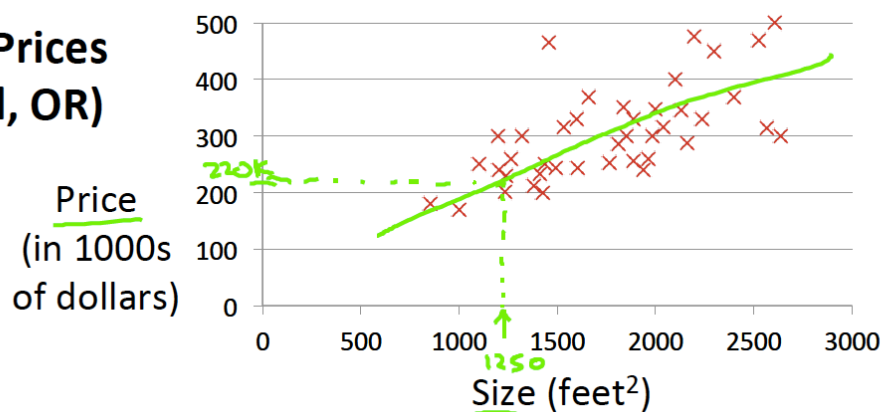


Housing Prices (Portland, OR)



Supervised Learning

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

Regression Problem

Predict real-valued output

Classification: Discrete-valued output

Andrew Ng

Training set of housing prices (Portland, OR)

| Size in feet ² (<u>x</u>) | Price (\$) in 1000's (<u>y</u>) |
|--|-----------------------------------|
| → 2104 | 460 |
| 1416 | 232 |
| → 1534 | 315 |
| 852 | 178 |
| ... | ... |

} m = 47

Notation:

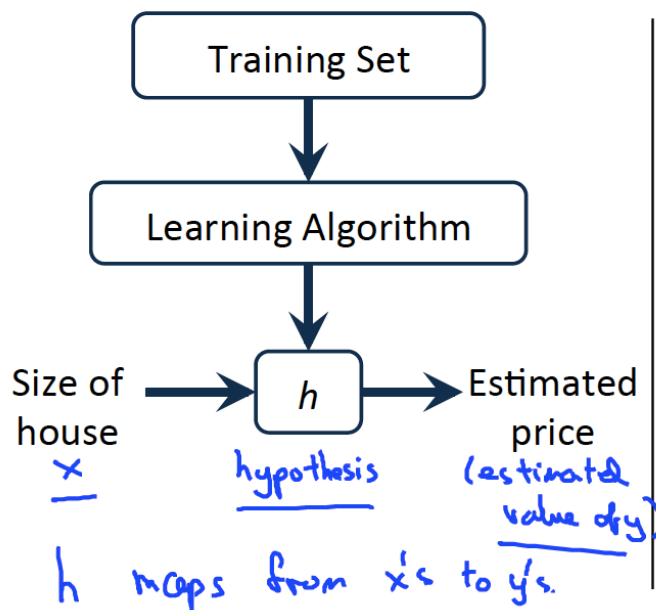
- **m** = Number of training examples
- **x**'s = “input” variable / features
- **y**'s = “output” variable / “target” variable

(**x**, **y**) - one training example

(**x**⁽ⁱ⁾, **y**⁽ⁱ⁾) - ith training example

$$\left\{ \begin{array}{l} x^{(1)} = 2104 \\ x^{(2)} = 1416 \\ y^{(1)} = 460 \end{array} \right.$$

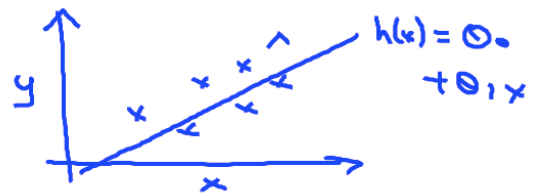
Andrew Ng



How do we represent h ?

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

Shorthand: $h(x)$



Linear regression with one variable. (x)
Univariate linear regression.
one variable

Andrew Ng

损失函数（衡量一个模型的好坏）

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

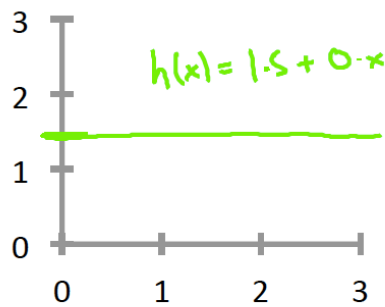
$m = 47$

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

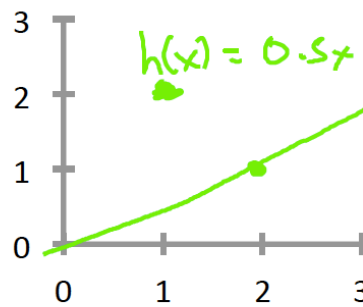
θ_i 's: Parameters

How to choose θ_i 's?

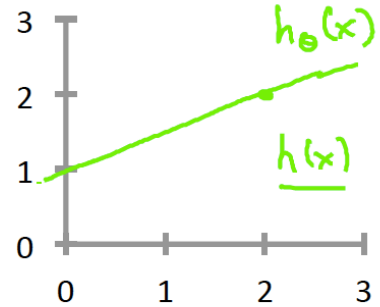
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



$$\begin{aligned} \rightarrow \theta_0 &= 1.5 \\ \rightarrow \theta_1 &= 0 \end{aligned}$$

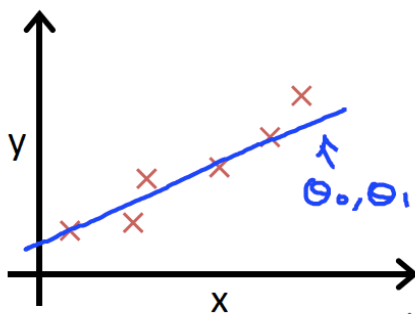


$$\begin{aligned} \rightarrow \theta_0 &= 0 \\ \rightarrow \theta_1 &= 0.5 \end{aligned}$$



$$\begin{aligned} \rightarrow \theta_0 &= 1 \\ \rightarrow \theta_1 &= 0.5 \end{aligned}$$

Andrew Ng



$(x^{(i)}, y^{(i)})$

$$\begin{aligned} \text{minimize}_{\theta_0, \theta_1} \quad & \frac{1}{2m} \sum_{i=1}^m \left(h_{\theta}(x^{(i)}) - y^{(i)} \right)^2 \\ & \uparrow \\ & \text{#training examples} \\ & h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)} \end{aligned}$$

