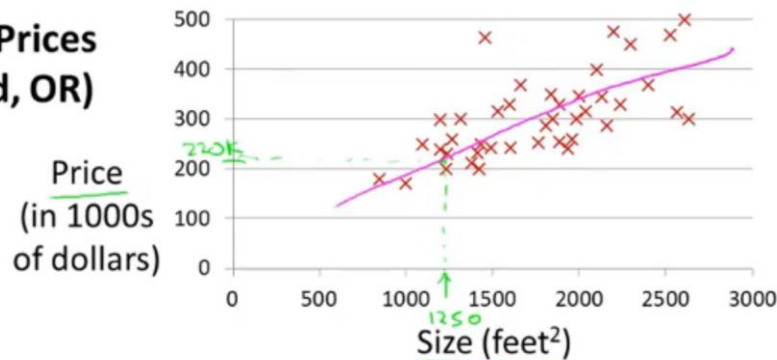


Linear regression with one variable

1. Model representation

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

Training set: data set given

(x, y) : one training example;

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

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:

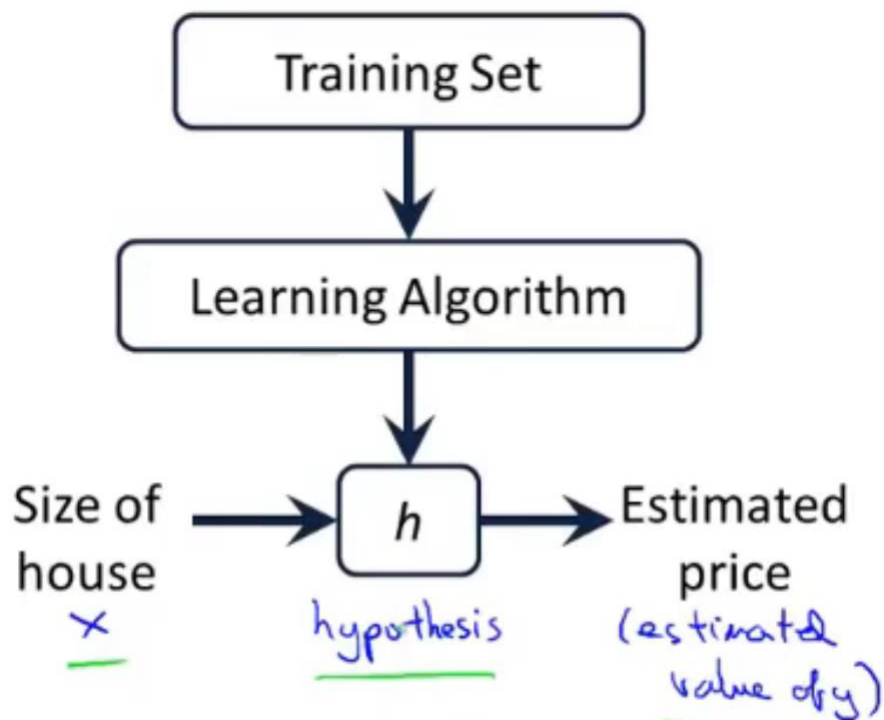
→ m = Number of training examples

→ x 's = “input” variable / features

→ y 's = “output” variable / “target” variable

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

h maps from x 's to y 's



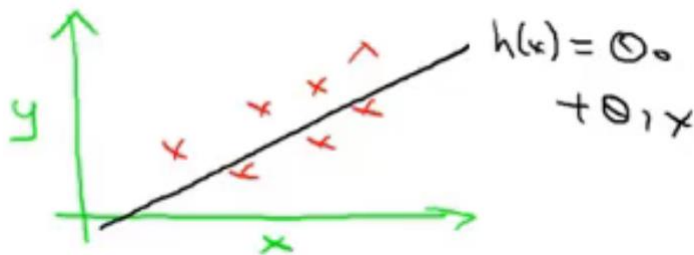
How do we represent h ?

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

shorthand: $h(x)$

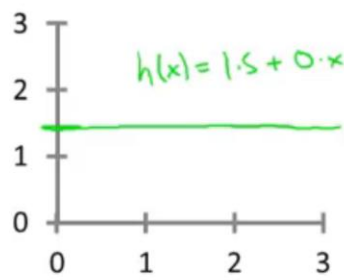
Linear regression with one variable.

