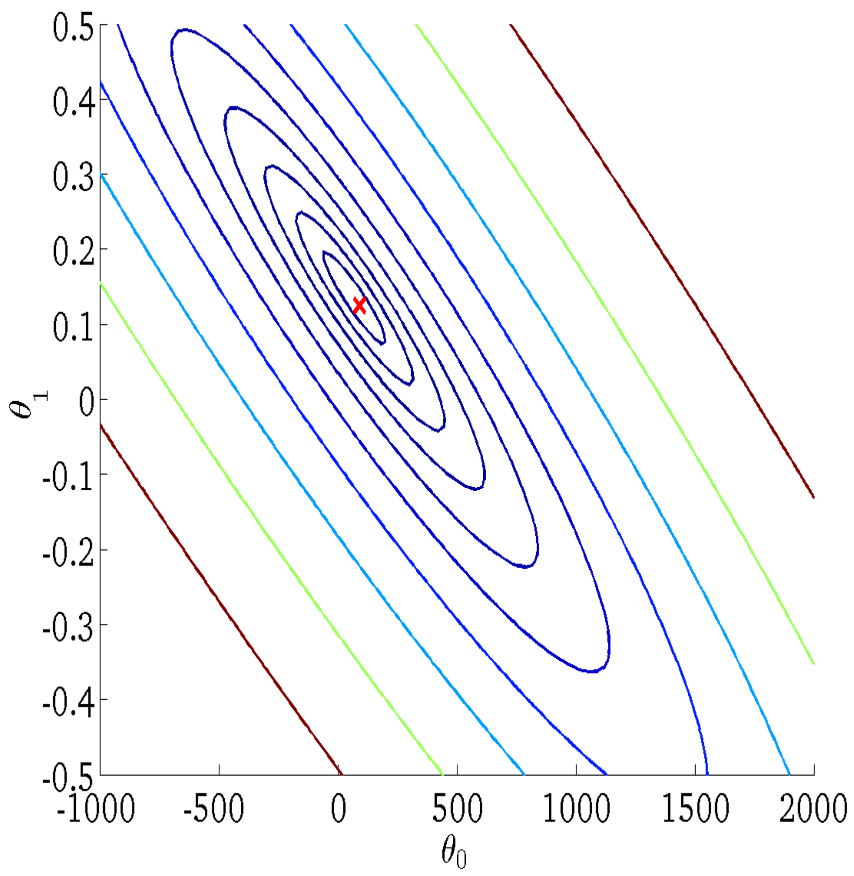
$h_\theta(x)$

(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)

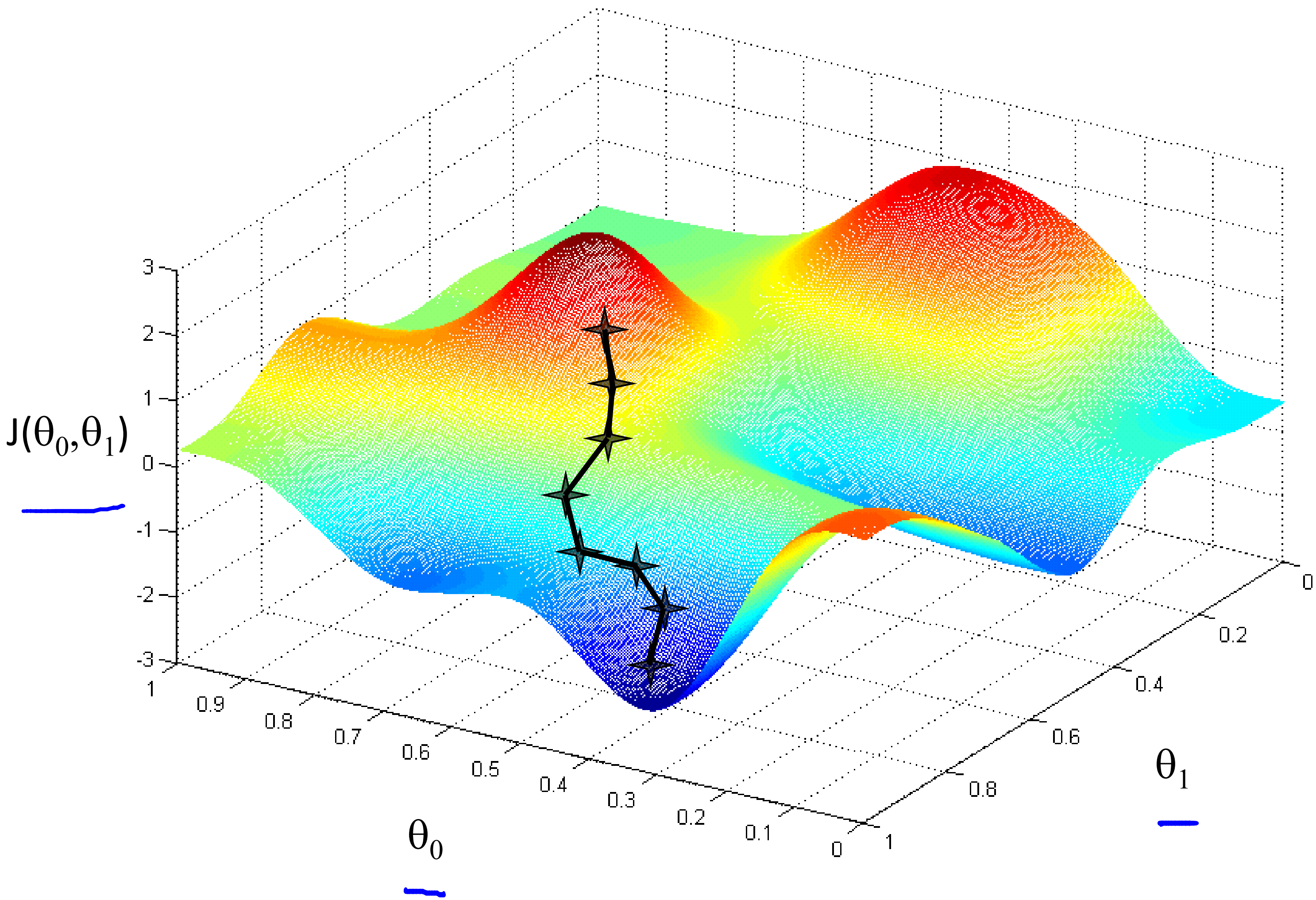


Have some function $J(\theta_0, \theta_1)$

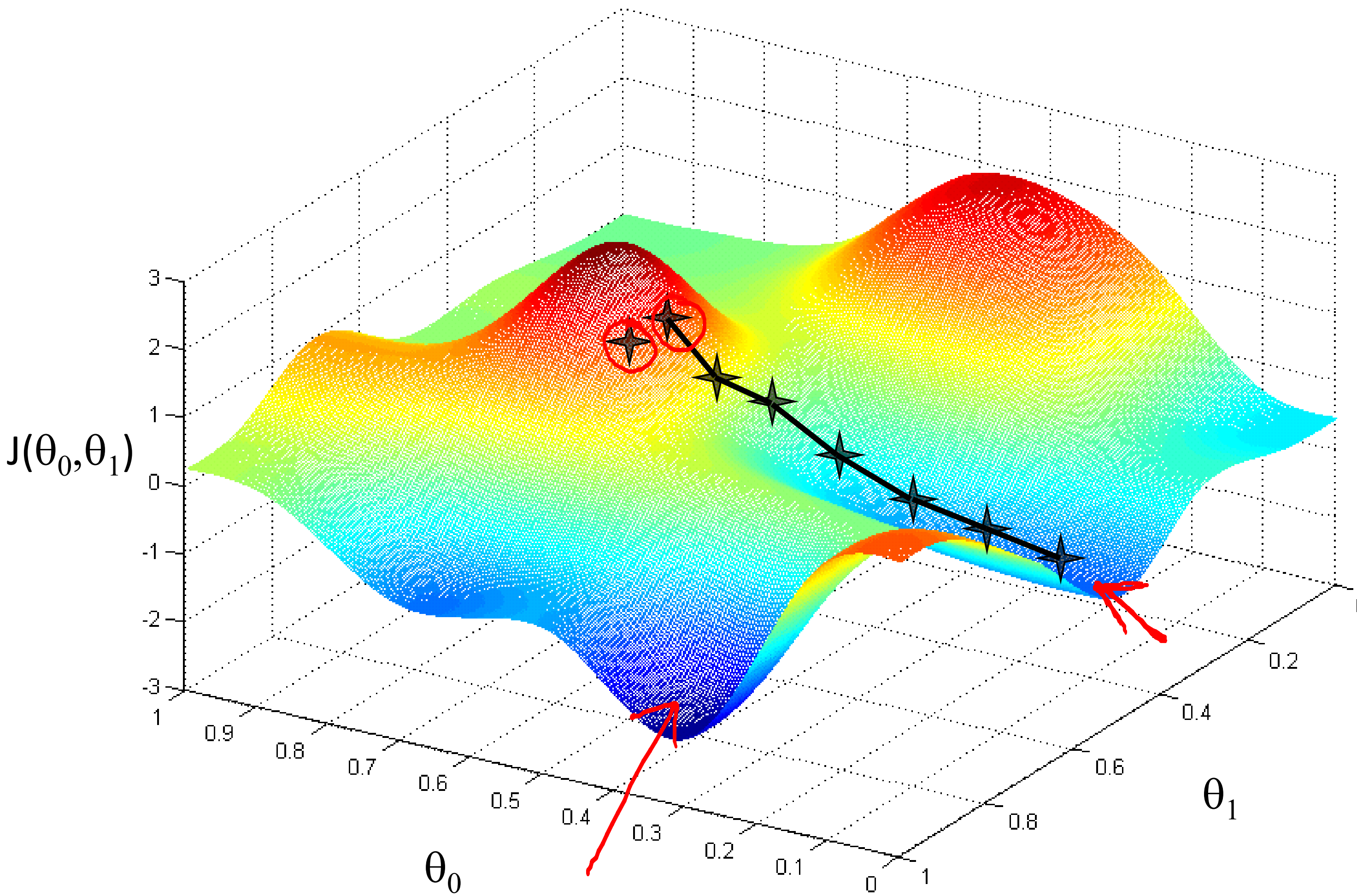
Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum



