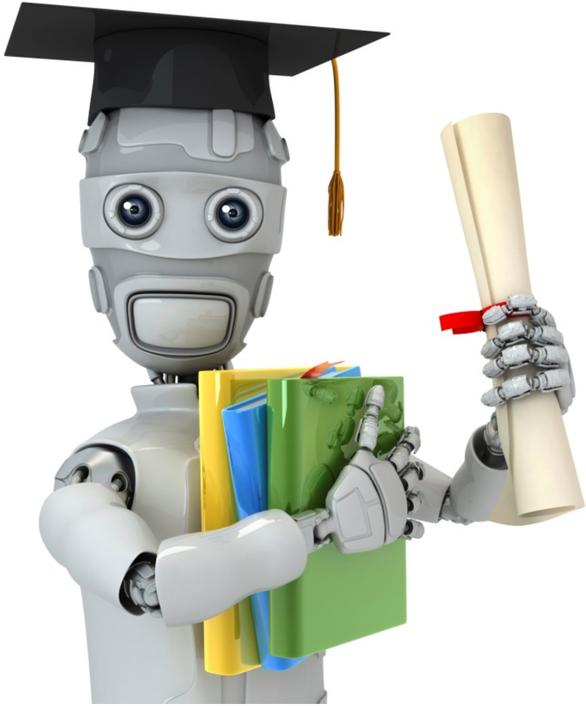


Machine Learning

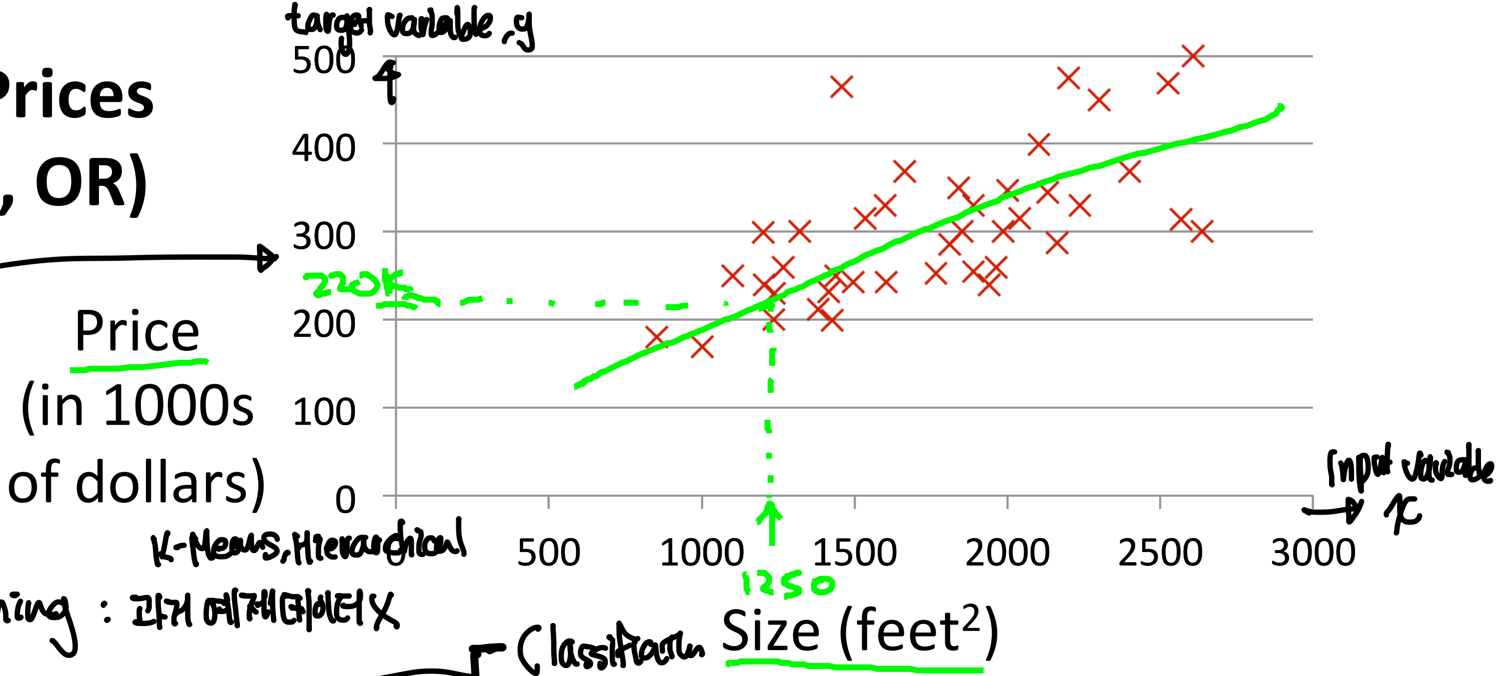
Linear regression with one variable

Model representation

Housing Prices (Portland, OR)

$y = ax + b$

입력 feature에 대한 연립방정식 찾기



Unsupervised learning : 과거 예제데이터 X

Supervised Learning

Given the "right answer" for each example in the data.

과거패턴에 대한 이해

Regression Problem

실수값 ≒ Classification 분류

Predict real-valued output

Classification: Discrete-valued output

Training set of housing prices (Portland, OR)

Size in feet^2 (x)	Price ($) in 1000's (y)
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

한번에 한번만

(x, y) - one training example

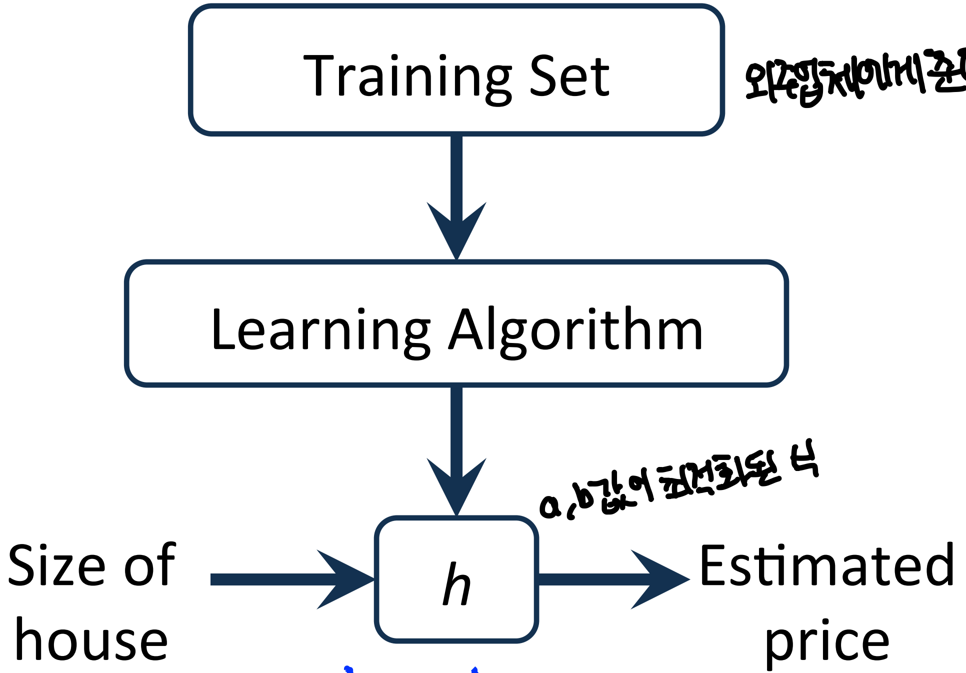
$(x^{(i)}, y^{(i)})$ - i^{th} training example

위첨자 : training data 에서 몇번째 변수가

$x^{(1)} = 2104$

$x^{(2)} = 1416$

$y^{(1)} = 460$



외접체에게준다

a,b값이 최적화된 식

x

hypothesis

가설 느낌

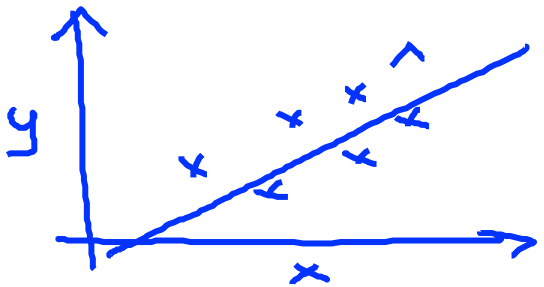
(estimated value of y)

h maps from x's to y's.