Univariate linear regression.

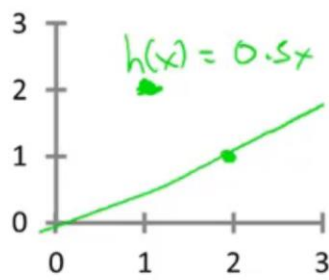


2. Cost function

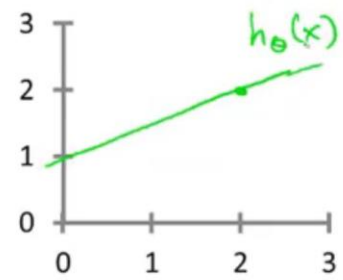
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



→ $\theta_0 = 1.5$
→ $\theta_1 = 0$

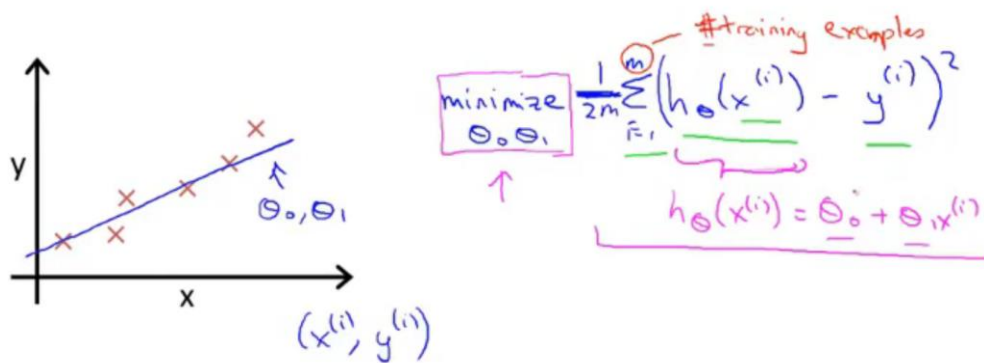


→ $\theta_0 = 0$
→ $\theta_1 = 0.5$



→ $\theta_0 = 1$
→ $\theta_1 = 0.5$

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



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

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

Cost function

Squared error function

3. Cost function intuition I

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$



Cost Function:

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

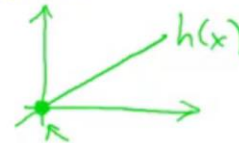
Goal: minimize $J(\theta_0, \theta_1)$

Simplified

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

$$\theta_0 = 0$$

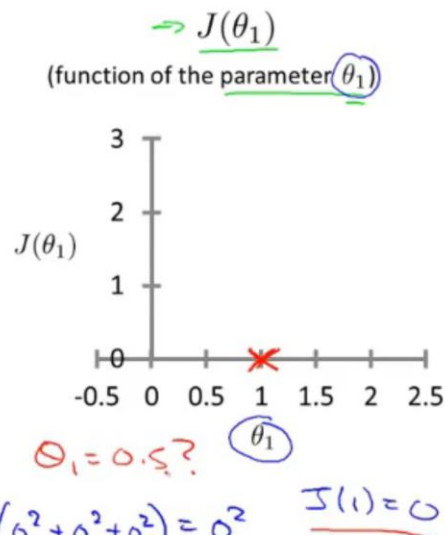
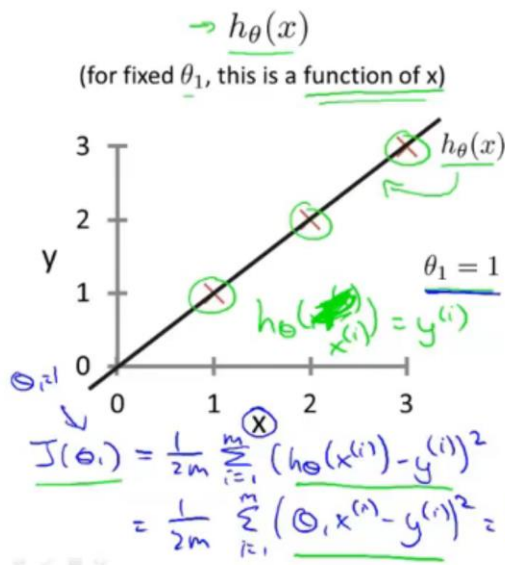
$$\theta_1$$



