CENG 463
Machine Learning

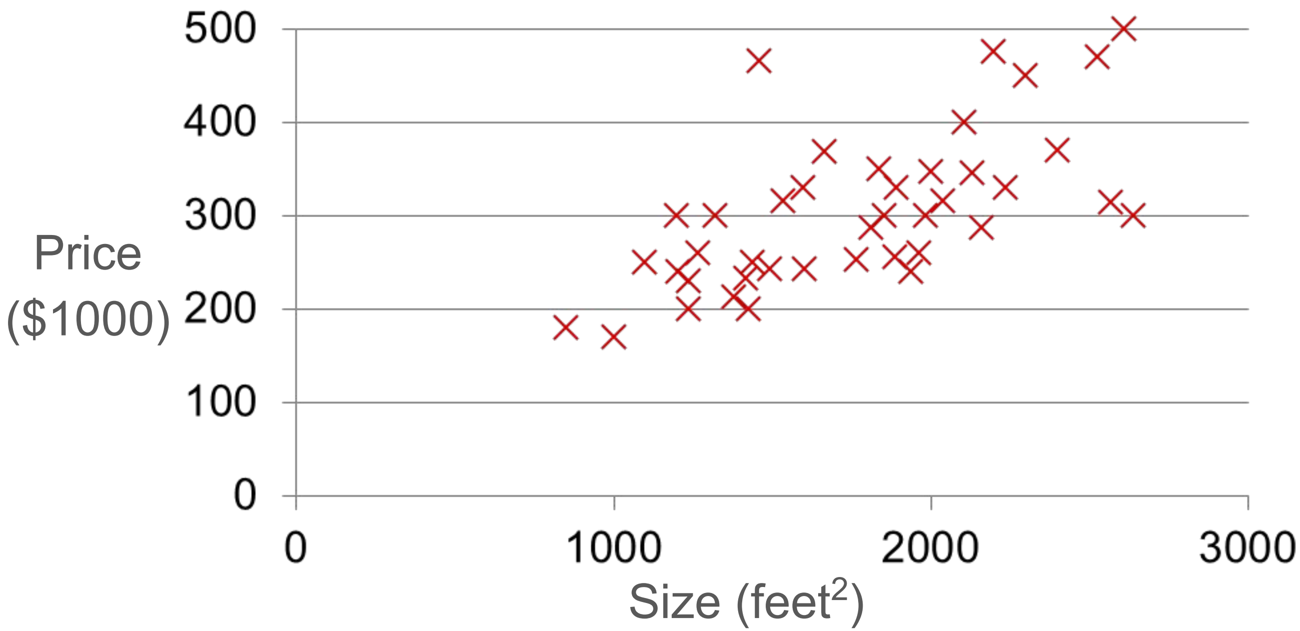
Lecture 04 - Linear Regression

Slides were prepared using the material provided in Stanford's Machine Learning Course by Andrew Ng.

Model with One Variable

Example: House prices according to area

- <u>Supervised Learning:</u> The “right answer” for each example in the data is given.
- <u>Regression Problem:</u> Predict real-valued output



Model with One Variable

Training set for house pricing:

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

Notation:

m = number of training examples

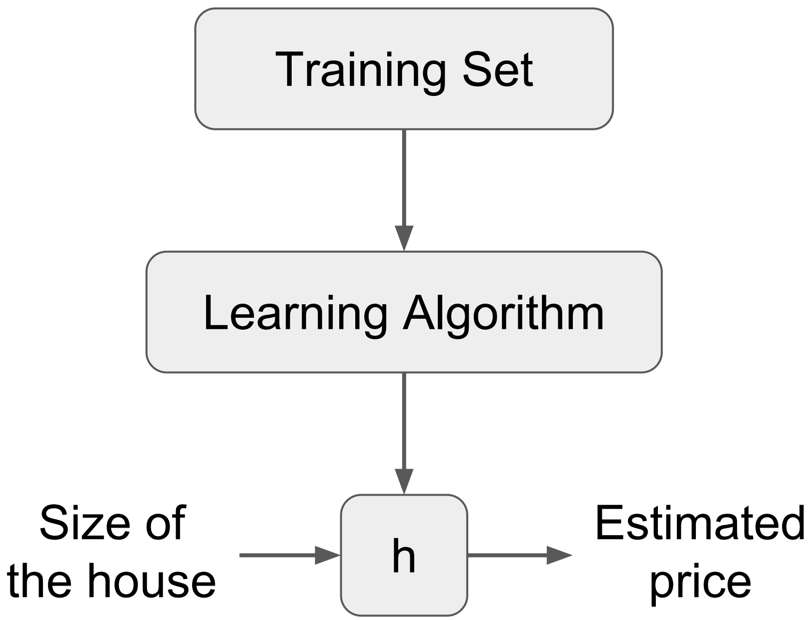
x = "input" variable / features

y = "output" variable / "target" variable

(x,y) = one training example

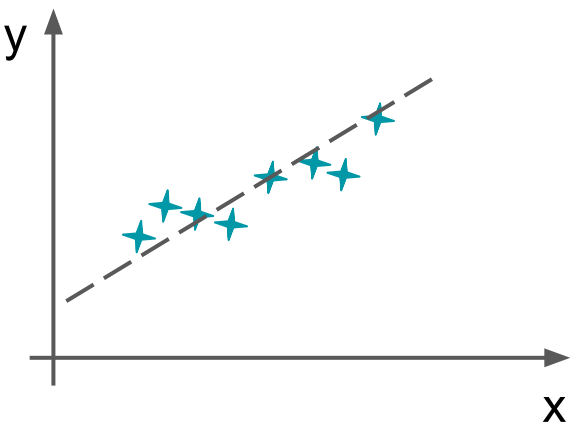
($x^{(i)}$, $y^{(i)}$) = i^{th} training example

Model with One Variable



h is a function from **x** to **y**.

How can we represent **h**?



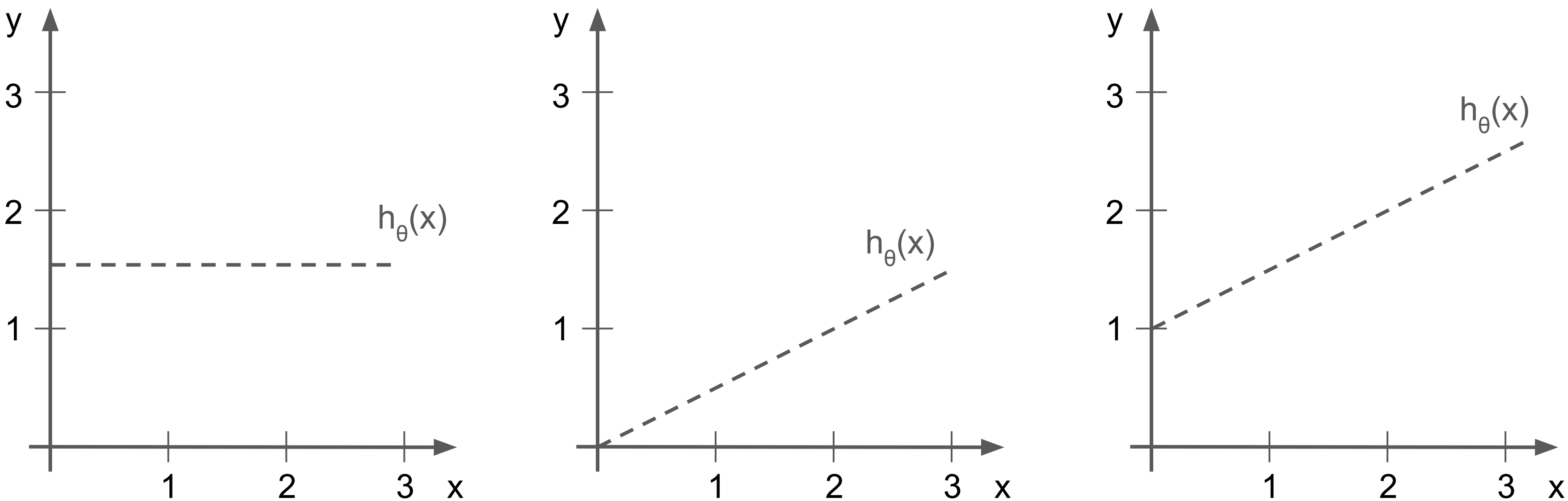
$$h_\theta(x) = \theta_0 + \theta_1 x$$

Linear regression with one variable.

i.e. Univariate linear regression.

Model with One Variable

- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
- θ_i's are **parameters**.



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

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

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

Cost Function

- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
- θ_i's are **parameters**. How to choose them?
- Choose θ_i's so that $\mathbf{h_\theta(x)}$ is close to $\mathbf{y}$ for our training examples.

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

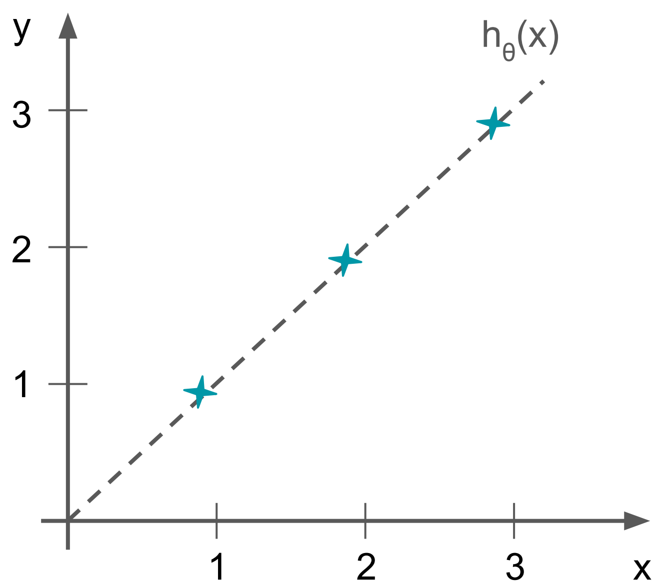
$$\min J(\theta_0, \theta_1)$$

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

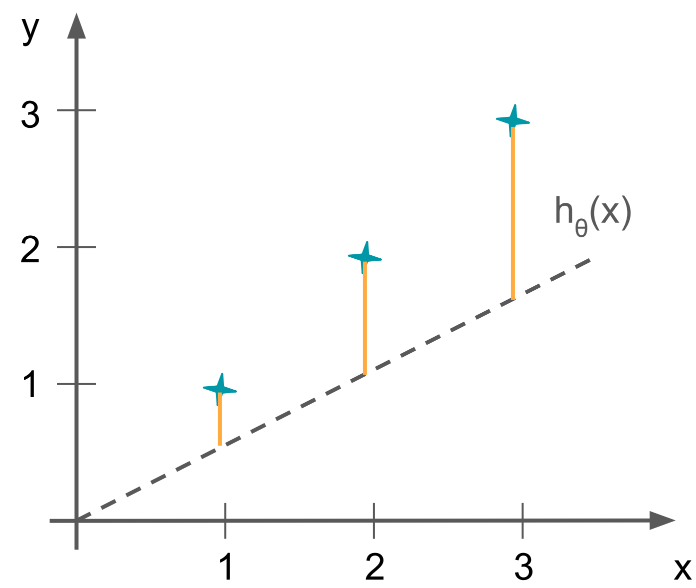
Cost Function

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

What is J(0,1)?

