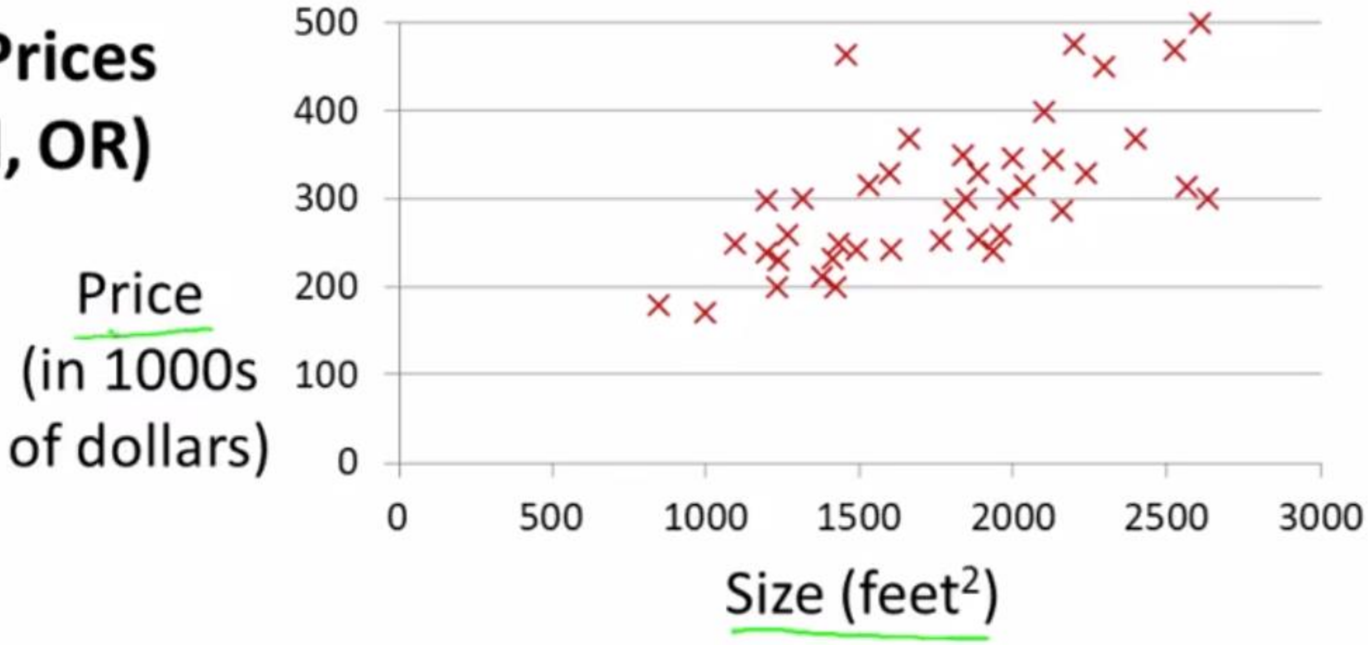
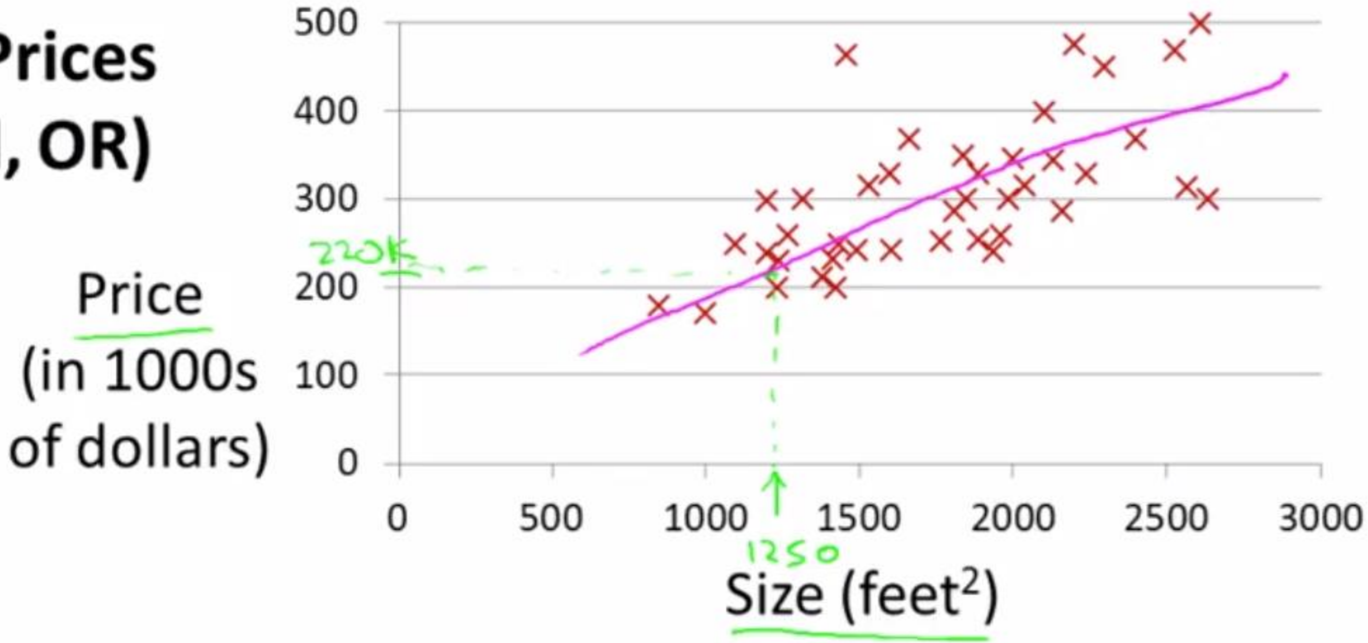
Model and Cost Function

Model Representation

Housing Prices (Portland, OR)



Housing Prices (Portland, OR)



Supervised Learning

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

Regression Problem

Predict real-valued output

Training set of housing prices (Portland, OR)

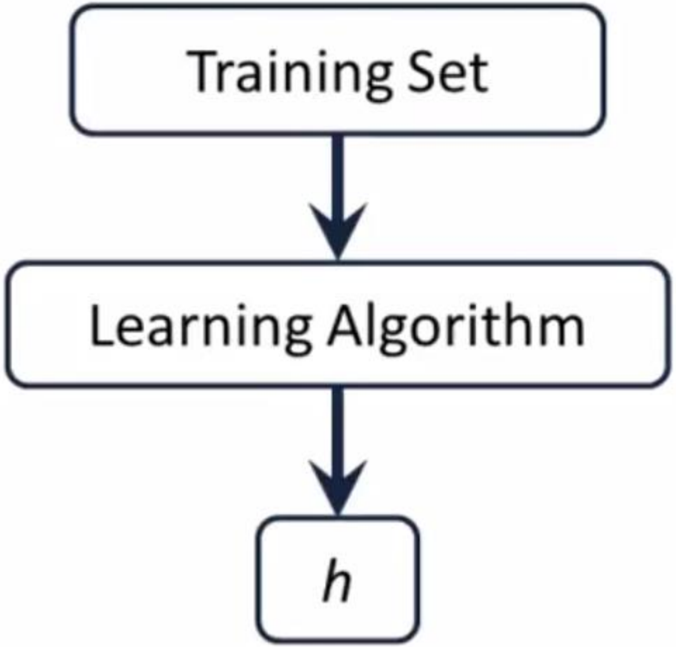
Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Notation:

m = Number of training examples

x’s = “input” variable / features

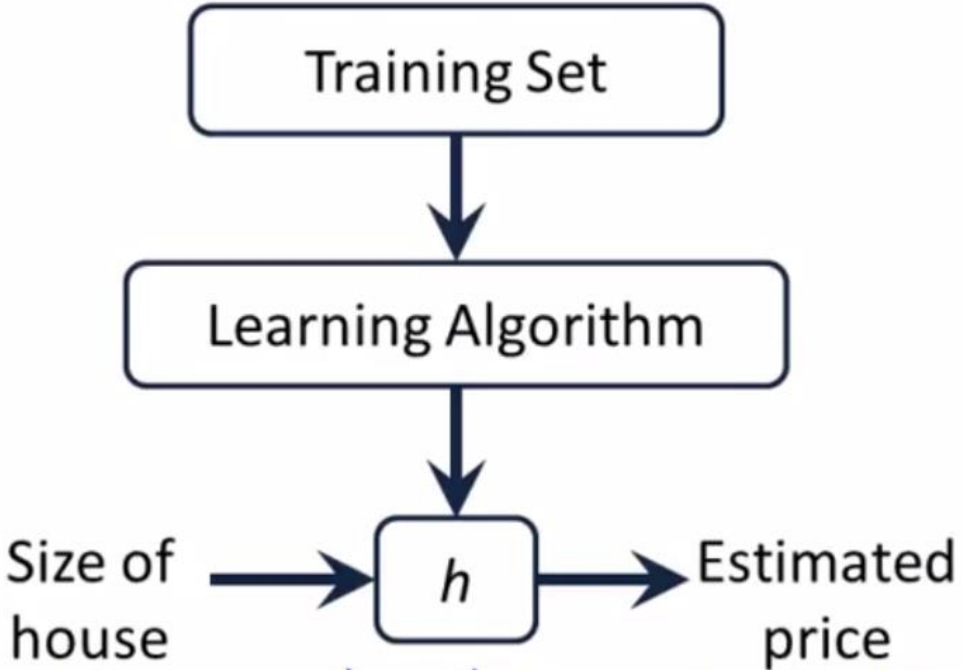
y’s = “output” variable / “target” variable



Training Set

Learning Algorithm

h



Training Set

Learning Algorithm

Size of house

h

Estimated price

x

hypothesis

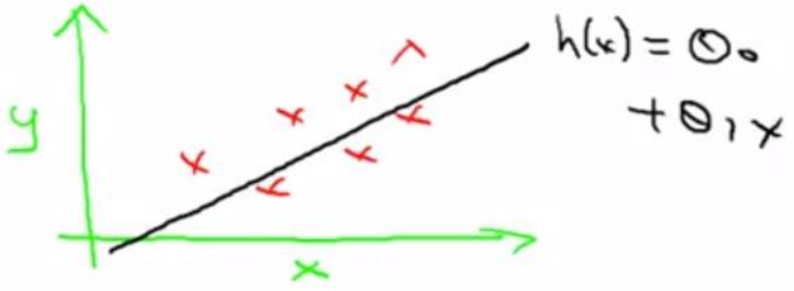
(estimated value of y)

h maps from x's to y's.

How do we represent *h* ?

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

Shorthand: h(x)

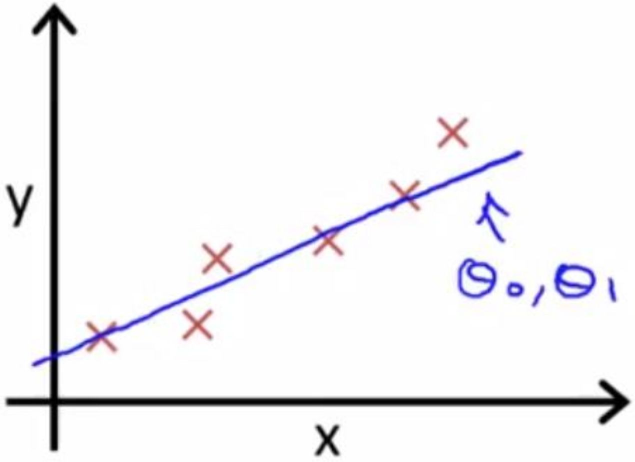


Linear regression with one variable. (x)

Univariate linear regression.