How do we represent h ?

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

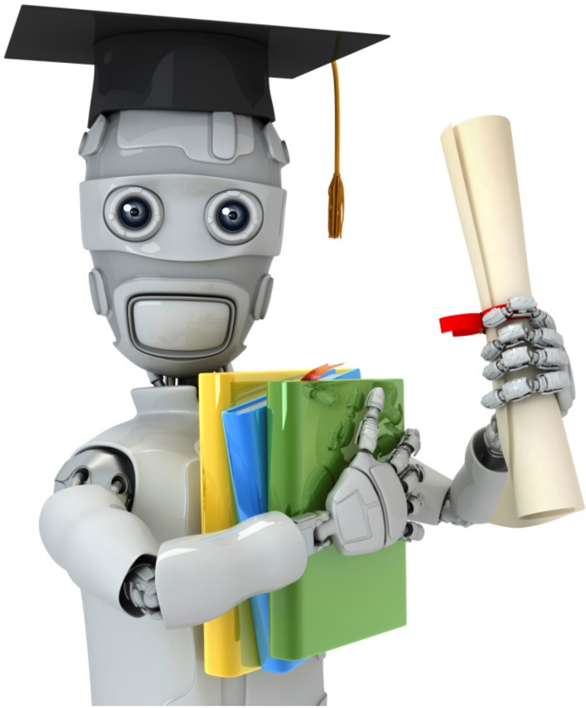
Shorthand: $h(x)$



$h(x) = \theta_0 + \theta_1 x$

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

Andrew Ng



Machine Learning

Linear regression with one variable

Cost function

Training Set

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

$m = 47$

$y = ax + b$

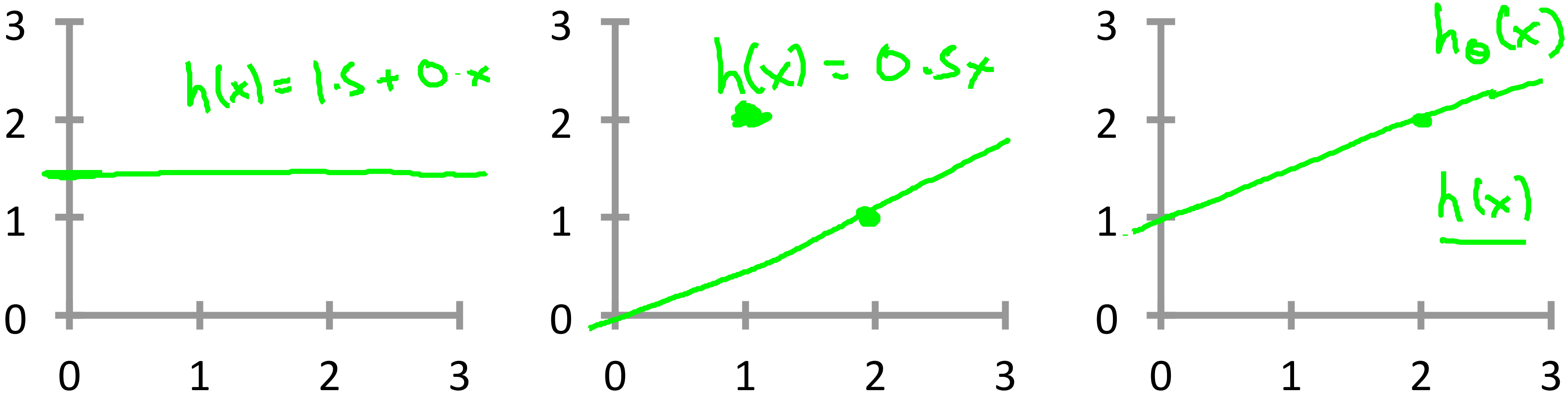
Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

θ_i's: Parameters

How to choose θ_i's ?

파라미터

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

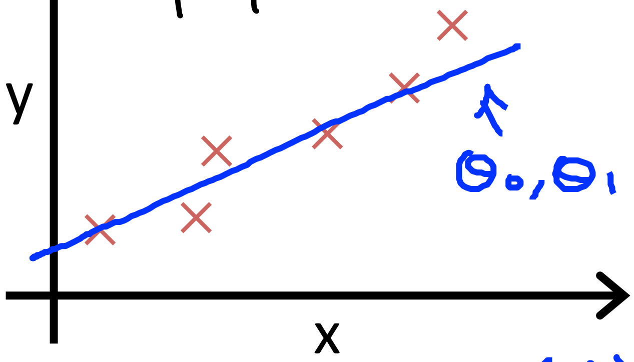


$\theta_0 = 1.5$
$\theta_1 = 0$

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

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

- 47개의 training set - 47개의 방정식
- training set > 파라미터의 수 ~ 일반적인 상황 = 해가 없다
- ~~차와의~~ 차이들을 모두 더해 ~~작은다~~ 값을 찾는다



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

m ~ # training examples

예측값 (미정값)

주어진값 (실제값)

Cost Function 비용최소화대상

$\frac{1}{2 \times 47}$ ~ 평균적인 개념

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

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

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

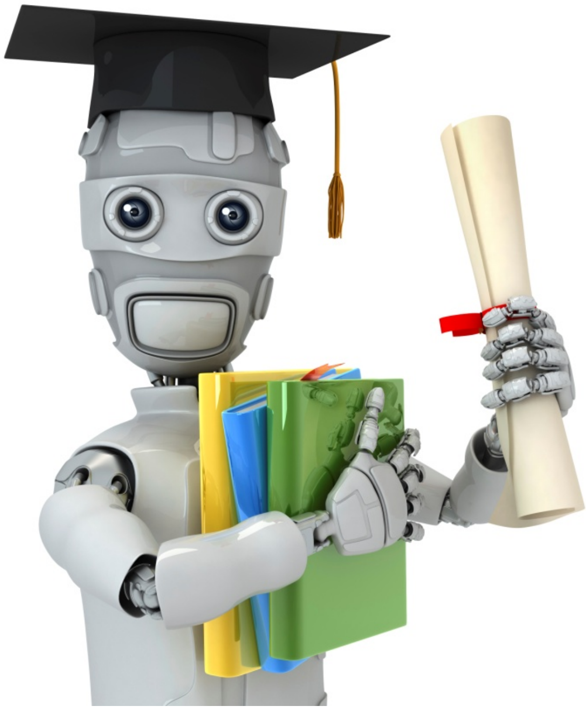
y절편 기울기

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

x, y

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

Squared error function (Objective func)



Machine Learning

Linear regression with one variable

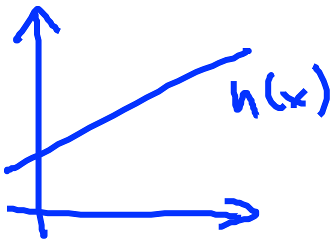
Cost function intuition I

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$



Cost Function:

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

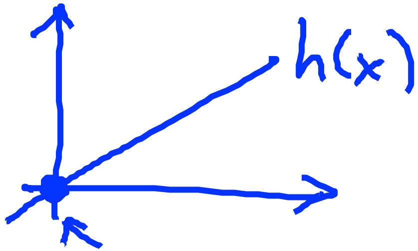
Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Simplified

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

$\theta_0 = 0$

$$\theta_1$$



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

$\theta, x^{(i)}$

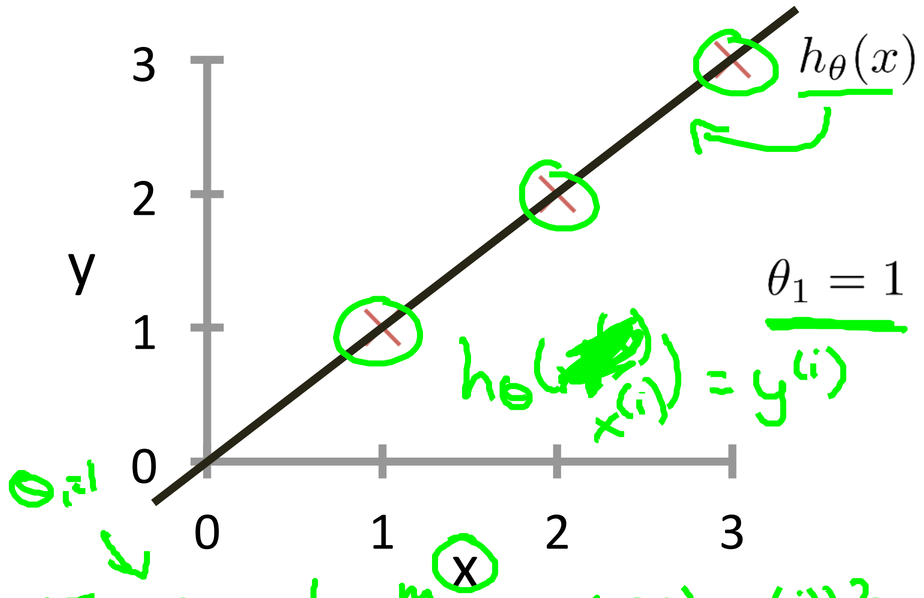
$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$