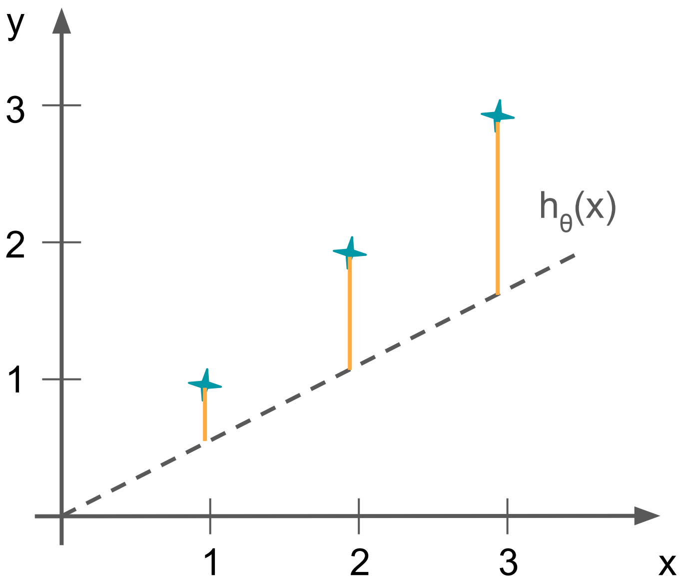


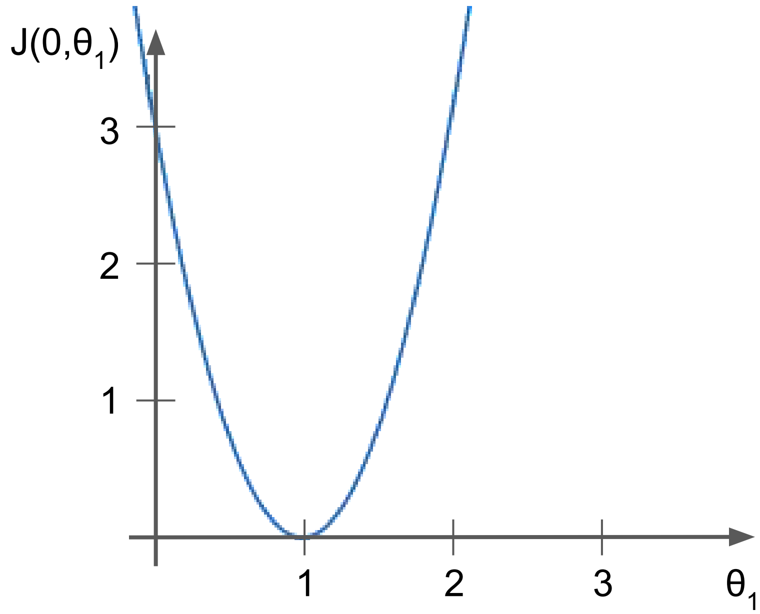
What is J(0,0.5)?



Cost Function

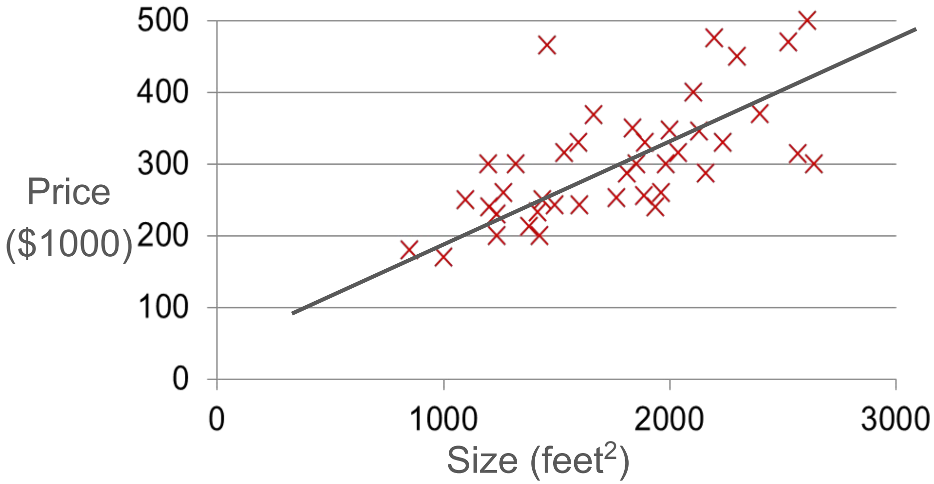
Change of cost function according to θ_1:

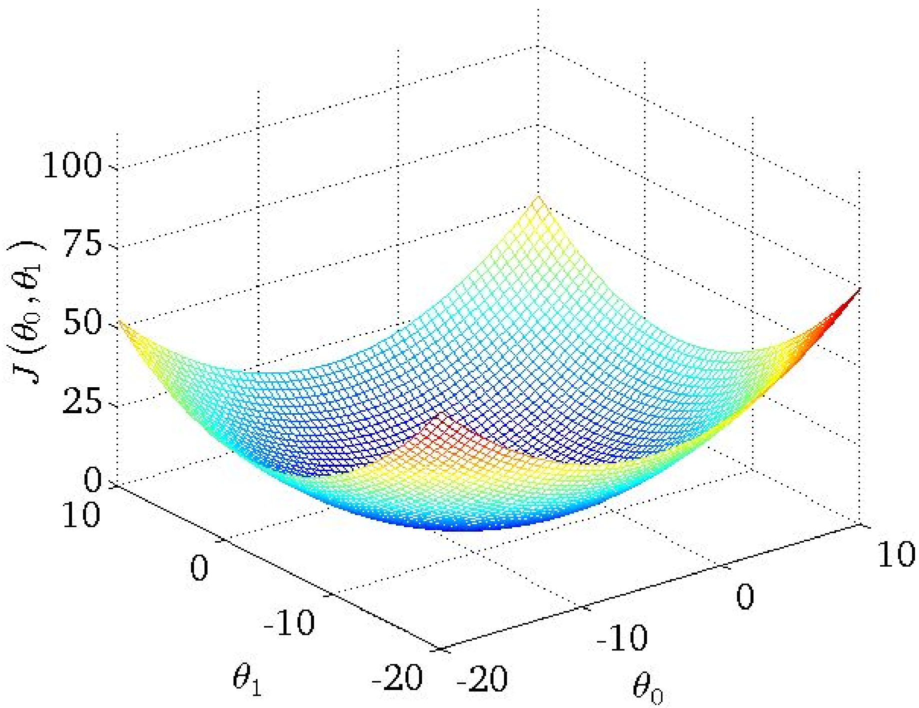




Cost Function

Change of cost function according to θ_0 and θ_1:

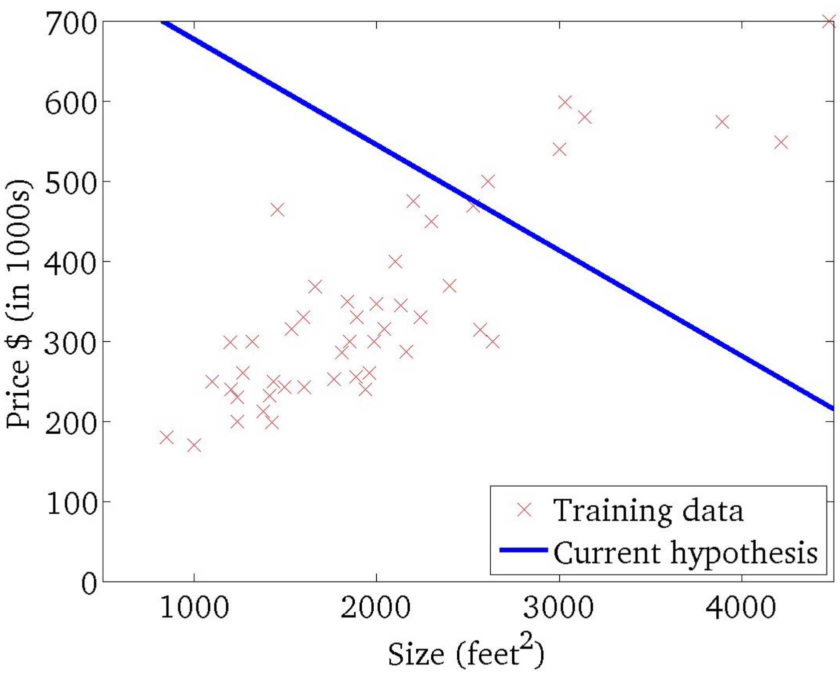




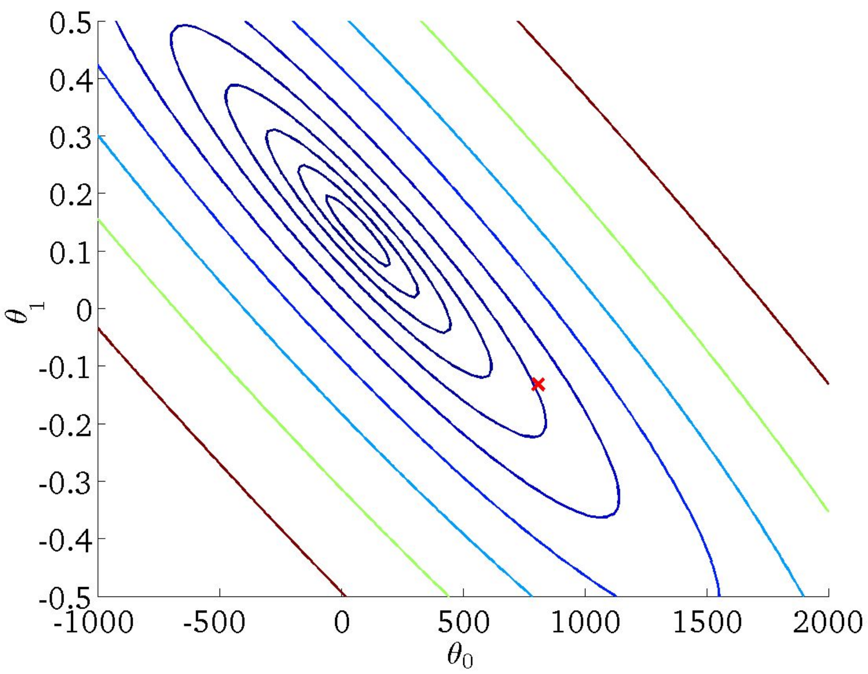
Cost Function

Cost with changing θ_0 and θ_1 (shown with a contour plot)