└ one variable

Cost Function: How to choose θs?



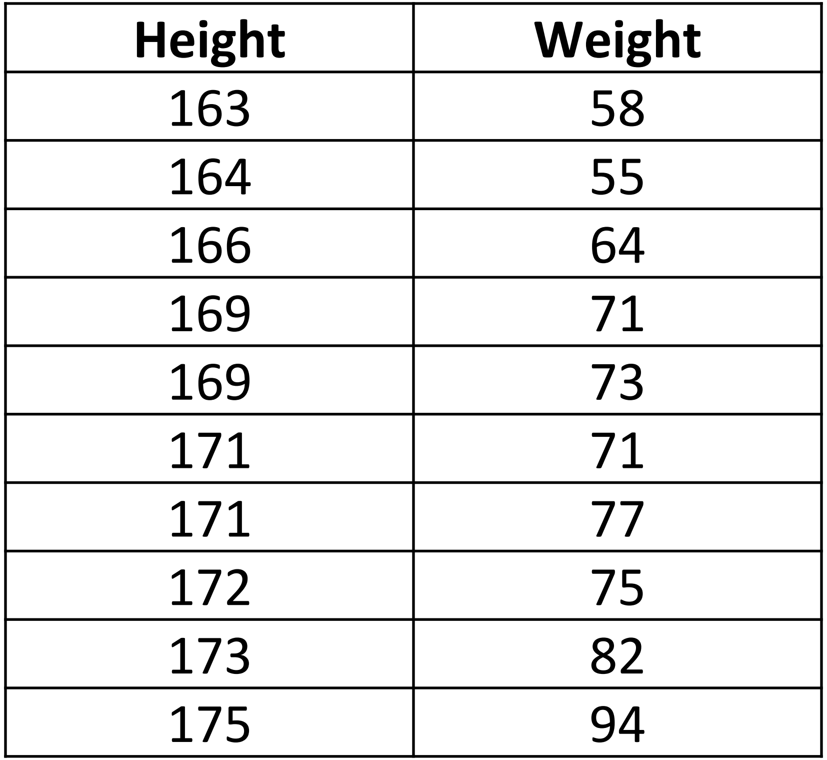
Idea: Choose θ_0, θ_1 so that
$h_\theta(x)$ is close to y for our
training examples (x, y)

2:44 / 8:12

Andrew Ng

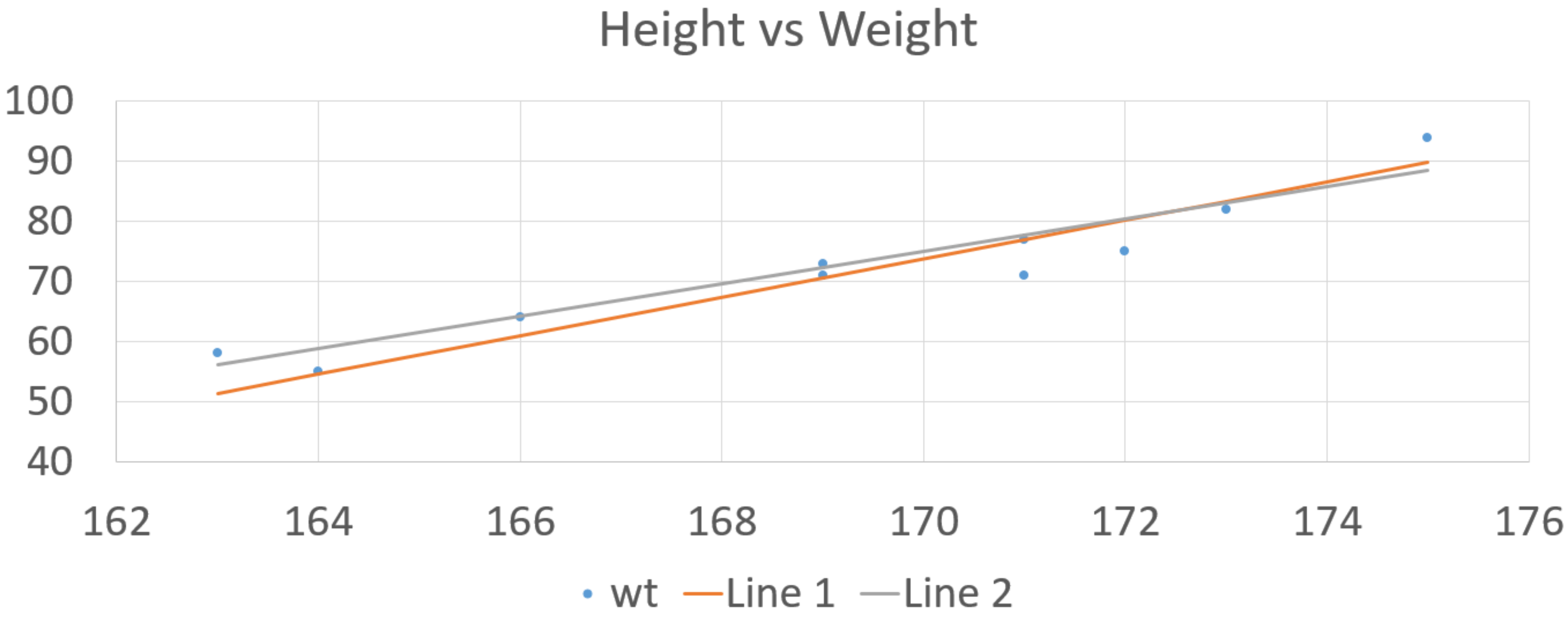
Sample relations

- Assume we want to find the relation between the weight and the height of the students in this university.
- We have the following data:



Height	Weight
163	58
164	55
166	64
169	71
169	73
171	71
171	77
172	75
173	82
175	94

Sample relations



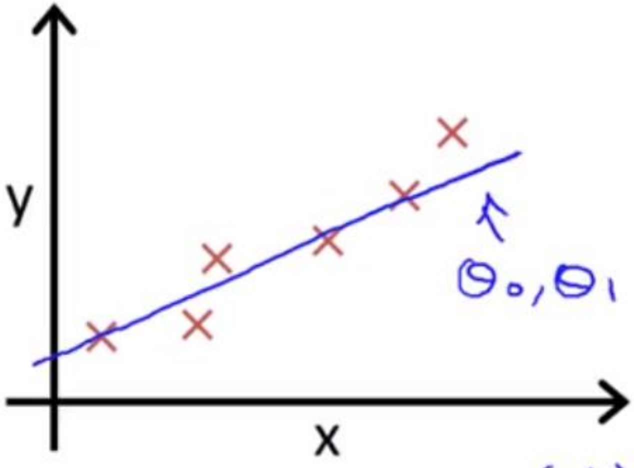
Line 1: Weight = -470.2 + 3.2*Height

Line 2: Weight = -384+2.7*Height

Which line is better?

xi	y_i	$\hat{y}$	e_i	e_i^2
163	58	51.4	6.6	43.56
164	55	54.6	0.4	0.16
166	64	61	3	9
169	71	70.6	0.4	0.16
169	73	70.6	2.4	5.76
171	71	77	-6	36
171	77	77	0	0
172	75	80.2	-5.2	27.04
173	82	83.4	-1.4	1.96
175	94	89.8	4.2	17.64
				141.28

y_i	$\hat{y}$	e_i	e_i^2
58	56.1	1.9	3.61
55	58.8	-3.8	14.44
64	64.2	-0.2	0.04
71	72.3	-1.3	1.69
73	72.3	0.7	0.49
71	77.7	-6.7	44.89
77	77.7	-0.7	0.49
75	80.4	-5.4	29.16
82	83.1	-1.1	1.21
94	88.5	5.5	30.25
			126.27



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

Idea: Choose $\underline{\theta_0}, \underline{\theta_1}$ so that
$\underline{h_\theta(x)}$ is close to $\underline{y}$ for our training examples $\underline{(x, y)}$

x, y

#training examples

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

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

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

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

Squared error function

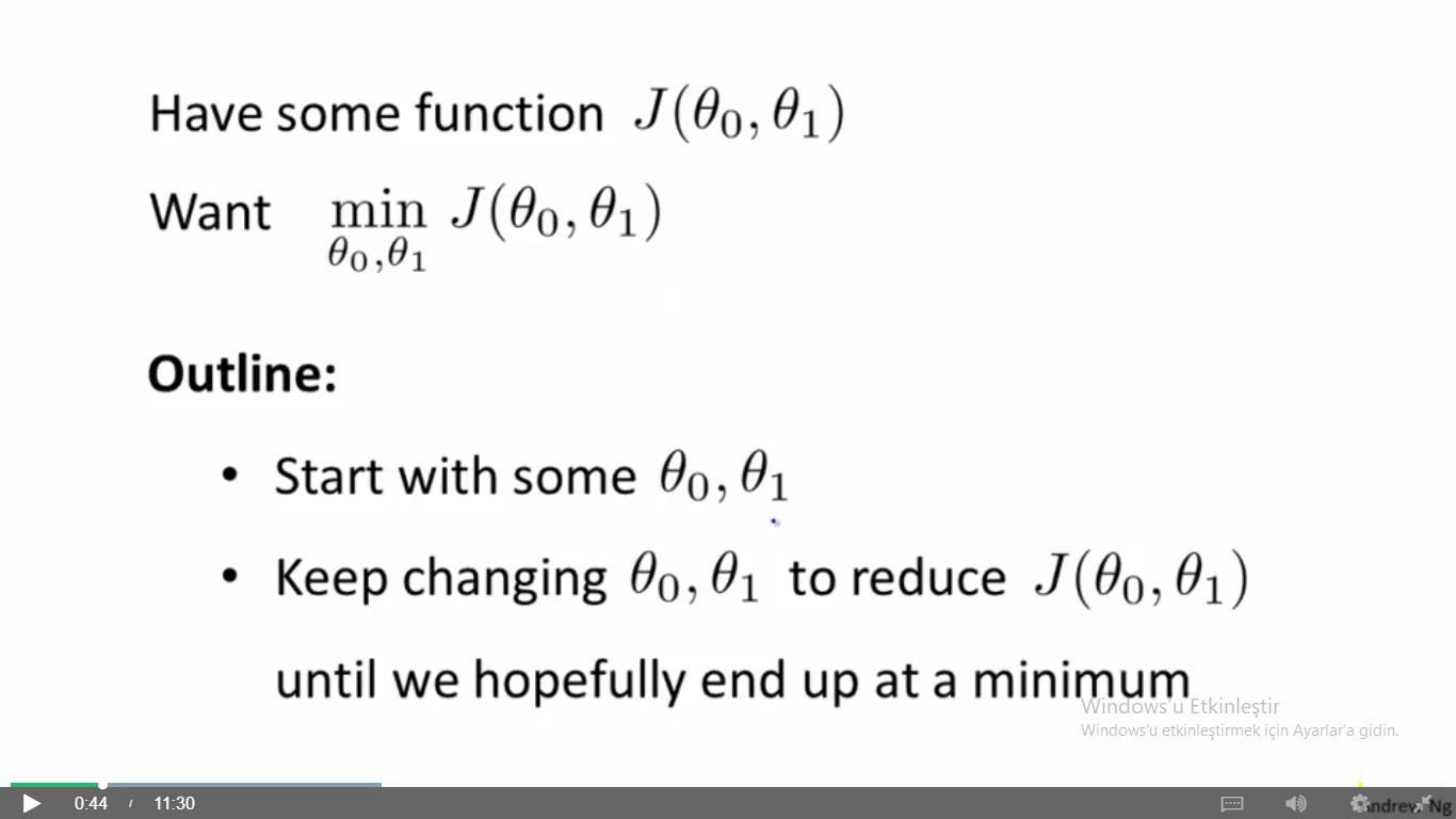
Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

