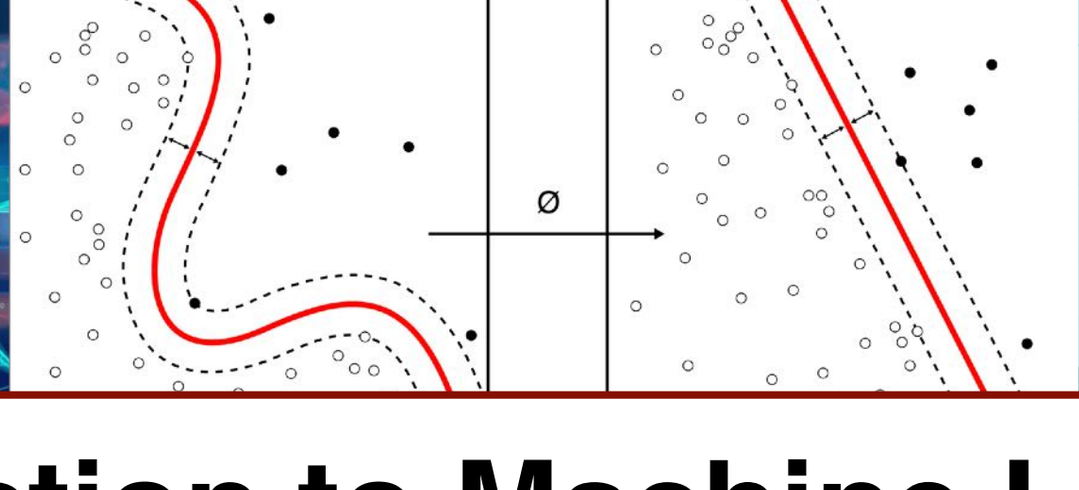
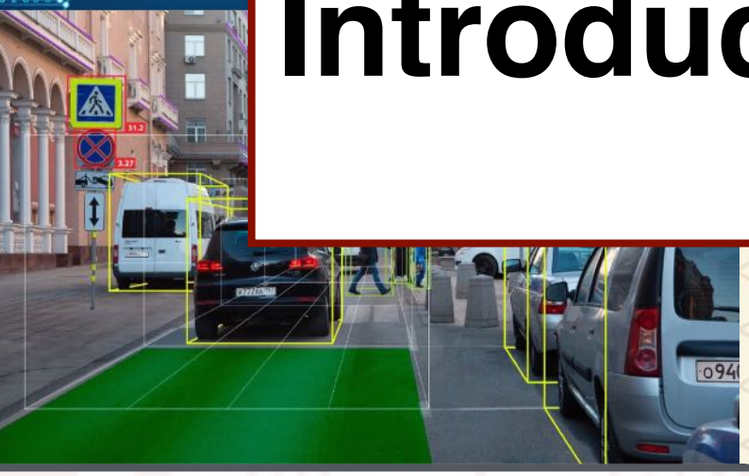
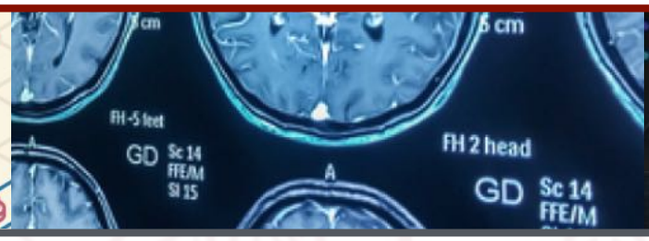
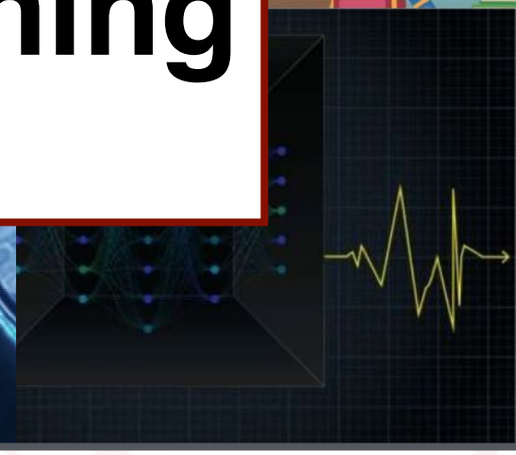