Andrew Ng

$\rightarrow h_\theta(x)$

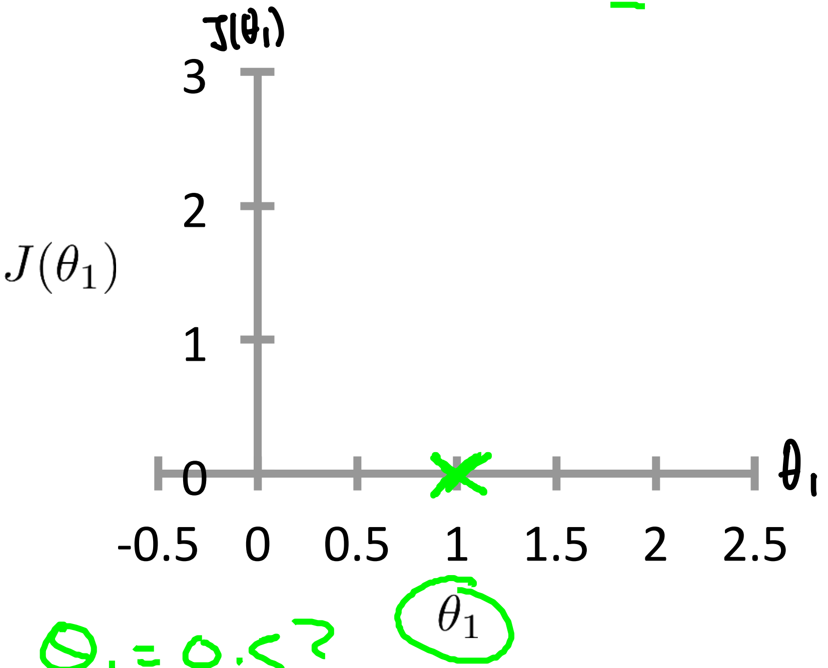
m=3 데이터셋

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



$\rightarrow J(\theta_1)$, cost function

(function of the parameter θ_1)



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

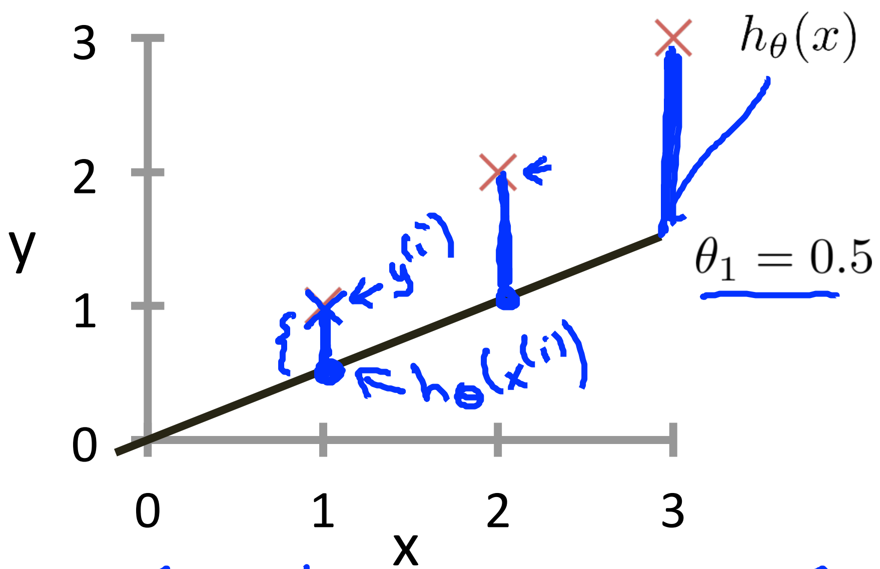
$$= \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_1 x^{(i)} - y^{(i)}\right)^2 = \frac{1}{2m}\left(0^2+0^2+0^2\right) = 0^2$$

$\theta_1 = 0.5$?

$J(1) = 0$

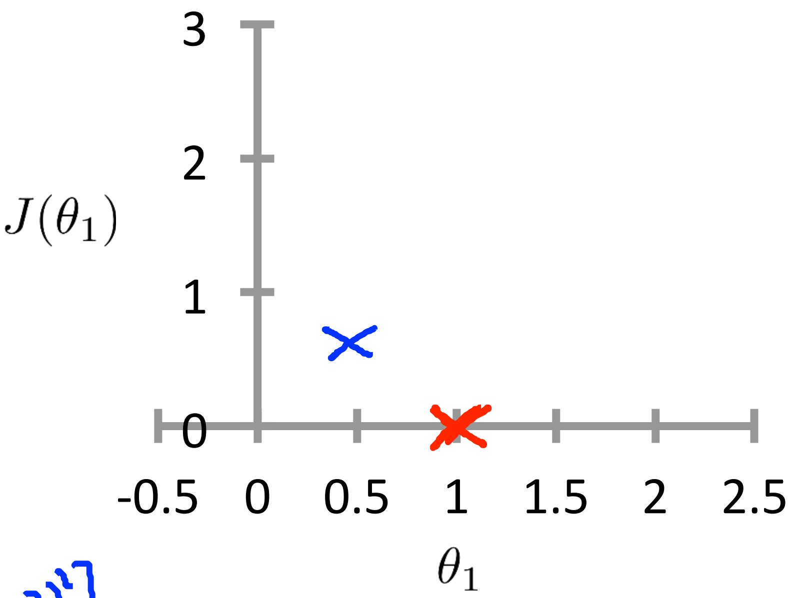
$h_\theta(x)$

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



$J(\theta_1)$

(function of the parameter θ_1)



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

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

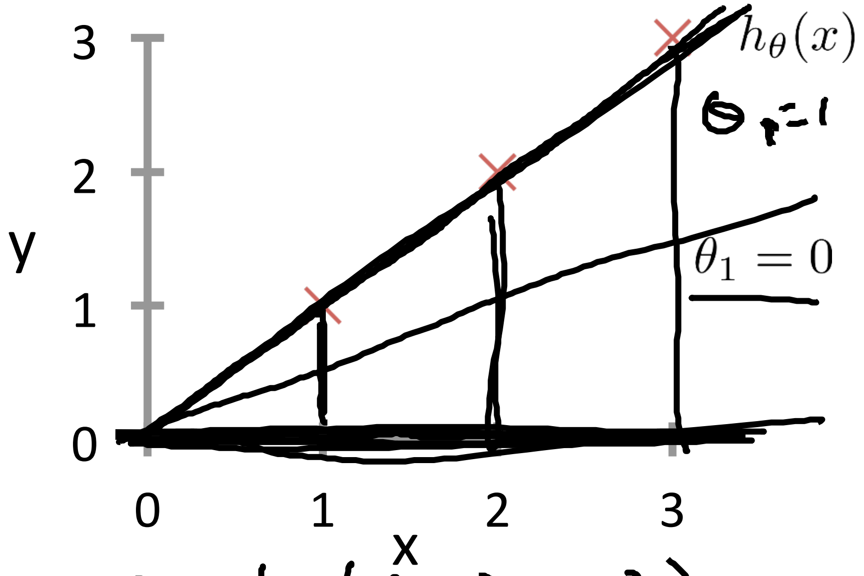
$\theta_1 = 0$?

$J(0) = $?