$h_\theta(x)$



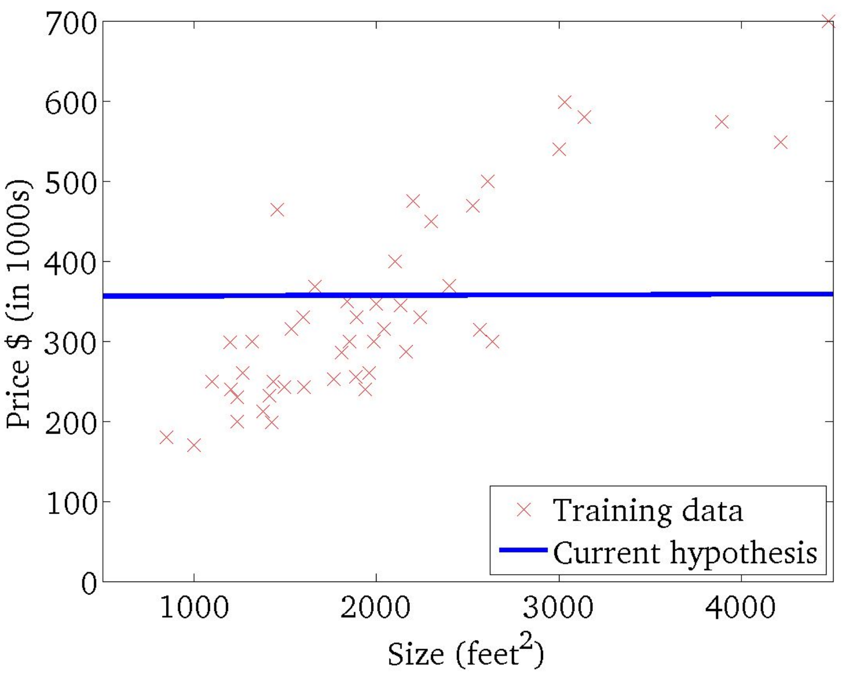
$J(\theta_0,\theta_1)$



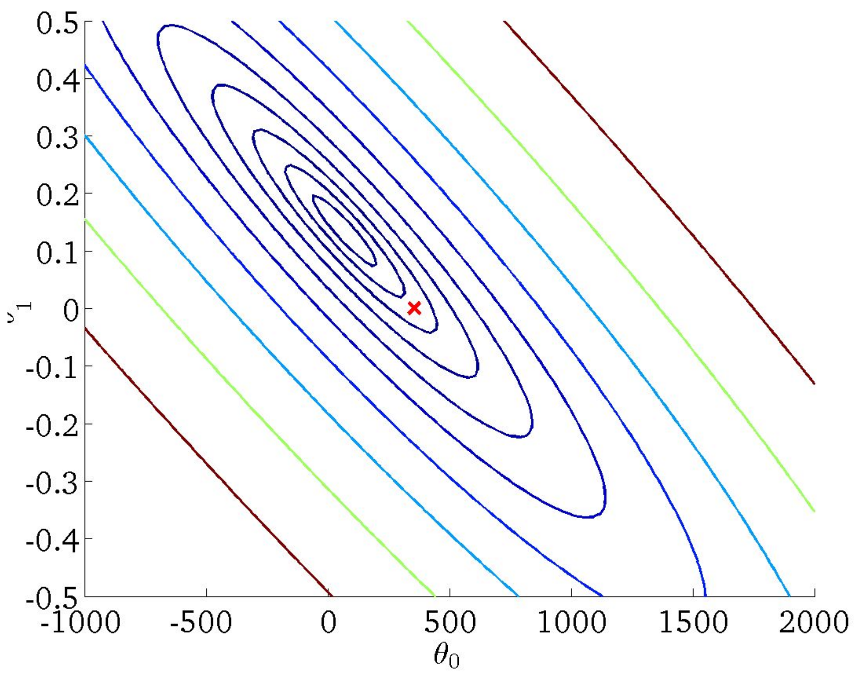
Cost Function

Cost with changing θ_0 and θ_1 (shown with a contour plot)

$h_\theta(x)$



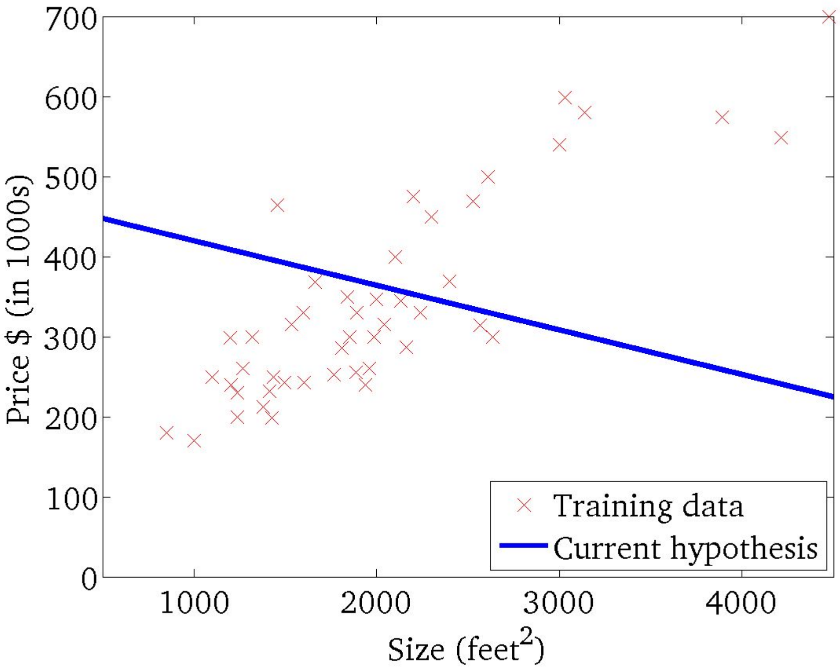
$J(\theta_0,\theta_1)$



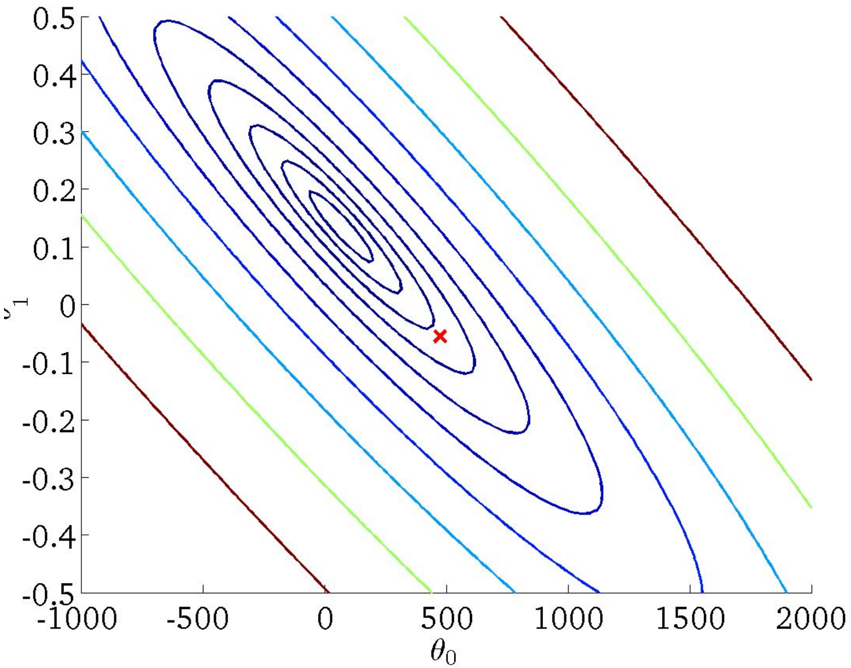
Cost Function

Cost with changing θ_0 and θ_1 (shown with a contour plot)