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

Idea: Choose θ_0, θ_1 so that $h_{\theta}(x)$ is close to y for our training examples (x, y)

minimize $J(\theta_0, \theta_1)$
 θ_0, θ_1
 Cost function
 Squared error function

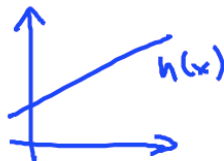
Andrew Ng

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$



Cost Function:

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

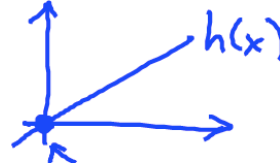
Goal: minimize $J(\theta_0, \theta_1)$
 θ_0, θ_1

Simplified

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

$$\theta_0 = 0$$

$$\theta_1$$



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

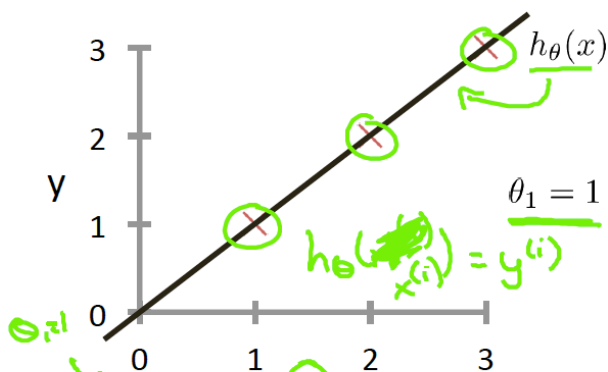
minimize $J(\theta_1)$
 θ_1

Andrew Ng

注：1/2m而不是1/m是为了方便计算（个人理解）。平方损失函数

$\rightarrow h_{\theta}(x)$

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



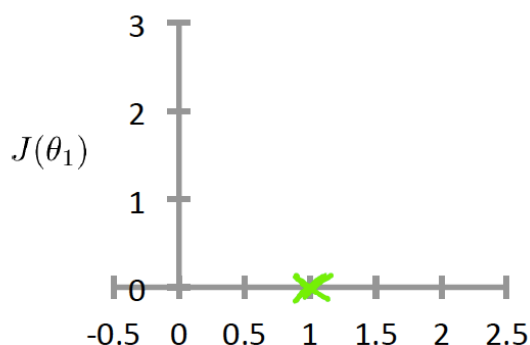
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$= \frac{1}{2m} \sum_{i=1}^m (\theta_1 x^{(i)} - y^{(i)})^2 = \frac{1}{2m} (0^2 + 0^2 + 0^2) = 0^2$$

$$J(1) = 0$$

$\rightarrow J(\theta_1)$

(function of the parameter θ_1)

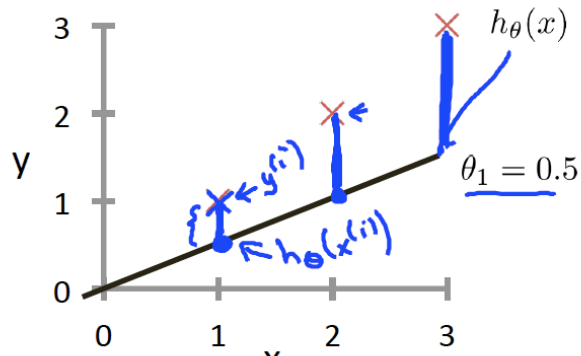


$$\theta_1 = 0.5?$$

Andrew Ng

$$h_{\theta}(x)$$

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

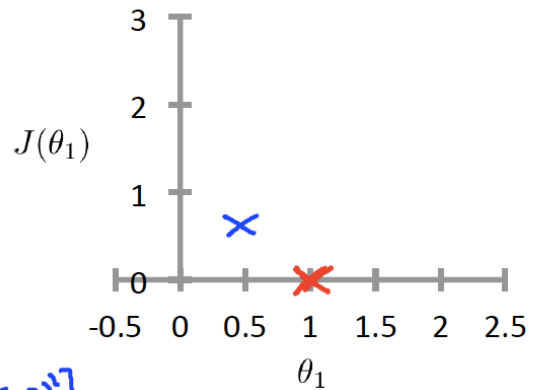


$$J(0.5) = \frac{1}{2m} [(0.5-1)^2 + (1-2)^2 + (1.5-3)^2]$$

$$= \frac{1}{2 \times 3} (3.5) = \frac{3.5}{6} \approx 0.58$$

$$J(\theta_1)$$

(function of the parameter θ_1)