3
2
1
0
-1
-2
-3
$J(\theta_0,\theta_1)$
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

Gradient descent algorithm

Learning Rate: α

$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$
$$\}$$

Correct: Simultaneous update

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

Incorrect:

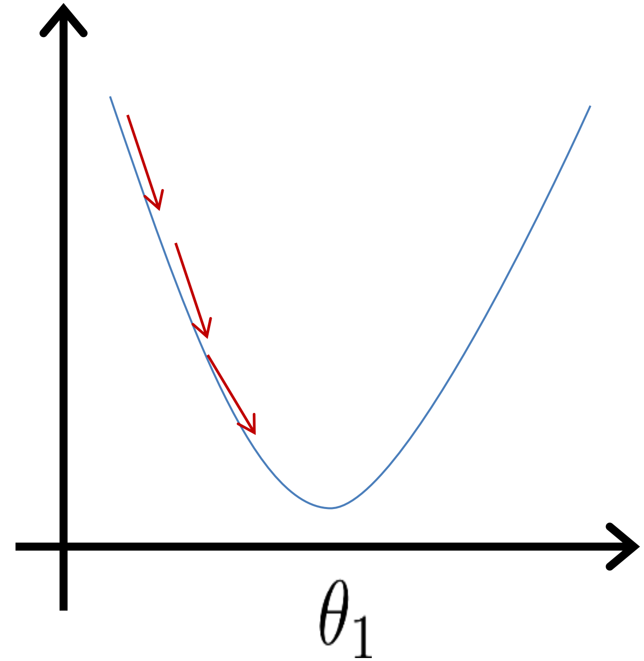
$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$

Gradient descent algorithm

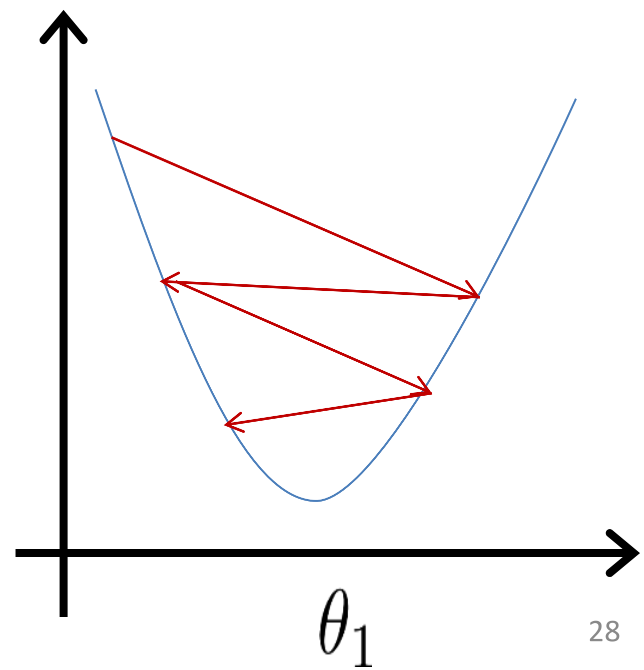
$$
\begin{aligned}
&\text{repeat until convergence } \{\\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad \text{(simultaneously update } j = 0 \text{ and } j = 1)\\
&\}
\end{aligned}
$$

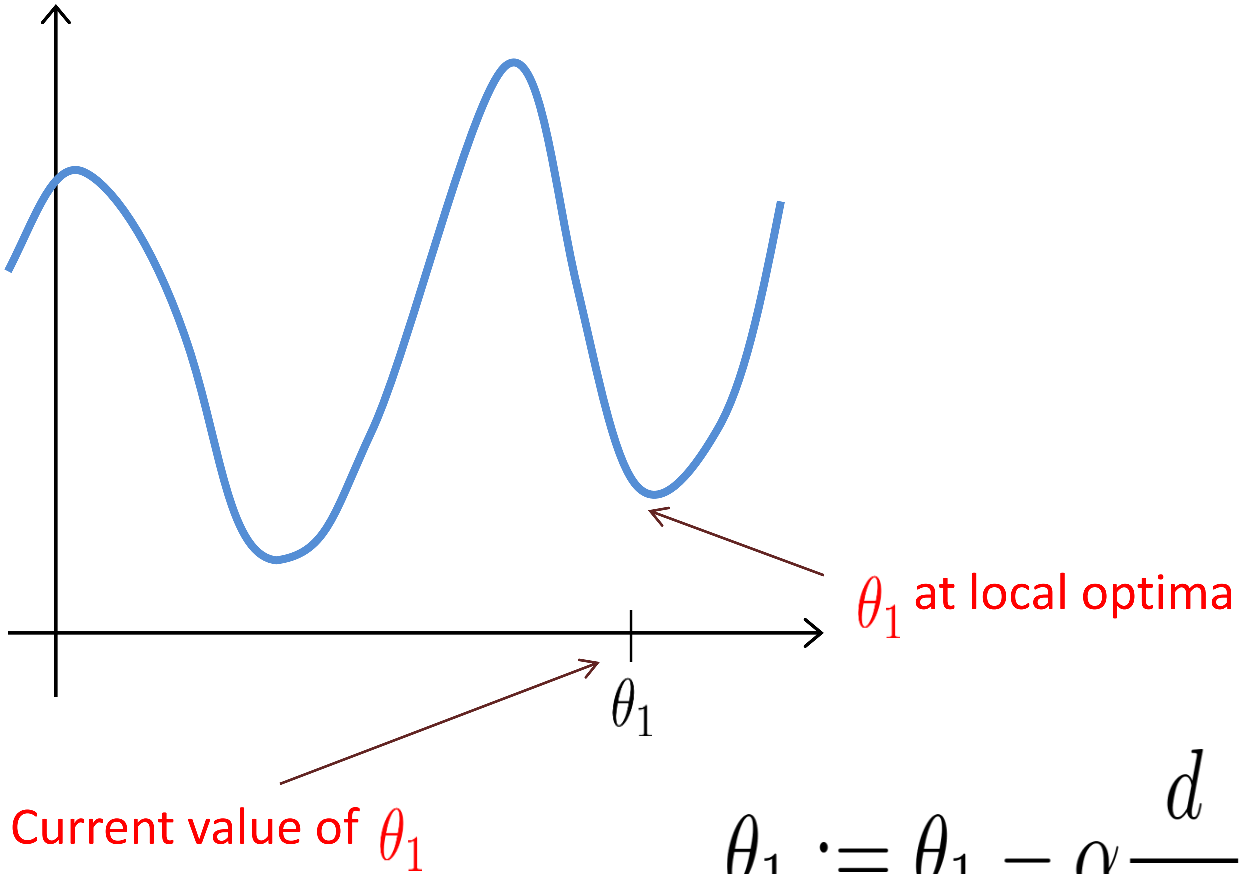
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

If α is too small, gradient descent can be slow.