$h_\theta(x)$



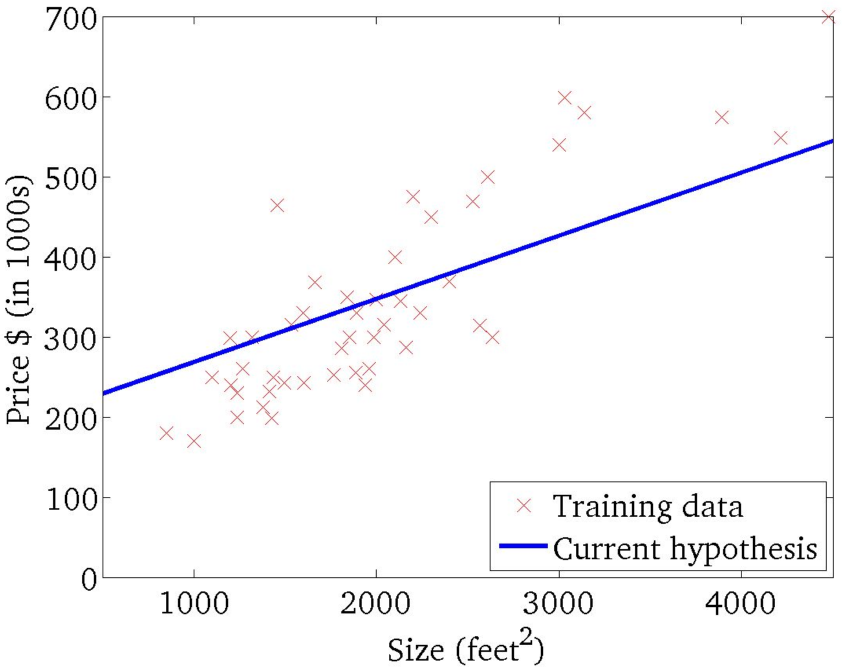
$J(\theta_0,\theta_1)$



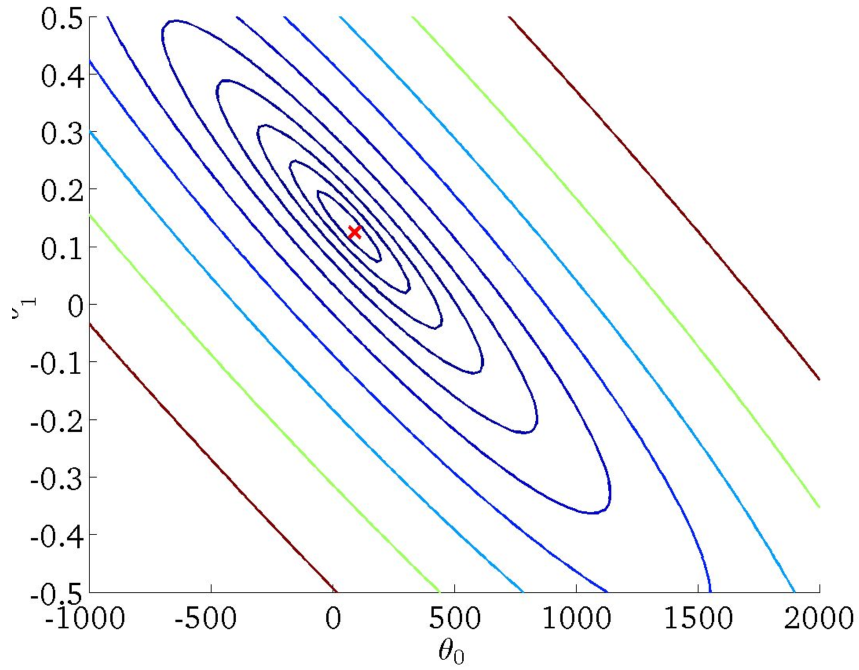
Cost Function

Cost with changing θ_0 and θ_1 (shown with a contour plot)

$h_\theta(x)$

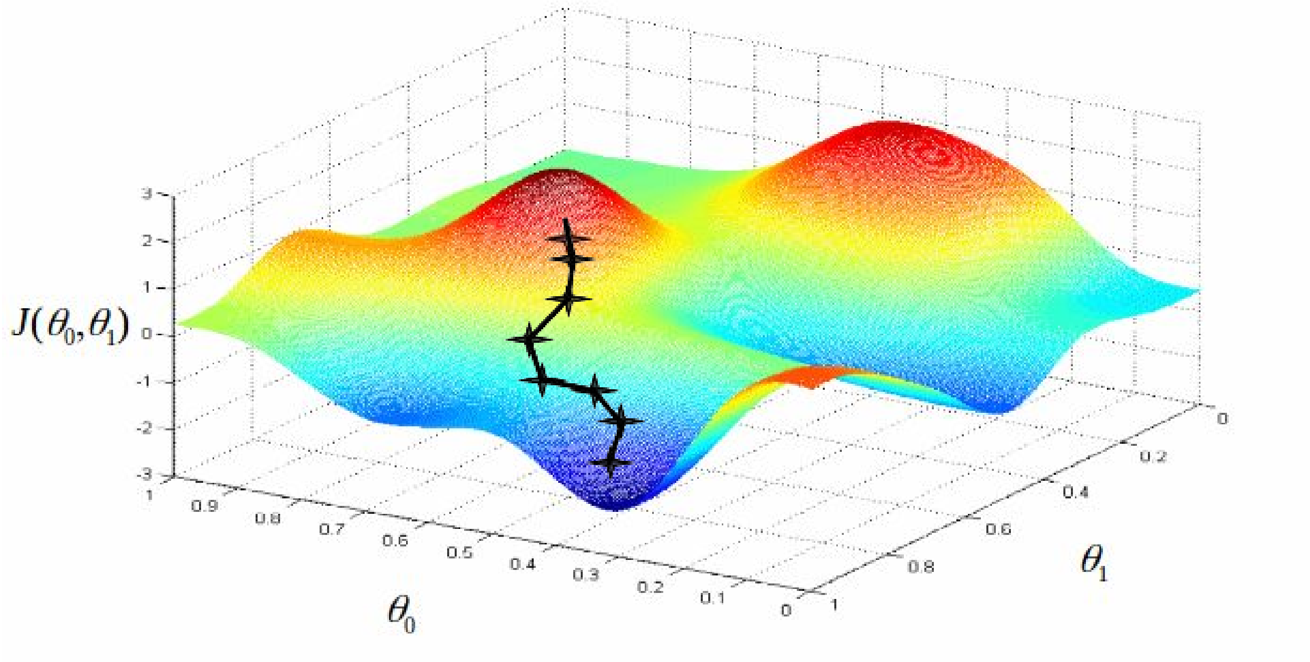


$J(\theta_0,\theta_1)$



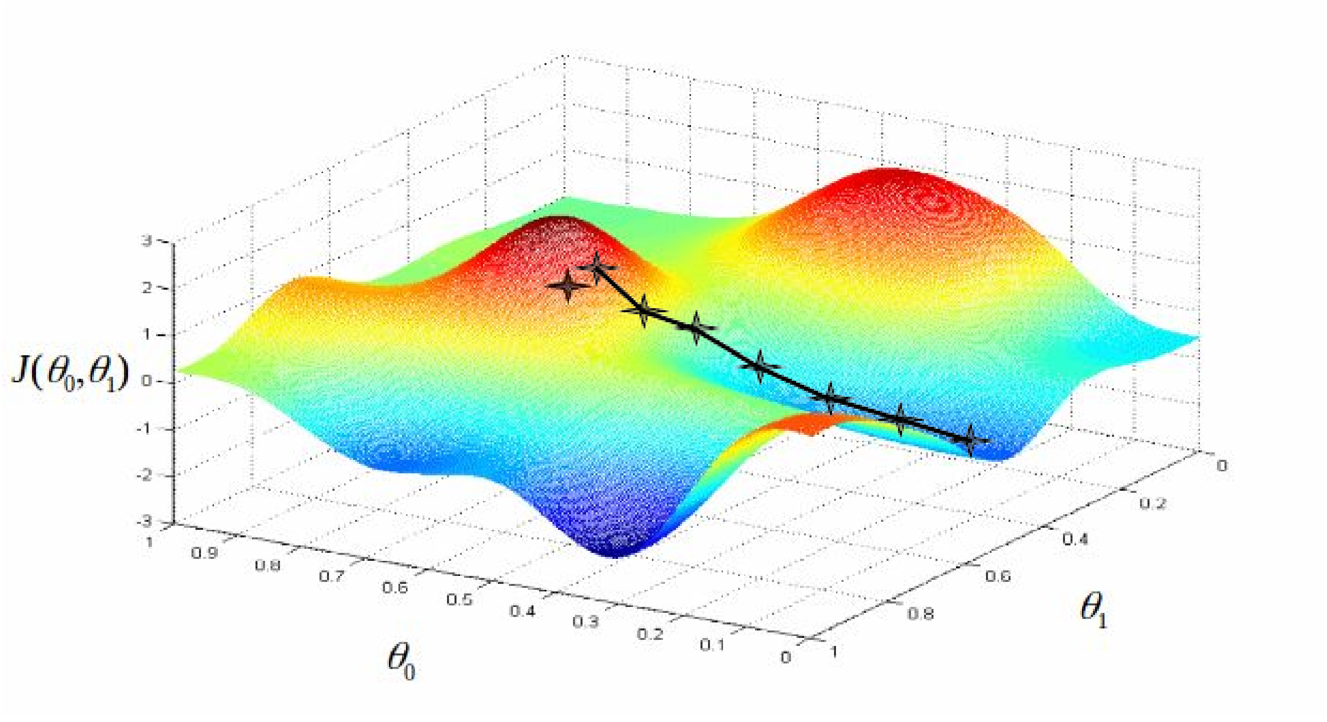
Gradient Descent

- We have some function $J(\theta_0,\theta_1)$ that we want to minimize.
  - Find θ_0, θ_1 parameters that minimize J:
    - Start with some θ_0, θ_1.
    - Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until hopefully we end up at a minimum.



Gradient Descent

- Gradient descent does **NOT** guarantee to reach the **global minimum**!



Gradient Descent

- Gradient descent algorithm for two-parameter case:

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$ *(for j=0 and j=1)*

}

This is the derivative term, indicates the direction of step.

This number is 'learning rate', controls the size of the step we take.

