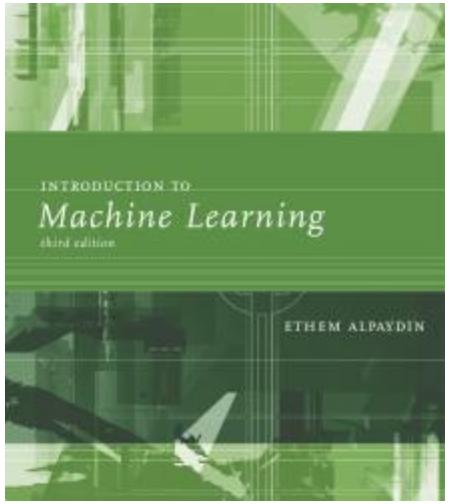


Linear Regression

Lecture notes by Kevyn Collins-Thompson
Applied Machine Learning (Coursera)

Lecture notes by Ethem Alpaydın
Introduction to Machine Learning (Boğaziçi Üniversitesi)

Lecture notes by Andrew NG
Machine Learning by Stanford University (Coursera)

Linear regression with one variable

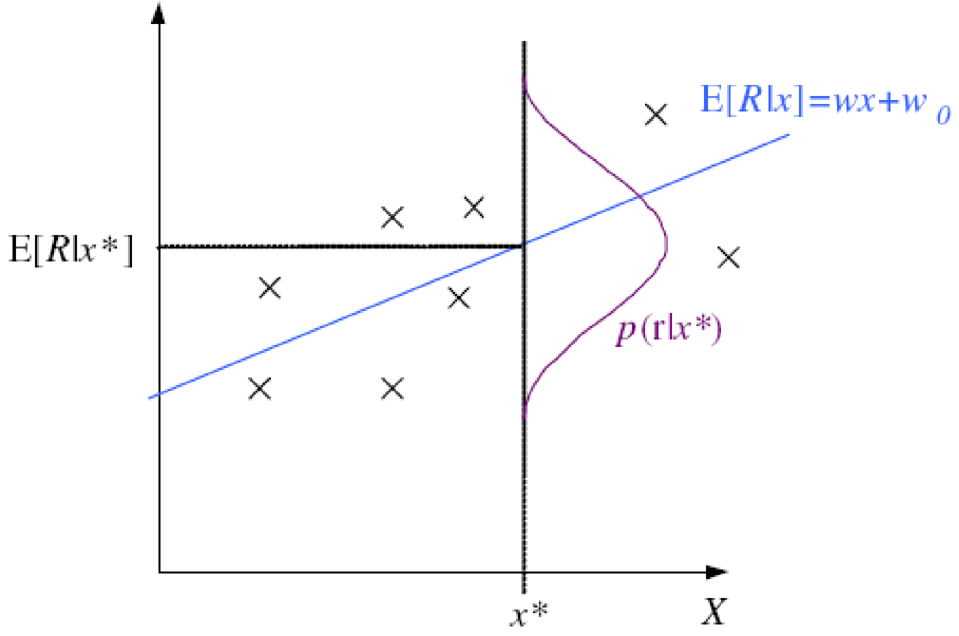
Regression

$$r = f(x) + \varepsilon$$

$$\text{estimator}: g(x \mid \theta)$$

$$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$$p(r \mid x) \sim \mathcal{N}(g(x \mid \theta), \sigma^2)$$



$$\mathcal{L}(\theta \mid \mathcal{X}) = \log \prod_{t=1}^{N} p(x^t, r^t)$$

$$= \log \prod_{t=1}^{N} p(r^t \mid x^t) + \log \prod_{t=1}^{N} p(x^t)$$

Regression: From LogL to Error

$$\mathcal{L}(\theta|\mathcal{X}) = \log \prod_{t=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\left[r^t - g\left(x^t|\theta\right)\right]^2}{2\sigma^2}\right]$$

$$= -N \log \sqrt{2\pi}\sigma - \frac{1}{2\sigma^2} \sum_{t=1}^{N} \left[r^t - g\left(x^t|\theta\right)\right]^2$$

$$E(\theta|\mathcal{X}) = \frac{1}{2} \sum_{t=1}^{N} \left[r^t - g\left(x^t|\theta\right)\right]^2$$

Most frequently used error function
$E = -logl$

θ minimize the error function are called the least squares estimates.