Introduction to Machine Learning

SCP8084699 - LT Informatica







Linear Regression, Gradient Descent

Prof. Lamberto Ballan

University of Padova, A.Y. 2022/23 March 9, 2023

DIPARTIMENTO MATEMATICA
Department of Mathematics “Tullio Levi-Civita”

Summary

- Inductive (learning) Bias
- Bias-Variance Tradeoff

- Linear Regression *Today*
  - (Univariate) Linear Regression model
  - Cost Function - Intuition
  - Examples

Linear Regression

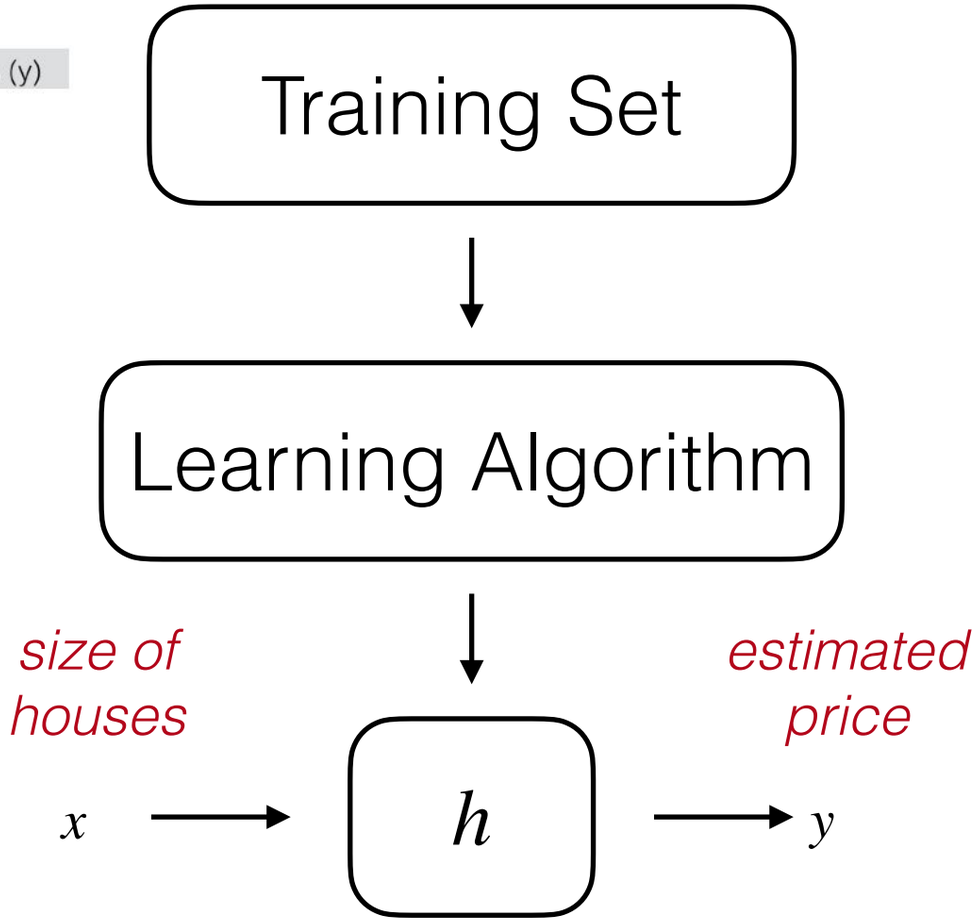
- **Example:** housing prices in Portland (OR)
  - Notation:
    - m = Number of training examples
    - x's = "input" variable / features
    - y's = "output" variable / target variable

Size in feet^2 (x)	Price ($) in 1K's (y)
2104	460
1416	232
1534	315
852	178
…	…

Linear Regression

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

Size in feet² (x)	Price ($) in 1K's (y)
2104	460
1416	232
1534	315
852	178
…	…

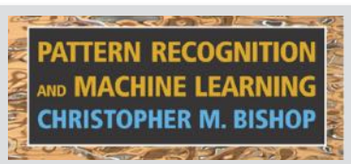


h maps from x's to y's

How do we represent h ?