<u>Correct: Simultaneous update</u>

$$temp_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$temp_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := temp_0$$
$$\theta_1 := temp_1$$

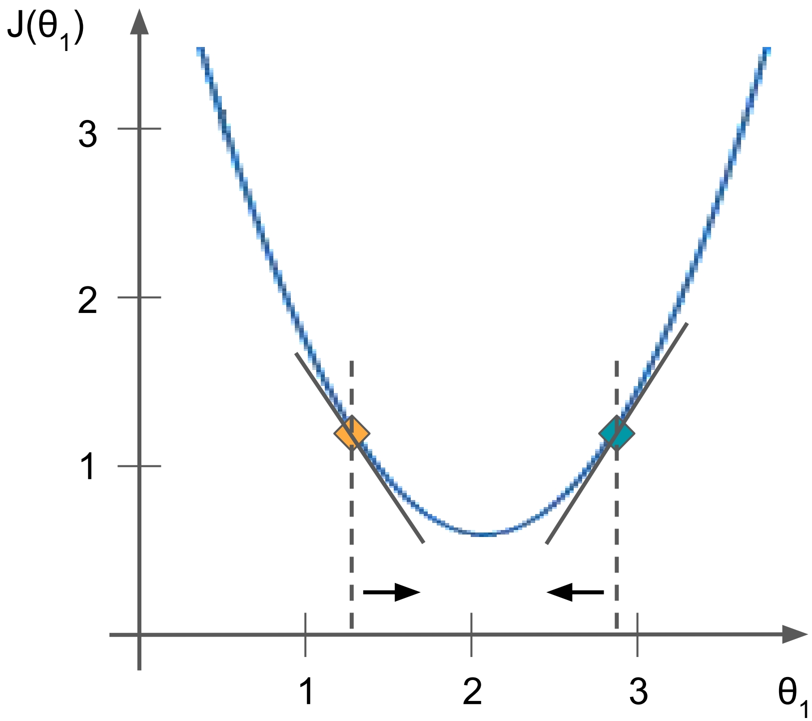
<u>Incorrect: Successive update</u>

$$temp_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := temp_0$$
$$temp_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := temp_1$$

Gradient Descent

- Let's assume a cost function with one variable (θ_1):

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

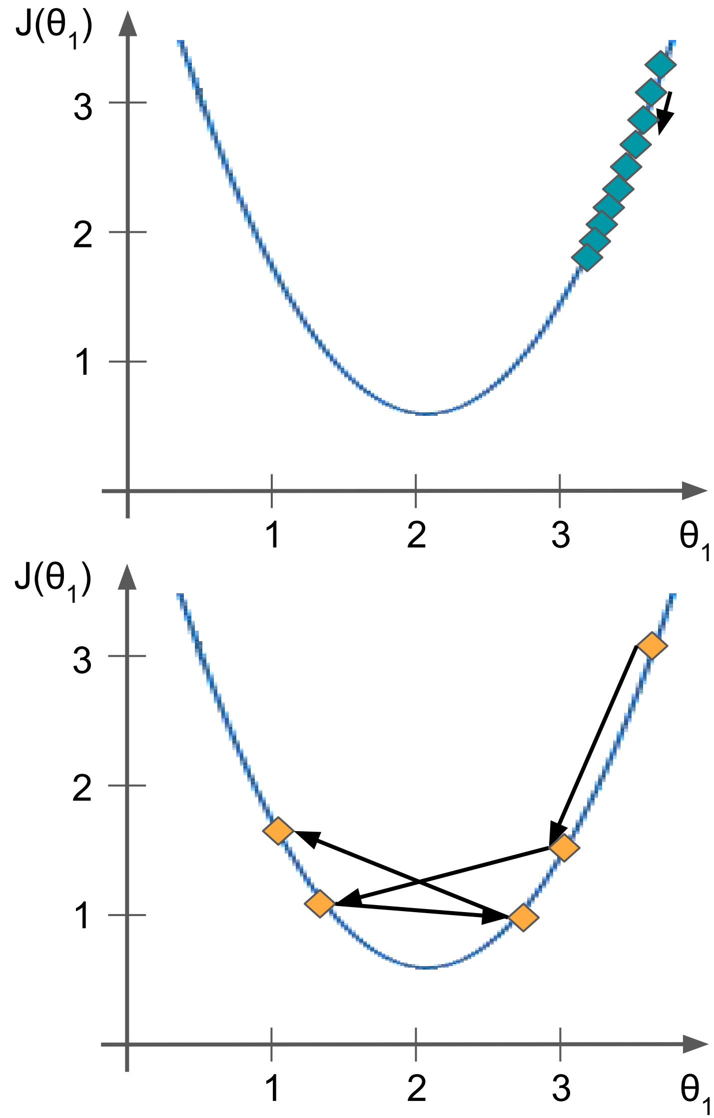


- The derivative gives the slope at that point.
  - Positive slope decreases the value of θ_1.
  - Negative slope increases θ_1.

Gradient Descent

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

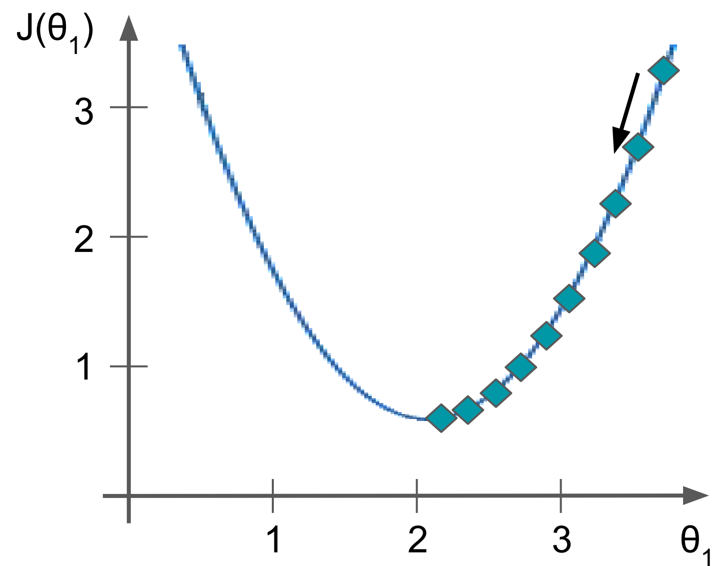
- If **α** is too small, gradient descent can be slow.
- If **α** is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.
- What does gradient descent do if we are already at the local minimum point?



Gradient Descent

- Gradient descent can converge to a local minimum, even with the learning rate α <u>fixed</u>.
- As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.

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



Gradient Descent for Linear Regression

- We merge two things we have learned:

<u>Gradient Descent</u>

repeat until convergence {

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

}

(for j=0 and j=1)

<u>Linear Regression</u>

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

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

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

Gradient Descent for Linear Regression

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

$$= \partial \frac{\frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2}{\partial \theta_j}$$

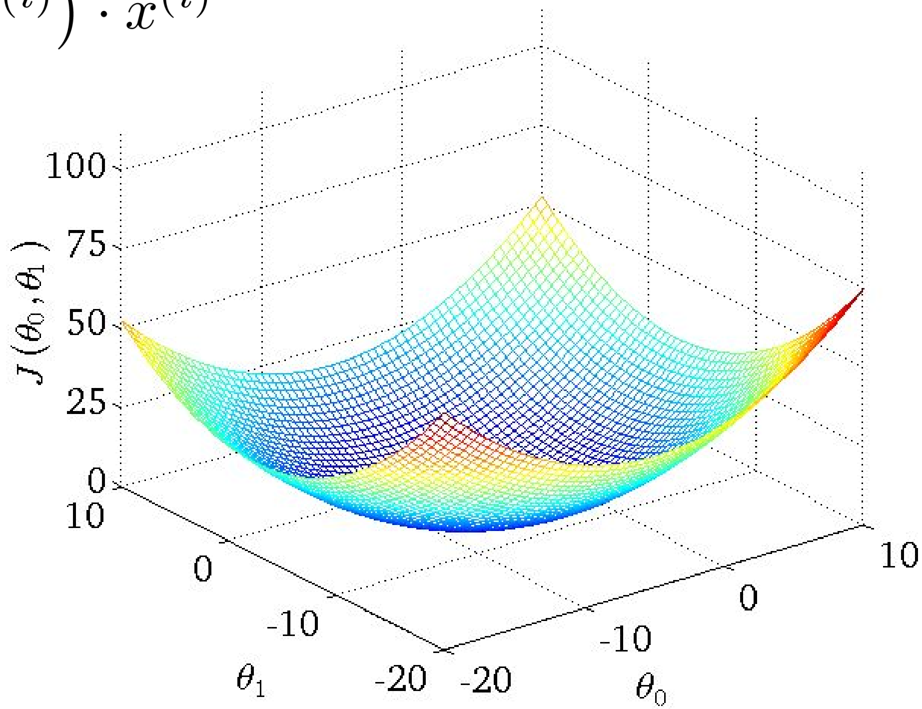
$$j = 0: \frac{\partial}{\partial \theta_0} J(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

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

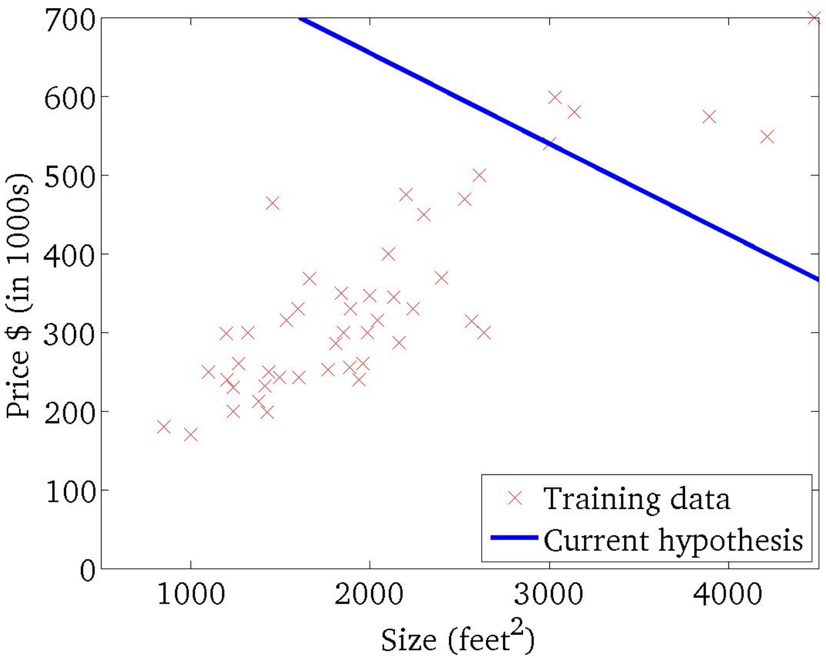
Gradient Descent for Linear Regression

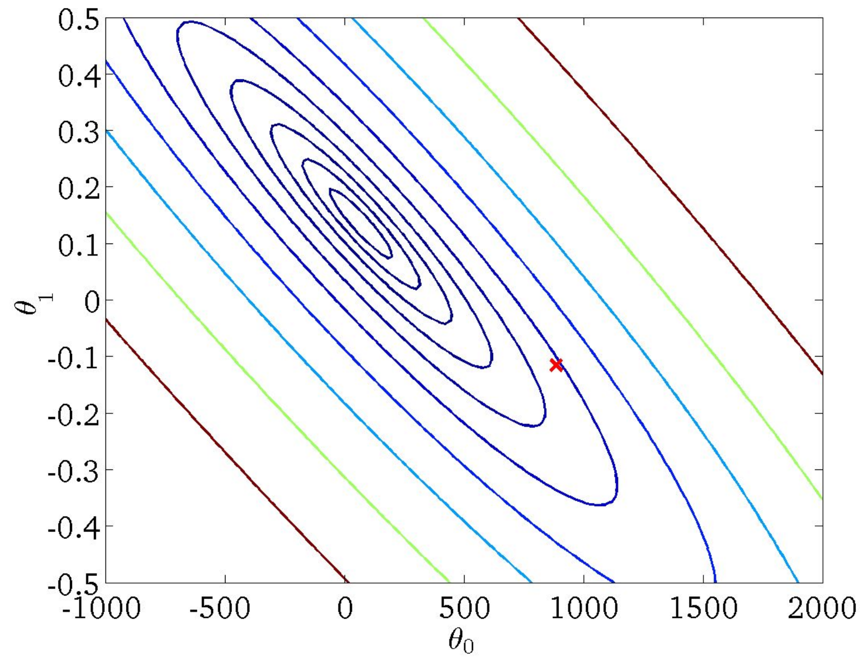
$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \\
&\quad \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)} \\
&\}
\end{aligned}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

Linear regression has a 'convex' (i.e. bowl-shaped) cost function, so <u>we expect to reach global minimum</u>.