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

Andrew Ng



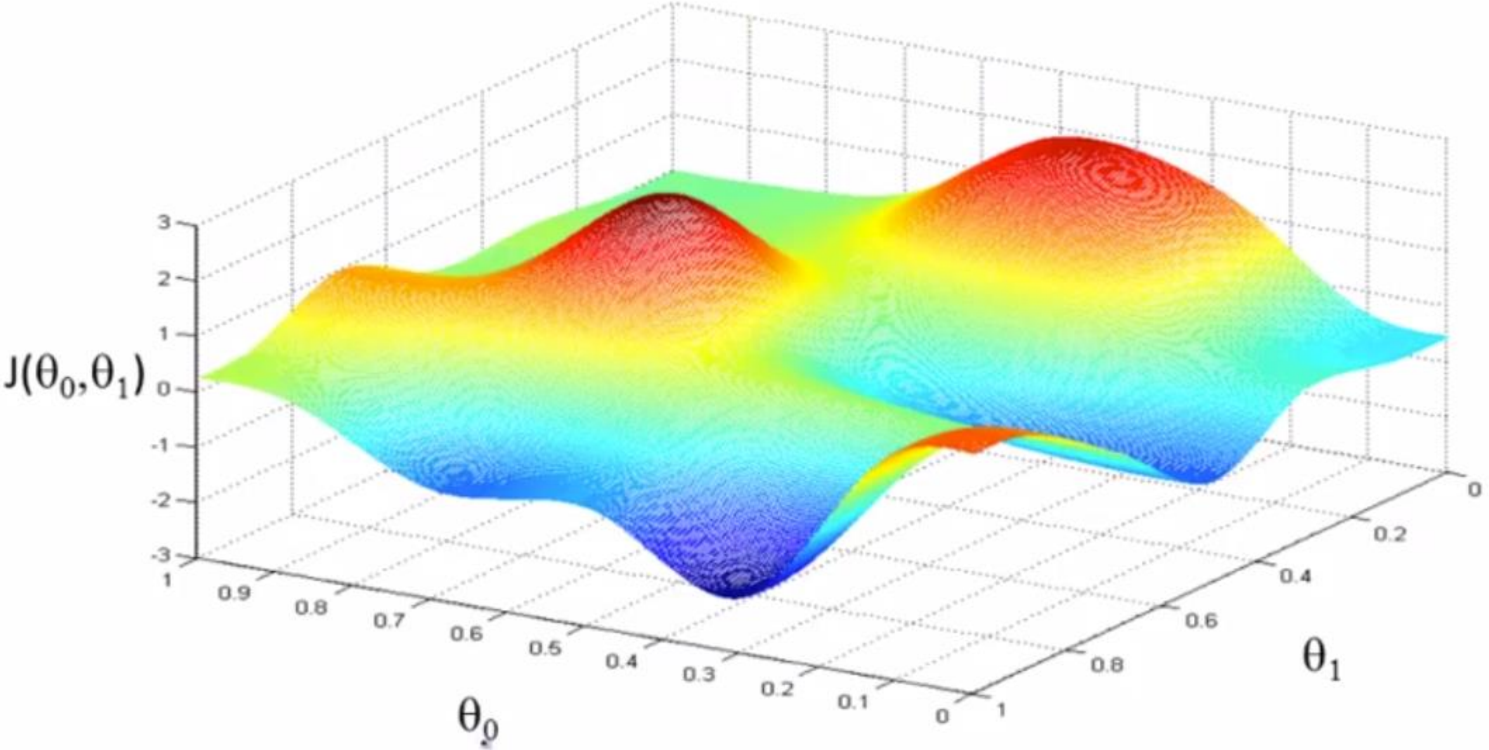
Have some function $J(\theta_0, \theta_1)$

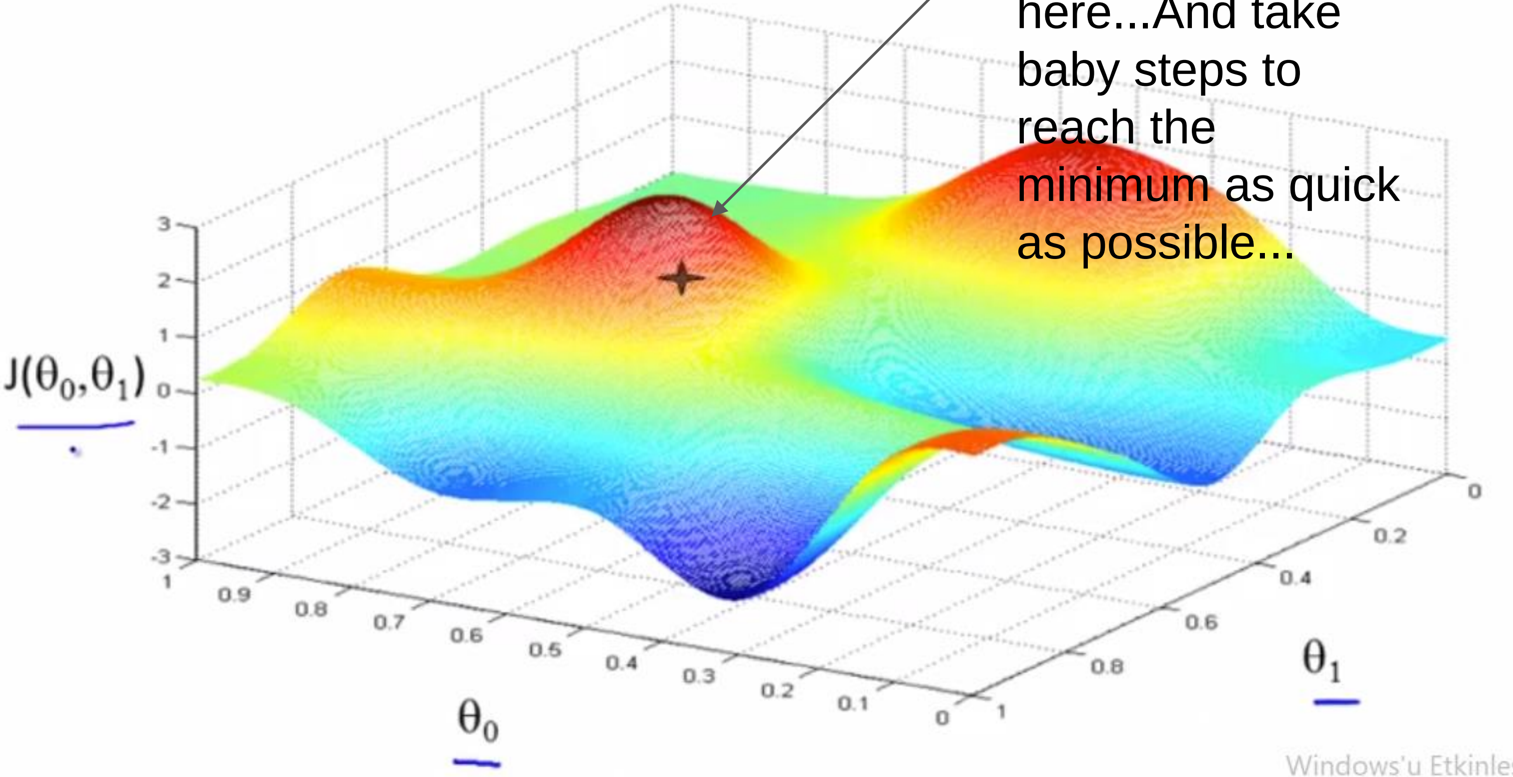
Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

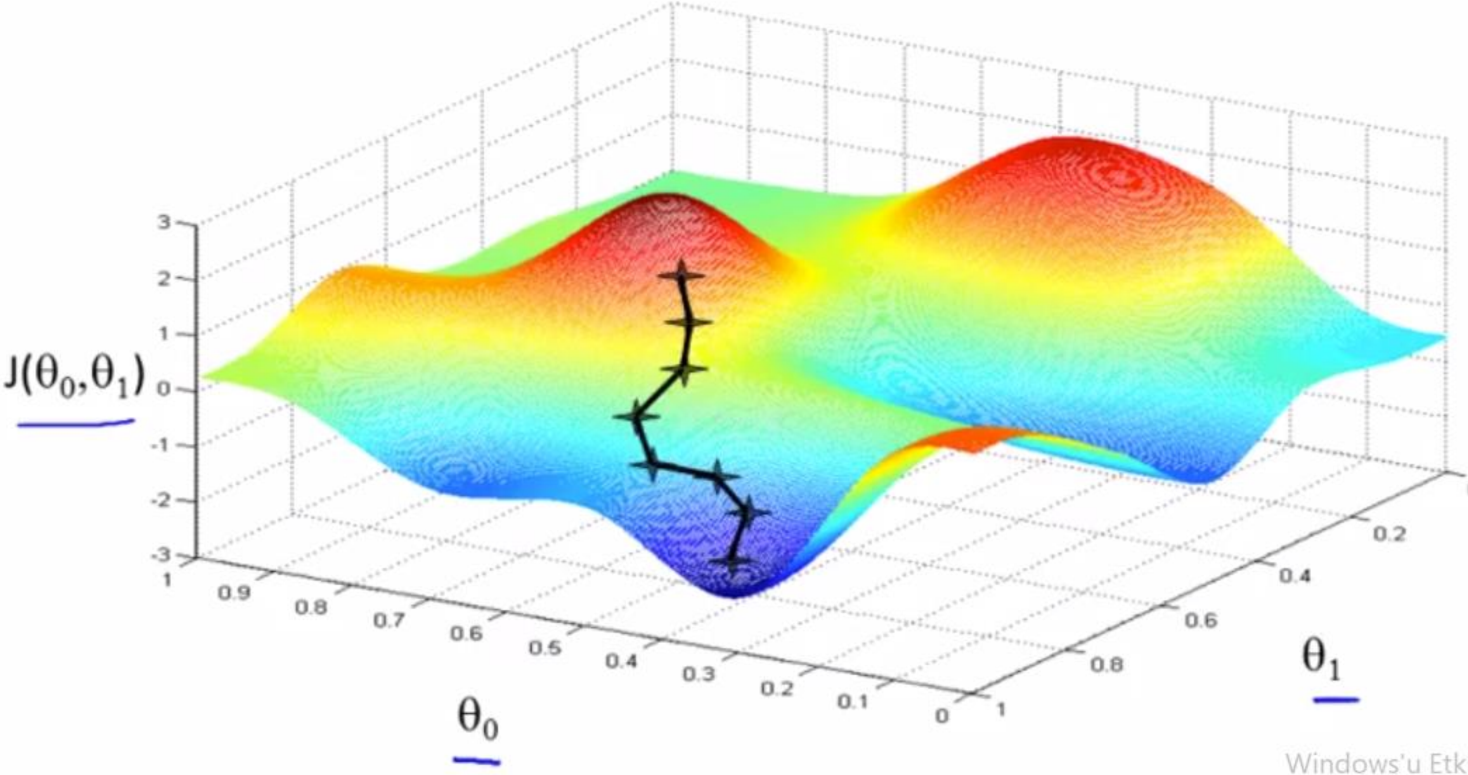
  until we hopefully end up at a minimum