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

Univariate linear regression model *(i.e. one variable)*



Chapter 1.0-1.1; 3.1

Linear Regression

- Training set:

Size in feet^2 (x)	Price ($) in 1K's (y)
2104	460
1416	232
1534	315
852	178
…	…

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

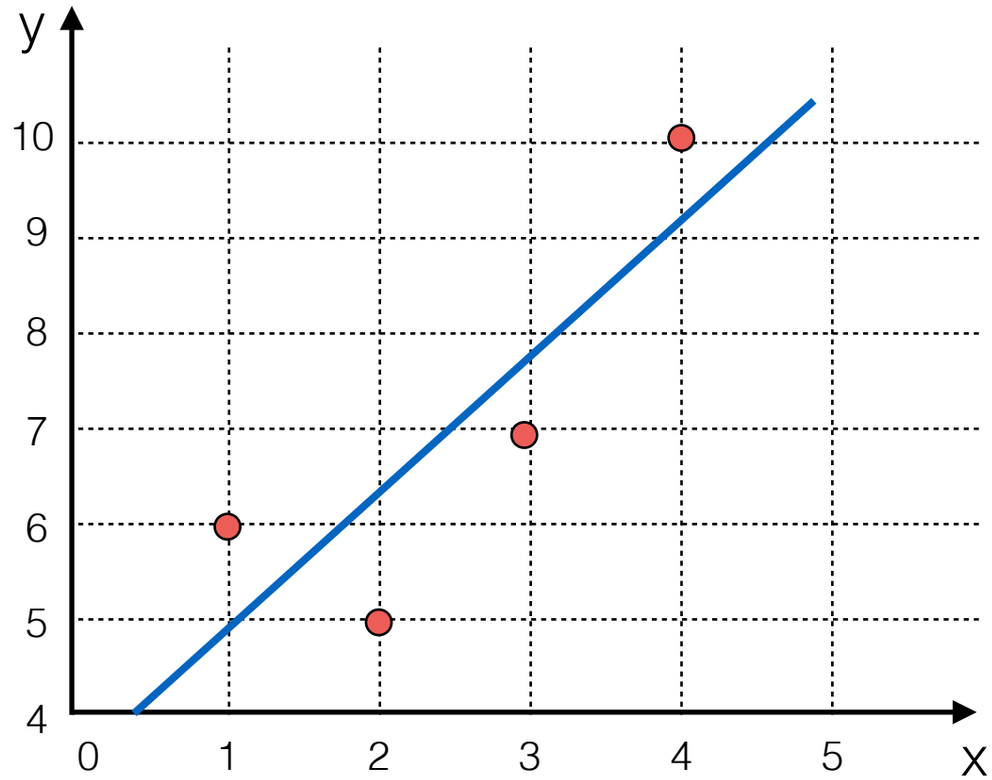
θ_i *'s parameters*

- How to chose the parameters?

Linear Regression

- Let's start with a simple example:

x	y
1	6
2	5
3	7
4	10



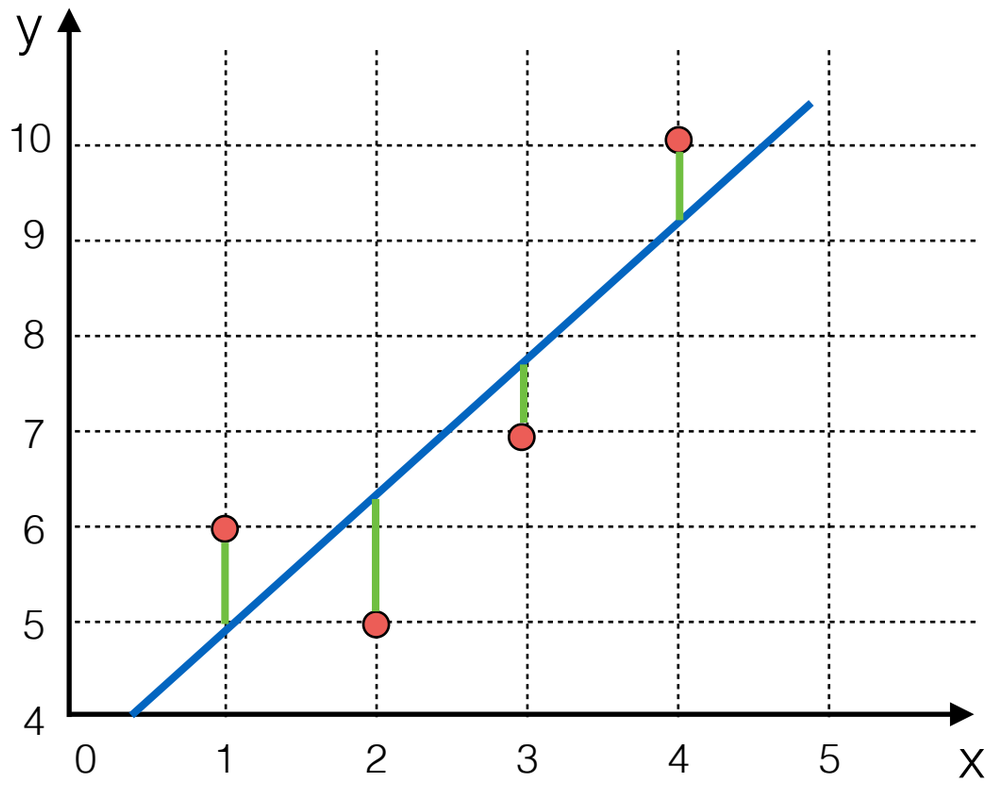
$$h_\theta(\mathrm{x}) = \theta_0 + \theta_1\ \mathrm{x}$$

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

Linear Regression

m training examples

x	y
1	6
2	5
3	7
4	10



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

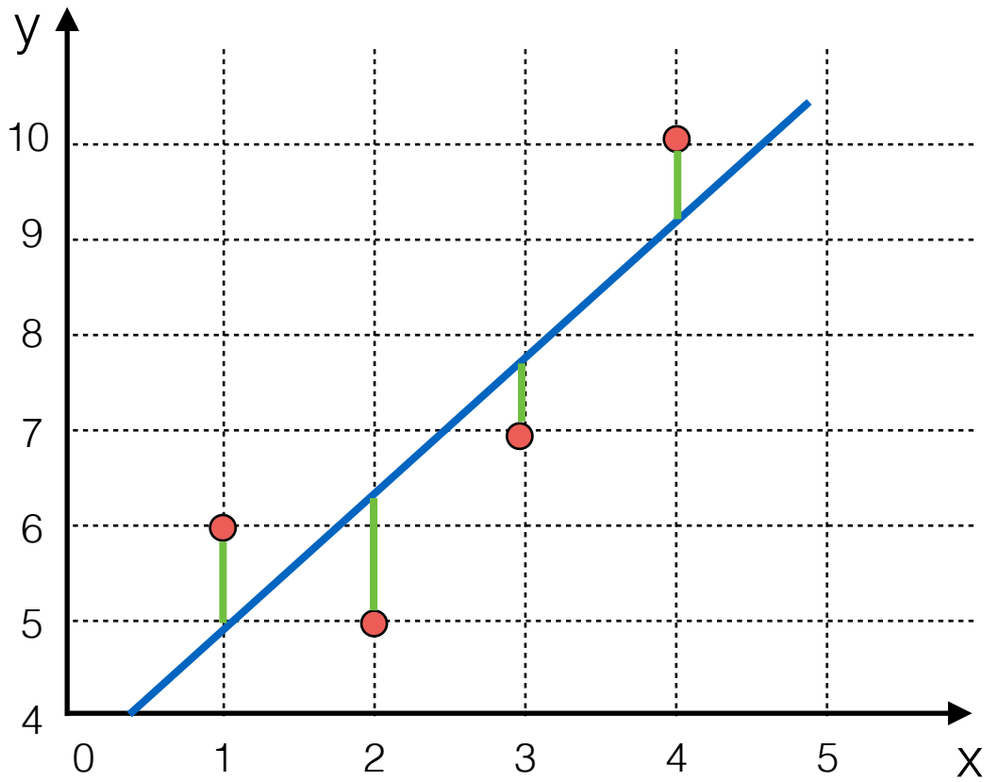
- Idea: choose θ_0,θ_1 so that $h_\theta(\mathrm{x})$ is close to y for our training examples $\{(x^{(\mathrm{i})}, y^{(\mathrm{i})})\}$
- Least squares: sum of squared distances (residuals)