Linear Regression

$$g\left(x^t \mid w_1, w_0\right) = w_1 x^t + w_0$$

$$\sum_t r^t = N w_0 + w_1 \sum_t x^t$$

$$\sum_t r^t x^t = w_0 \sum_t x^t + w_1 \sum_t \left(x^t\right)^2$$

$$\mathbf{A} = \begin{bmatrix} N & \sum_t x^t \\ \sum_t x^t & \sum_t \left(x^t\right)^2 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} \sum_t r^t \\ \sum_t r^t x^t \end{bmatrix} \Rightarrow \mathbf{w} = \mathbf{A}^{-1}\mathbf{y}$$

Gradient Descent

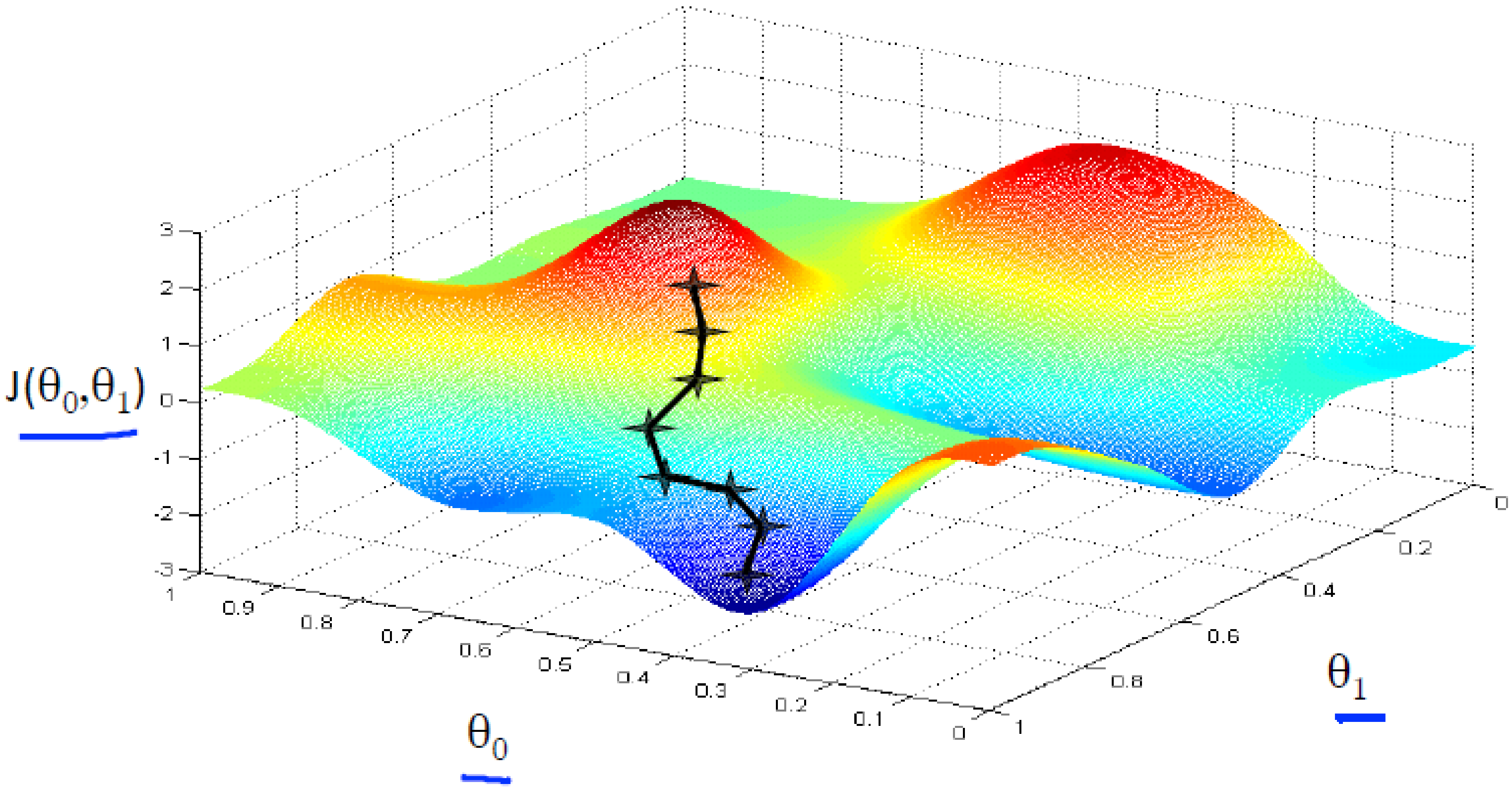
Have some function $J(\theta_0, \theta_1)$