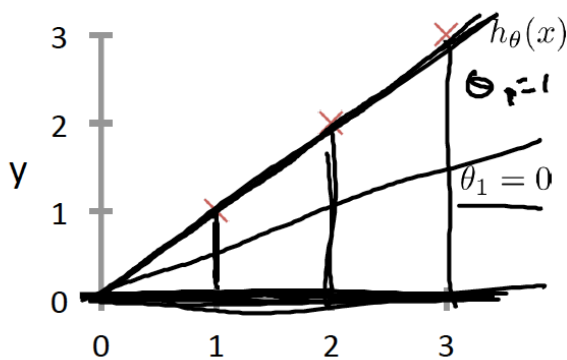
$$\theta_1 = 0?$$

$$J(0) = ?$$

Andrew Ng

$$h_{\theta}(x)$$

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

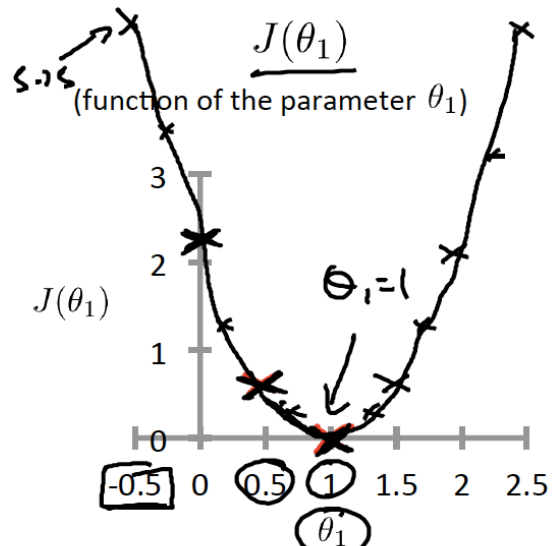


$$J(1) = \frac{1}{2m} (1^2 + 2^2 + 3^2)$$

$$= \frac{1}{6} \cdot 14 \approx 2.3$$

$$h(x) = -0.5x$$

$$\text{Minimize } J(\theta_1)$$



Andrew Ng

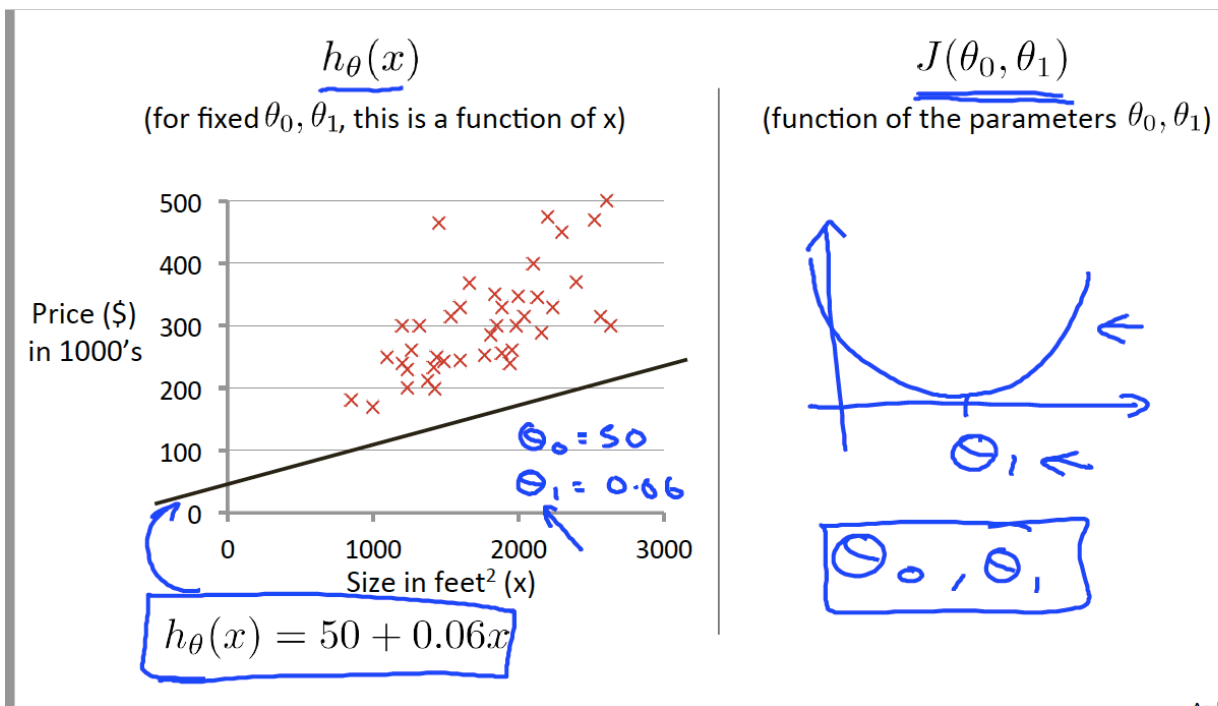
Hypothesis: $h_{\theta}(x) = \theta_0 + \theta_1 x$