Andrew Ng

$h_\theta(x)$

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

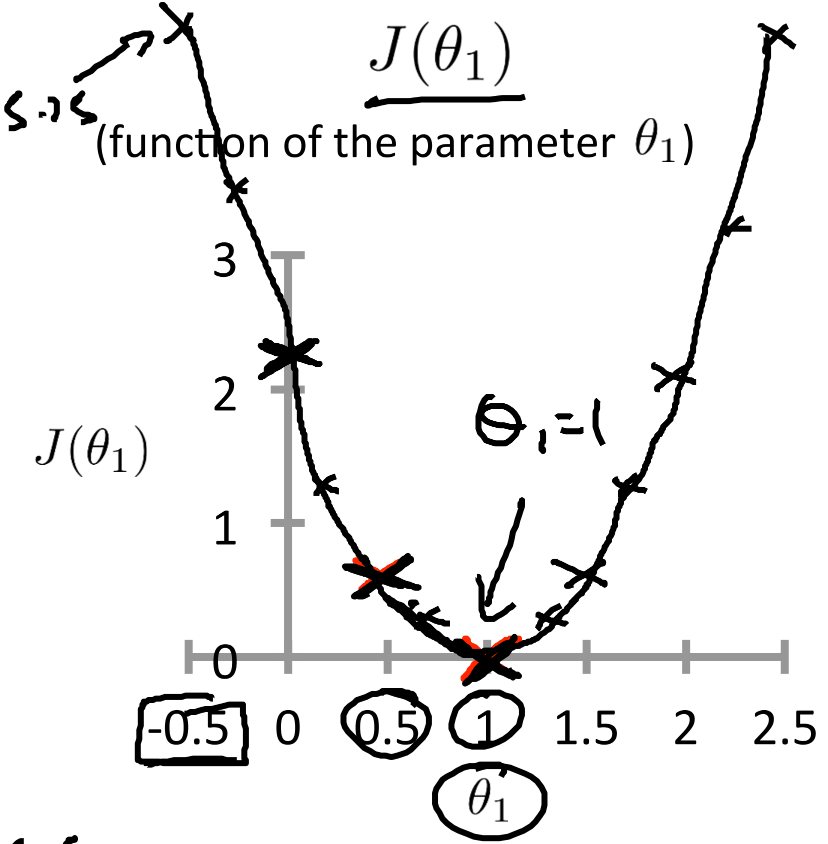


$$J(0) = \frac{1}{2m}(1^2 + 2^2 + 3^2)$$

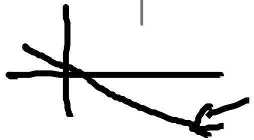
$$= \frac{1}{6} \cdot 14 \approx 2.3$$

$J(\theta_1)$

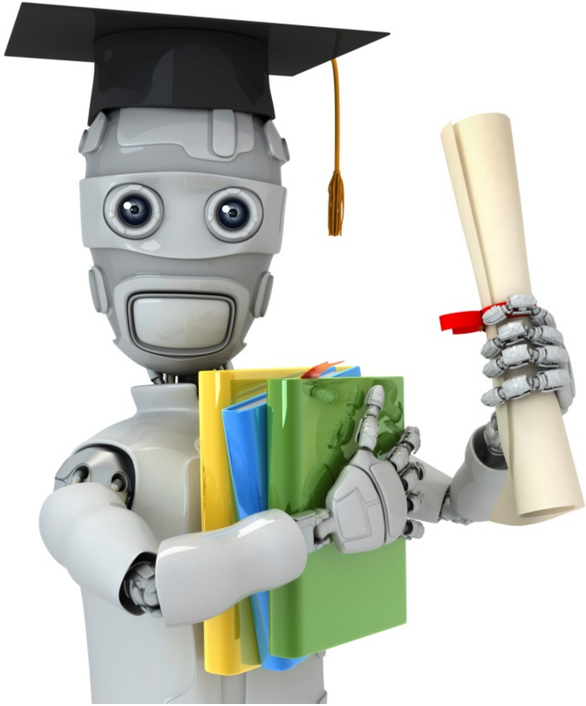
(function of the parameter θ_1)



$h(x) = -0.5x$



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



Machine Learning

Linear regression with one variable

Cost function intuition II

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

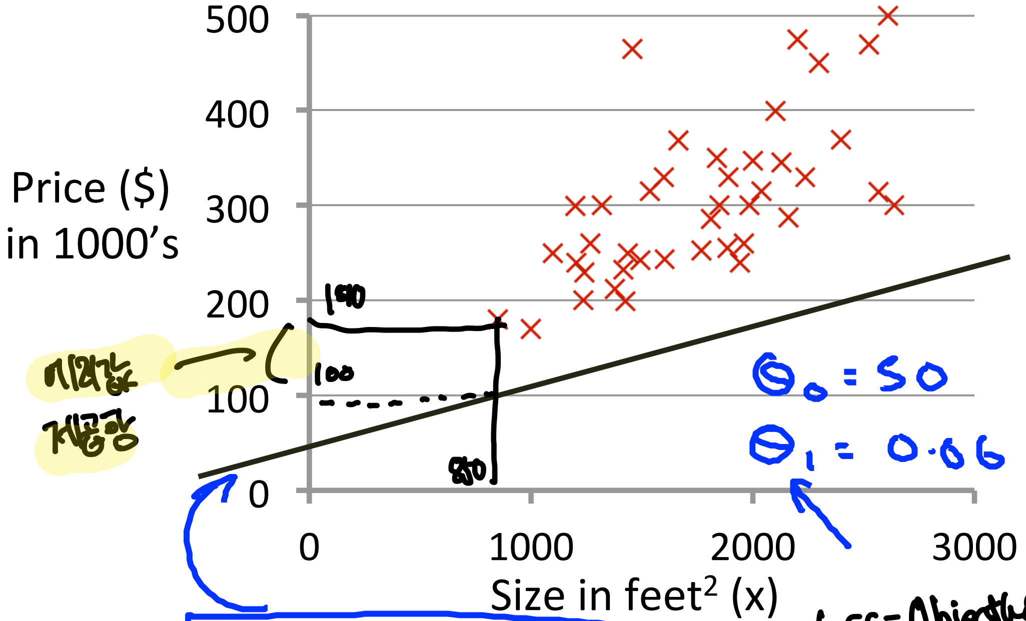
Parameters: θ_0, θ_1

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)