If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



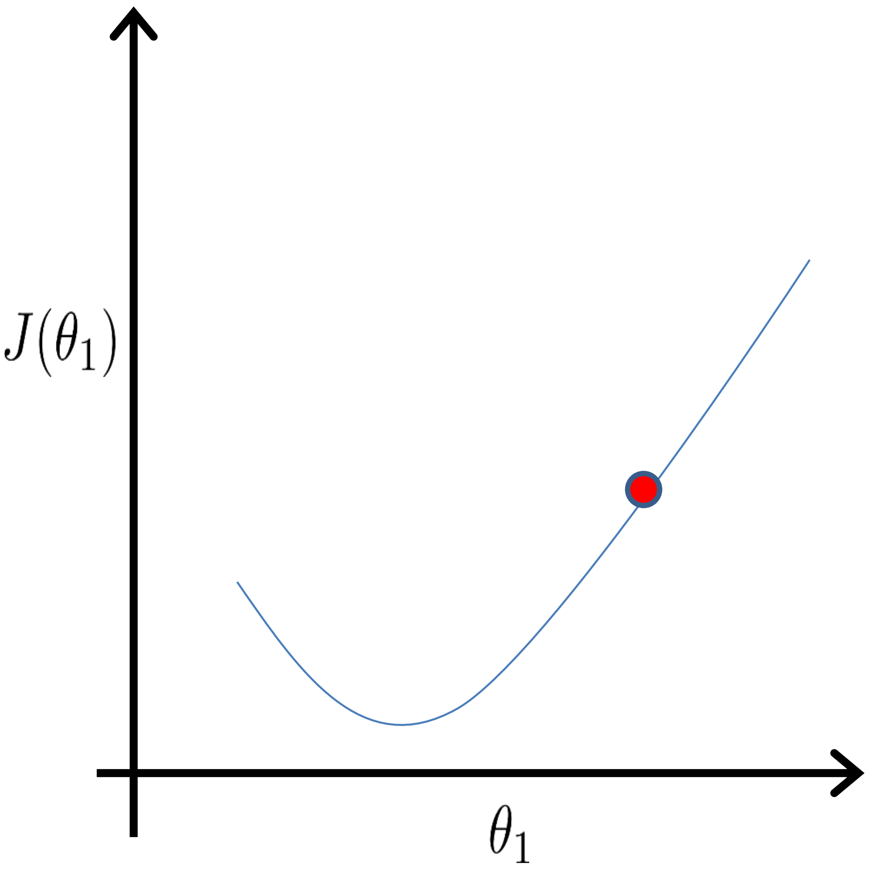


$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

Gradient descent can converge to a local minimum, even with the learning rate α fixed.

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.



Gradient descent algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
&\qquad \text{(for } j = 1 \text{ and } j = 0\text{)} \\
&\}
\end{aligned}
$$

Linear Regression Model

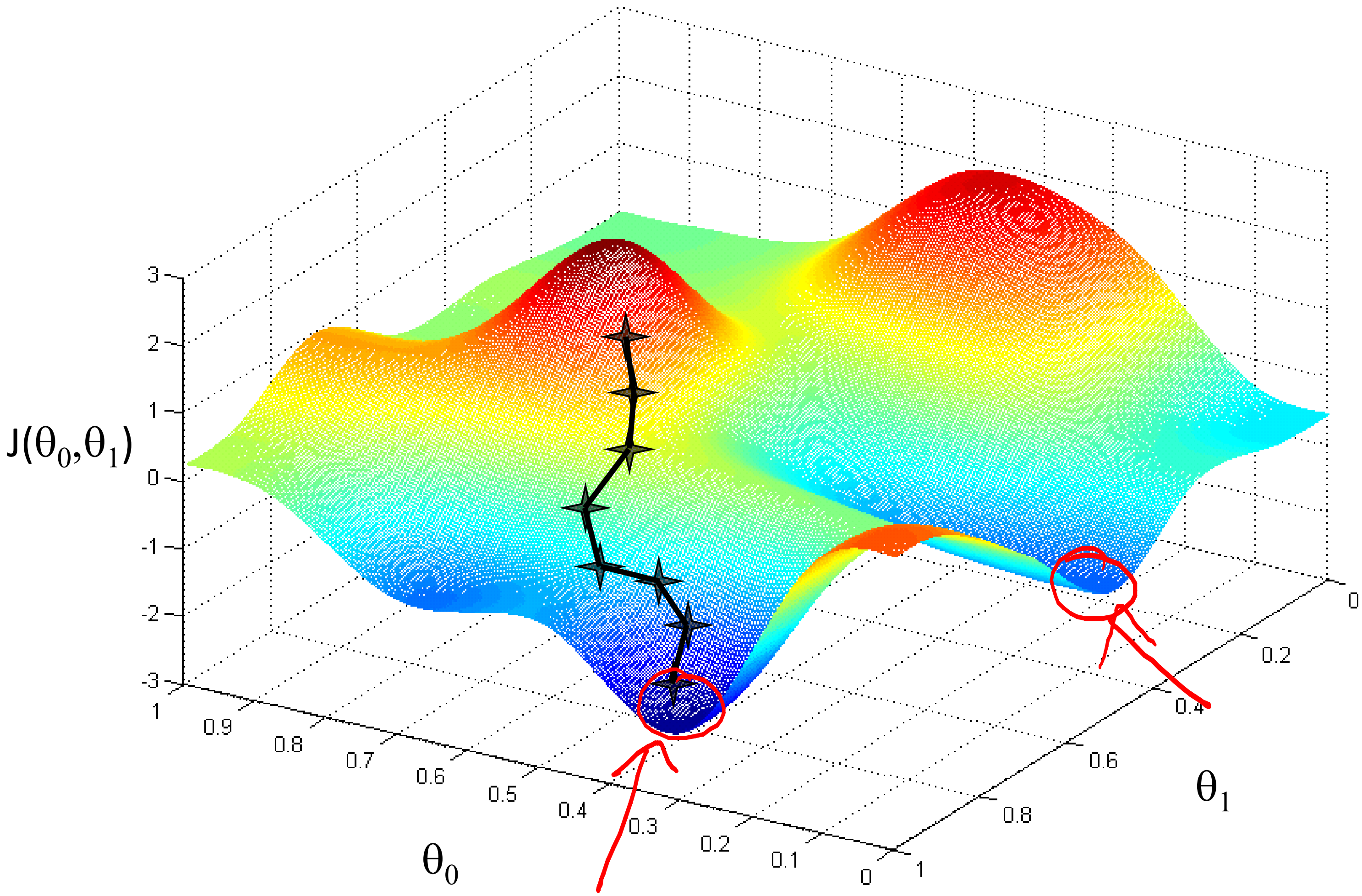
$$h_\theta(x) = \theta_0 + \theta_1 x$$

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

Gradient descent algorithm

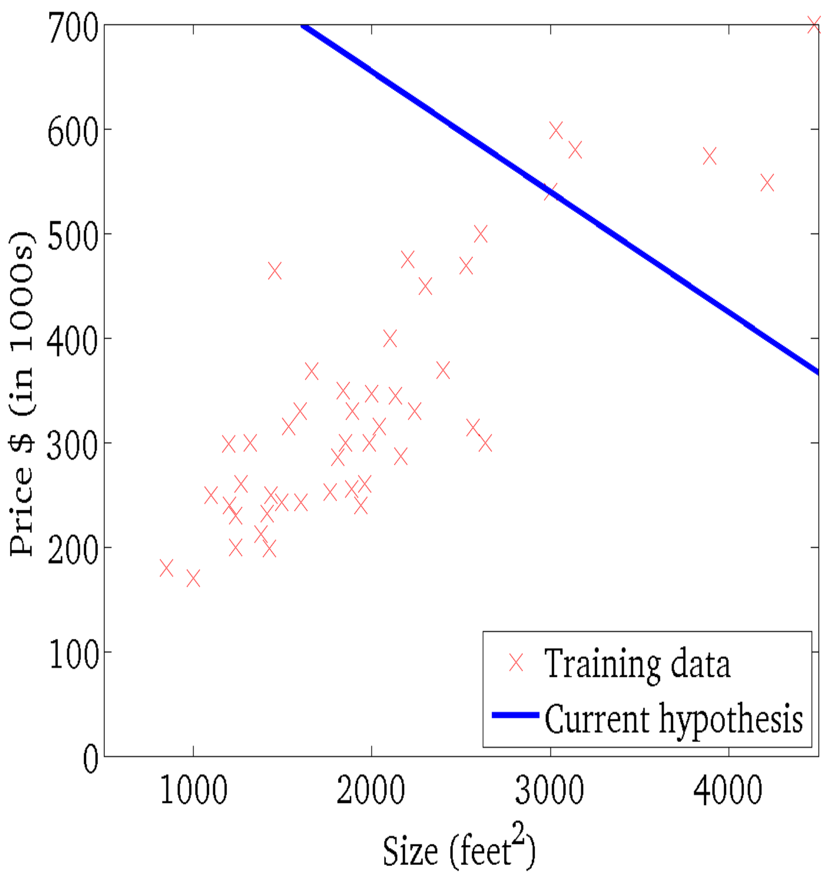
$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \left.\begin{aligned}
\theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\
\theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}
\end{aligned}\right\} \\
&\}
\end{aligned}
$$

update θ_0 and θ_1 simultaneously


3
2
1
0
-1
-2
-3
$J(\theta_0,\theta_1)$
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