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

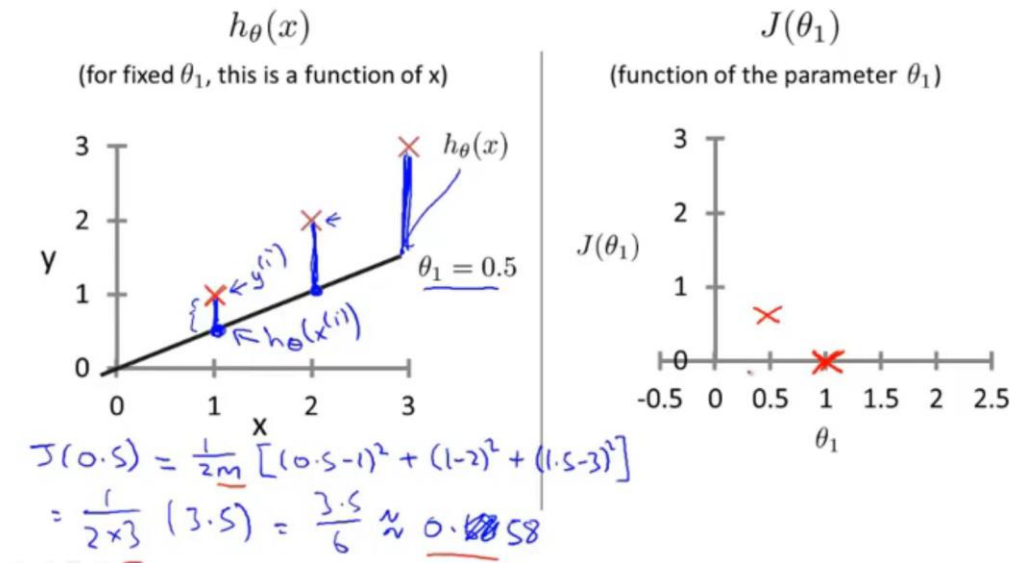
minimize $J(\theta_1)$

theta 1 = 1:

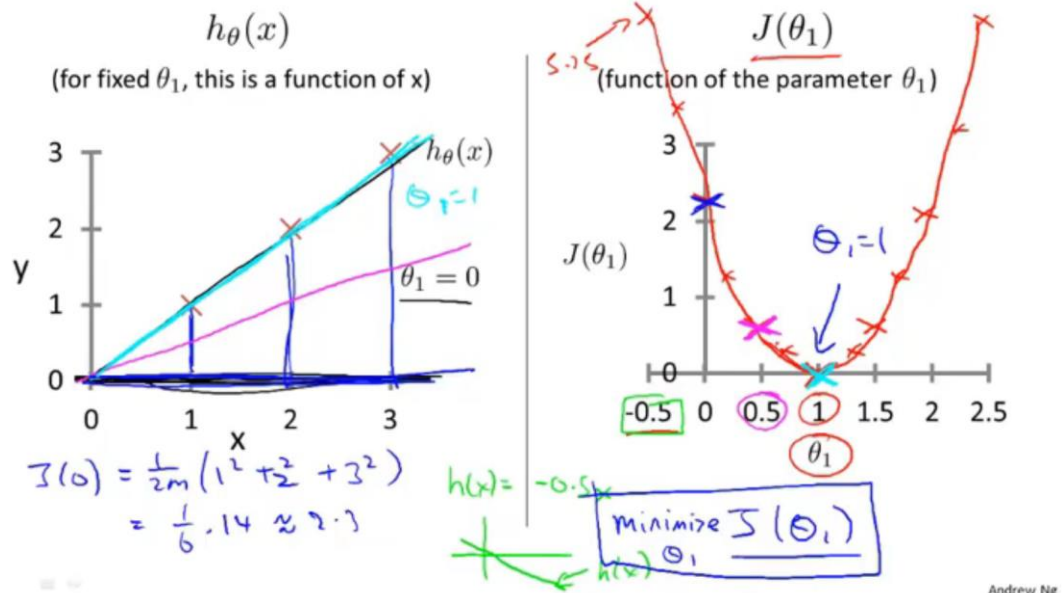


Andrew Ng

theta 1 = 0.5:

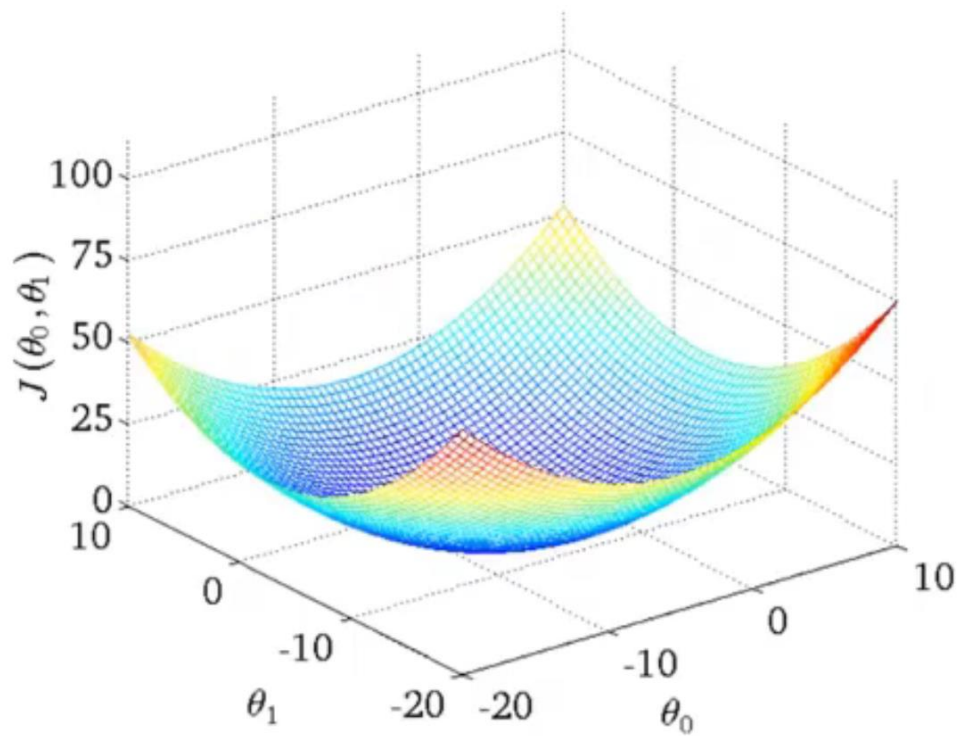


$\theta_1 = 0$:

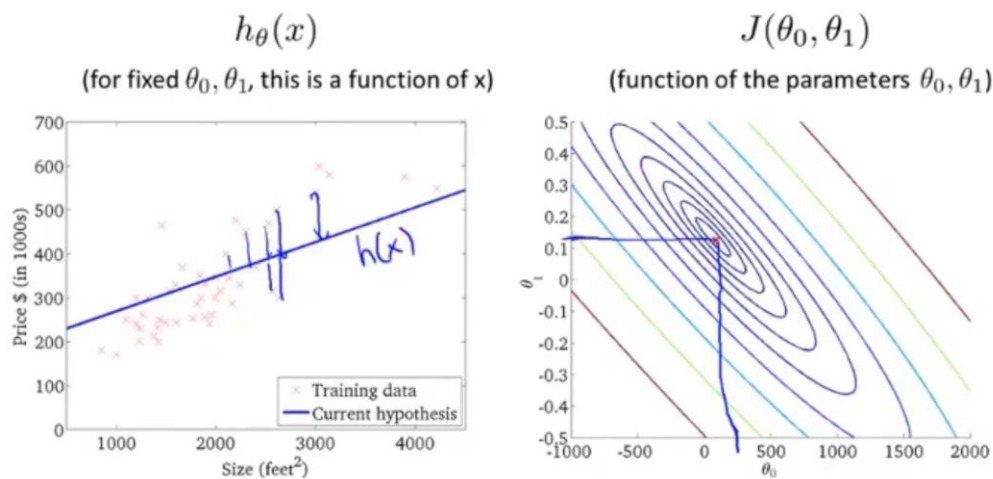


4. Cost function intuition II

bowl shape (3-D surface plot)



The minimum, the bottom of the bowl is this point right there, this middle of these concentric ellipses.