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

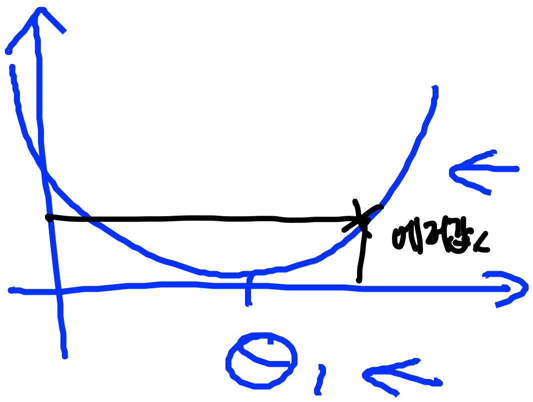
$\theta_0 = 50$

$\theta_1 = 0.06$

loss=Objective function

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



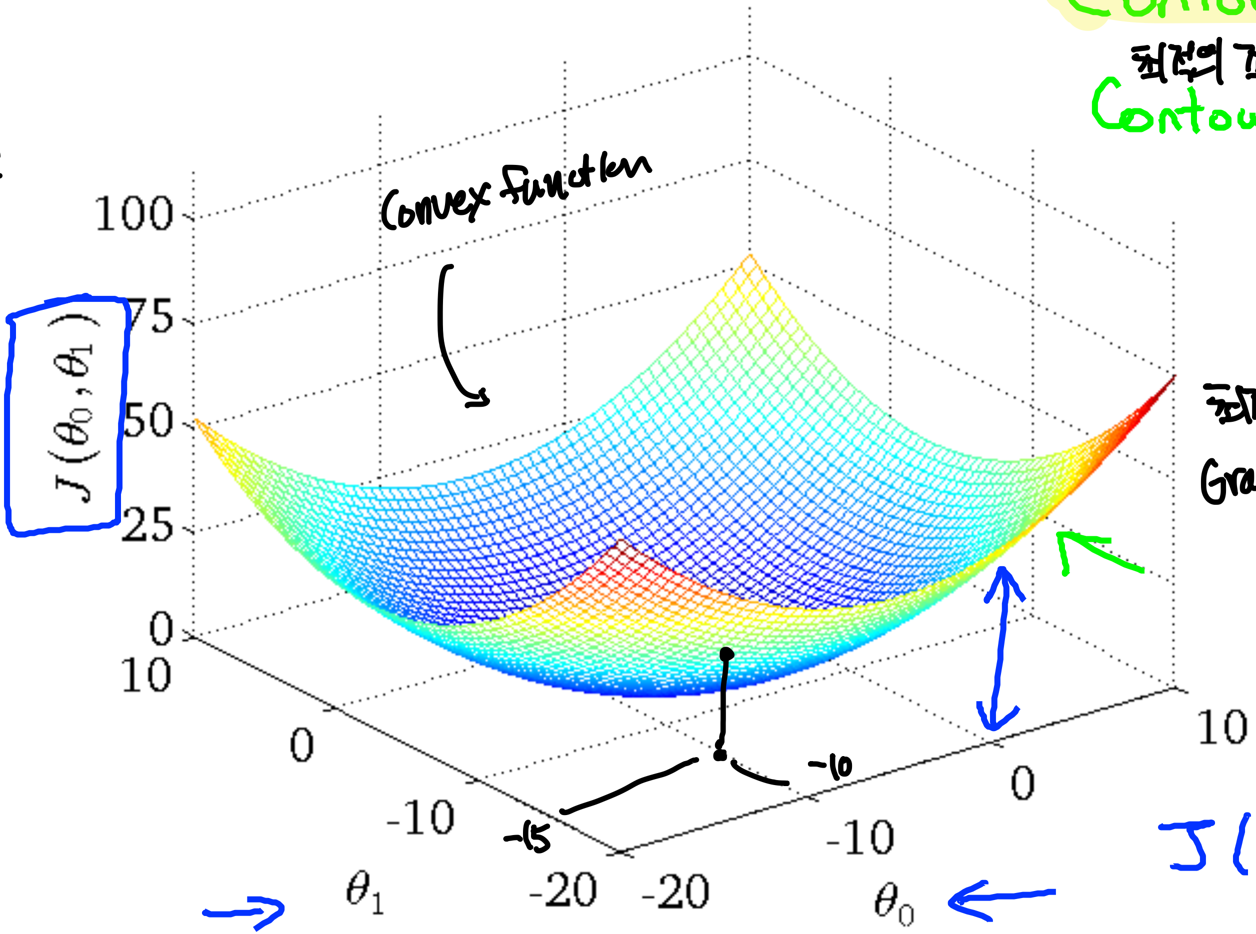
$$\boxed{\theta_0, \theta_1}$$

Contour plots

최적의 조합을 찾는것 ~ optimal

Contour figures -

이차원적인 입력값



최적화 ~ Optimization

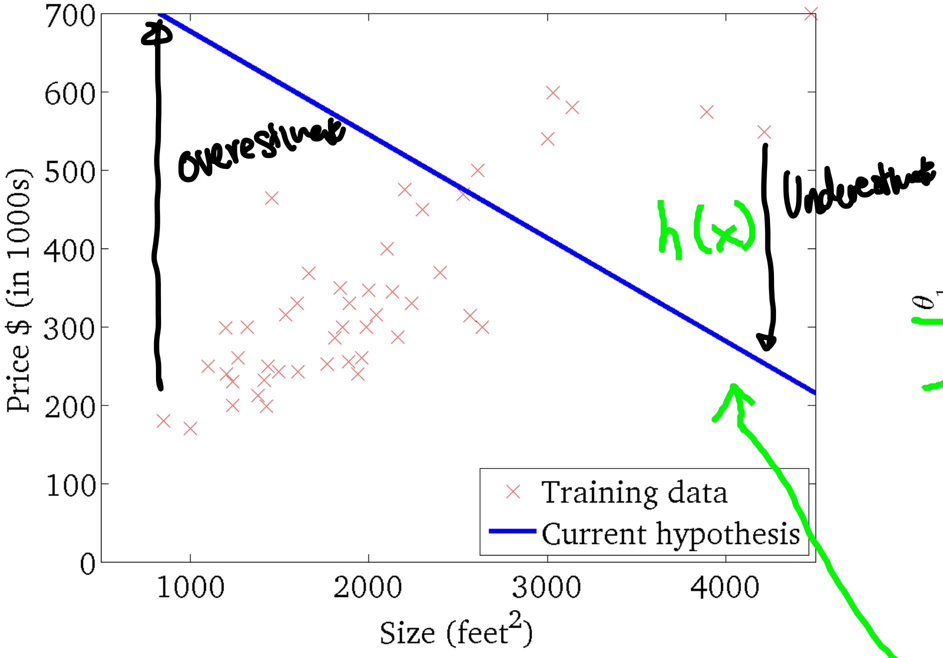
Gradient Descent

$J(\theta_0, \theta_1)$

Andrew Ng

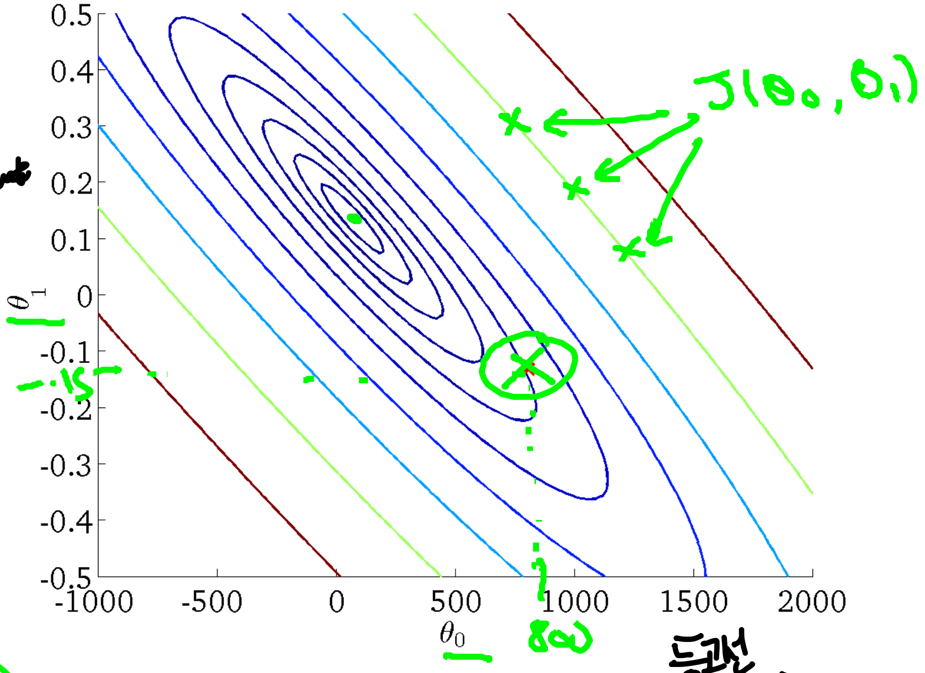
$h_\theta(x)$

(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)