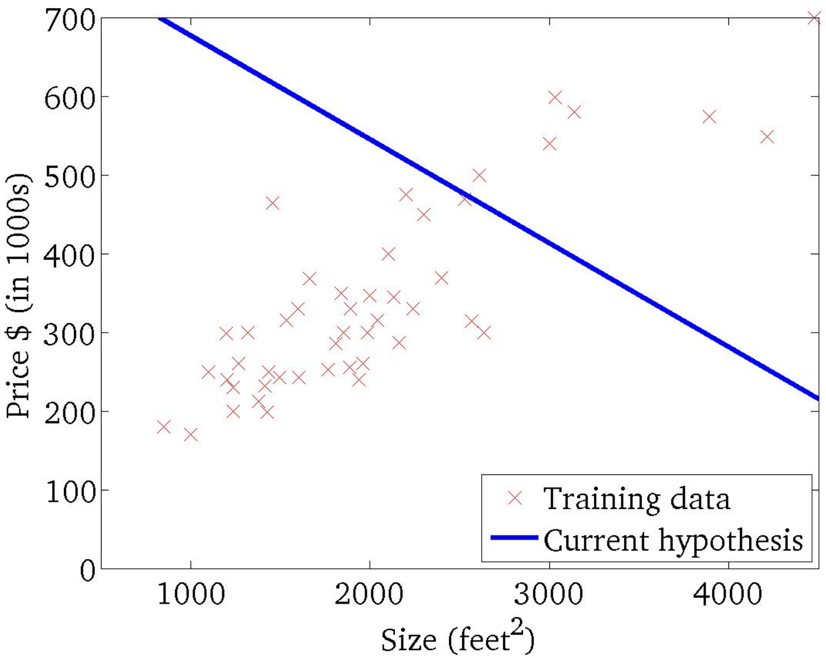


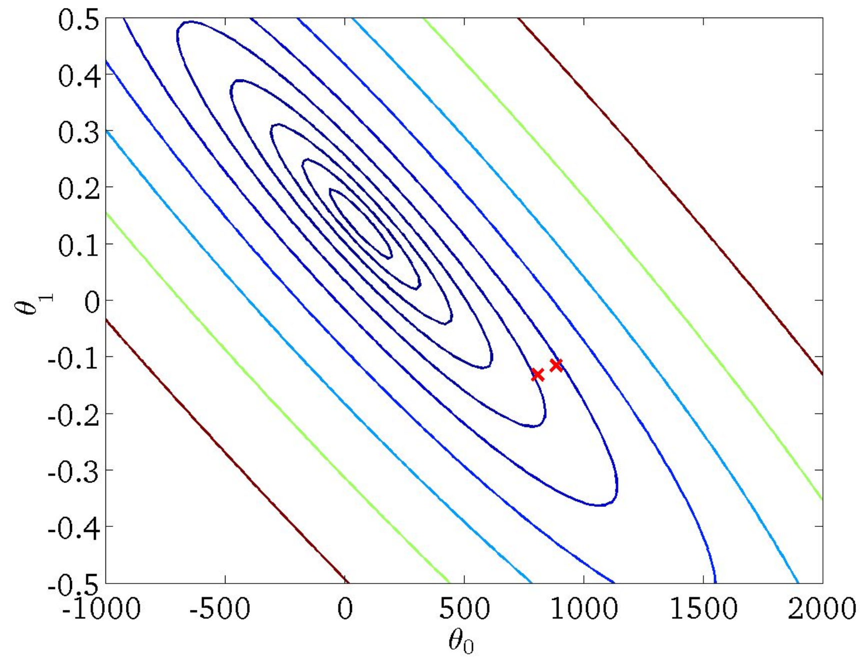
Gradient Descent in Action



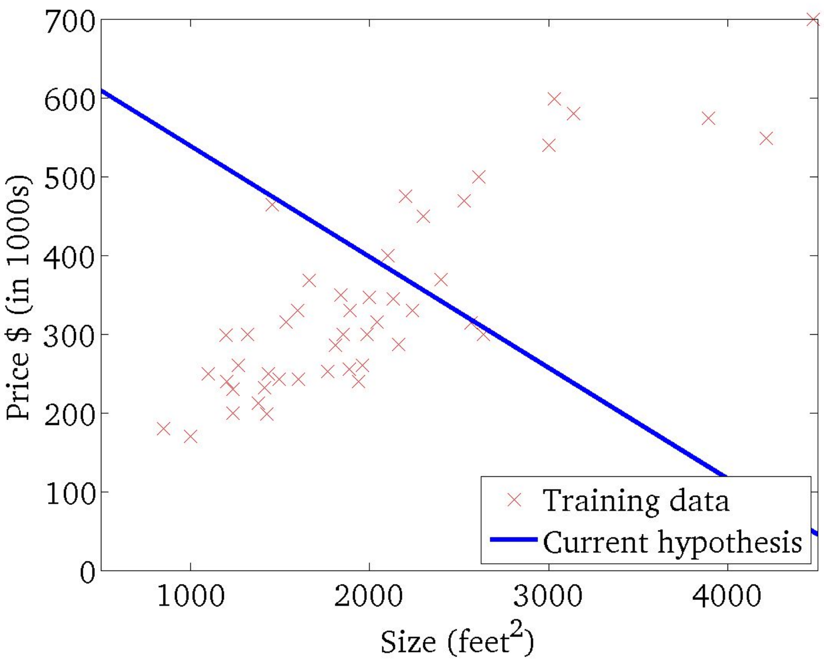


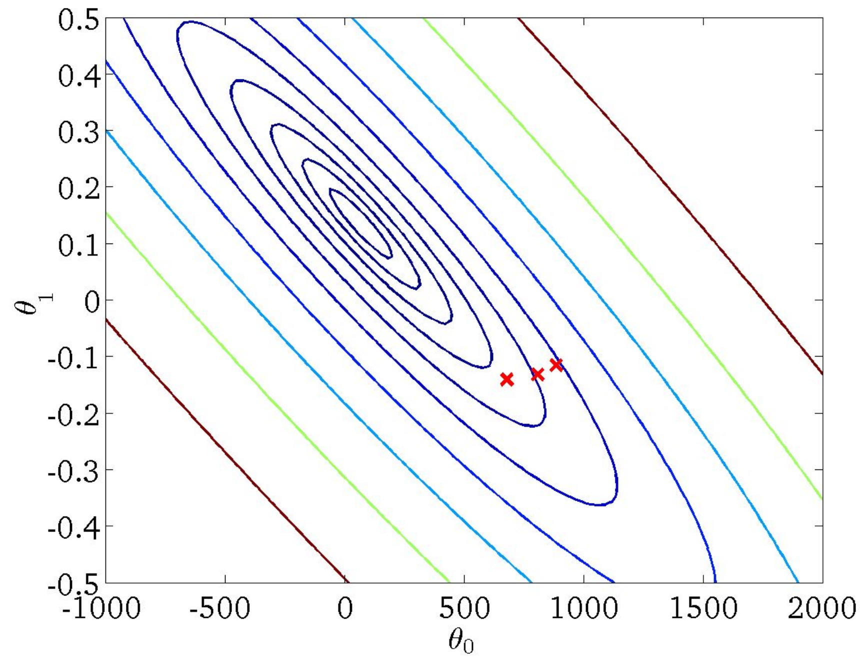
Gradient Descent in Action



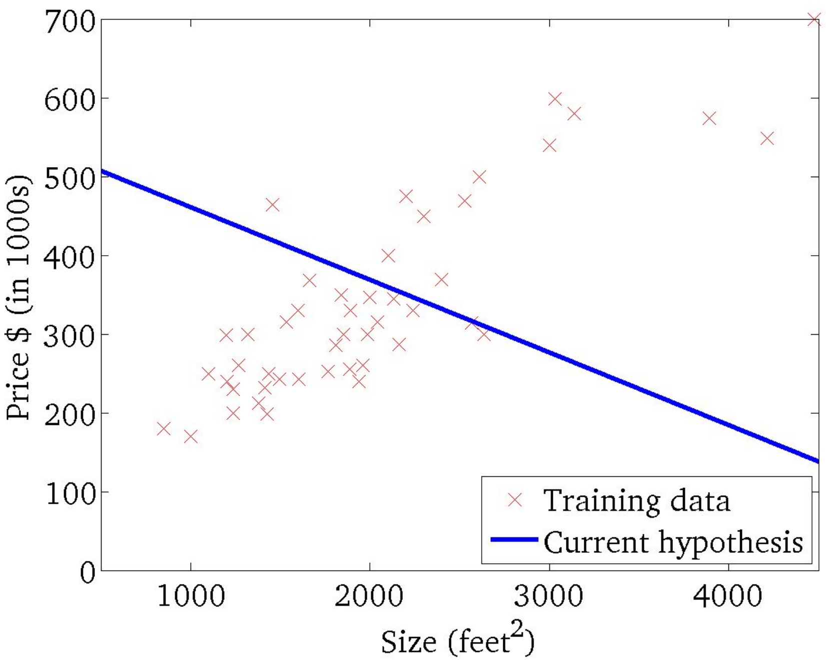


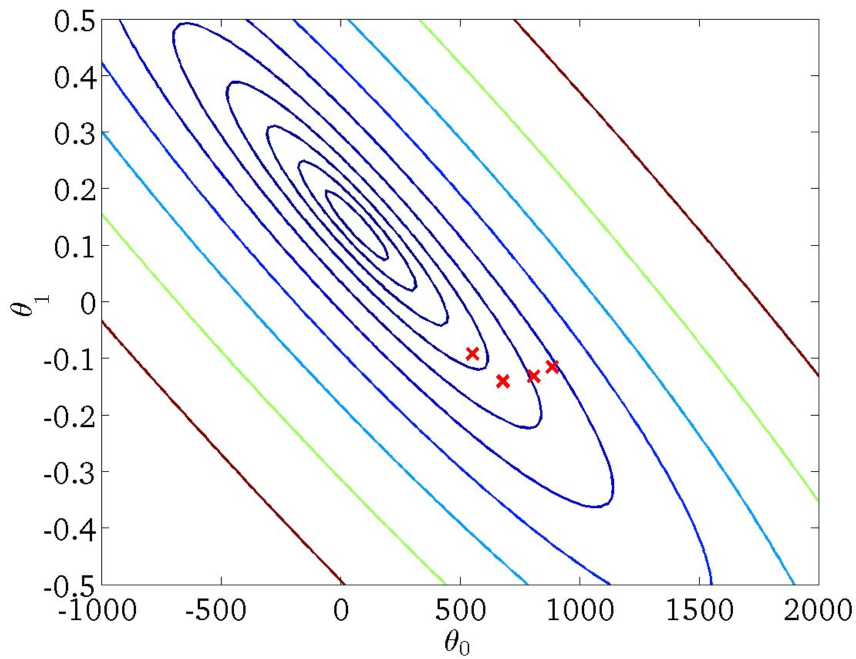
Gradient Descent in Action



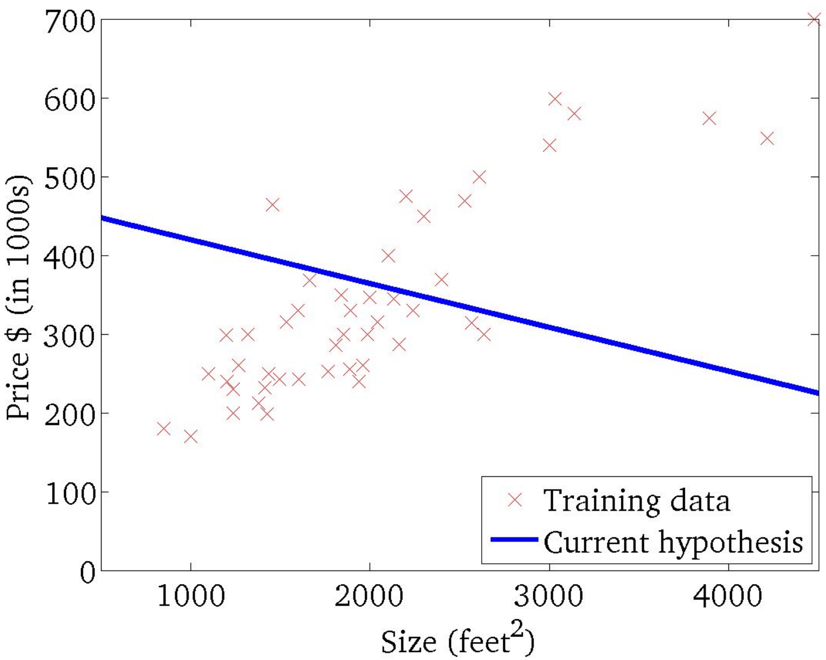