Want $\min\limits_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline

Start with some θ_0, θ_1

Keep chaning θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum



$J(\theta_0,\theta_1)$
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

```
repeat until convergence {
```

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

```
                    (for j=0 and j=1)
}
```

Simultaneously update

Gradient Descent

Simultaneously update

$$temp0 \coloneqq \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$temp1 \coloneqq \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 \ \coloneqq temp0$$

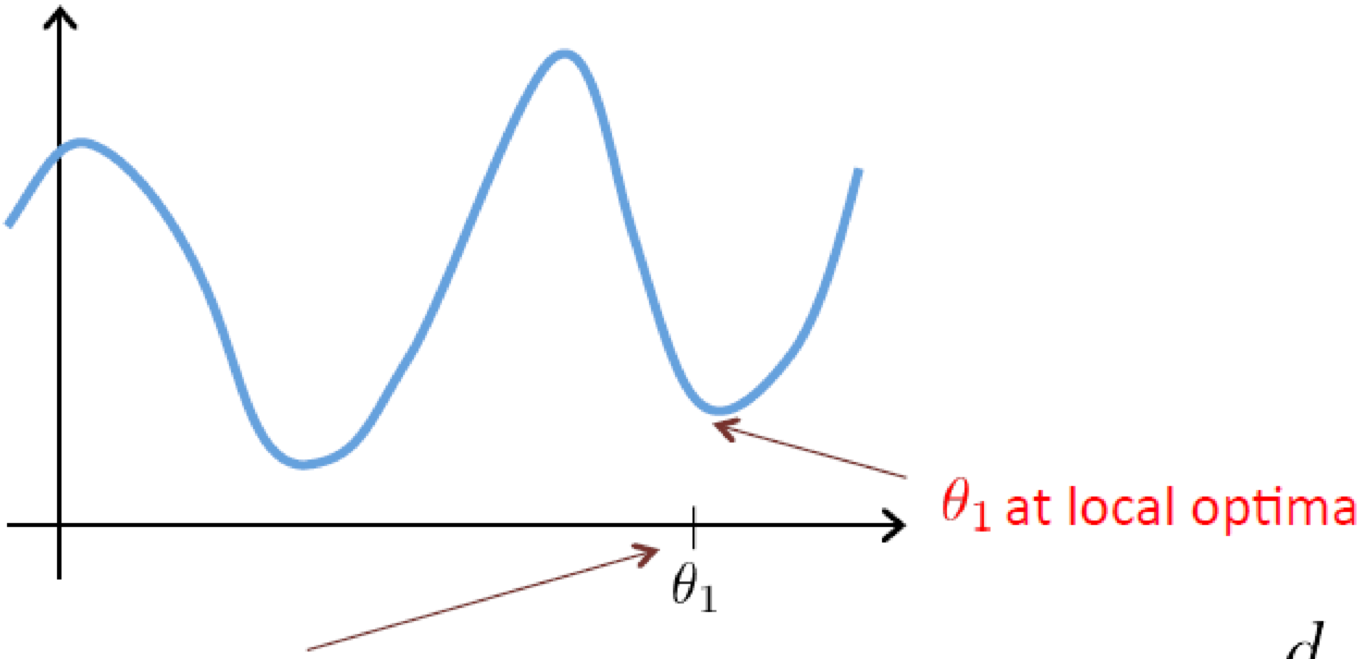
$$\theta_1 \ \coloneqq temp1$$

Gradient Descent

If α (learning rate) is too small, gradient descent can be slow.

If α (learning rate) is too large, gradient descent can overshoot the minimum. It may fail to converge.

Gradient Descent



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

Gradient Descent

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

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

Gradient descent for linear regression

Gradient Descent

```
repeat until convergence {
```

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

```
            (for j=0 and j=1)
}
```