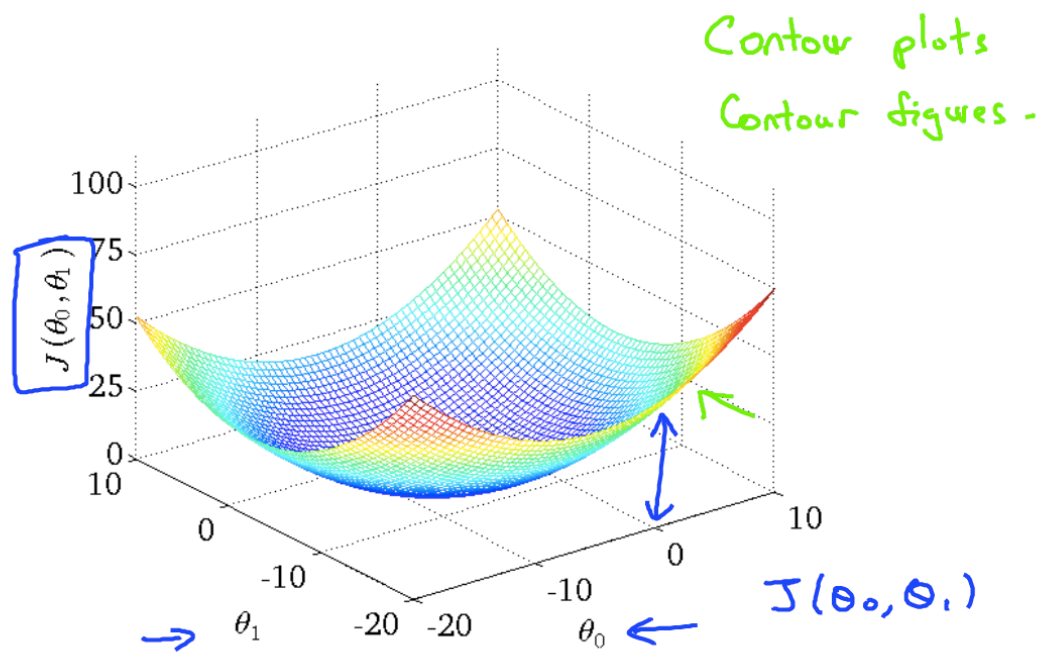
Parameters: θ_0, θ_1

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

Goal: minimize $J(\theta_0, \theta_1)$
 θ_0, θ_1

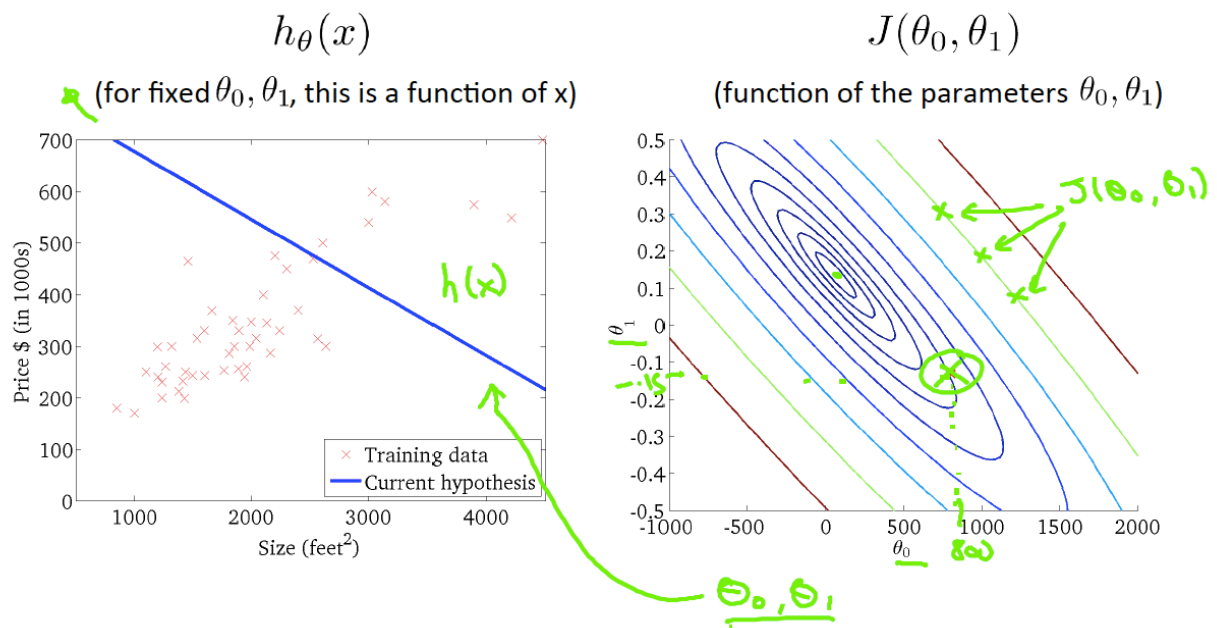
Andrew Ng





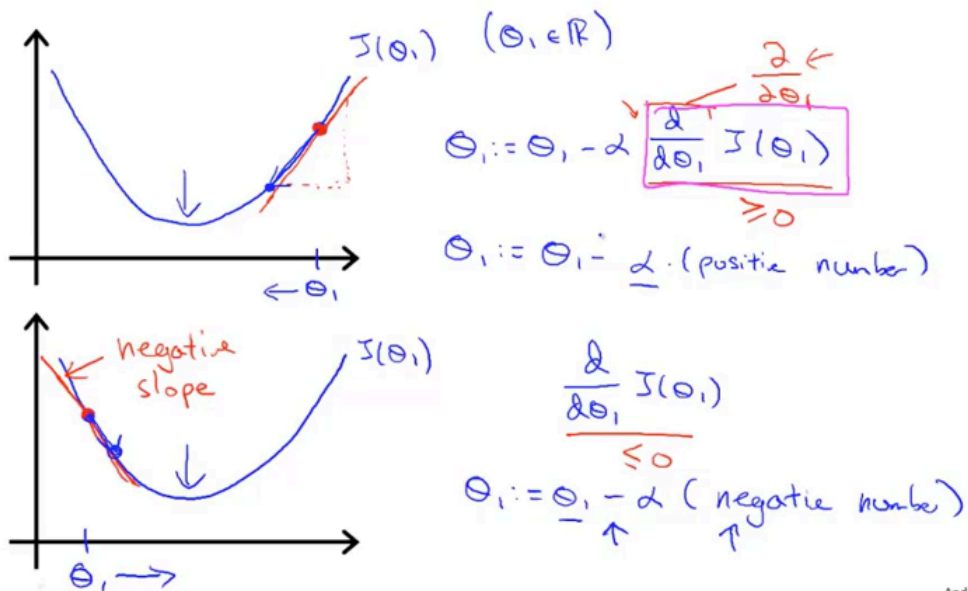
Andrew Ng

注：可采用梯度下降算法寻找最低点（即最小损失函数）



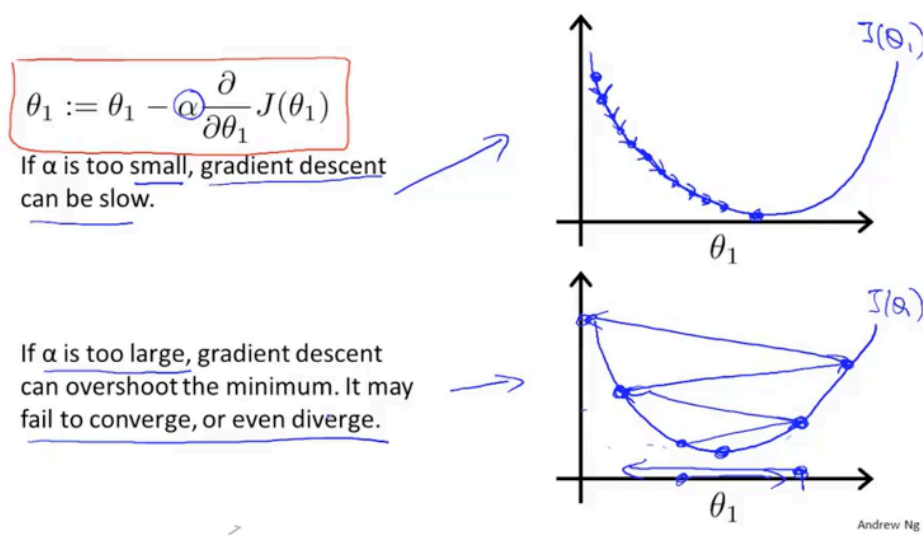
Andrew Ng

梯度下降算法寻找最小代价函数



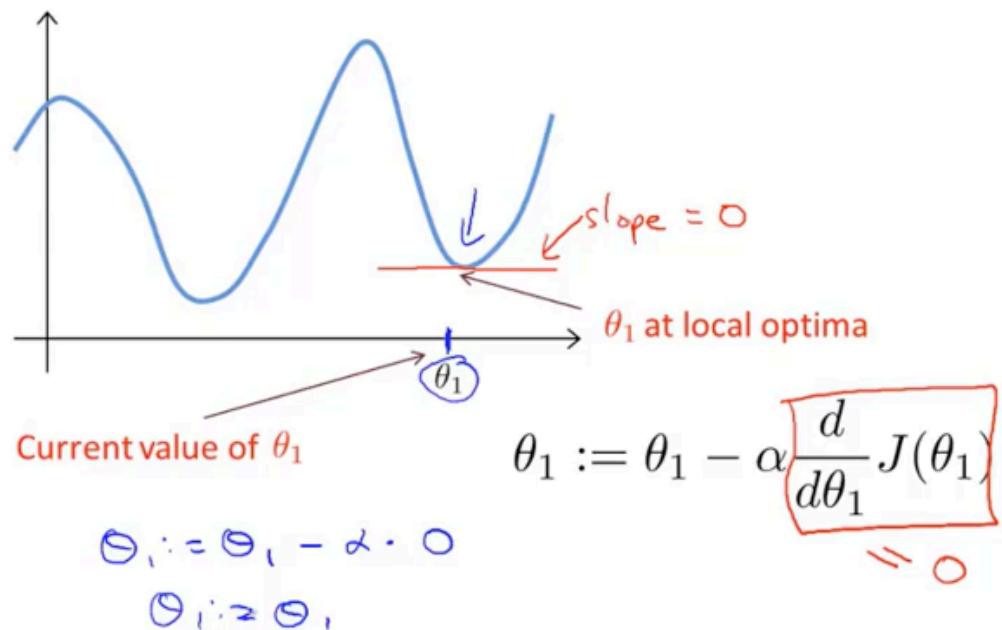
Andrew Ng

当a过小或者过大的时候对梯度下降的影响



Andrew Ng

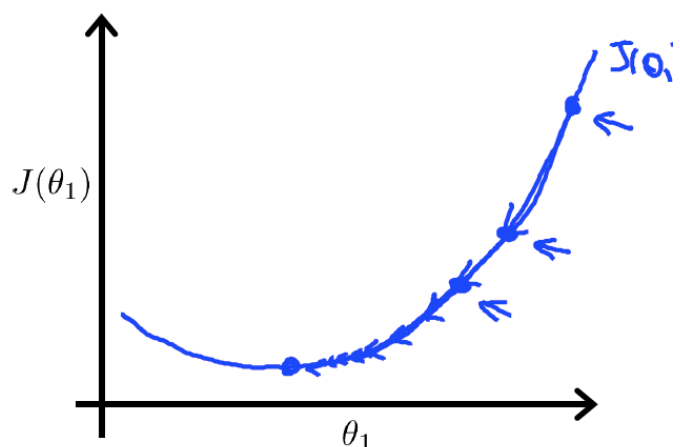
当代价函数处于局部最小时（即斜率为零）梯度下降不会改变原来参数的值



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

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

Linear Regression Model