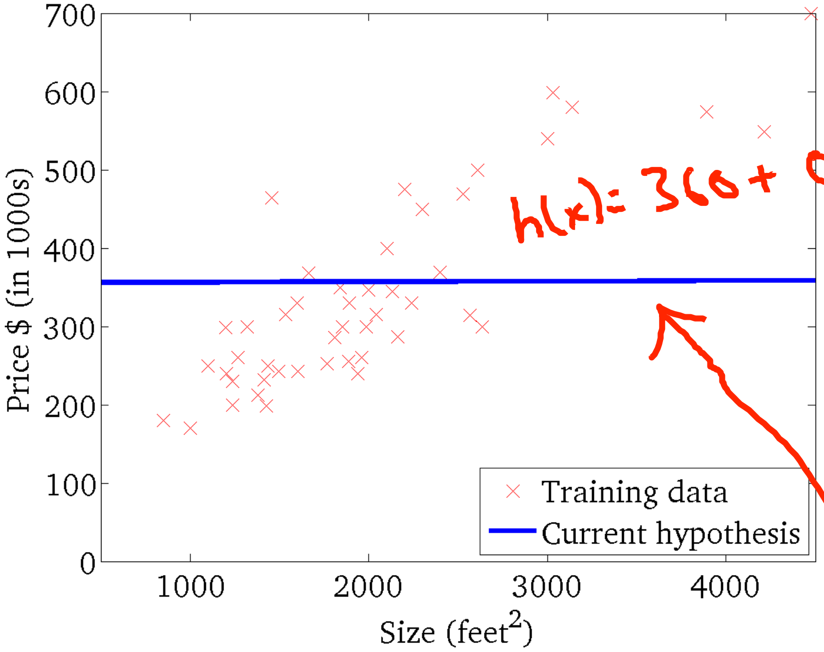


θ_0, θ_1

등고선
(contour)
≒ 3차원

$h_\theta(x)$

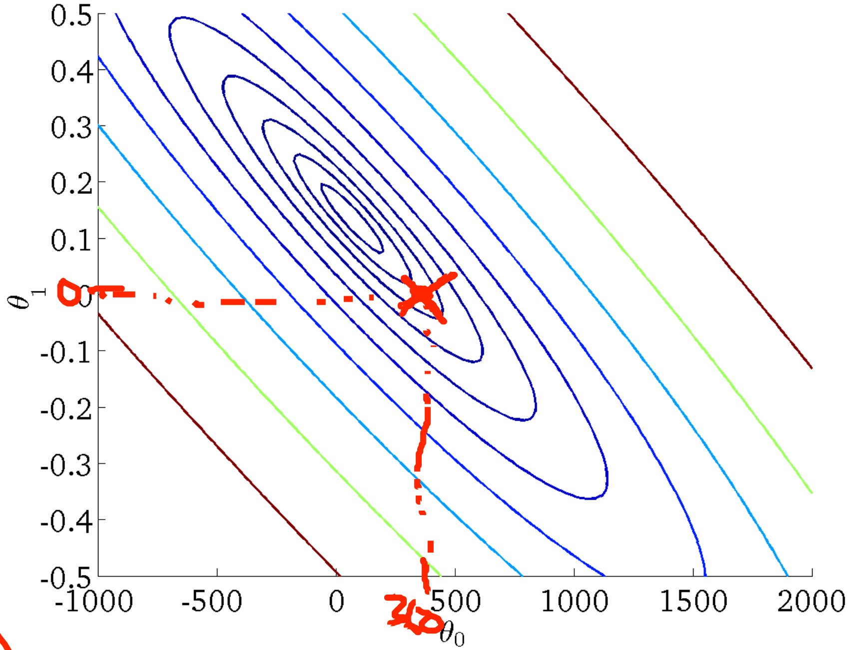
(for fixed θ_0, θ_1, this is a function of x)



$h(x) = 360 + 0 \cdot x$

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



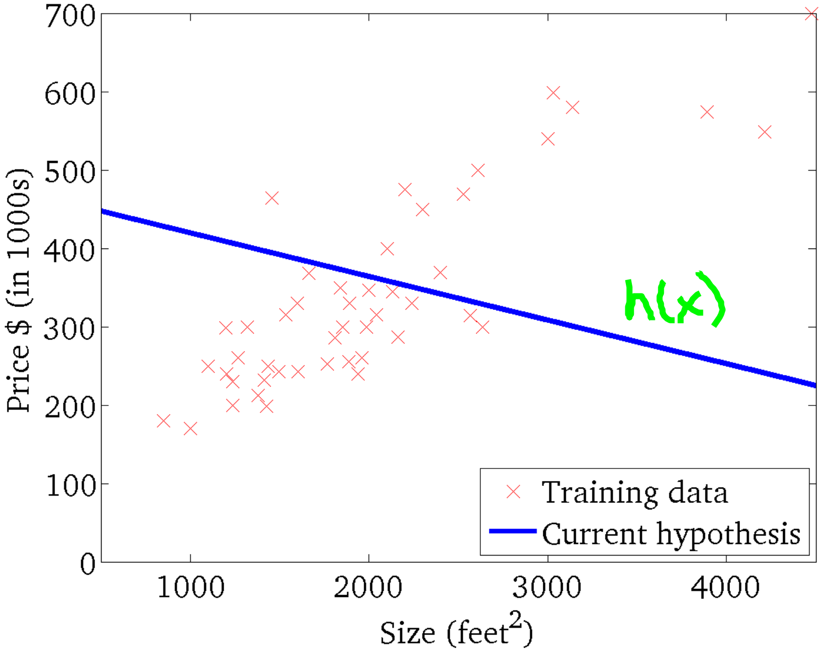
$\theta_0 = 360$

$\theta_1 = 0$

Andrew Ng

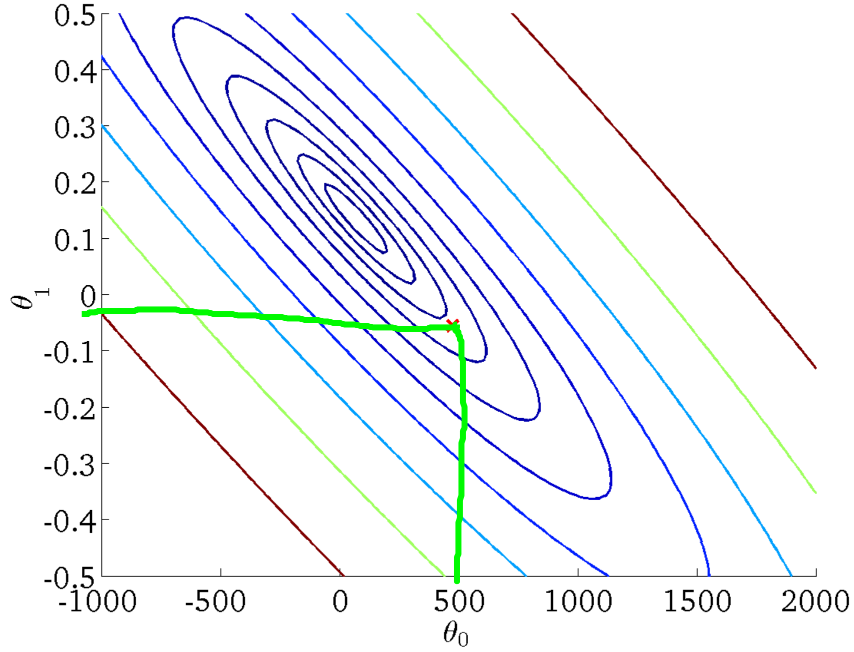
$h_\theta(x)$

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



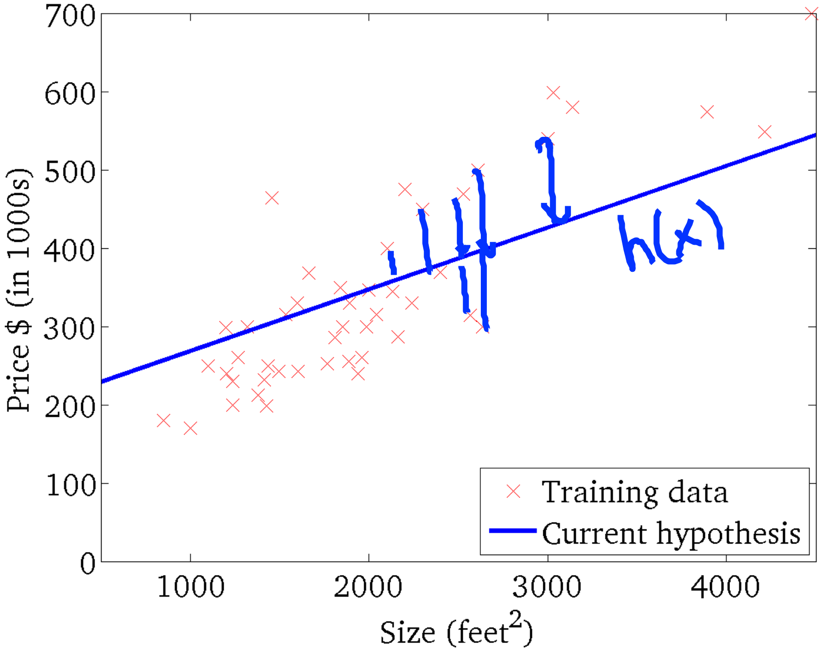
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