$$g\left(x^t \mid w_1, w_0\right) = w_1 x^t + w_0$$

Error function – linear regression

$$E(w|X) = \frac{1}{2} \sum_{t=1}^{N} [r^t - g(x^t|w)]^2$$

Gradient descent for linear regression

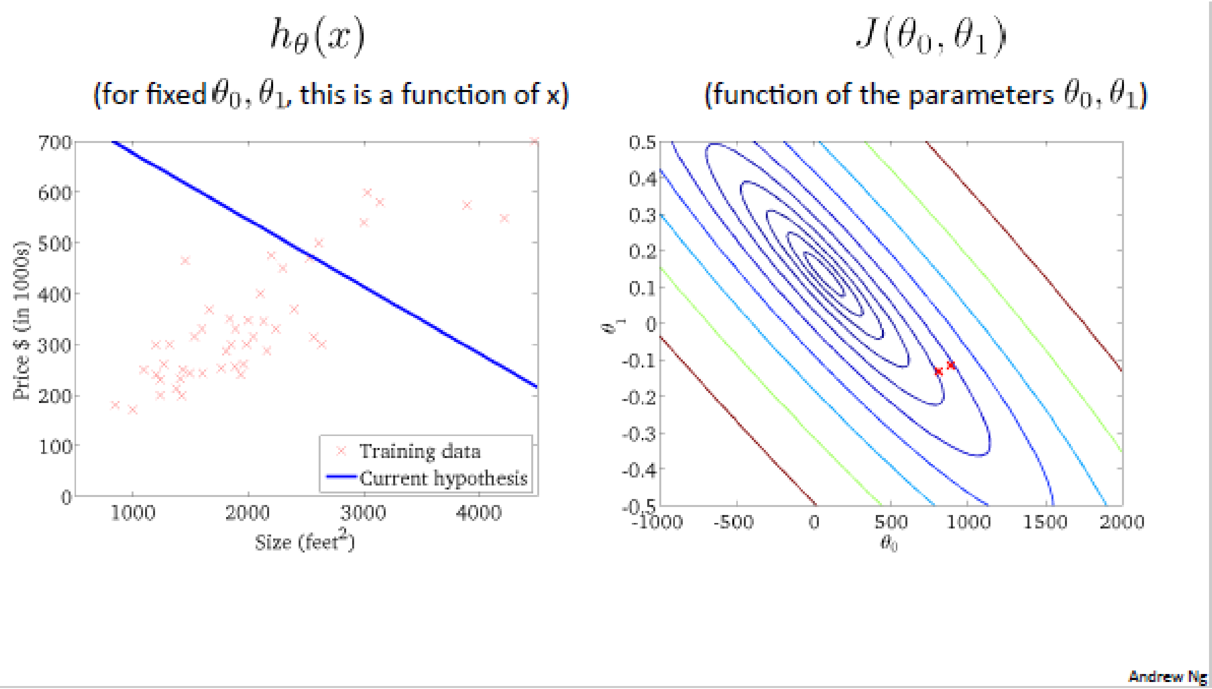