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

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

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) &= \frac{2}{2\theta_j} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \\ &= \frac{2}{2\theta_j} \frac{1}{2m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2 \end{aligned}$$

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

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

Gradient descent algorithm

repeat until convergence {

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

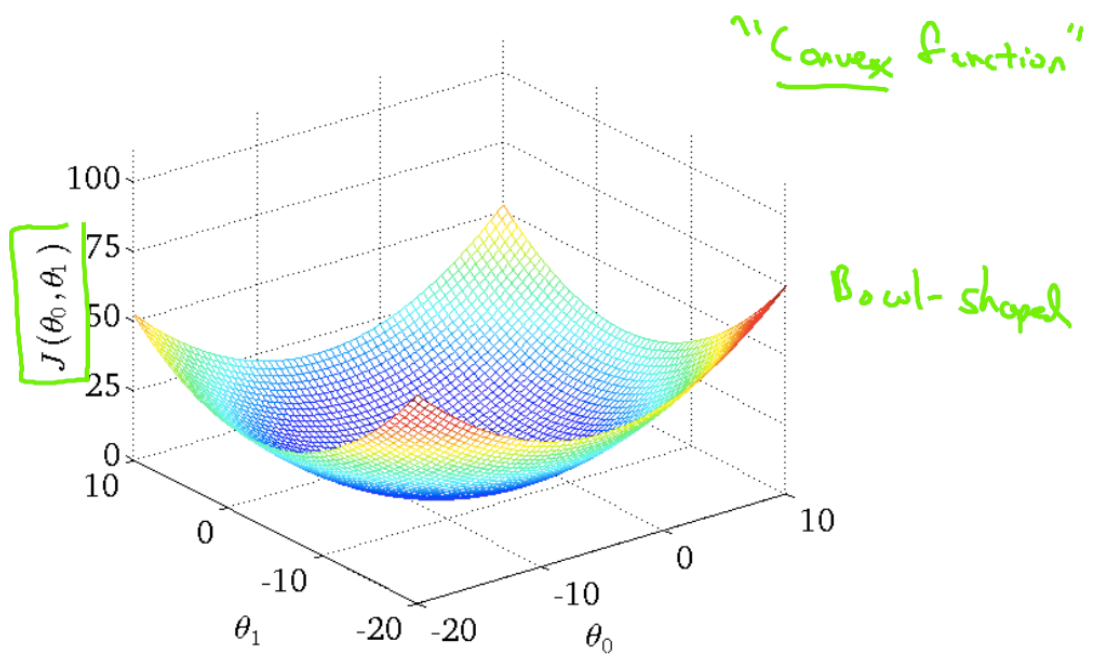
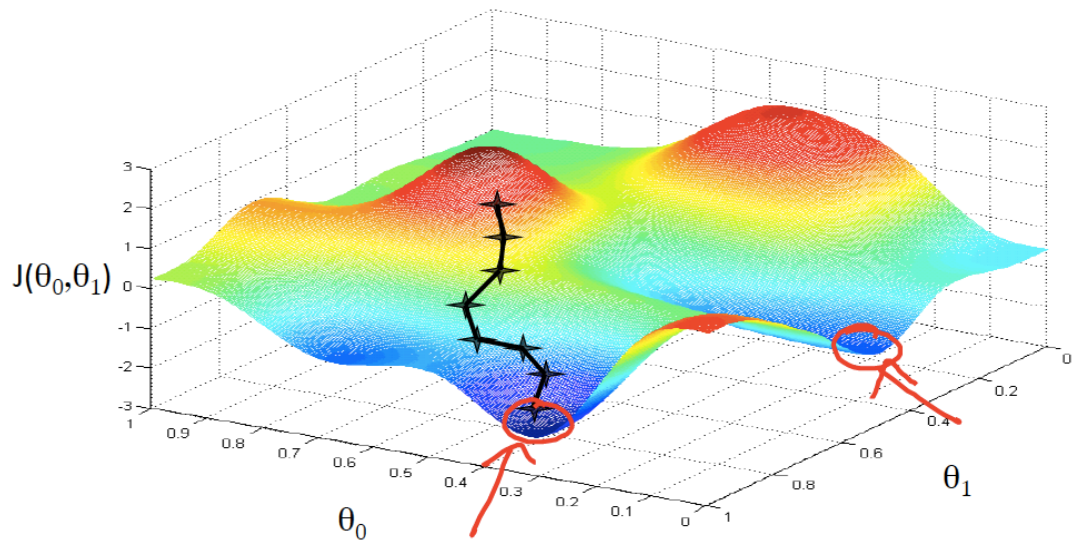
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