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

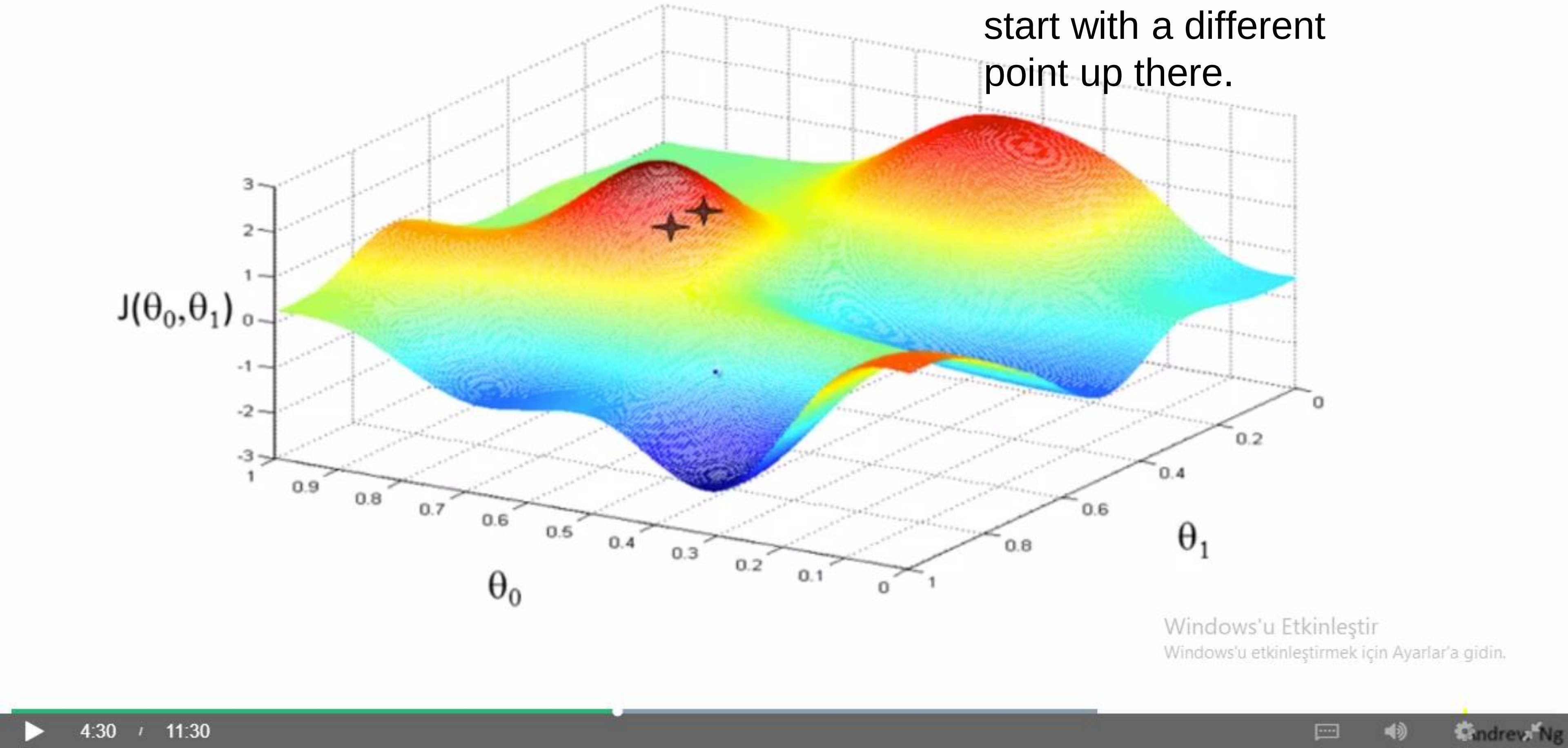


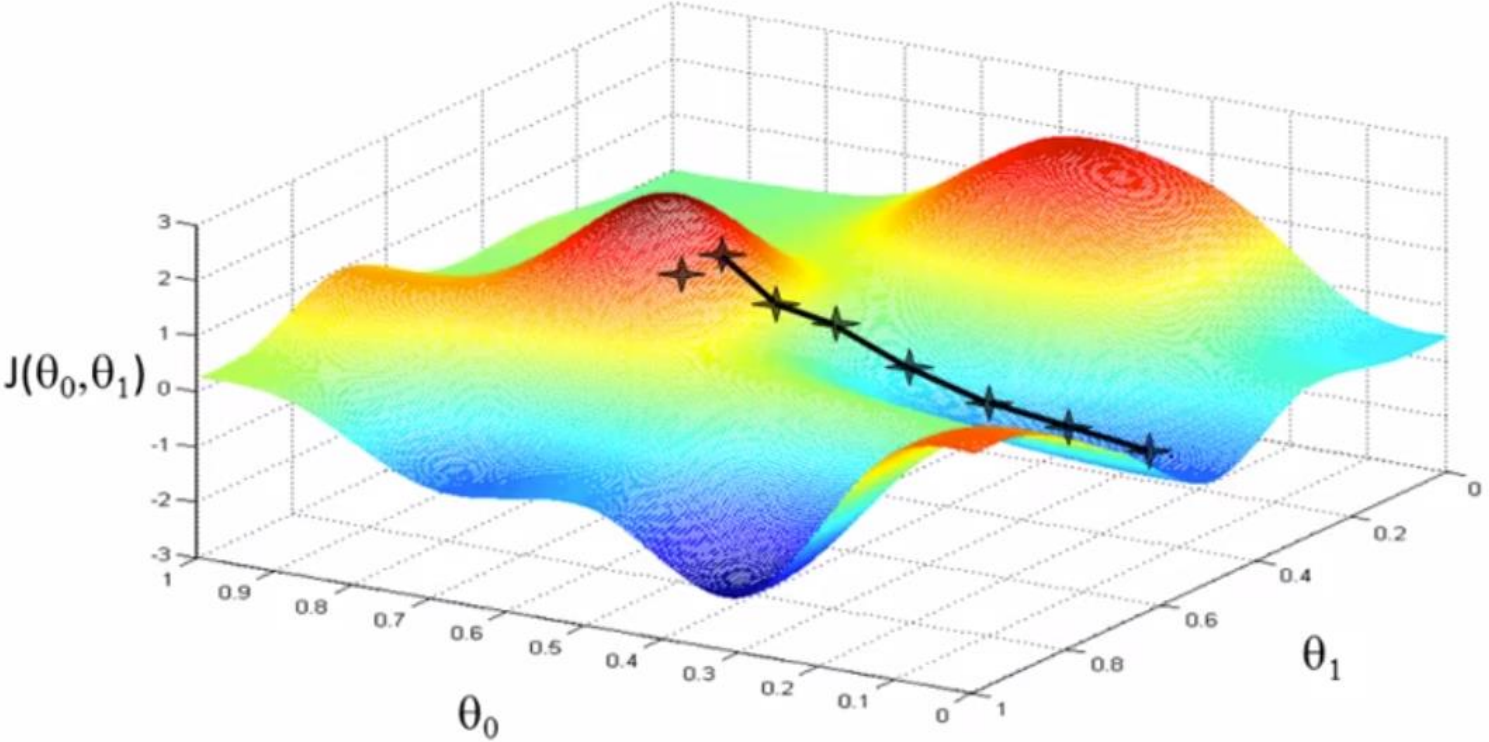
Start from here...And take baby steps to reach the minimum as quick as possible...

Andrew Ng


J(θ0,θ1)
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
θ0
0
0.2
0.4
0.6
0.8
1
θ1

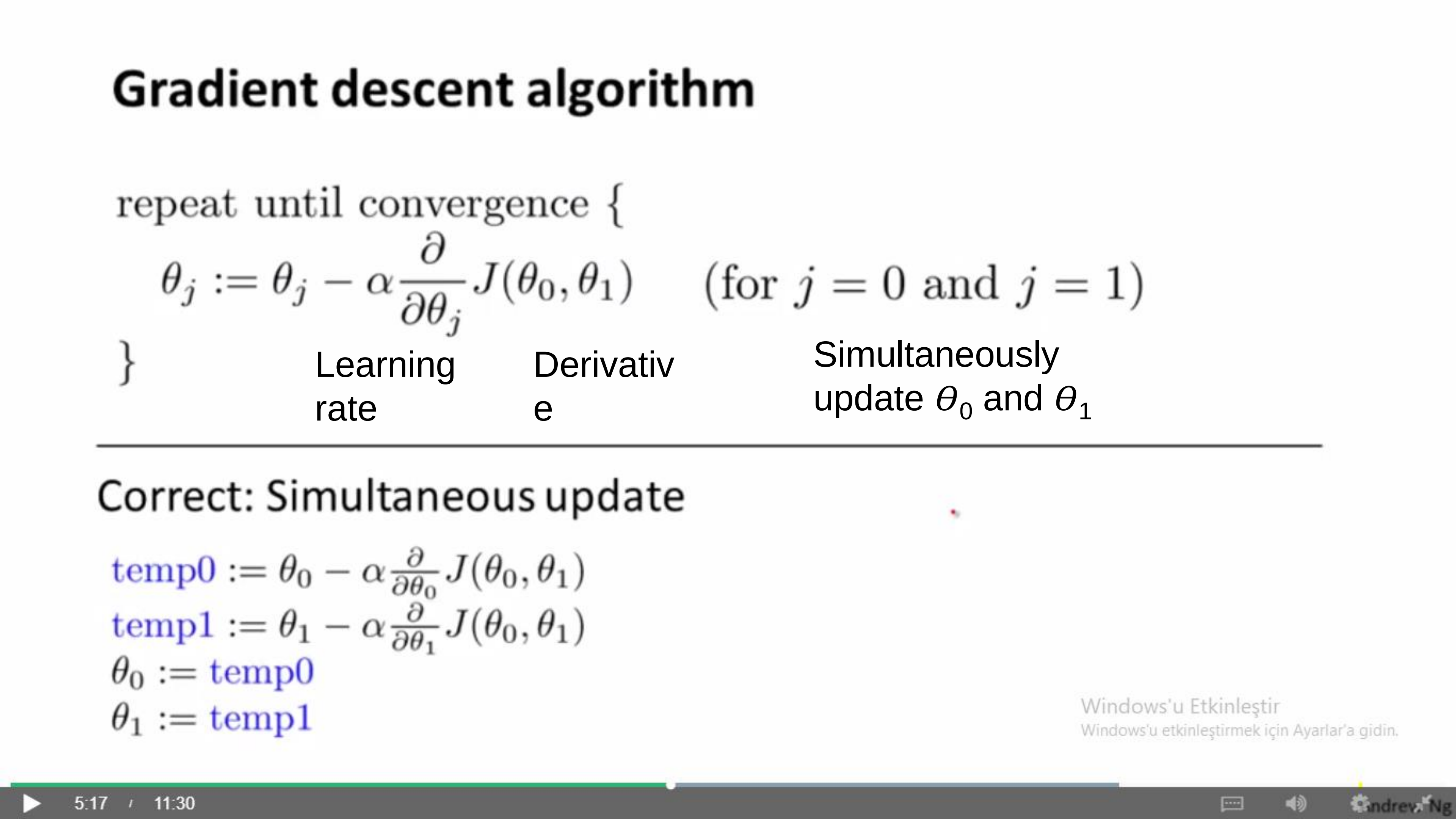
Now assume we start with a different point up there.