```
repeat until convergence {
```

$$w_0 := w_0 - \alpha \sum_{t=1}^{N} (g(x^t|w) - r^t)$$

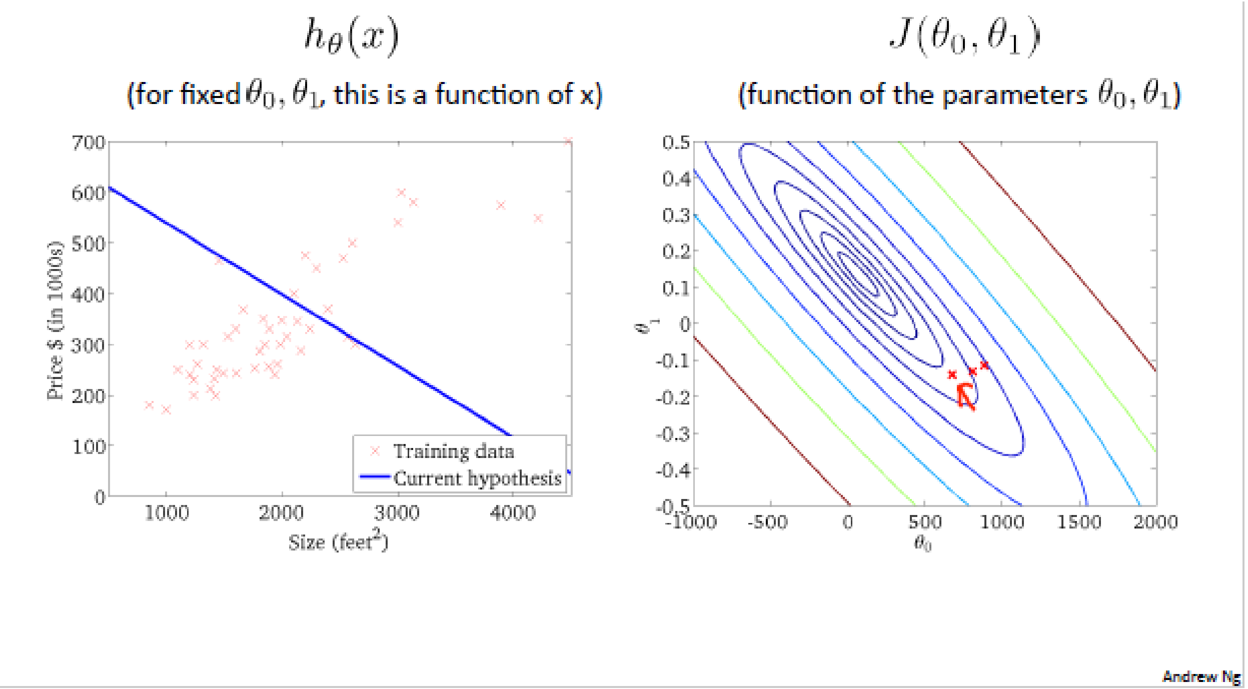
$$w_1 := w_1 - \alpha \sum_{t=1}^{N} (g(x^t|w) - r^t).x^t$$

```
}
```