}

update
 θ_0 and θ_1
simultaneously

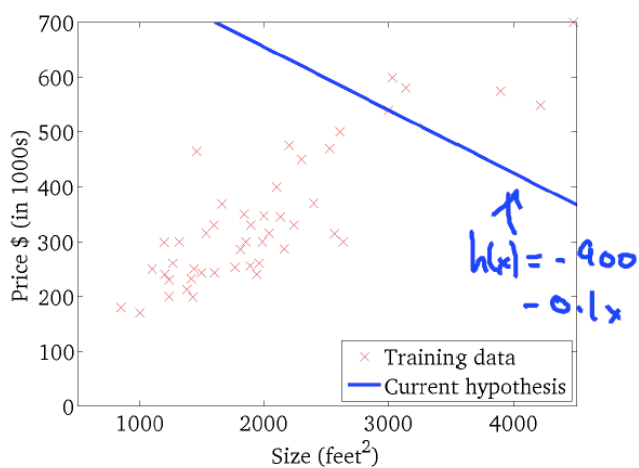
$$\frac{2}{2\theta_0} J(\theta_0, \theta_1)$$

$$\frac{2}{2\theta_1} J(\theta_0, \theta_1)$$



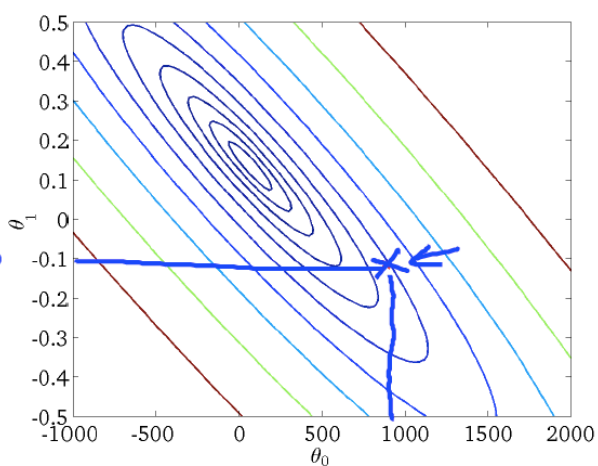
$$h_{\theta}(x)$$

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



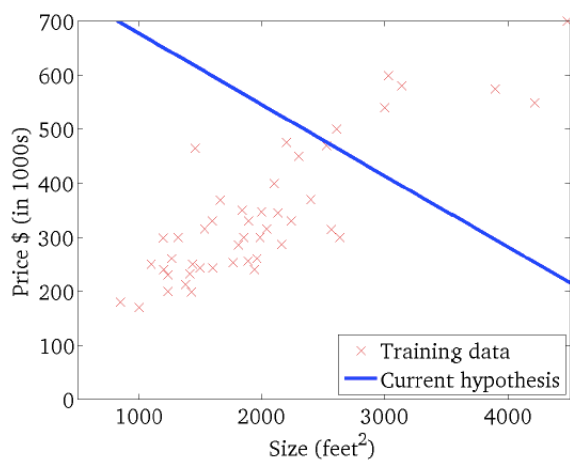
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



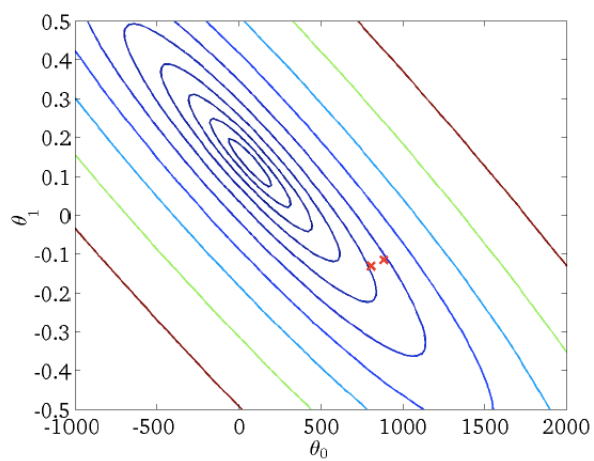
$$h_{\theta}(x)$$

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



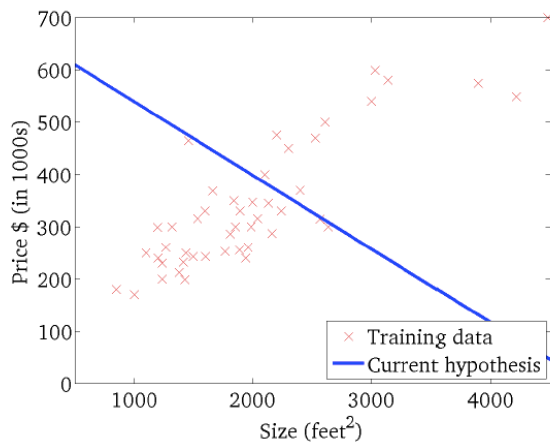
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