4:49 / 11:30

Andrew Ng



Gradient descent algorithm

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

Learning rate

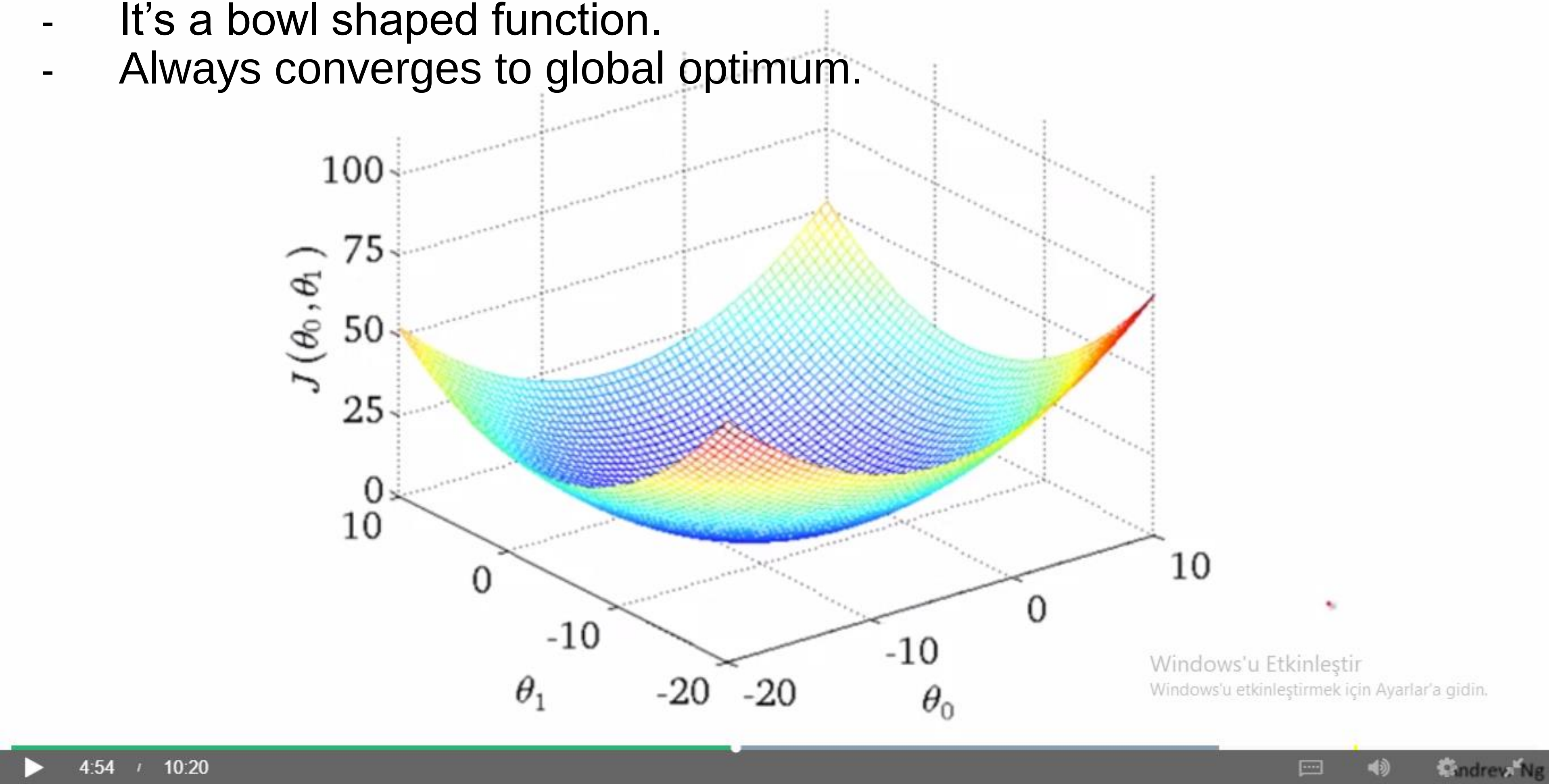
Derivative

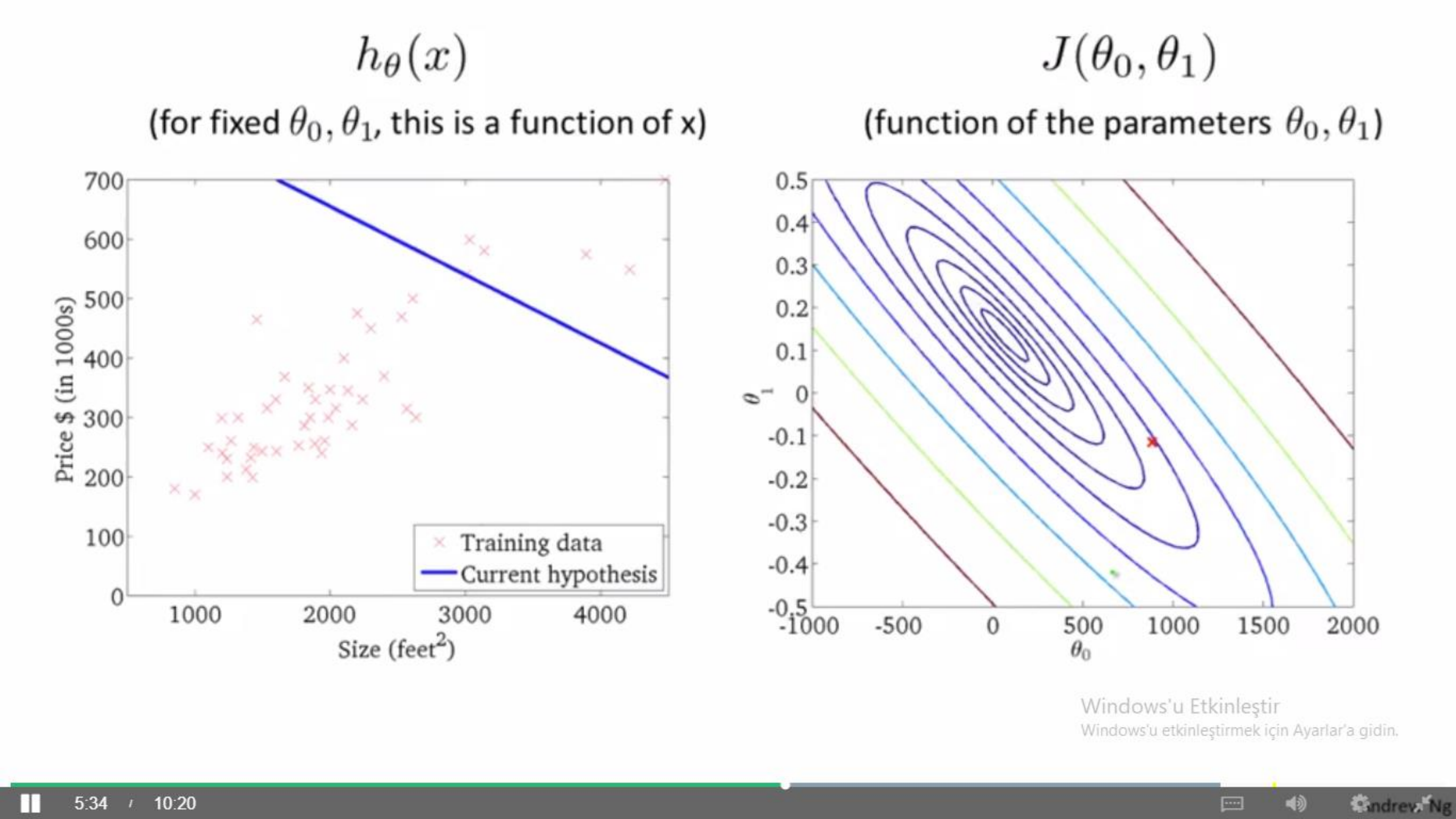
Simultaneously update θ_0 and θ_1

Correct: Simultaneous update

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

- Convex function.
- It’s a bowl shaped function.
- Always converges to global optimum.





$h_\theta(x)$

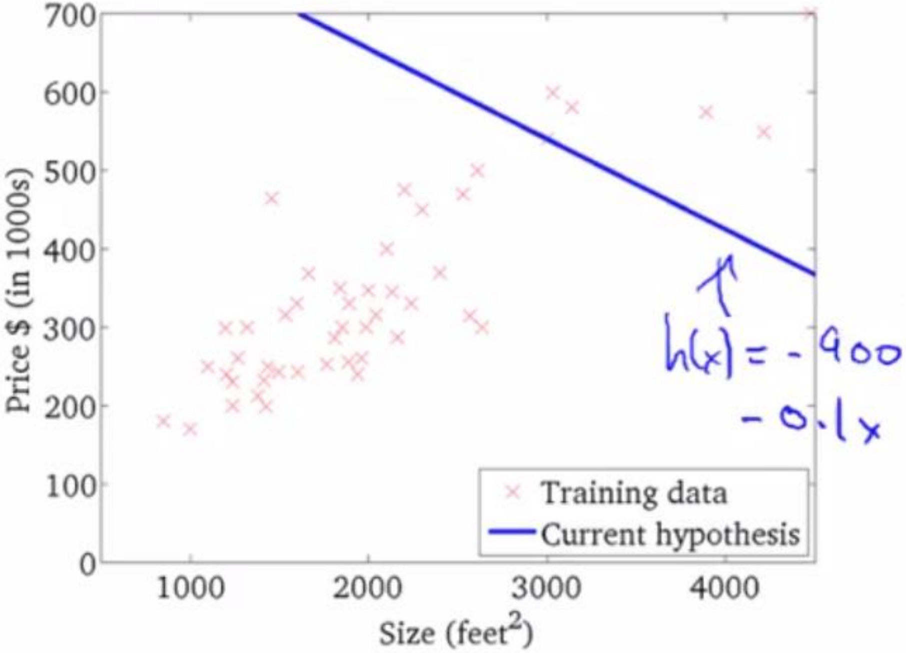
(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