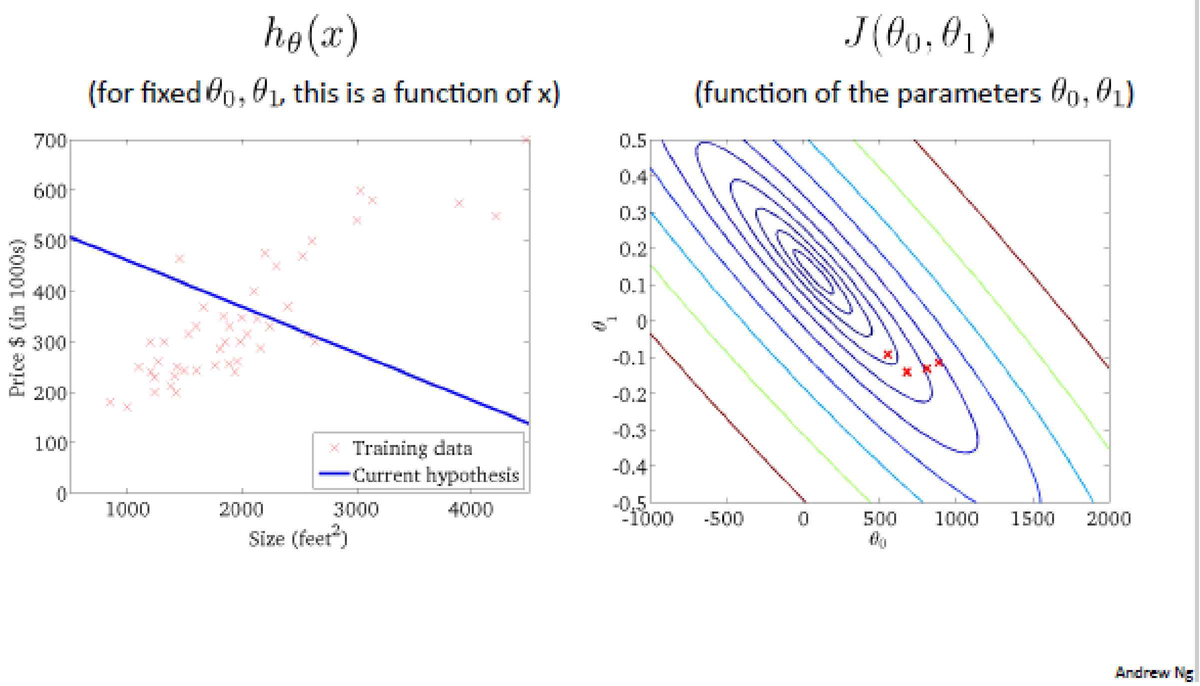
Simultaneously update

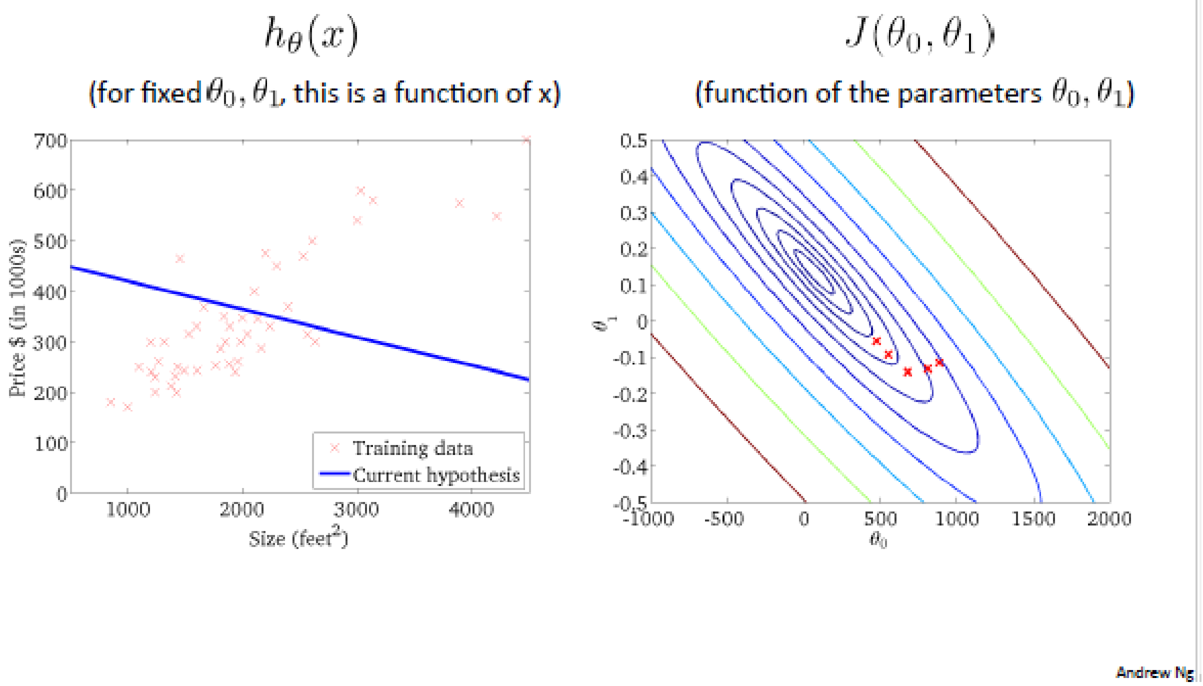


$h_\theta(x)$
(for fixed θ_0, θ_1, this is a function of x)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
$J(\theta_0, \theta_1)$
(function of the parameters θ_0, θ_1)
θ_1
θ_0
Andrew Ng



$h_\theta(x)$
(for fixed θ_0, θ_1, this is a function of x)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
$J(\theta_0, \theta_1)$
(function of the parameters θ_0, θ_1)
θ_1
θ_0
Andrew Ng

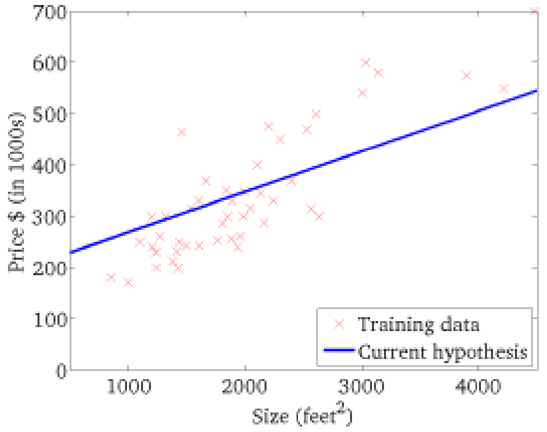

$h_\theta(x)$
(for fixed θ_0, θ_1, this is a function of x)
$J(\theta_0, \theta_1)$
(function of the parameters θ_0, θ_1)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
θ_0
θ_1
Andrew Ng


$h_\theta(x)$
(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)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
θ_0
θ_1
Andrew Ng