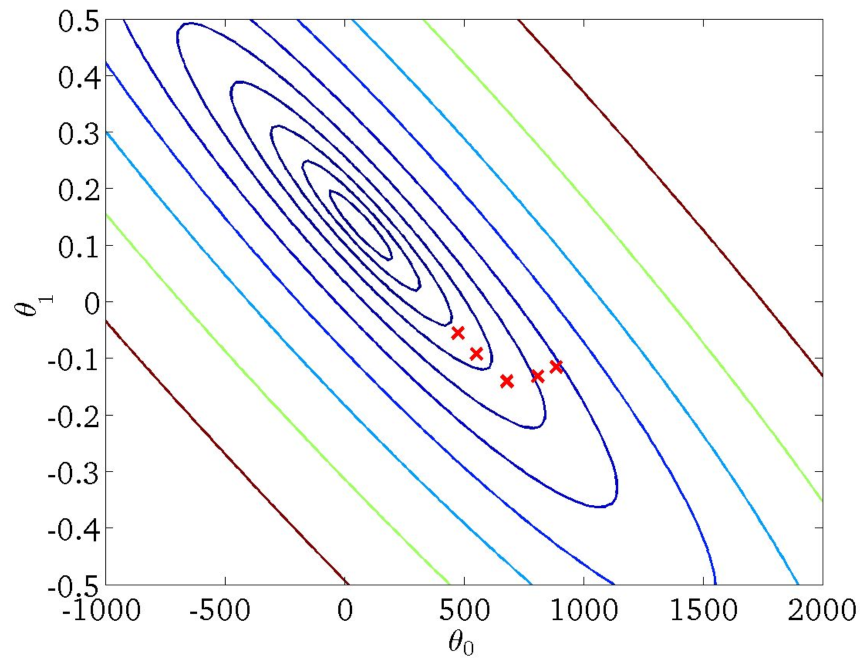
Gradient Descent in Action



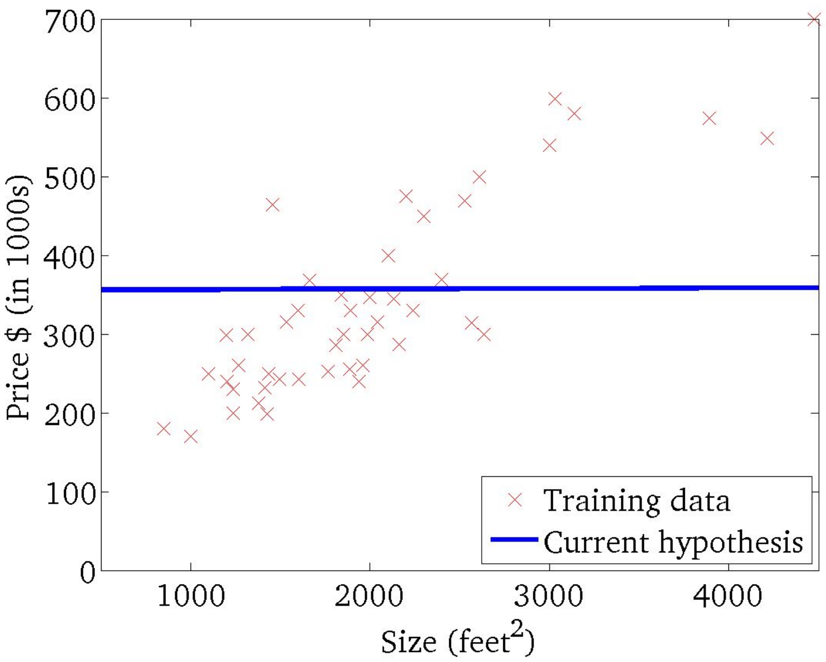


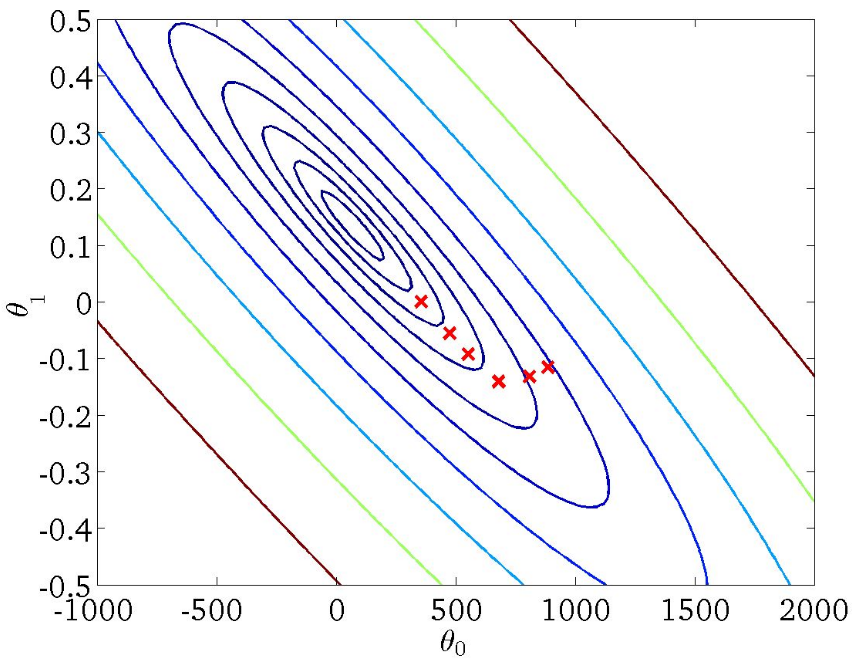
Gradient Descent in Action



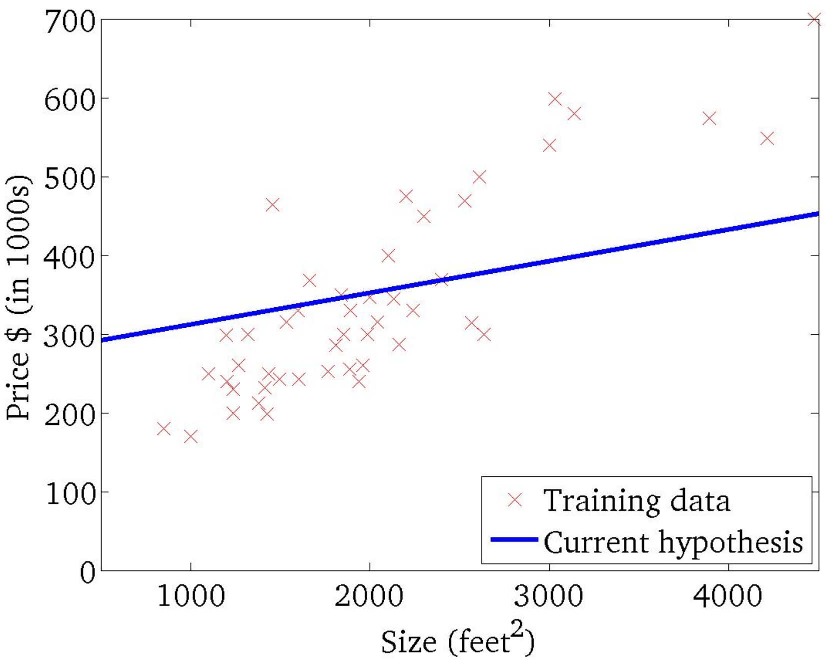


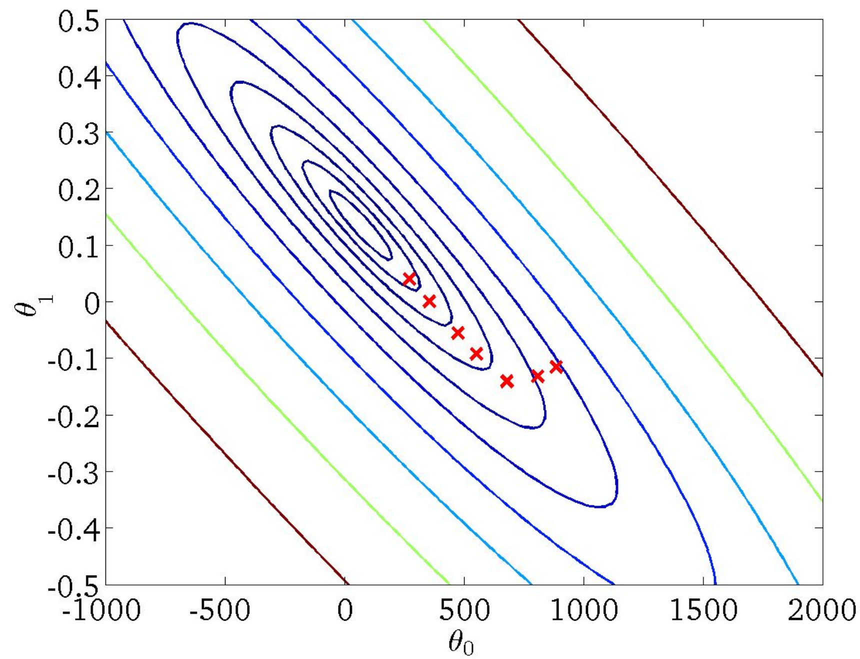
Gradient Descent in Action



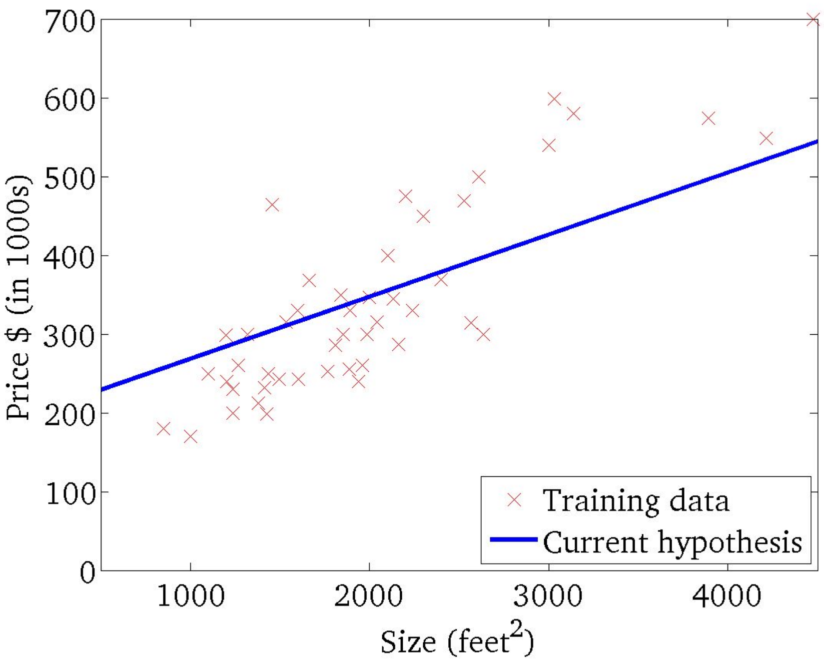