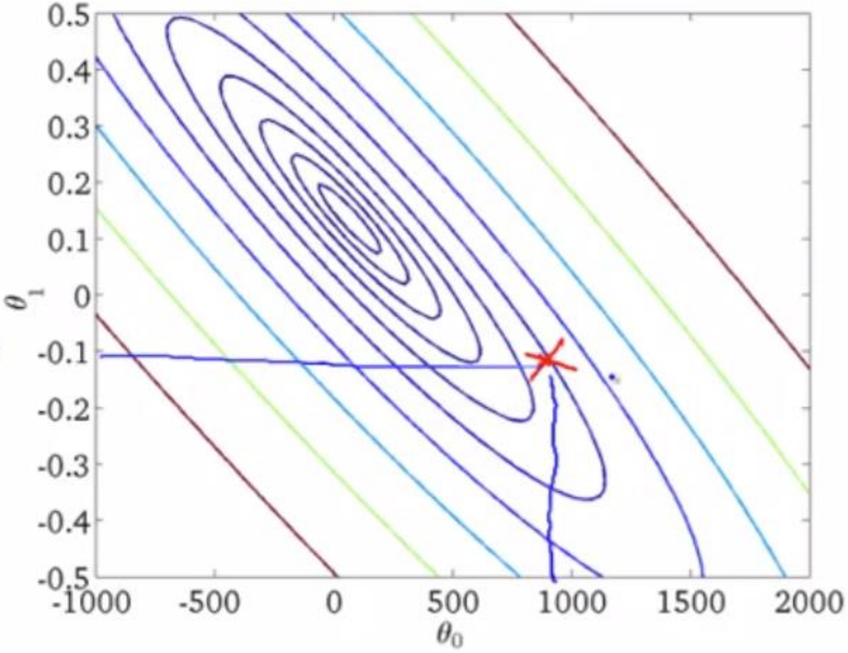
$h_\theta(x)$

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



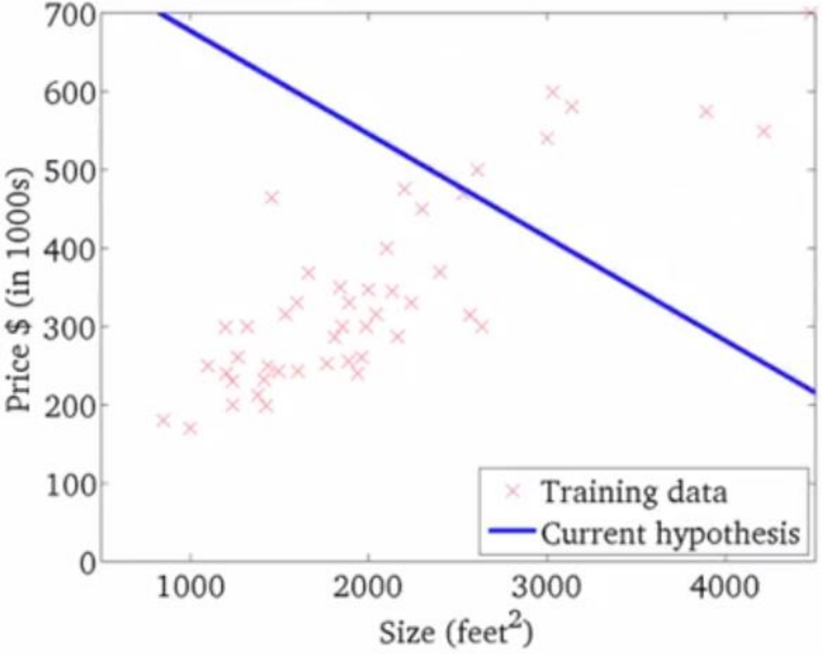
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



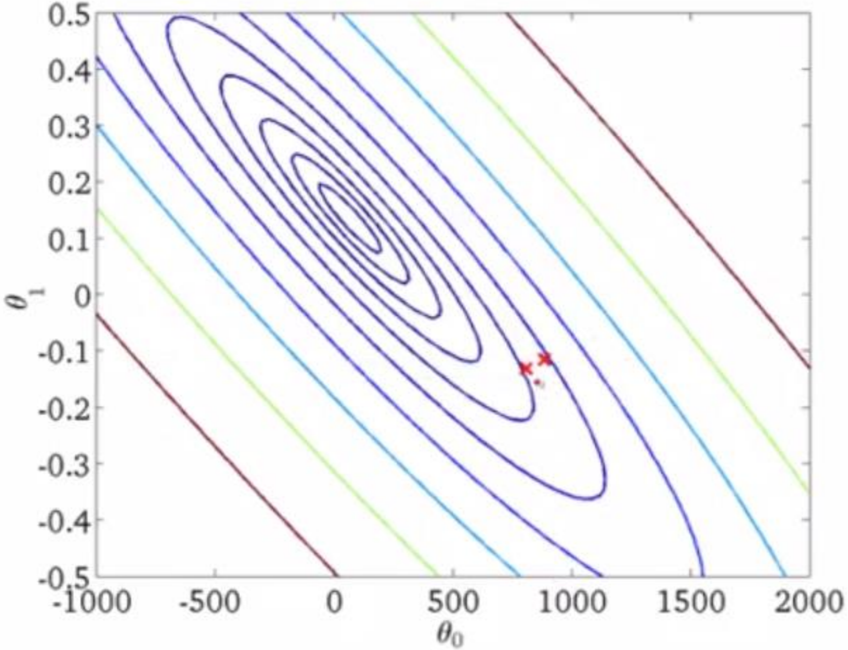
$h_\theta(x)$

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



$J(\theta_0, \theta_1)$

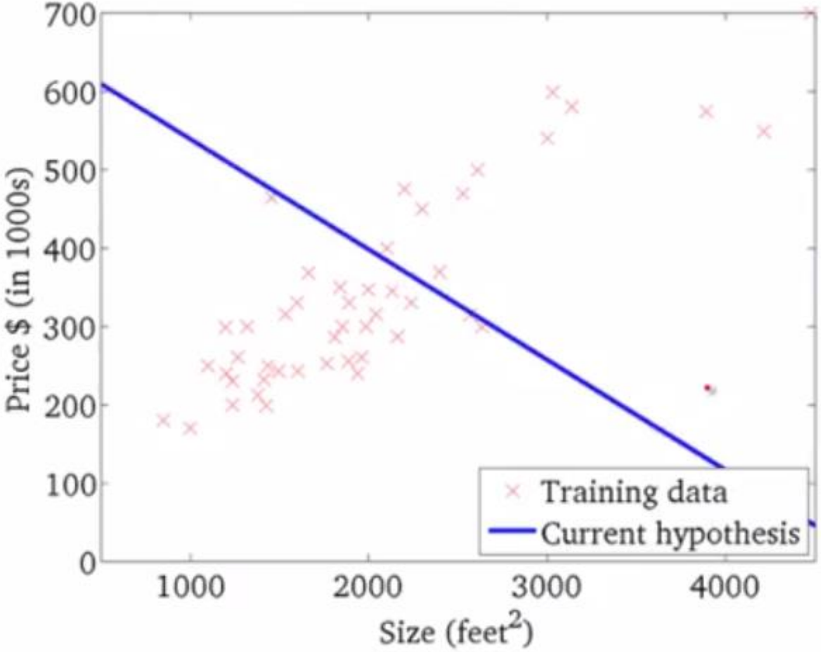
(function of the parameters θ_0, θ_1)



Andrew Ng

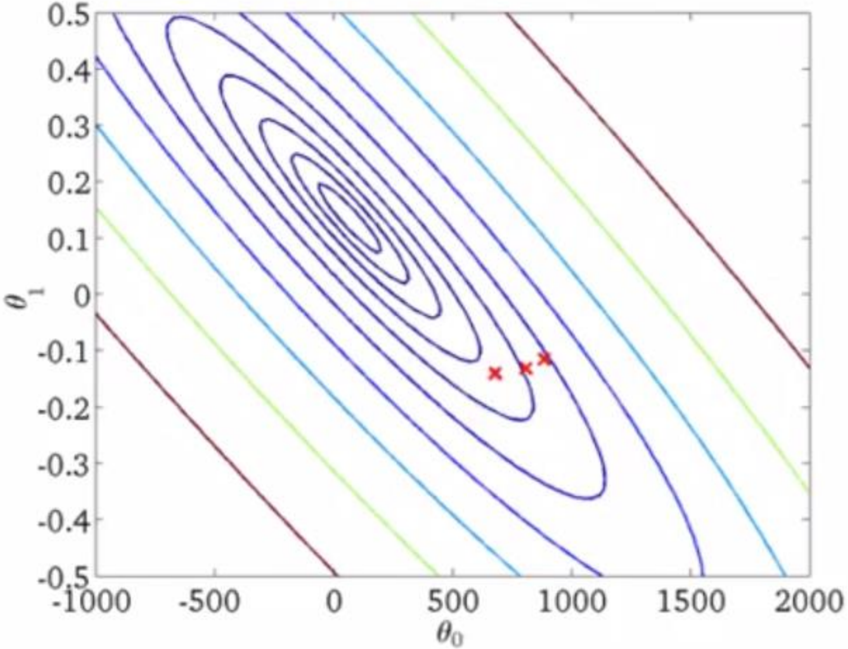
$h_\theta(x)$

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



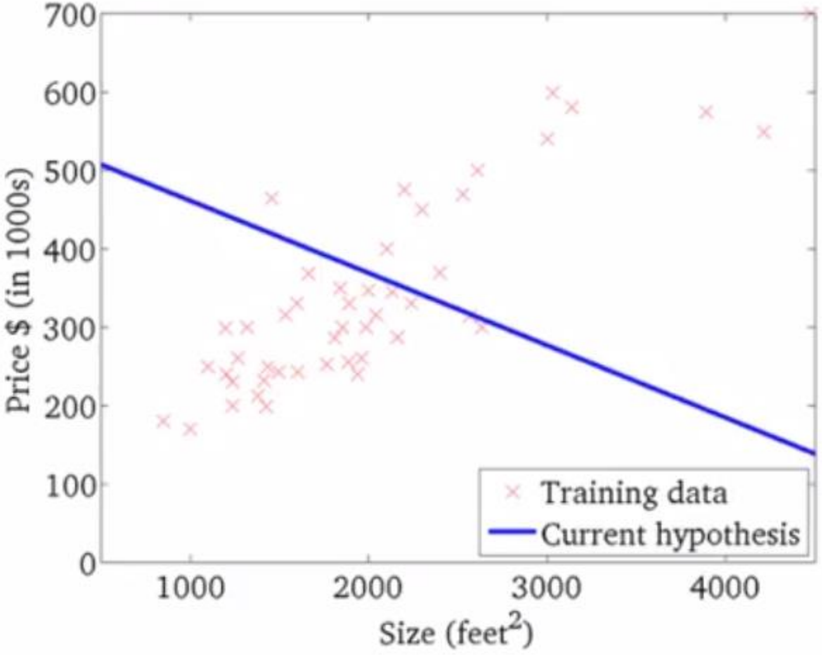
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