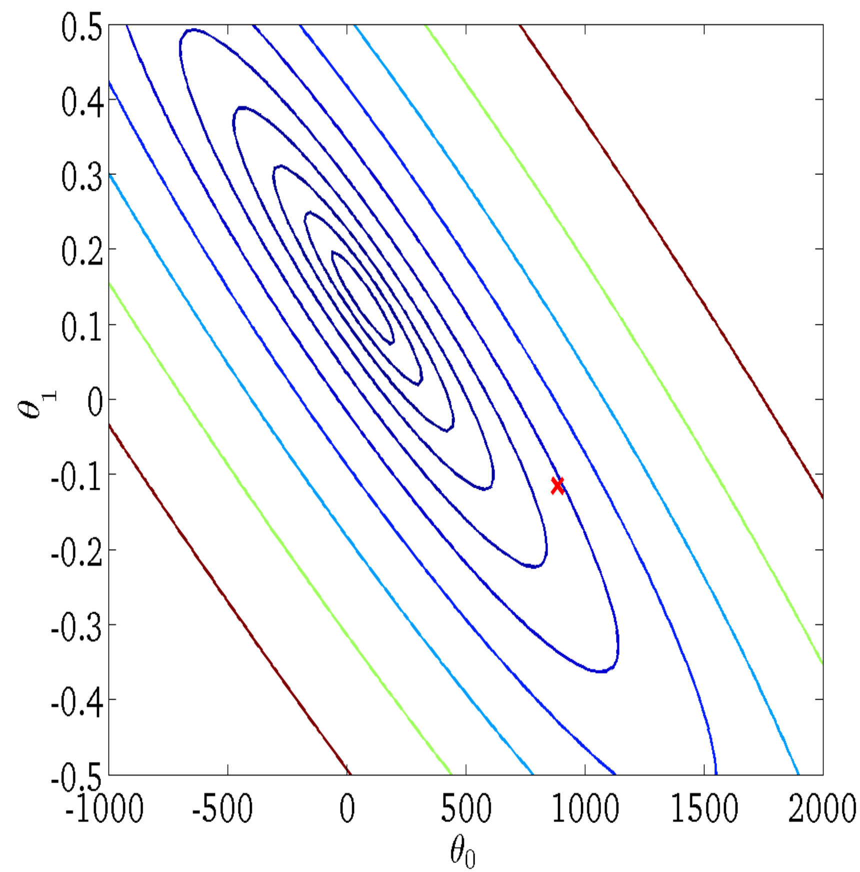
$h_\theta(x)$

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



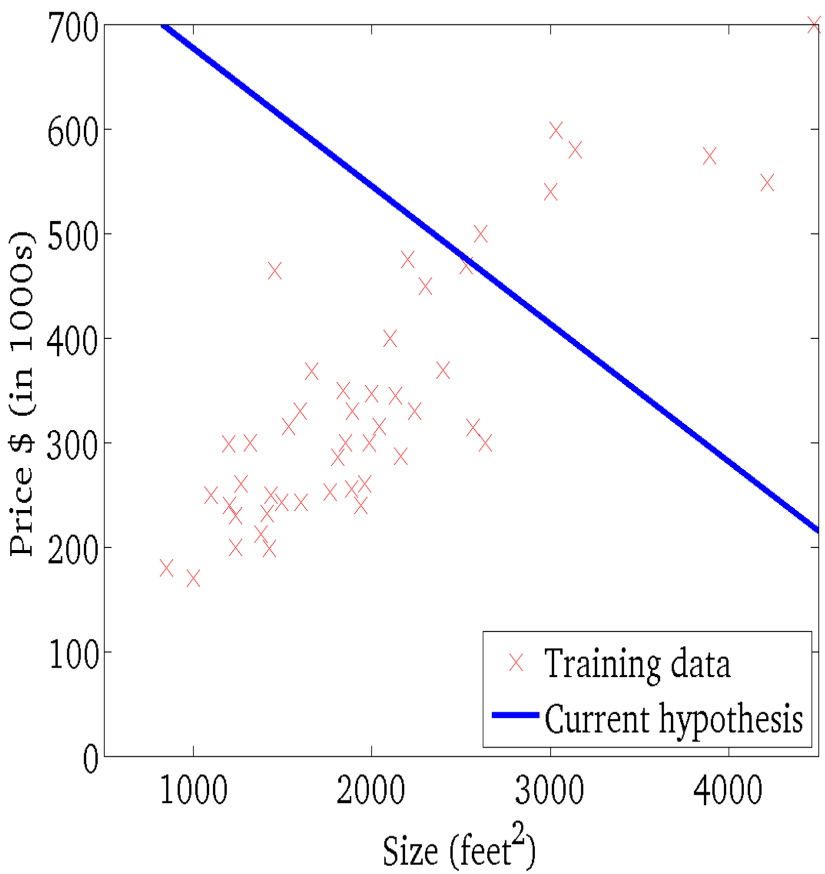
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



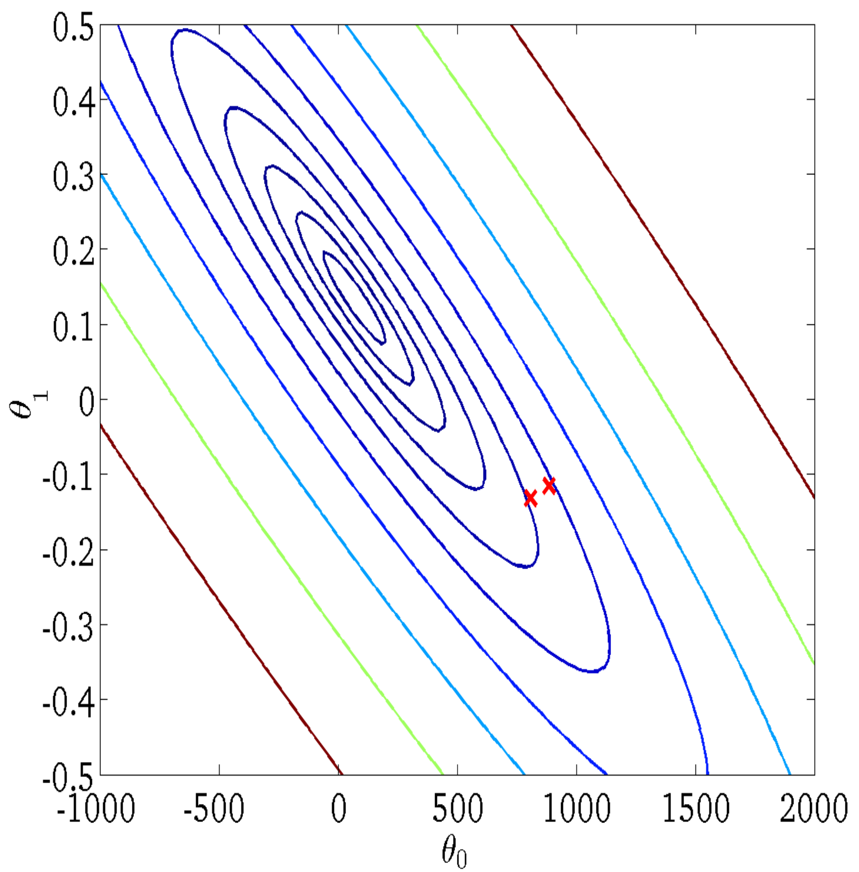
$$h_\theta(x)$$

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



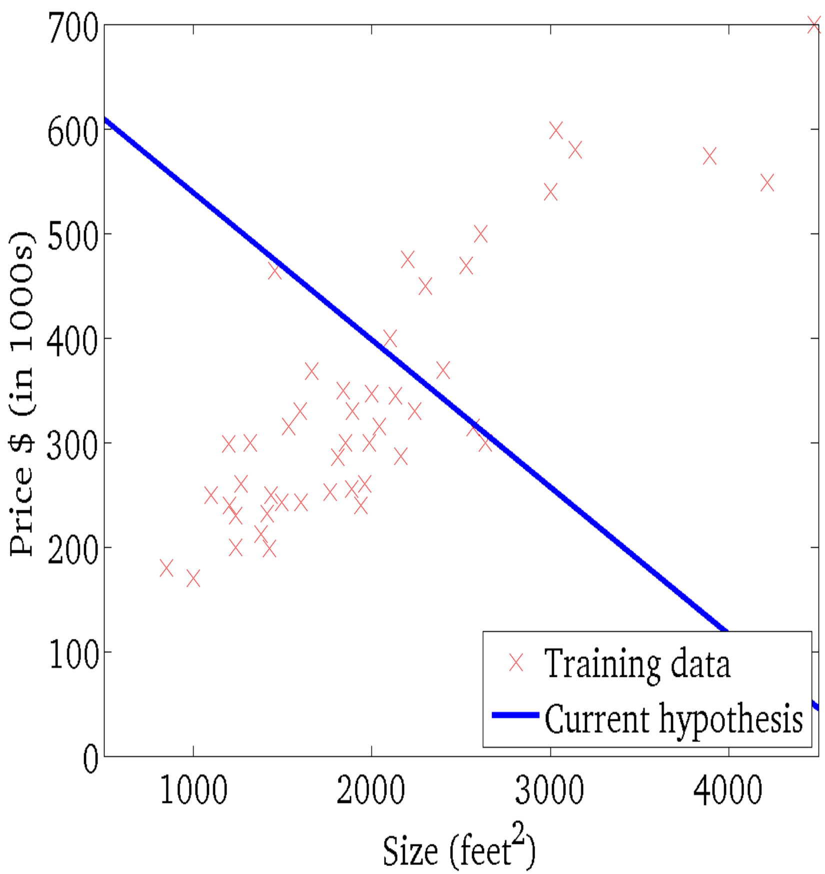
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