Cost Function - Intuition

m training examples

x	y
1	6
2	5
3	7
4	10

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



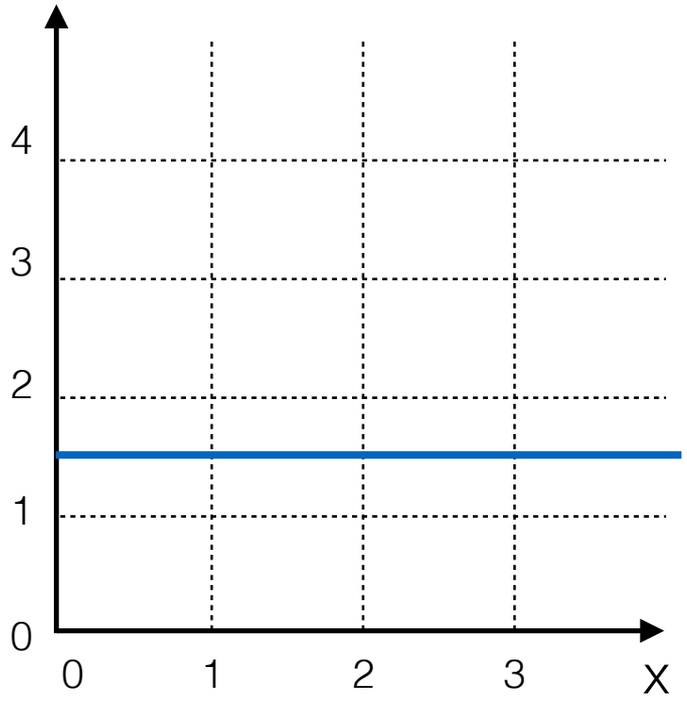
- Residuals: $\{h_\theta(\mathrm{x}^{(\mathrm{i})}) - \mathrm{y}^{(\mathrm{i})}\}$
- Cost function: $\mathrm{J}(\theta_0, \theta_1) = \dfrac{1}{2\mathrm{m}} \sum_{\mathrm{i}=1}^{\mathrm{m}} (h_\theta(\mathrm{x}^{(\mathrm{i})}) - \mathrm{y}^{(\mathrm{i})})^2$
- Objective: $\underset{\theta_0,\theta_1}{\text{minimize}}\ \mathrm{J}(\theta_0, \theta_1)$