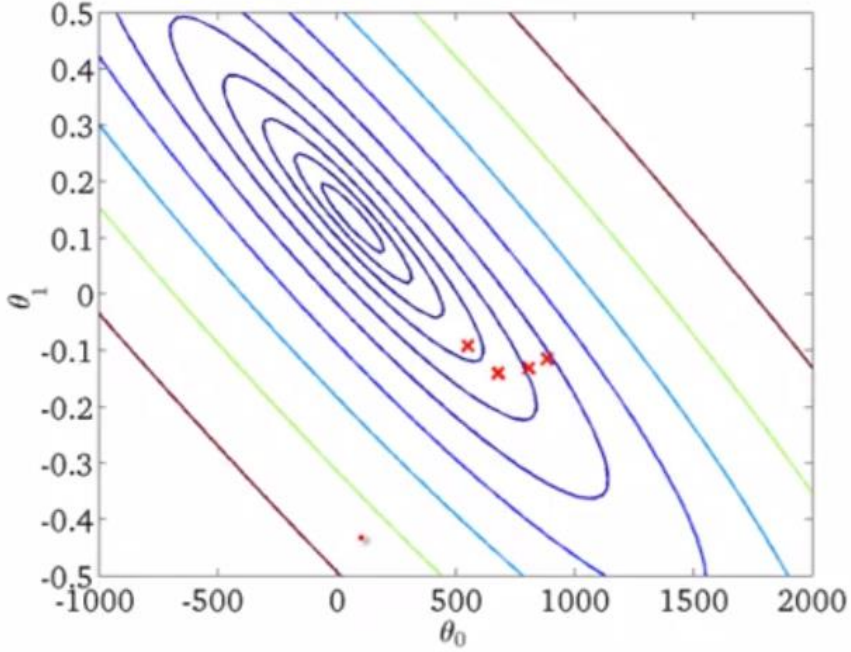
$h_\theta(x)$

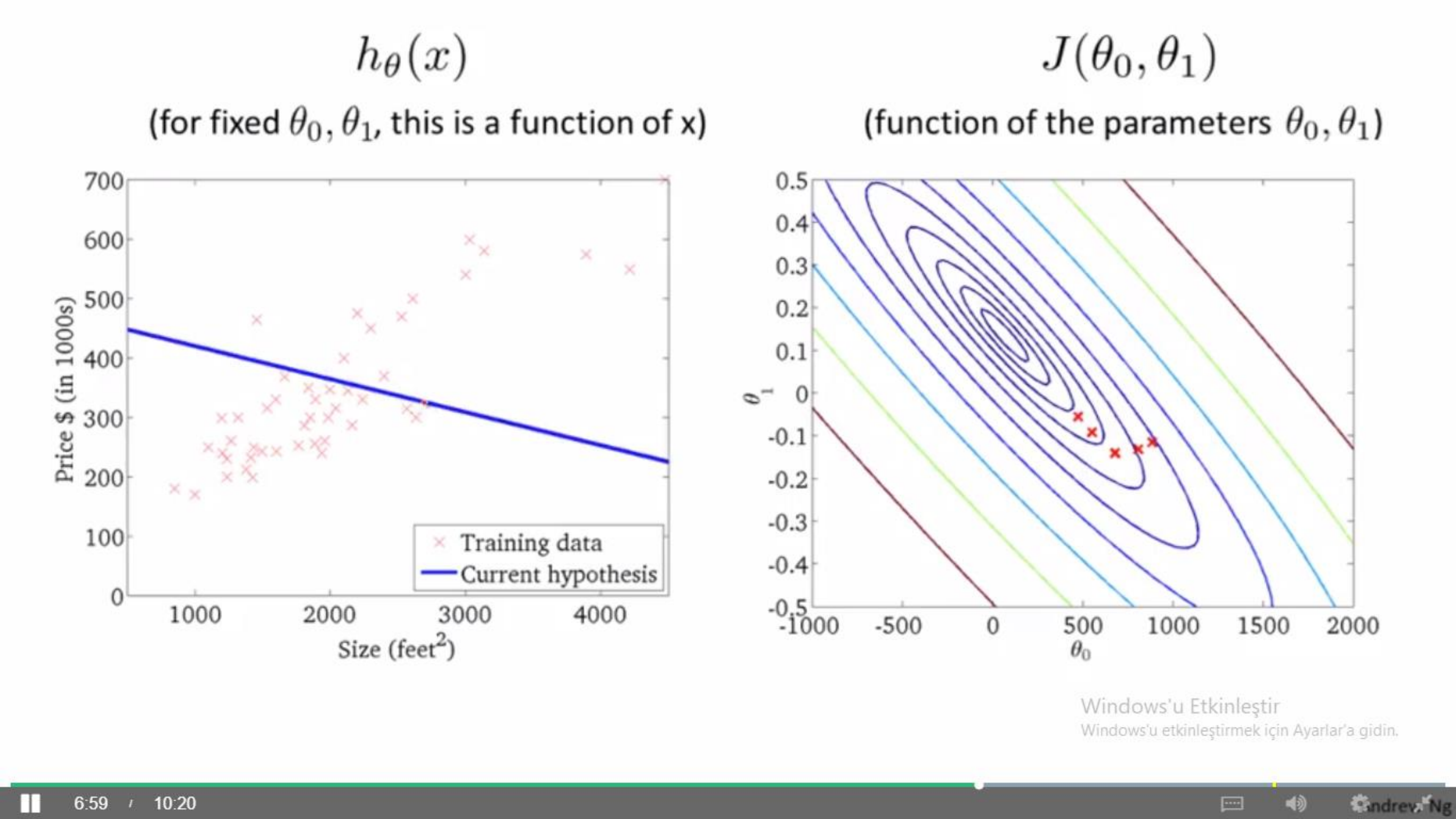
(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)





$h_\theta(x)$

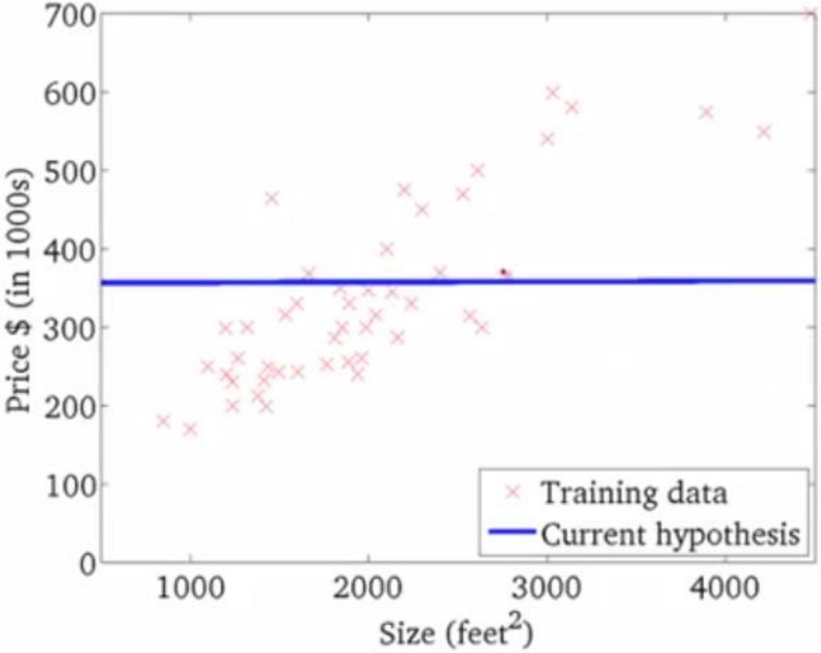
(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

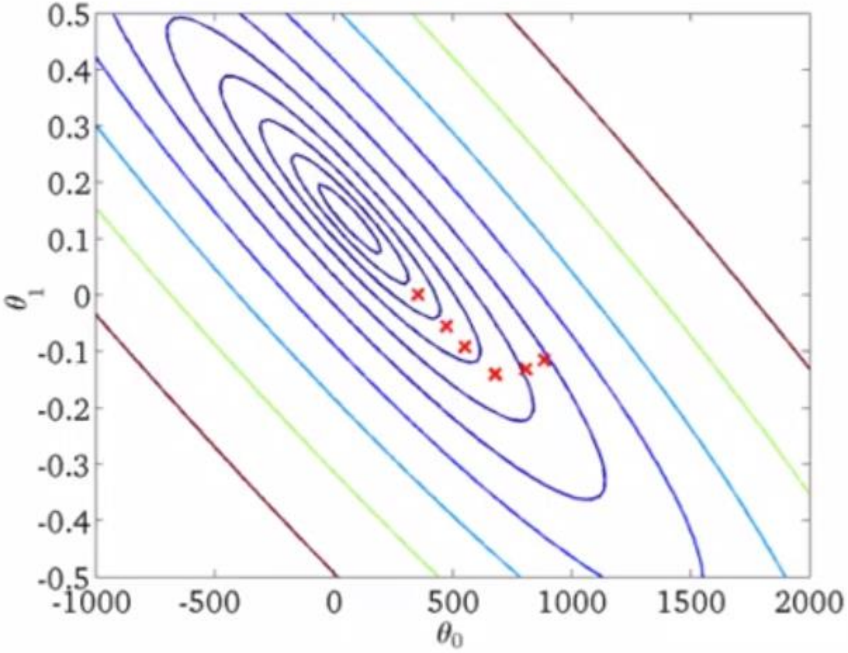
$h_\theta(x)$

(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)