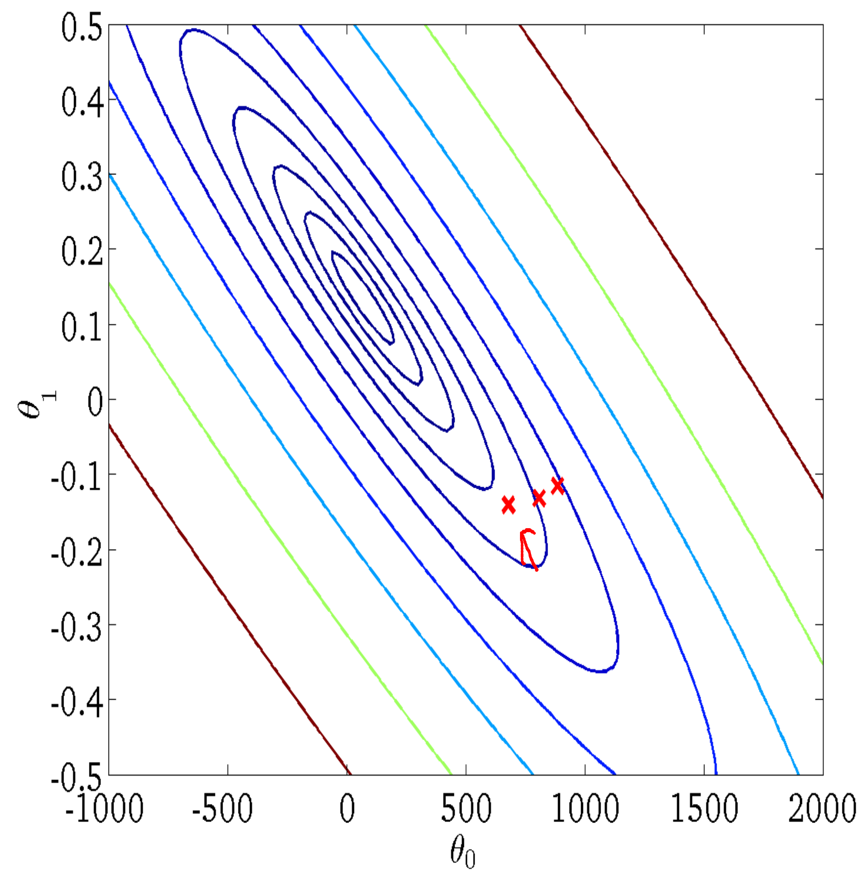
$$h_\theta(x)$$

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



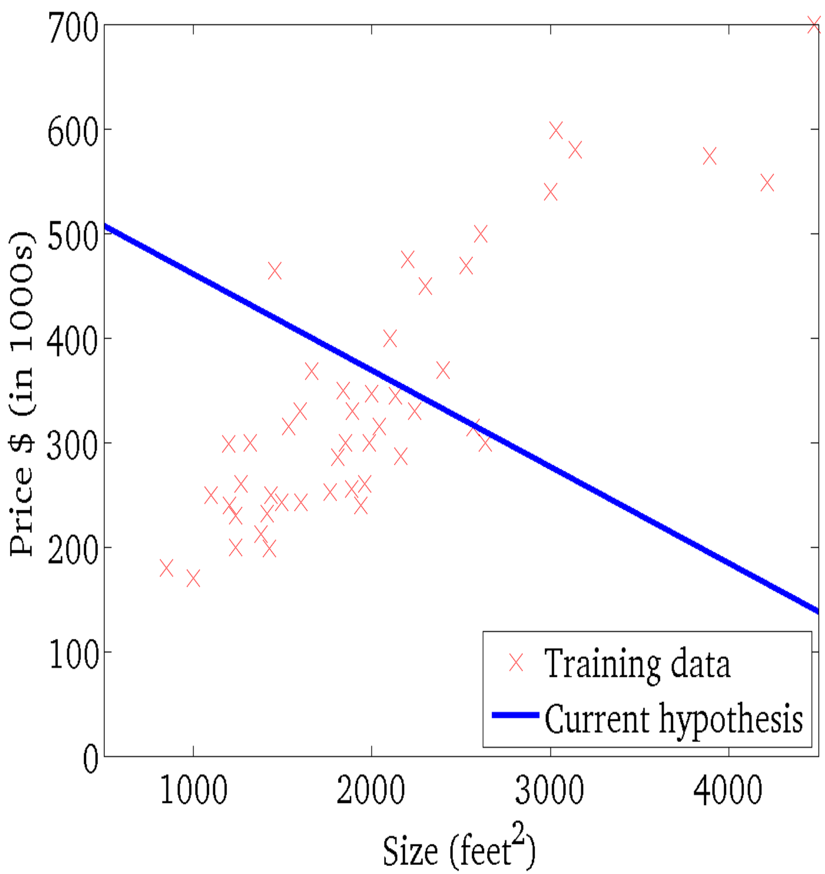
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