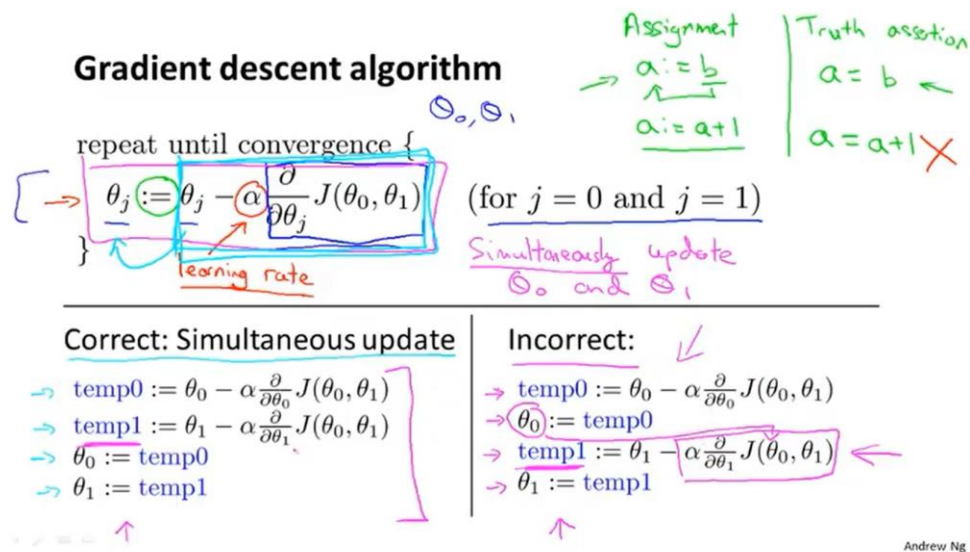


5. Gradient descent

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

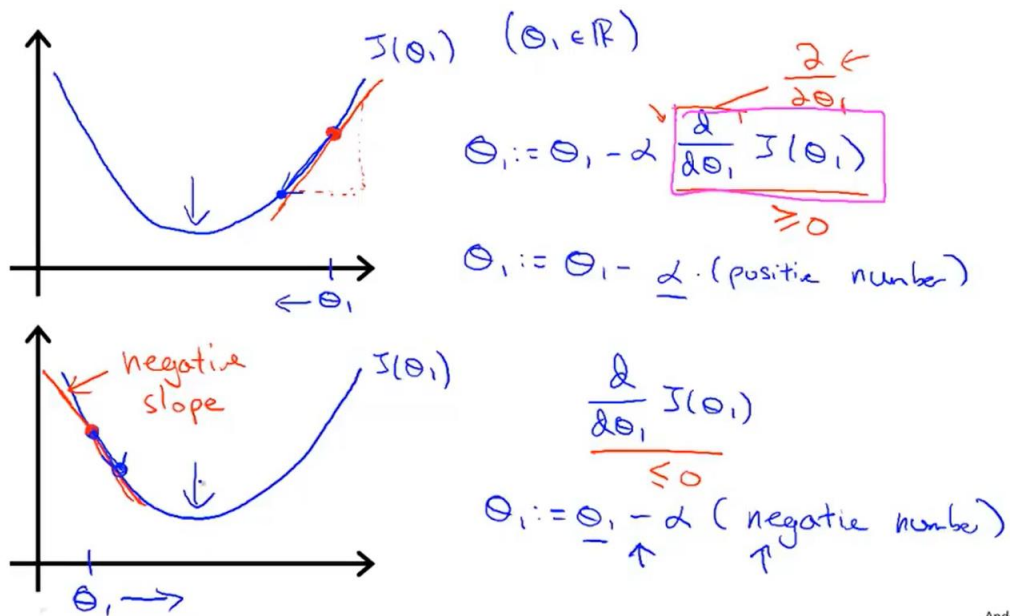
Simultaneous update!



learning rate: how big a step we take downhill with gradient descent.