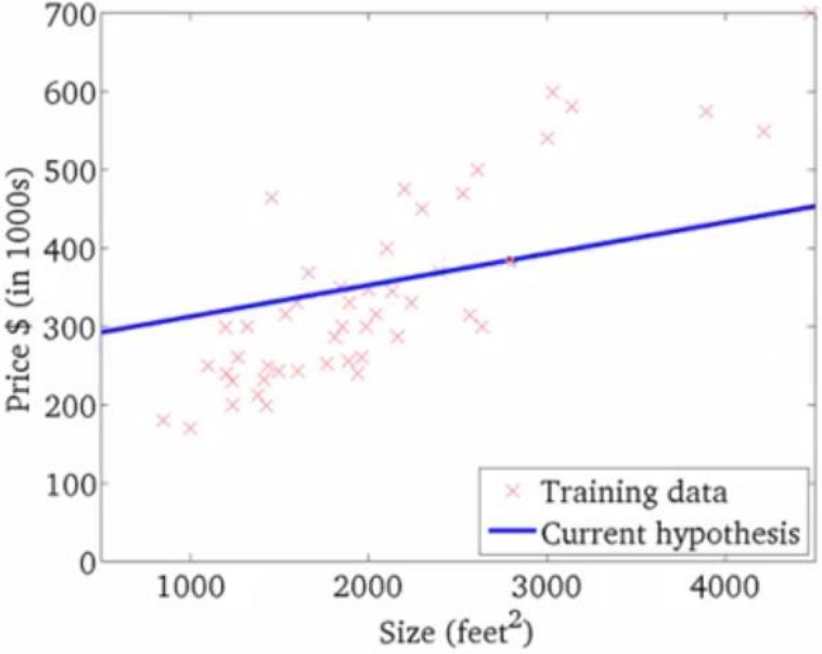


7:00 / 10:20

Andrew Ng

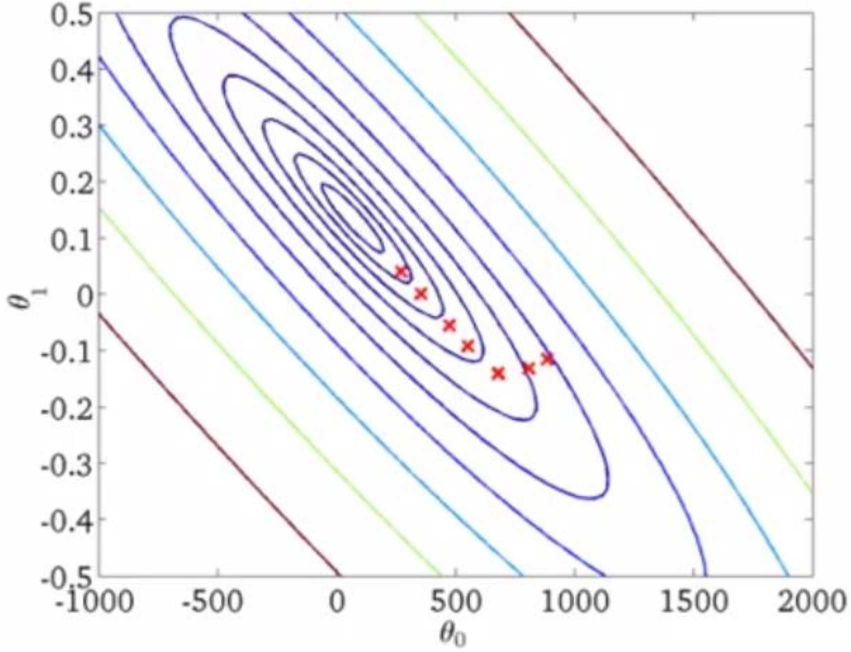
$h_\theta(x)$

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



$J(\theta_0, \theta_1)$

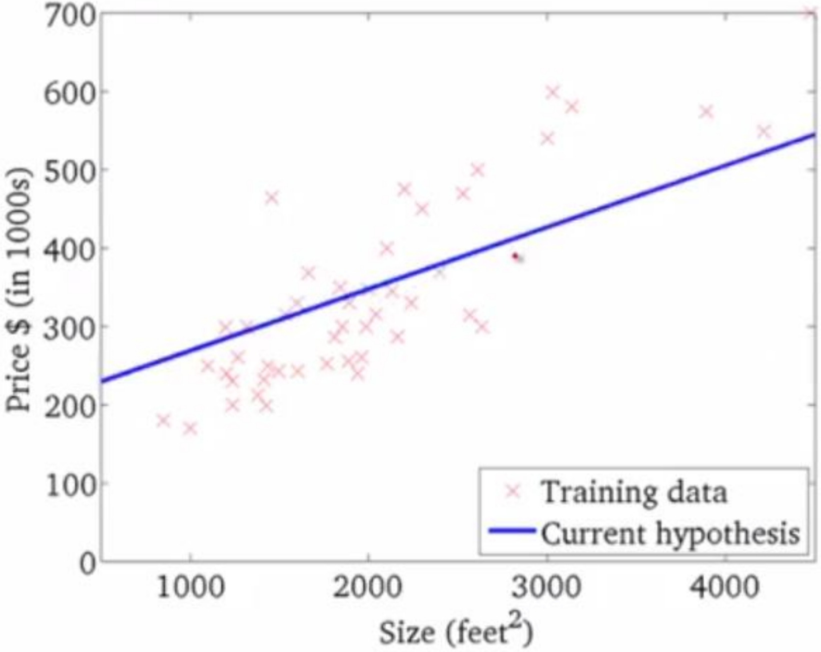
(function of the parameters θ_0, θ_1)



Andrew Ng

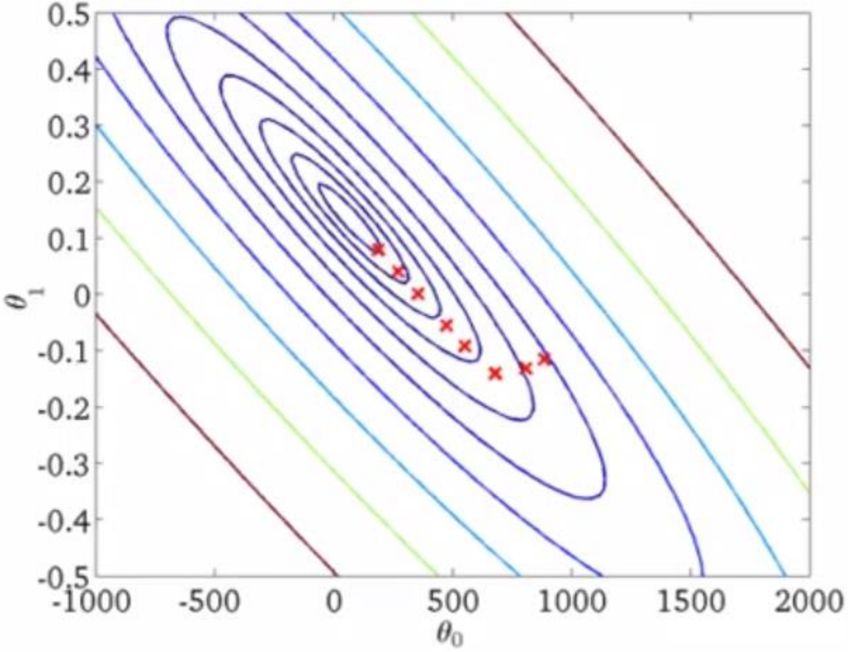
$h_\theta(x)$

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

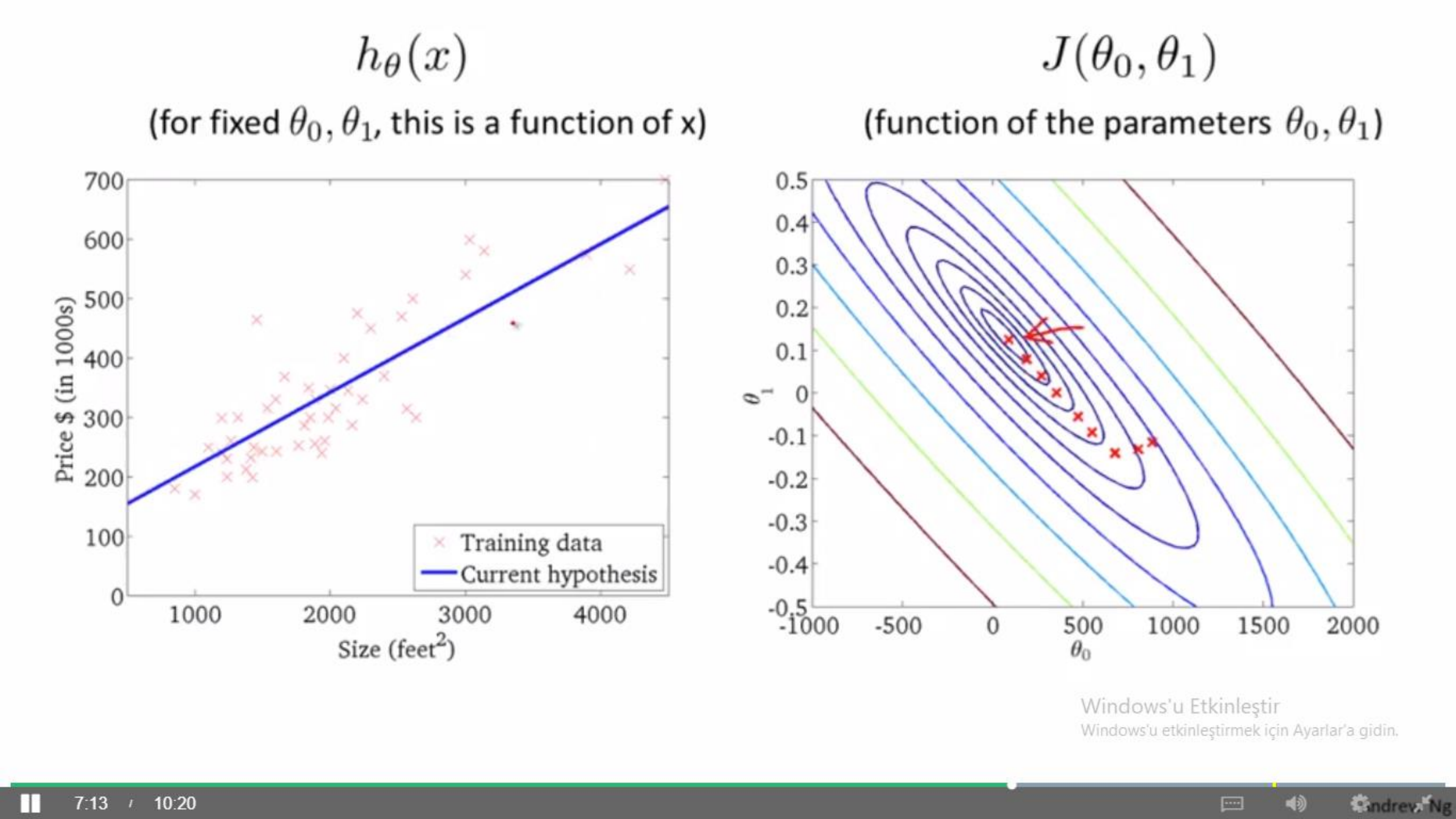


$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



Andrew Ng



$h_\theta(x)$

(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)

The Best Fitting Line (from END2991)

- "**best**" line: minimum SSE.
- "**least squares criterion**," which says
- "minimize the **S**um of the **S**quared prediction **E**rrors (SSE) or residual sum of squares.
- Find b_0 and b_1 that minimize:

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2$$

Method of Least Squares (from END2991)

- Differentiating SSE with respect to b_0 and b_1 and setting the resulting equations to zero we get:

- Solving these two equations will yield the computing formulas for b_0 and b_1 as follows:

$$\frac{\partial(SSE)}{\partial b_0} = -2\sum_{i=1}^{n}(y_i - b_0 - b_1 x_i) \qquad \frac{\partial(SSE)}{\partial b_1} = -2\sum_{i=1}^{n}(y_i - b_0 - b_1 x_i)x_i.$$

Method of Least Squares (from END2991)

$$nb_0 + b_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i, \qquad b_0 \sum_{i=1}^{n} x_i + b_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

$$b_1 = \frac{n \sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i \right) \left(\sum_{i=1}^{n} y_i \right)}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$b_0 = \frac{\sum_{i=1}^{n} y_i - b_1 \sum_{i=1}^{n} x_i}{n} = \bar{y} - b_1 \bar{x}.$$