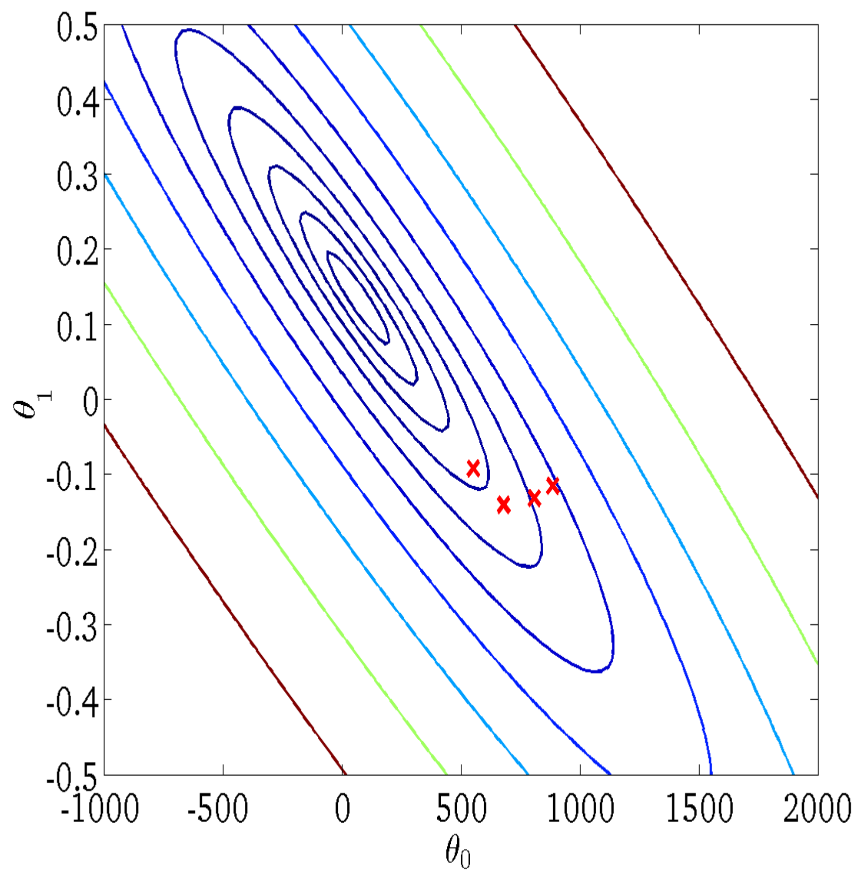
$$h_\theta(x)$$

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



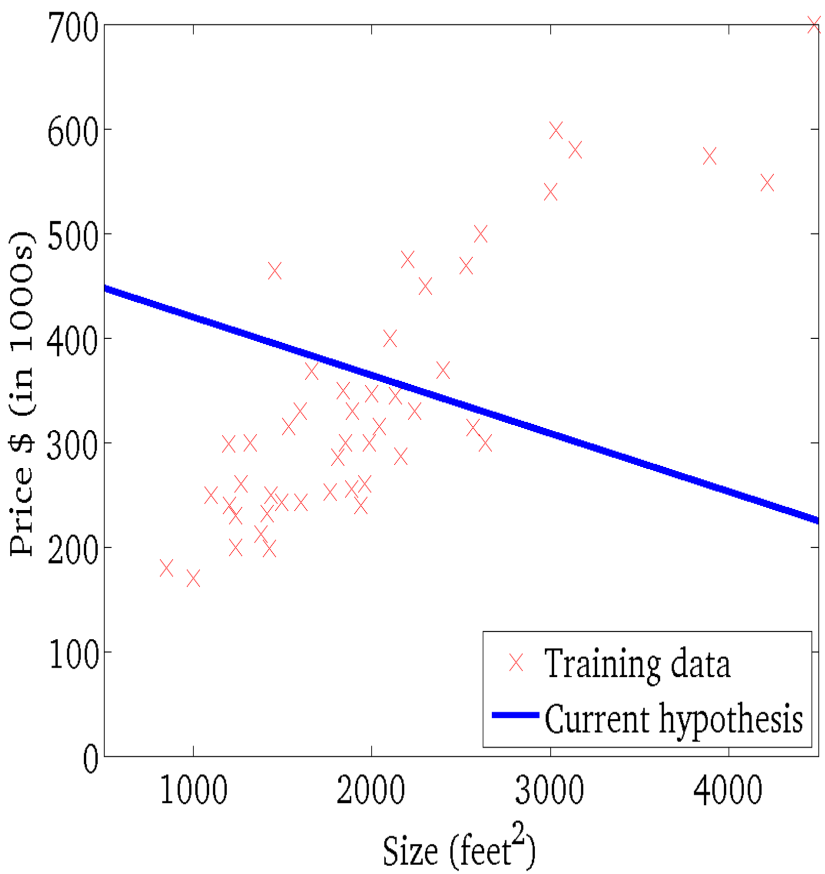
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



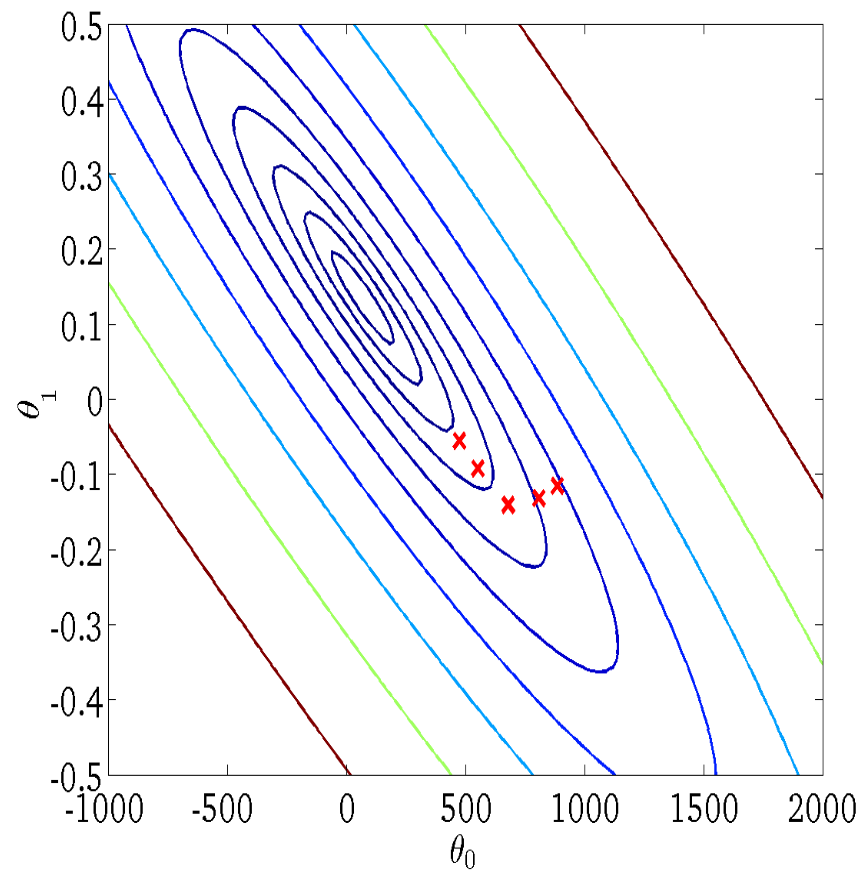
$h_\theta(x)$

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



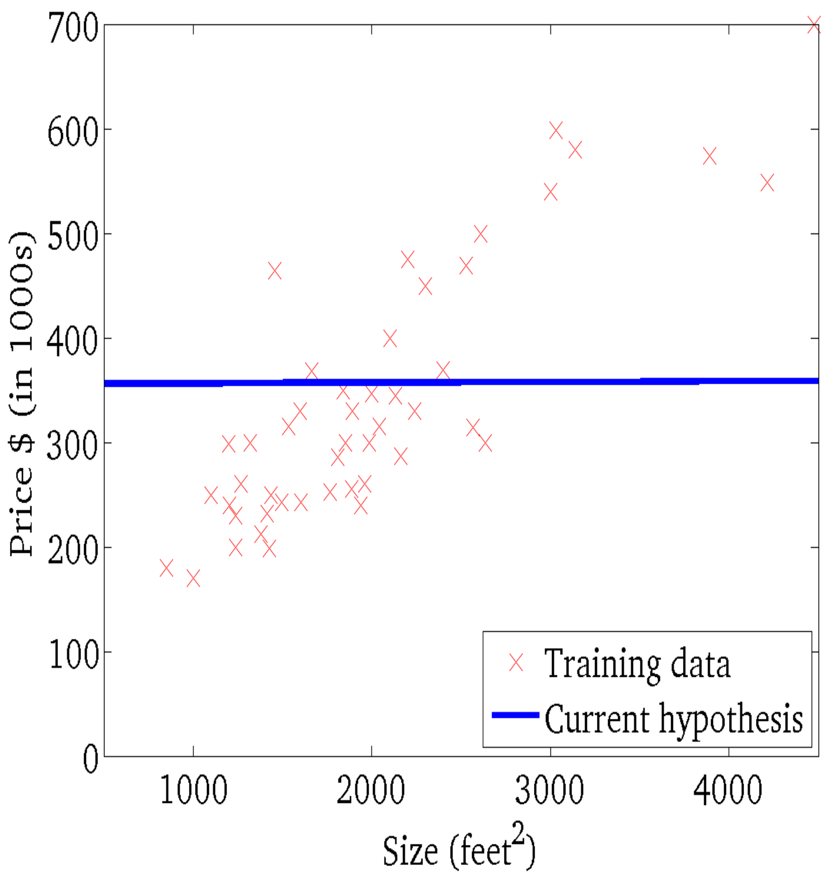
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