6. Gradient descent intuition

derivative term:

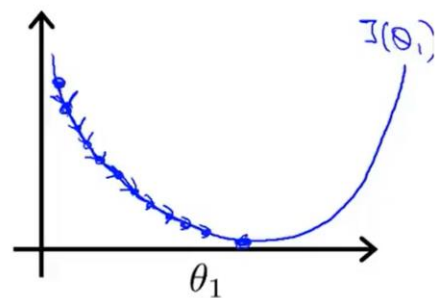


Andrew Ng

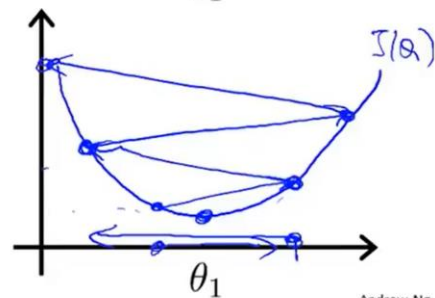
learning rate alpha:

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

If α is too small, gradient descent can be slow.

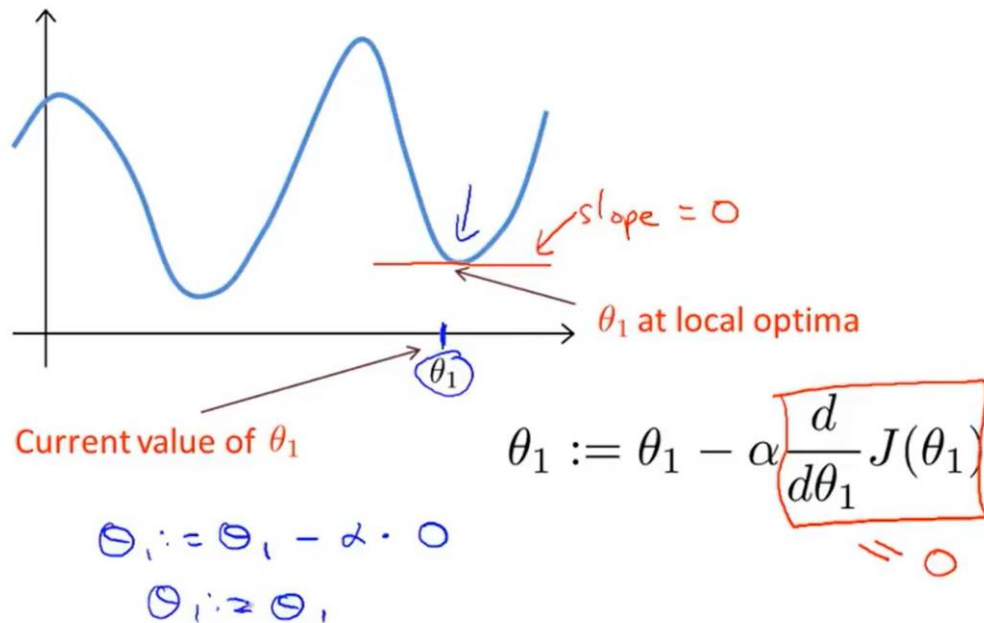


If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



Andrew Ng

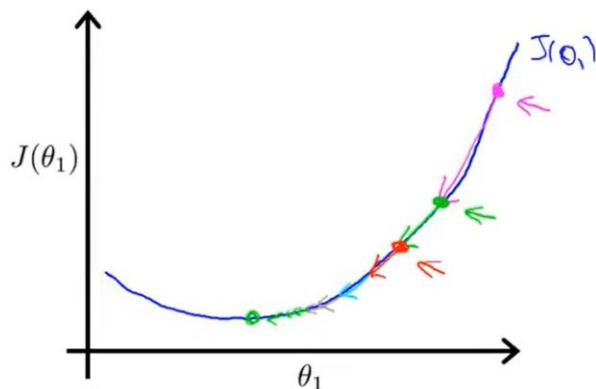
local optimum:



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.



Andrew Ng

7. Gradient descent for linear regression

repeat until convergence

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

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

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

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

"Batch" Gradient Descent

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