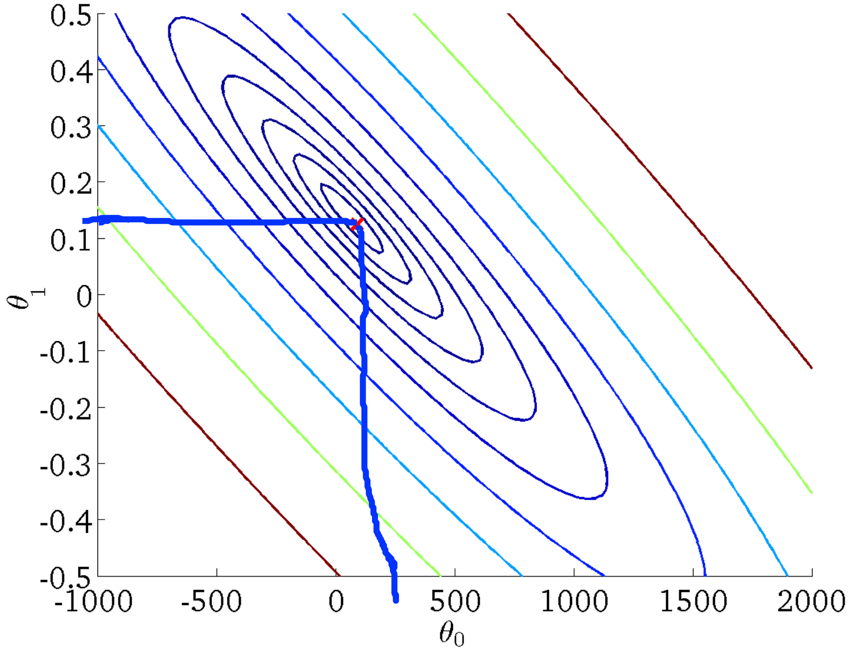
$h_\theta(x)$

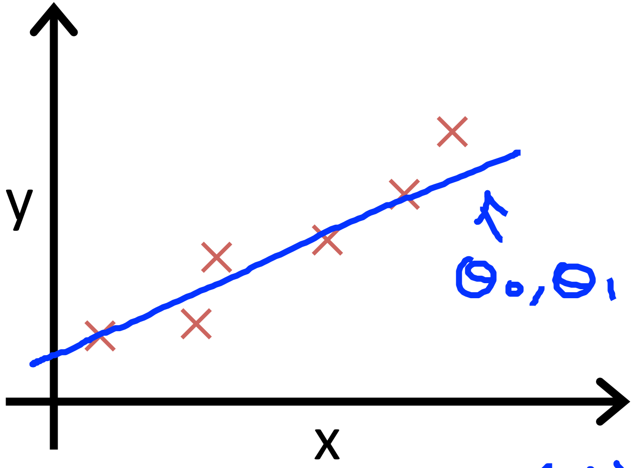
(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)





θ_0, θ_1

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

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

training examples

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

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

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

x, y

MAE / MSE

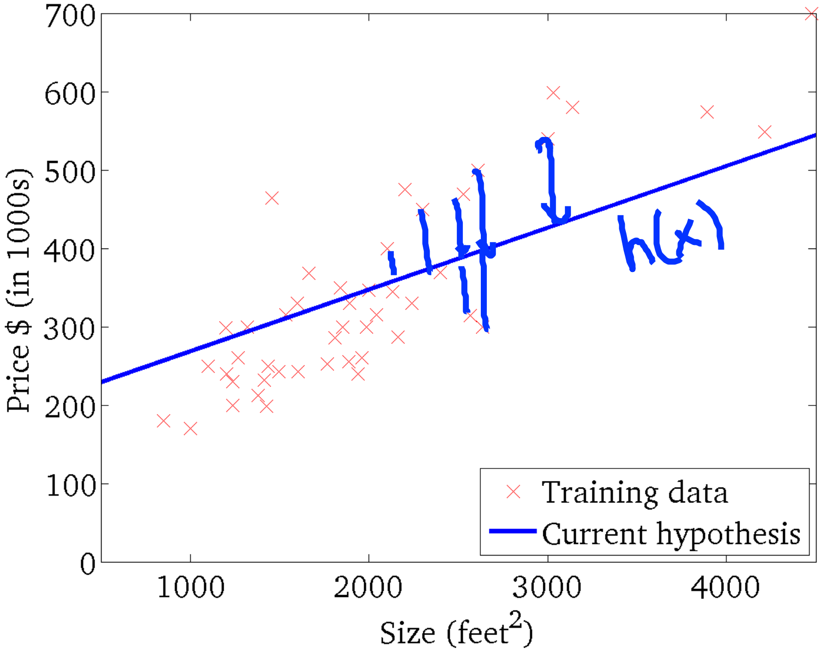
$$\underset{\theta_0, \theta_1}{\text{minimize}} \; \underbrace{J(\theta_0, \theta_1)}_{\text{Cost function}}$$

(제곱을 해서 Squared

Squred error fuction

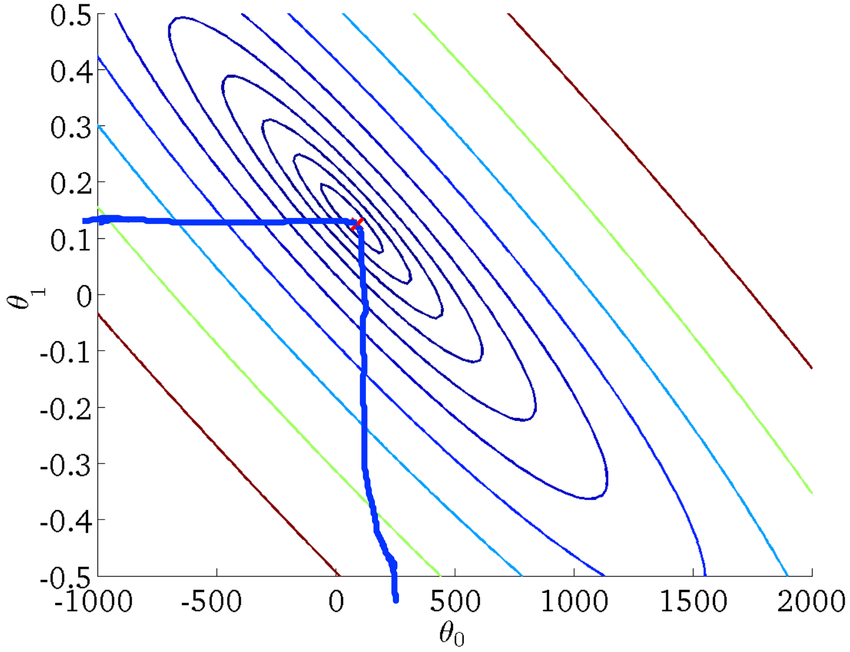
$h_\theta(x)$

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

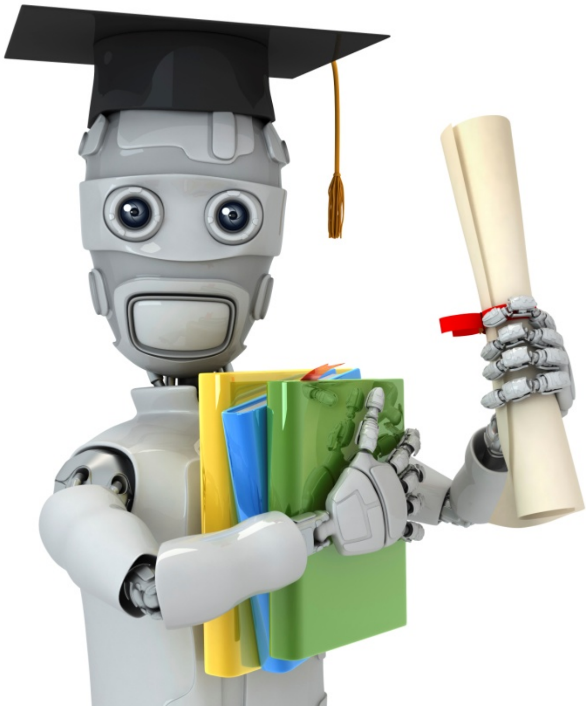


$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



Andrew Ng



Machine Learning

Linear regression with one variable

Gradient descent

Have some function $J(\theta_0, \theta_1)$ $J(\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$

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

Outline:

- Start with some θ_0, θ_1 (say $\theta_0 = 0, \theta_1 = 0$)
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

$y = 2x^2 - 4x + 5$ (e.g)