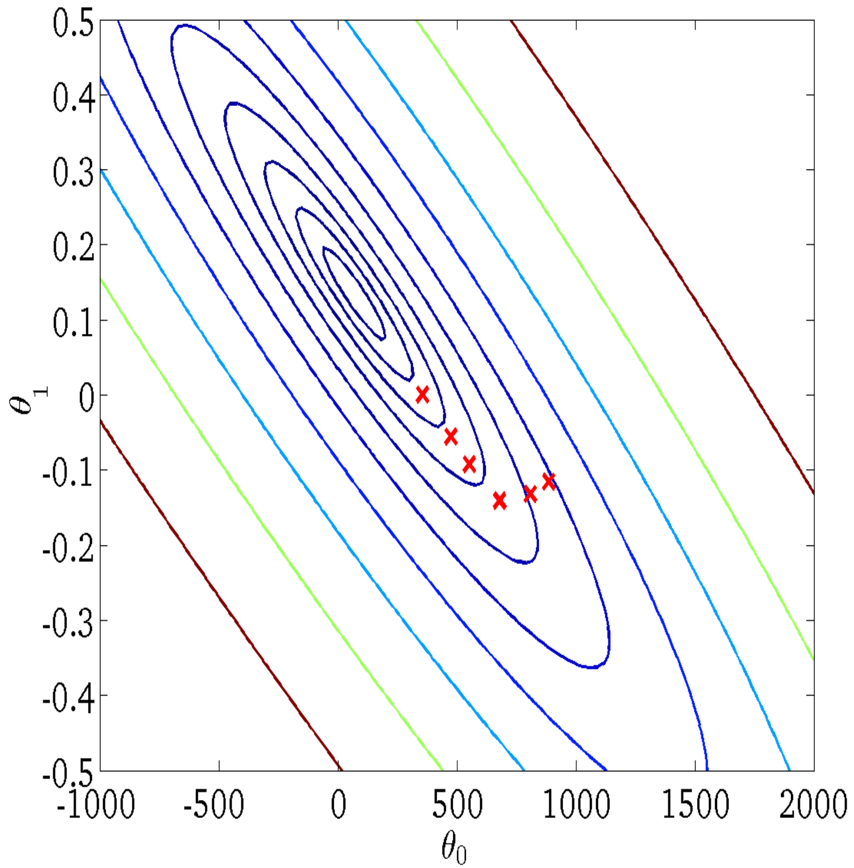
$$h_\theta(x)$$

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



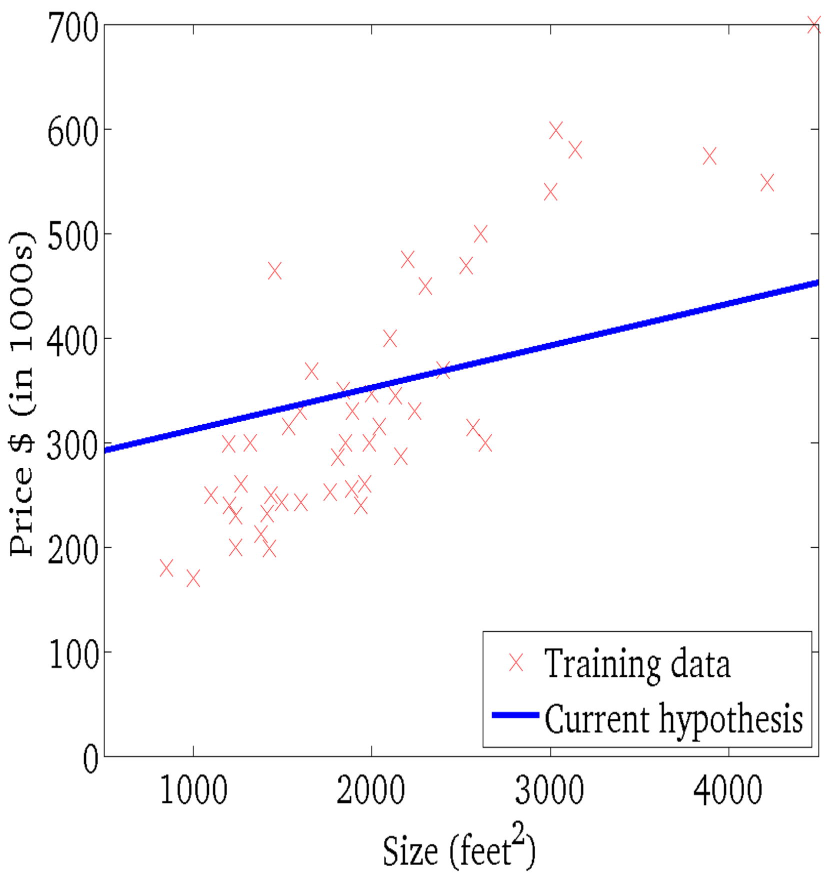
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



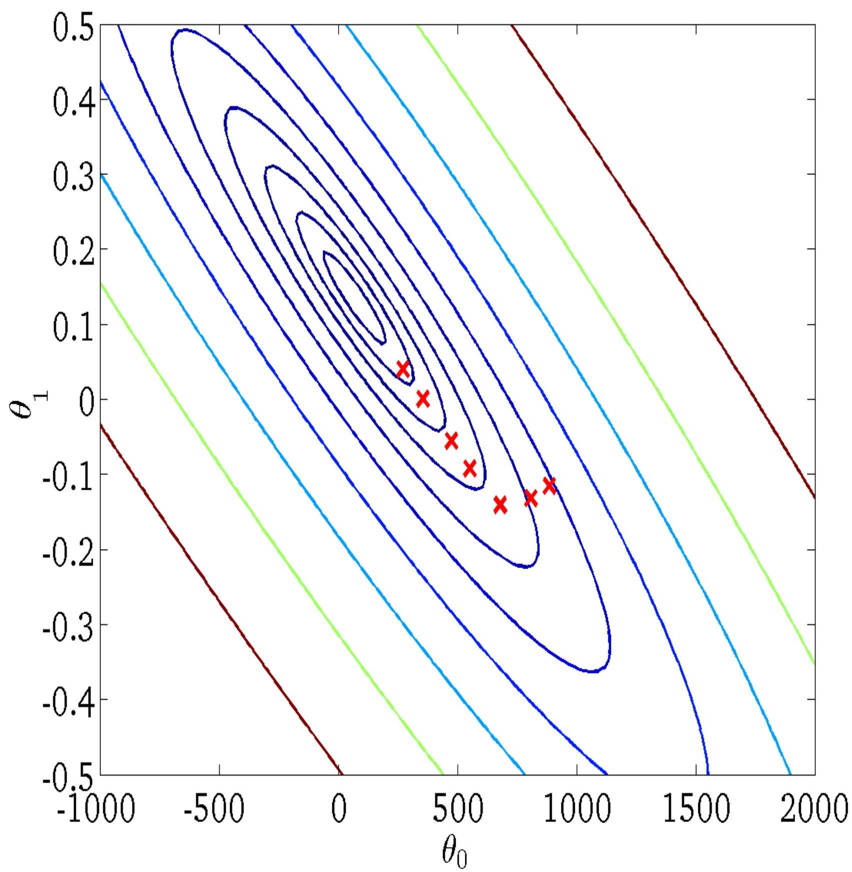
$$h_\theta(x)$$

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



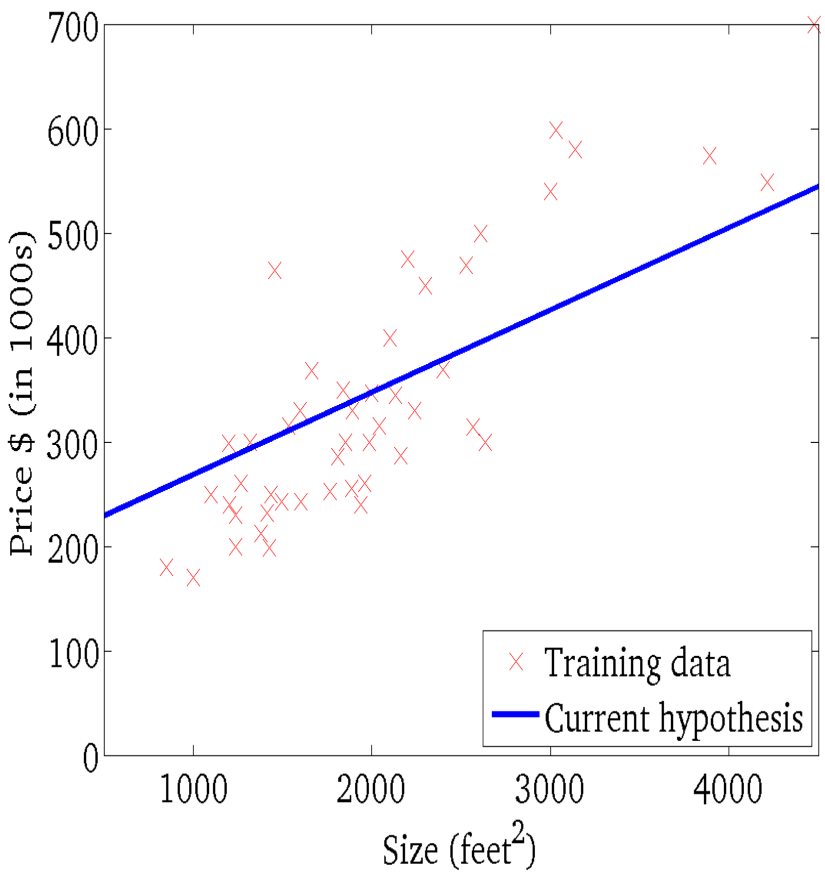
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