Cost Function - Intuition

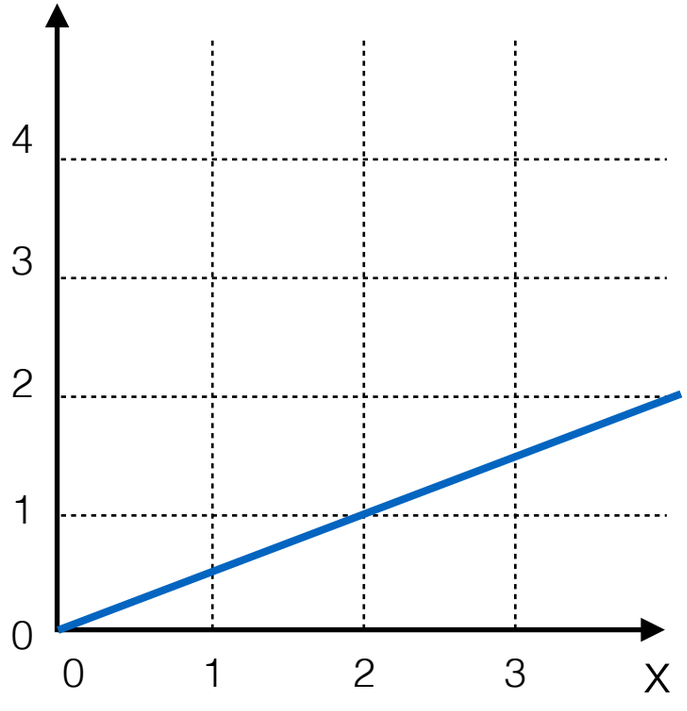
- How to choose θ_i's? *Let's see a few examples…*

$h_\theta(x) = \theta_0 + \theta_1 x$ $\qquad\qquad$ θ_0, θ_1

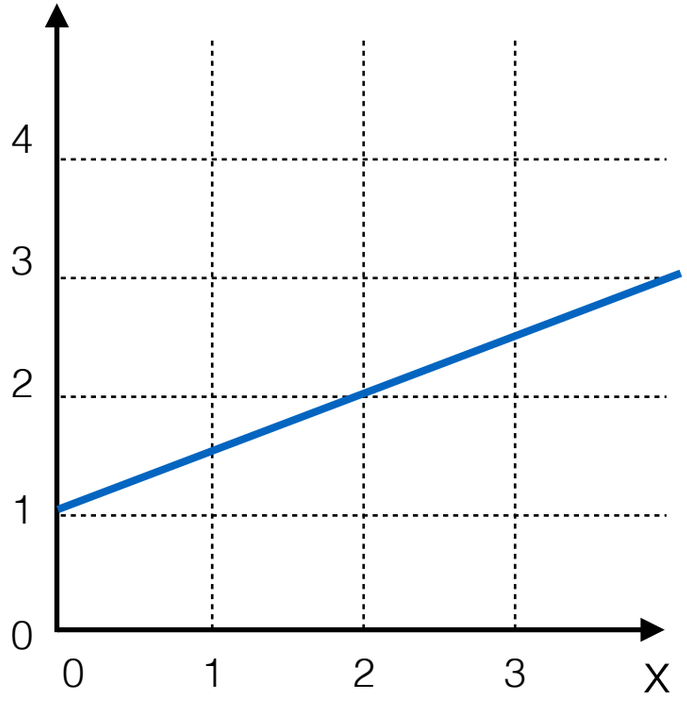
hypothesis $\qquad\qquad$ *parameters*



$\theta_0=3/2, \theta_1=0$



$\theta_0=0, \theta_1=1/2$



$\theta_0=1, \theta_1=1/2$

Adapted from slide by A. Ng

Cost Function - Intuition

- Hypothesis:

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

- Parameters: θ_0, θ_1

- Cost Function:

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

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

Simplified version

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

θ_1

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

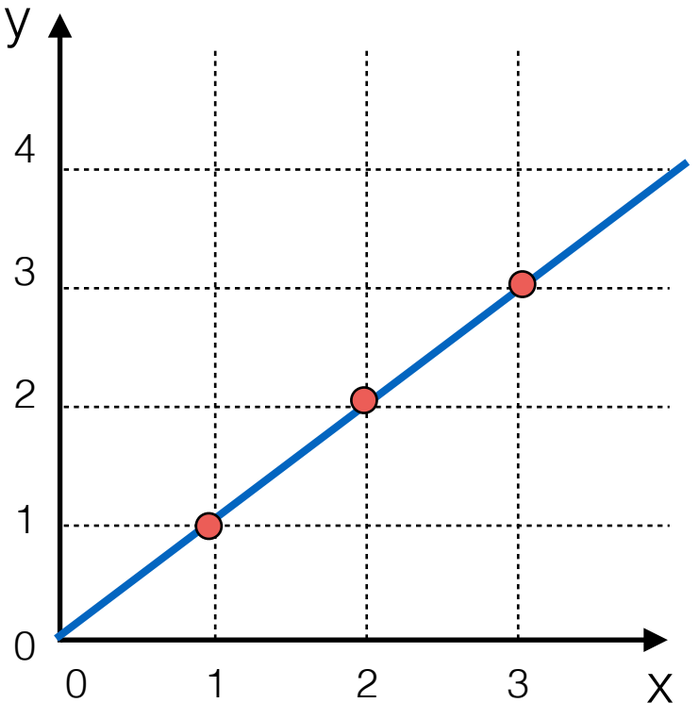
$\underset{\theta_1}{\text{minimize}}\ \mathrm{J}(\theta_1)$

Adapted from slide by A. Ng

Cost Function - Intuition

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

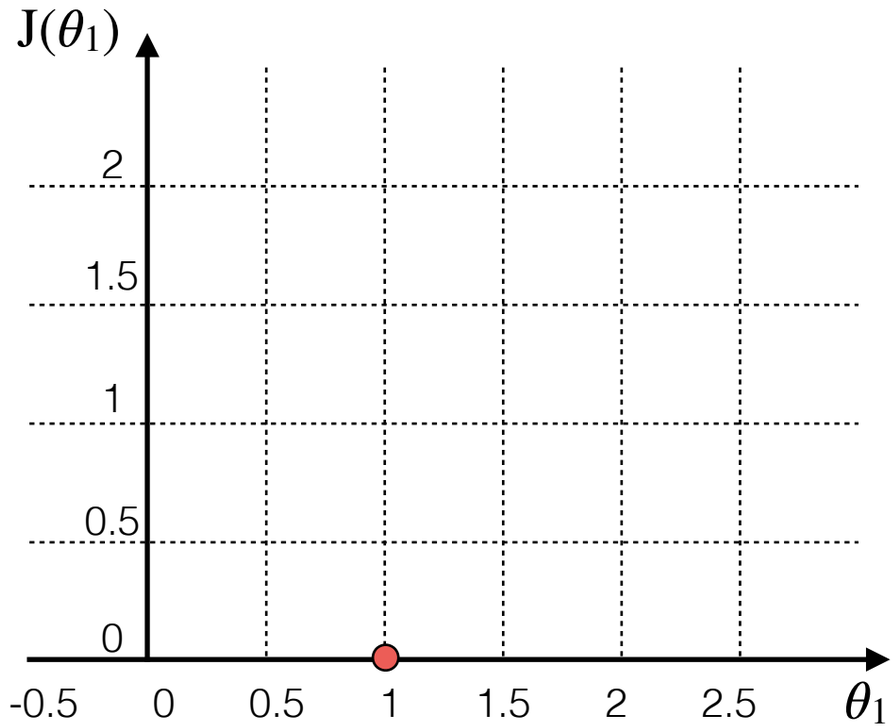
(for fixed θ_1 this is a function of x)



θ_1=1

$\mathrm{J}(\theta_1)$

(function of the parameter θ_1)