…

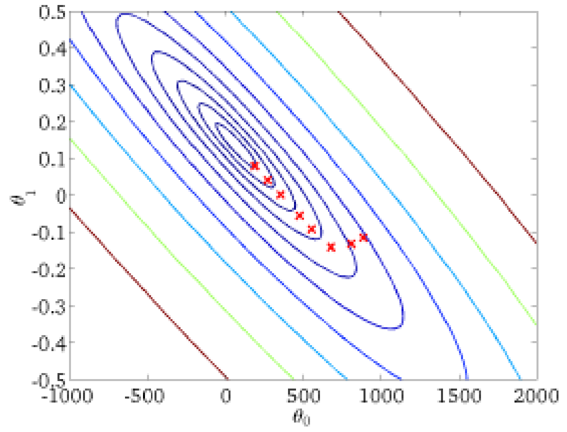
$h_\theta(x)$

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



$J(\theta_0, \theta_1)$

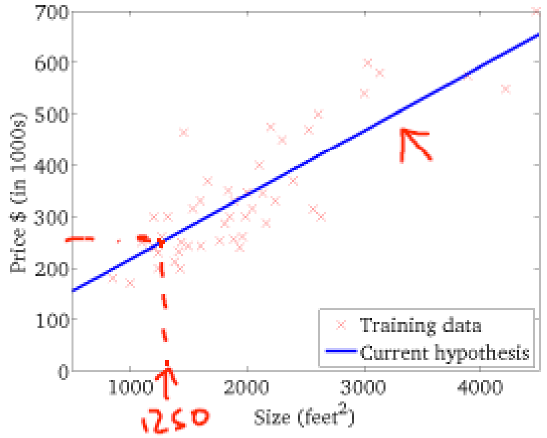
(function of the parameters θ_0, θ_1)



Andrew Ng

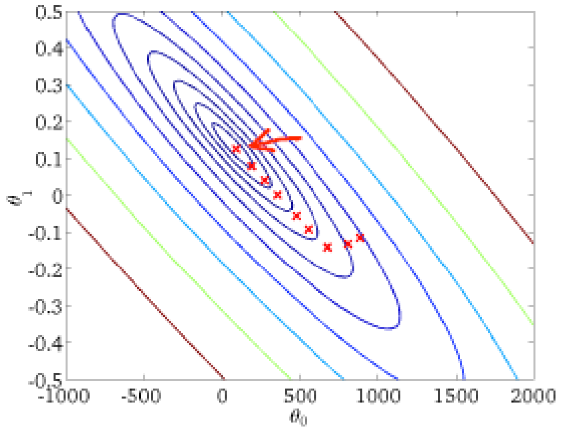
$h_\theta(x)$

(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)



Andrew Ng

Linear regression with multiple variables

Multivariate Regression

$$g(x^t|w_0, w_1, \dots, w_d) = w_0 + w_1 x_1^t + w_2\, x_2^t + \cdots + w_d x_d^t = w^T x^t$$

$$E(w|X) = \frac{1}{2}\sum_{t=1}^{N}[r^t - g(x^t|w)]^2$$

Multivariate Regression

- Normal Equation

$$
\begin{aligned}
\sum_t r^t &= N w_0 + w_1 \sum_t x_1^t + w_2 \sum_t x_2^t + \cdots + w_d \sum_t x_d^t \\
\sum_t x_1^t r^t &= w_0 \sum_t x_1^t + w_1 \sum_t (x_1^t)^2 + w_2 \sum_t x_1^t x_2^t + \cdots + w_d \sum_t x_1^t x_d^t \\
\sum_t x_2^t r^t &= w_0 \sum_t x_2^t + w_1 \sum_t x_1^t x_2^t + w_2 \sum_t (x_2^t)^2 + \cdots + w_d \sum_t x_2^t x_d^t \\
&\vdots \\
\sum_t x_d^t r^t &= w_0 \sum_t x_d^t + w_1 \sum_t x_d^t x_1^t + w_2 \sum_t x_d^t x_2^t + \cdots + w_d \sum_t (x_d^t)^2
\end{aligned}
$$

Multivariate Regression

- Normal Equation

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \cdots & x_d^1 \\ 1 & x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & & & & \\ 1 & x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d, \end{bmatrix}, \boldsymbol{r} = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^N \end{bmatrix}$$

$$X^T X w = X^T r \quad \rightarrow w = (X^T X)^{-1} X^T r$$

Example

$$w = (X^T X)^{-1} X^T r$$

x_0	Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) r
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad r = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

Gradient Descent

```
repeat until convergence {
```

$$w_j \coloneqq w_j - \alpha \sum_{t=1}^{N} (g(x^t|w) - r^t).\, x_j^t$$

```
}
```

Feature Scaling

Idea: Make sure features are on a similar scale

Mean normalization

$$x' = \frac{x - mean(x)}{\max(x) - \min(x)}$$

x'*: normalized value*

Debugging

- How to make sure that gradient descent is working correctly
- How to choose learning rate

Convergence Plot (number of iteration vs $J(\theta_0, \theta_1)$)