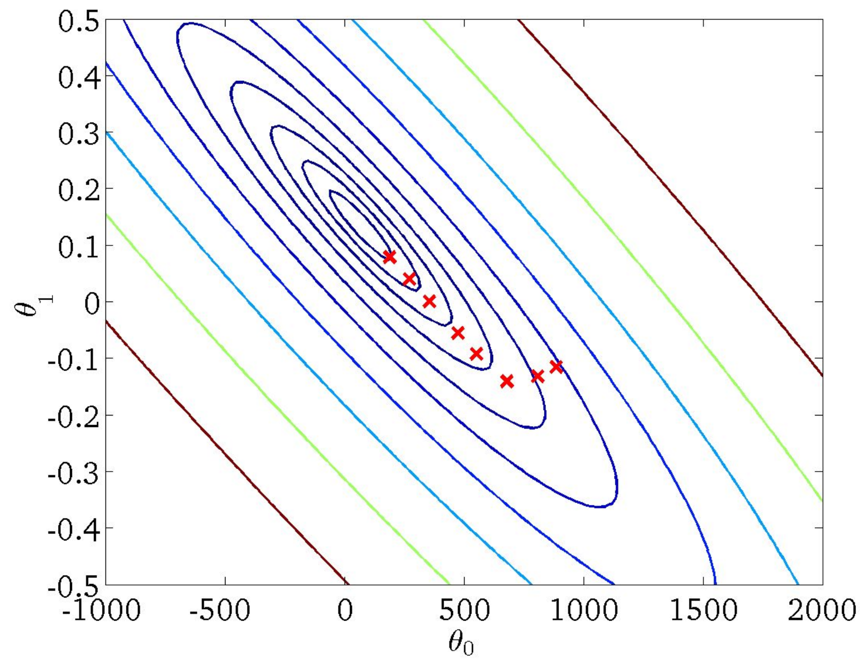
Gradient Descent in Action



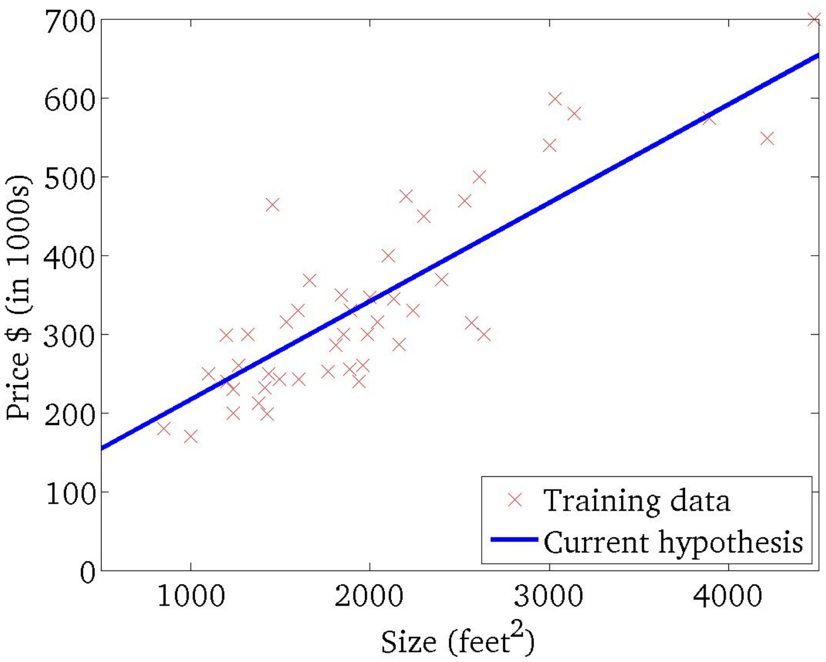


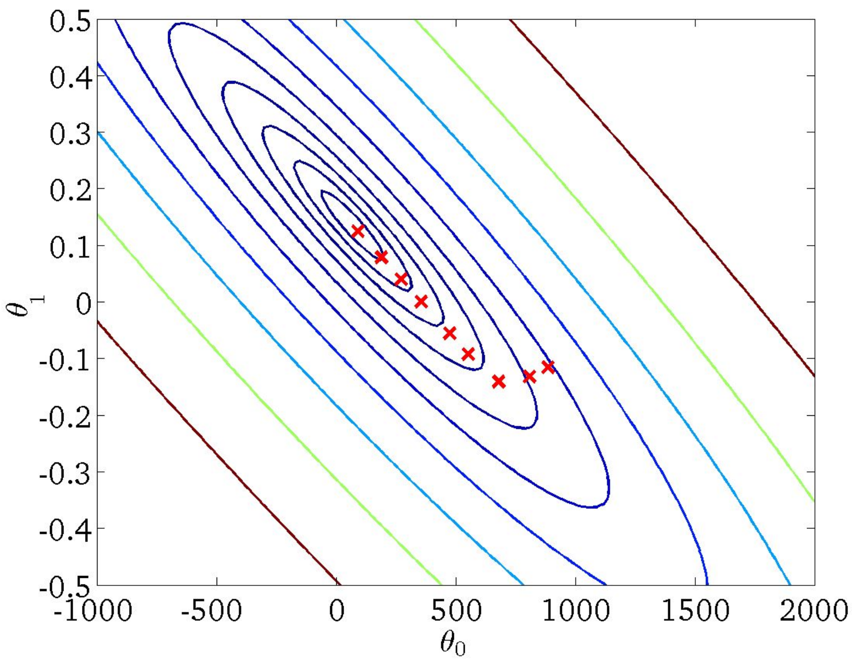
Gradient Descent in Action





Gradient Descent in Action





Summary

- We have learned about:
  - Linear Regression Model with One Variable
  - Cost Function
  - Gradient Descent
  - Gradient Descent with Linear Regression