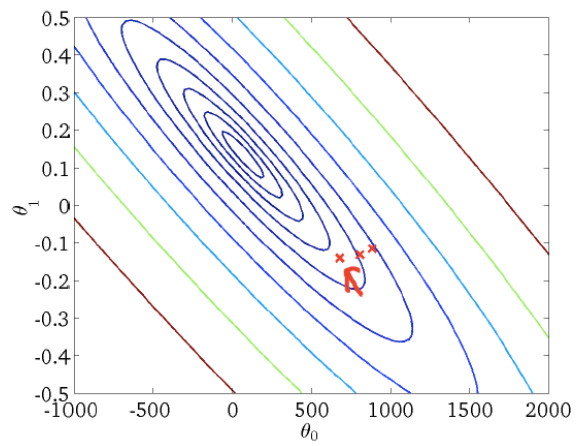
$$h_{\theta}(x)$$

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



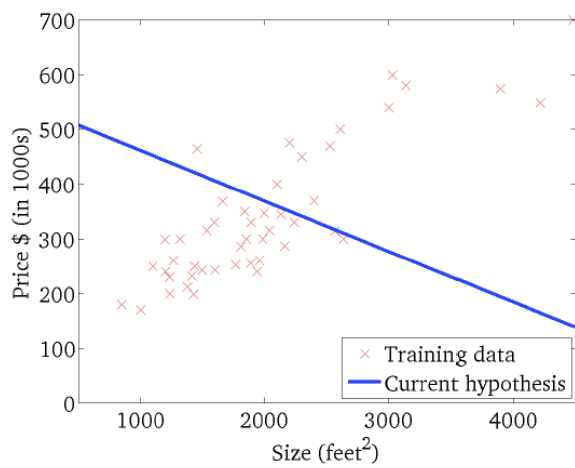
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



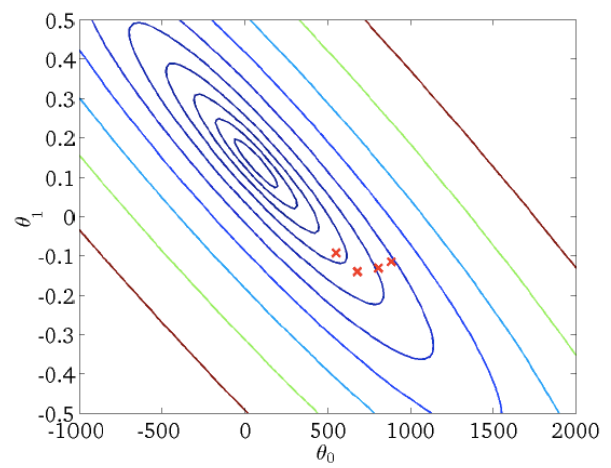
$$h_{\theta}(x)$$

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



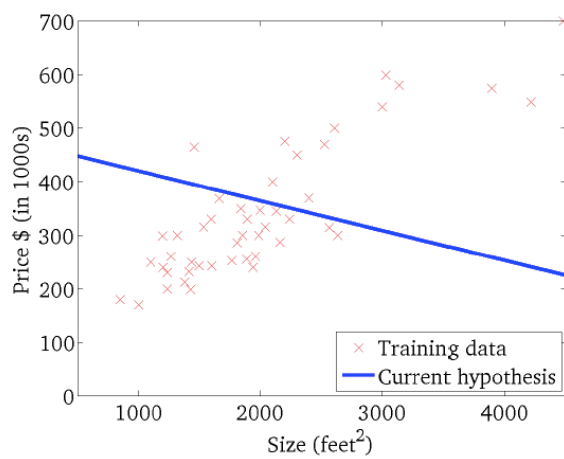
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



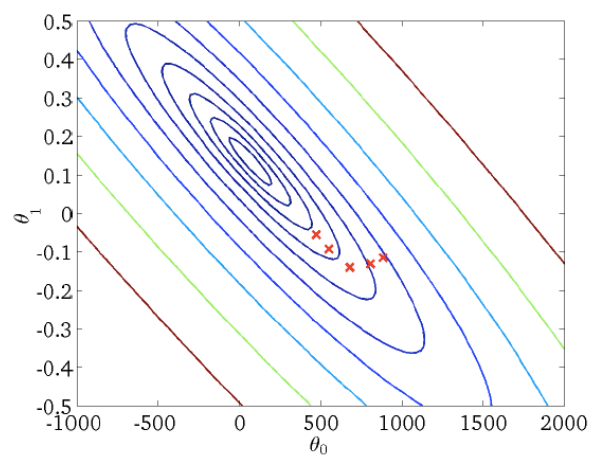
$$h_{\theta}(x)$$

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



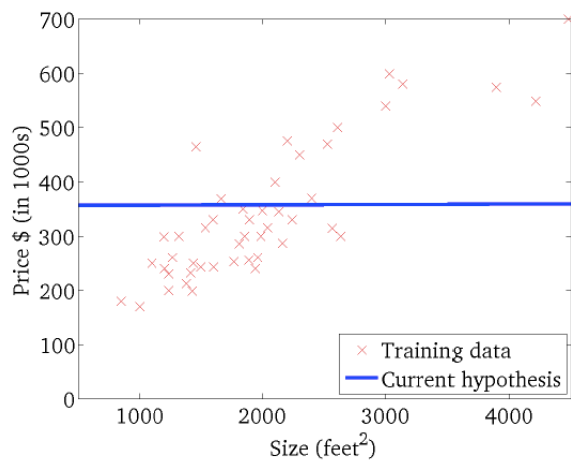
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



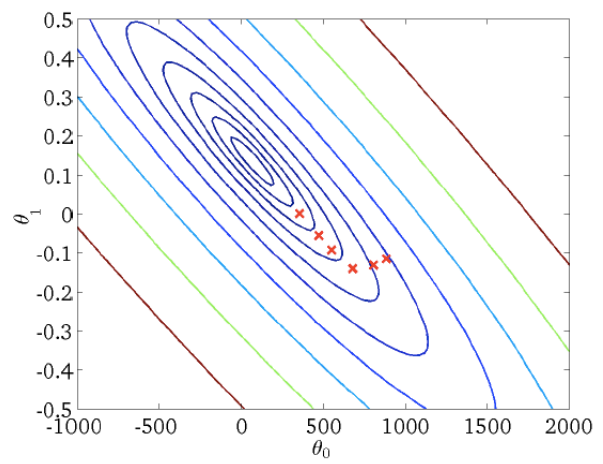
$$h_{\theta}(x)$$

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



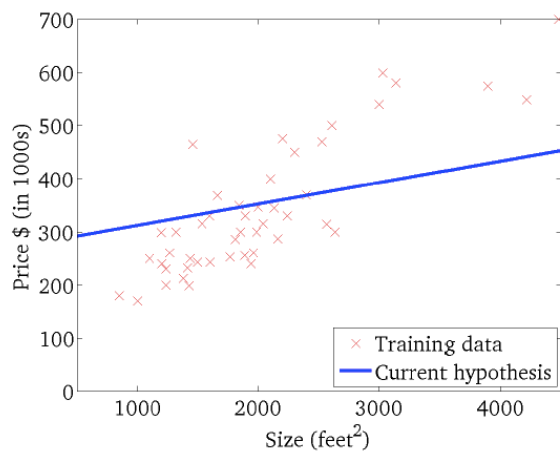
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