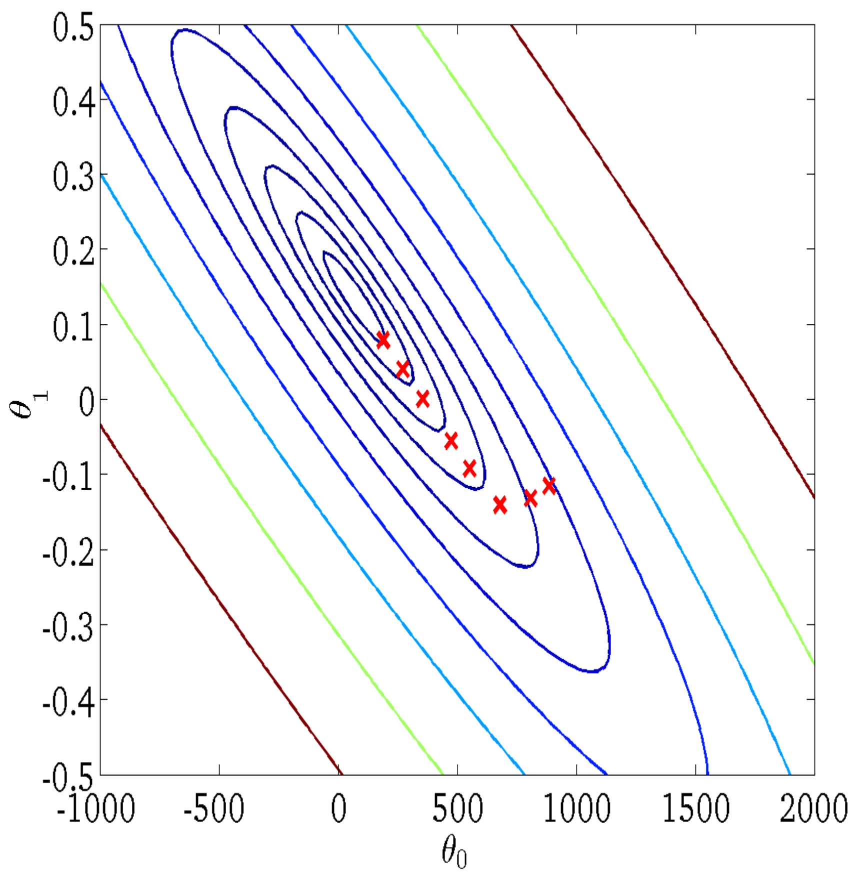
$h_\theta(x)$

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



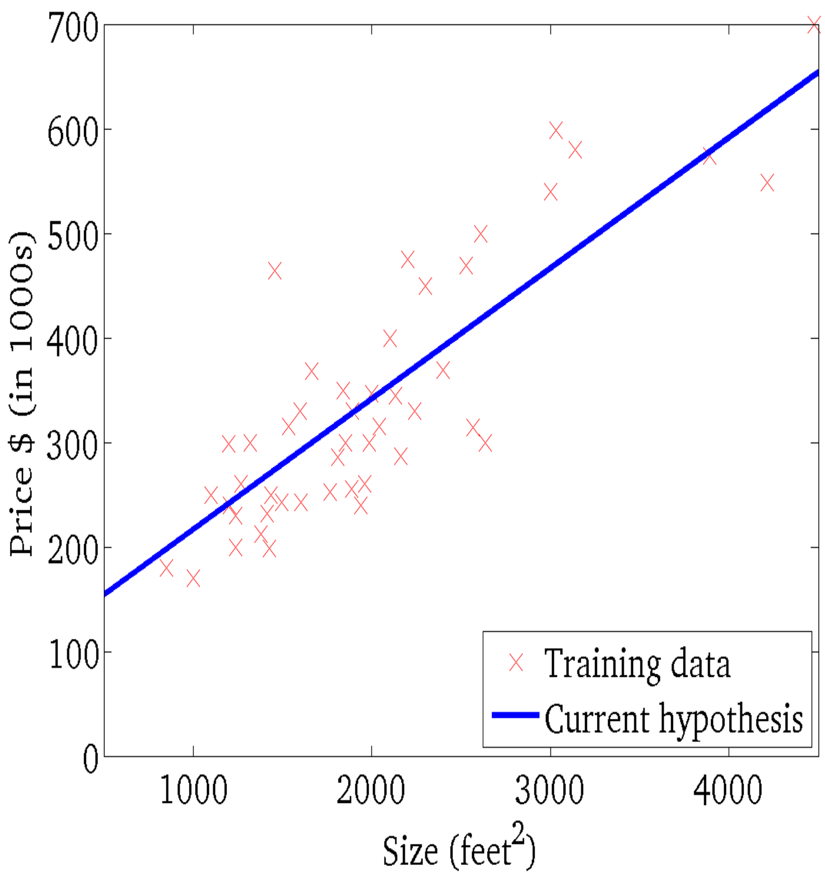
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



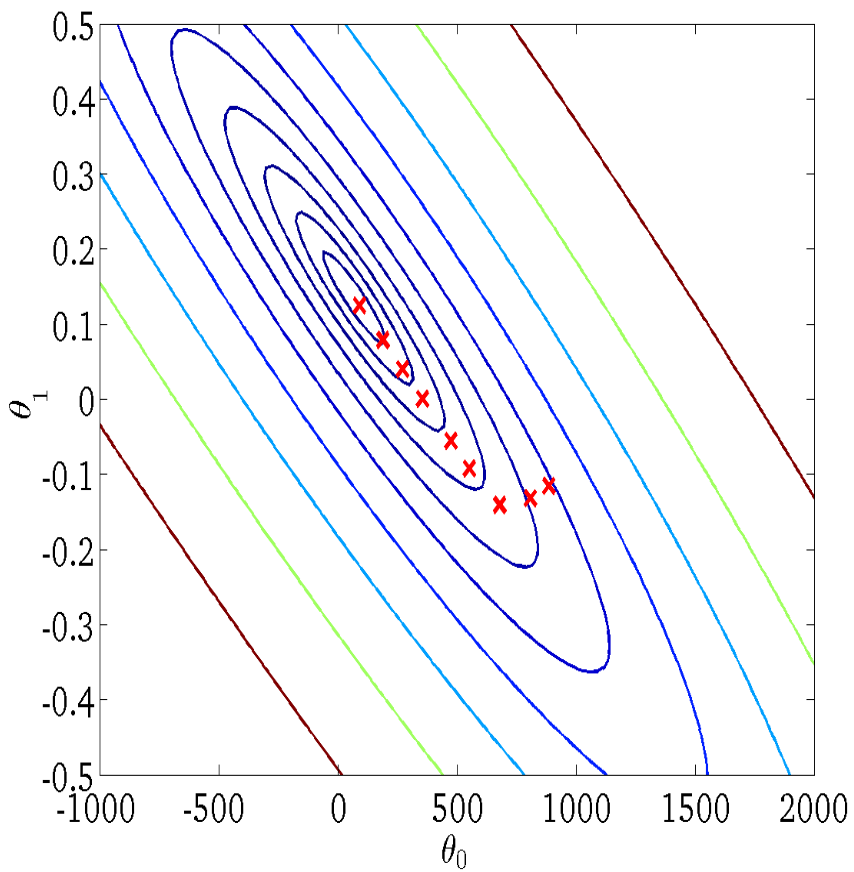
$$h_\theta(x)$$

(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)



Single feature (variable) : x

Size (feet^2) x	**Price (\$1000)** y
2104	460
1416	232
1534	315
852	178
…	…

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

Multiple features (variables).

Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

$$x^{(1)} = \begin{bmatrix} 2104 \\ 5 \\ 1 \\ 45 \end{bmatrix}$$

Notation:

n = number of features

$x^{(i)}$ = input (features) of i^{th} training example.

$x_j^{(i)}$ = value of feature j in i^{th} training example.

Hypothesis:

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

Now with multiple variables or features

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

Size (feet^2)

Number of bedrooms etc.

Or as per Bishop's book notations (page 138, section 3.1)

$$y(x,w) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4$$

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

Cost function:

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

Gradient descent:

Repeat $\{$

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

$\}$ (simultaneously update for every $j = 0, \ldots, n$)

Gradient Descent

Previously (n=1):

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial\theta_0}J(\theta)}$$

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

(simultaneously update θ_0, θ_1)

$\}$

New algorithm $(n \geq 1)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, \ldots, n$)

$\}$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}$$

$\ldots$

Linear Regression

$$y(x,w)= w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots\ldots + w_D x_D$$

Key Properties of Linear Regression

- y is a **linear** function of the parameters $w_0, w_1, w_2, \ldots w_D$
- y is a **linear** function of the input variables (features) $x_0, x_1, x_2, \ldots x_D$

Generalized Form of Linear Regression

- A notion of class of functions $\phi_i(x)$ is used to represent the regression function
- $y(x,w) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots\ldots + w_D x_D$

is represented as

$$y(x,w) = w_0 + w_1 \phi_1(x) + w_2 \phi_2(x) + w_3 \phi_3(x) + \ldots\ldots + w_D \phi_D(x)$$

Where $\phi_i(x) = x_i$

- $\phi_i(x)$ are called as basis functions for $i=1,2,3,..D$

Basis functions

- Linear basis functions $\phi_i(x)=x$ (Quadratic in x)

- Nonlinear basis functions

$\phi_i(x)=x^2$ (Quadratic in x)

$\phi_i(x)=x^3$ (Cubic in x)

What is linear in linear regression?

- The following expression is linear in W

$y(x,w)= w_0 + w_1 \phi_1(x)+ w_2 \phi_2(x)+ w_3 \phi_3(x)+ \ldots\ldots+w_D \phi_D(x)$

- The basis functions may be linear or nonlinear in x