$J(\theta_1) = 2\theta_1^2 - 4\theta_1 + 5$

$= 2(\theta_1 - 1)^2 + 3$

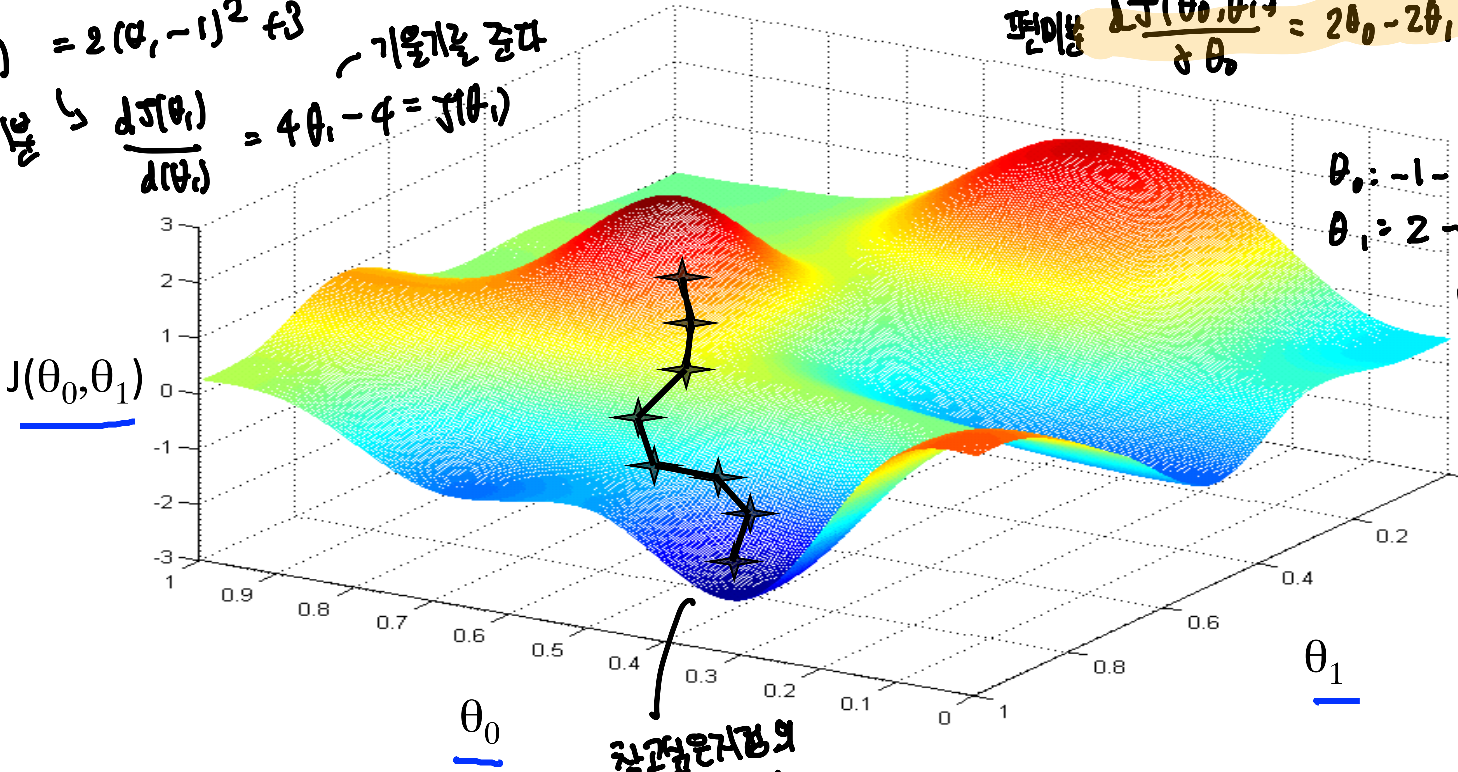
$Min(\theta_1)$

미분 ↳ $\frac{dJ(\theta_1)}{d(\theta_1)} = 4\theta_1 - 4 = J'(\theta_1)$ ← 기울기를 준다

두개의 변수: $J(\theta_0, \theta_1) = \theta_0^2 + 3\theta_1^2 - 2\theta_0\theta_1 + 4\theta_0 - 5\theta_1 + 3$

minimize $(\theta_0 = -1, \theta_1 = 2)$

편미분 $\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} = 2\theta_0 - 2\theta_1 + 4$ $\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} = 6\theta_1 - 2\theta_0 - 5$



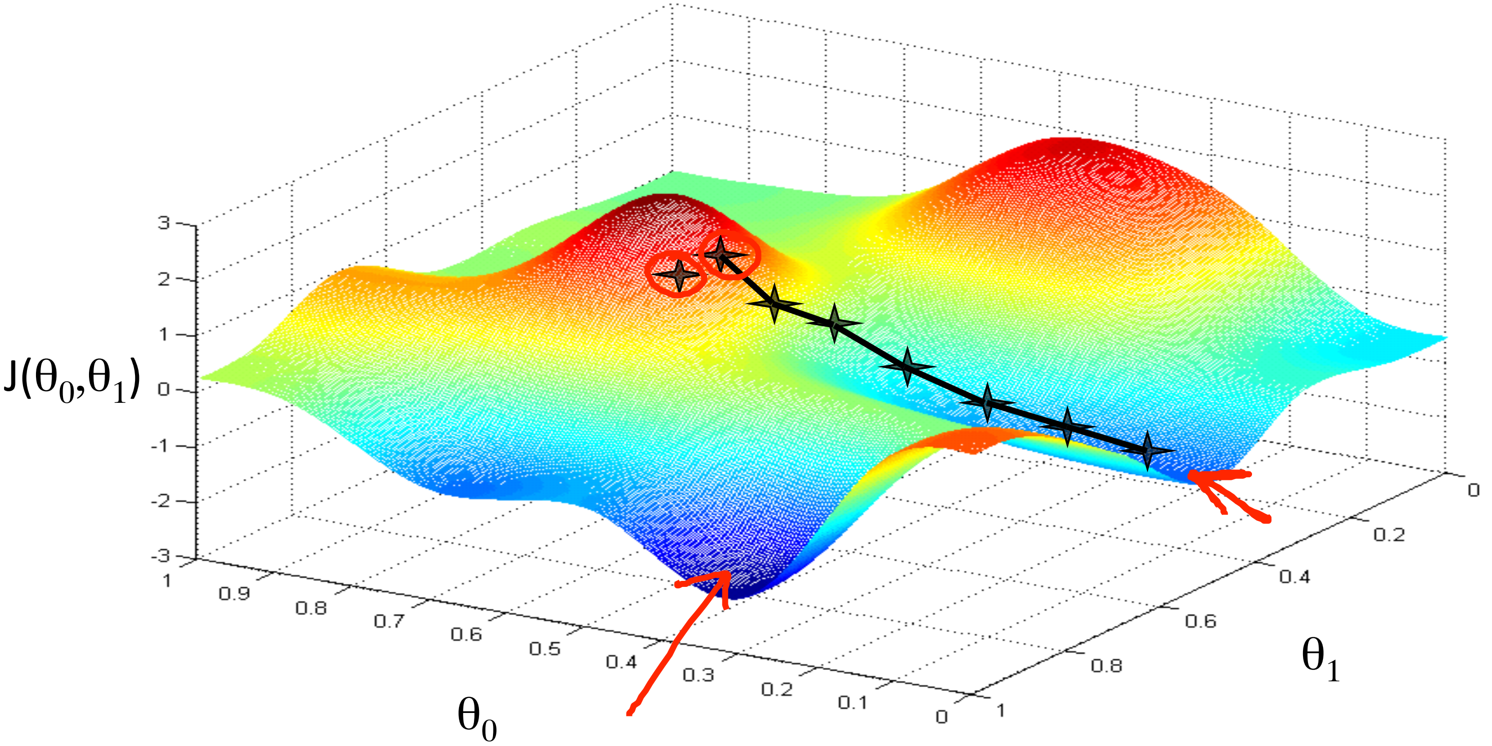
$\theta_0 : -1 - 0.1 \cdot (-2)$

$\theta_1 : 2 - 0.1(9)$

$\theta_0 = -0.8$

$\theta_1 = 1.1$

찾고싶은지점의 $\theta_0\ \theta_1$ 찾기



$J(\theta_0,\theta_1)$
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

Assignment

$a := b$

$a := a+1$

Truth assertion

$a = b$

$a = a+1$ ✗

θ_0, θ_1

repeat until convergence {

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

}

learning rate

Simultaneously update θ_0 and θ_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:

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