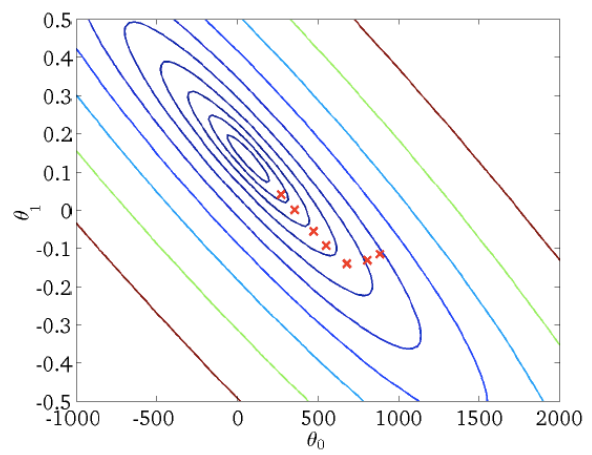
$$h_{\theta}(x)$$

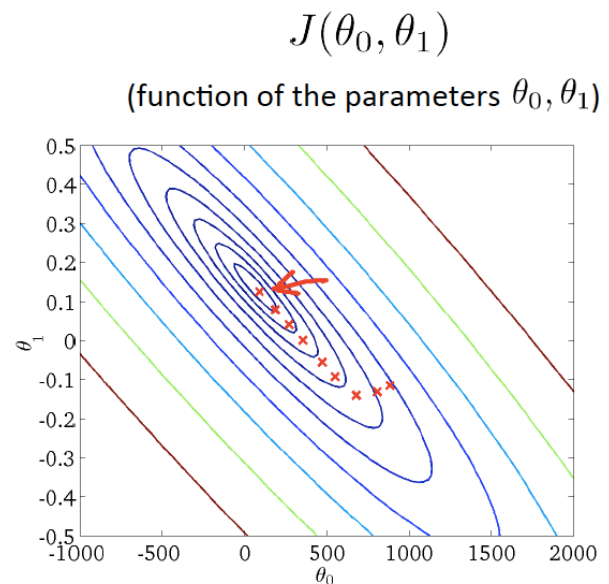
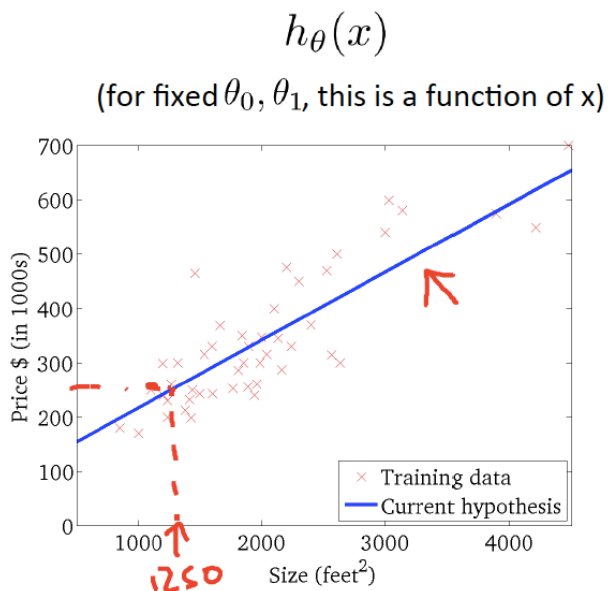
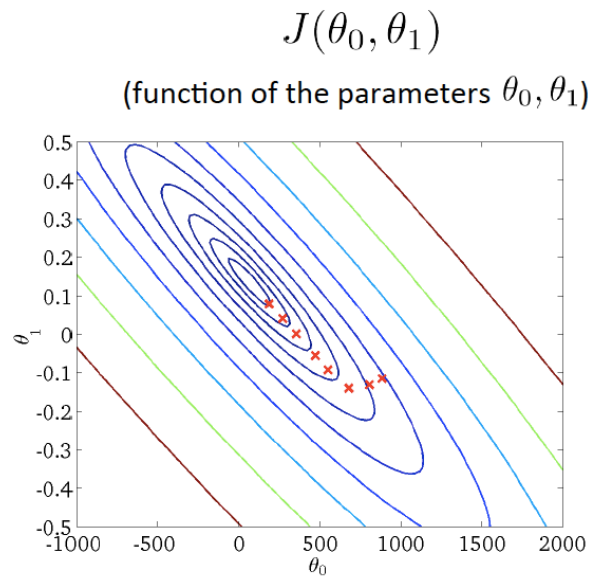
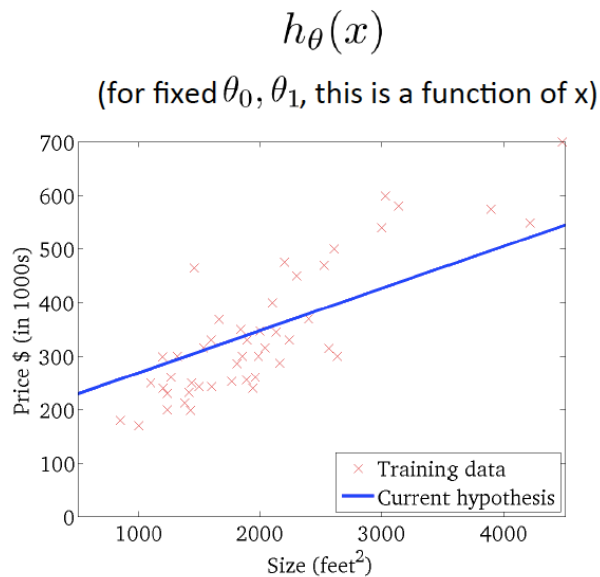
(for fixed θ_0, θ_1 , this is a function of x)



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

(function of the parameters θ_0, θ_1)





“Batch” Gradient Descent

“Batch”: Each step of gradient descent uses all the training examples.

$$\rightarrow \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$$

以上是'batch'梯度下降法的过程。