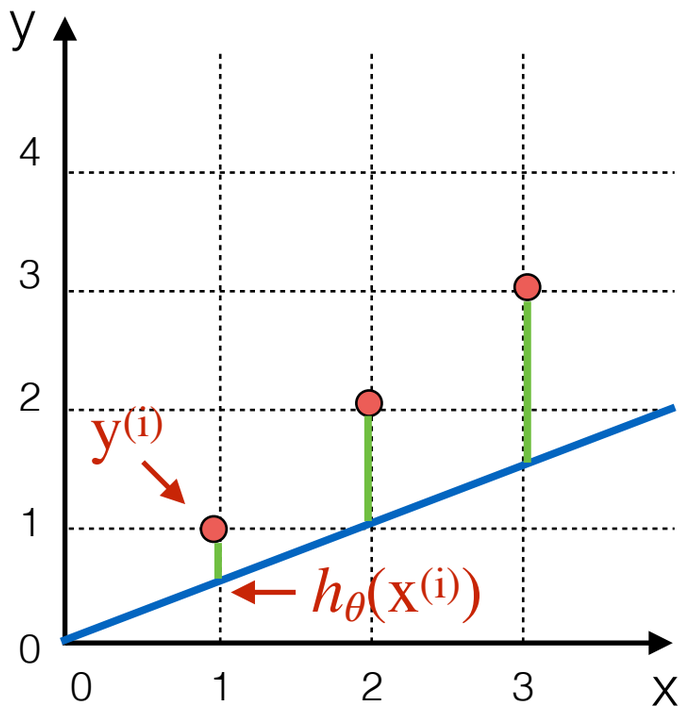


$$\mathrm{J}(\theta_1) = \frac{1}{2\mathrm{m}} \sum_{\mathrm{i}=1}^{\mathrm{m}} (h_\theta(\mathrm{x}^{(\mathrm{i})}) - \mathrm{y}^{(\mathrm{i})})^2 = \frac{1}{2\mathrm{m}} \sum_{\mathrm{i}=1}^{\mathrm{m}} (\theta_1\, \mathrm{x}^{(\mathrm{i})} - \mathrm{y}^{(\mathrm{i})})^2 \overset{\theta_1=1}{=} \frac{1}{2\mathrm{m}} (0+0+0)^2$$

Cost Function - Intuition

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

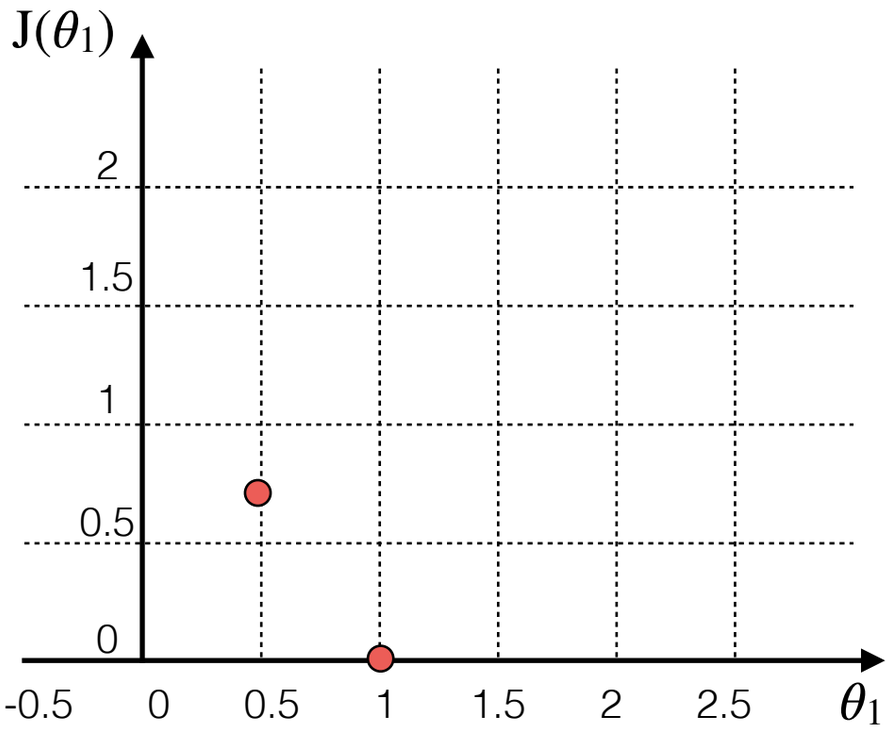
(for fixed θ_1 this is a function of x)



θ_1=1/2 ?

$\mathrm{J}(\theta_1)$

(function of the parameter θ_1)

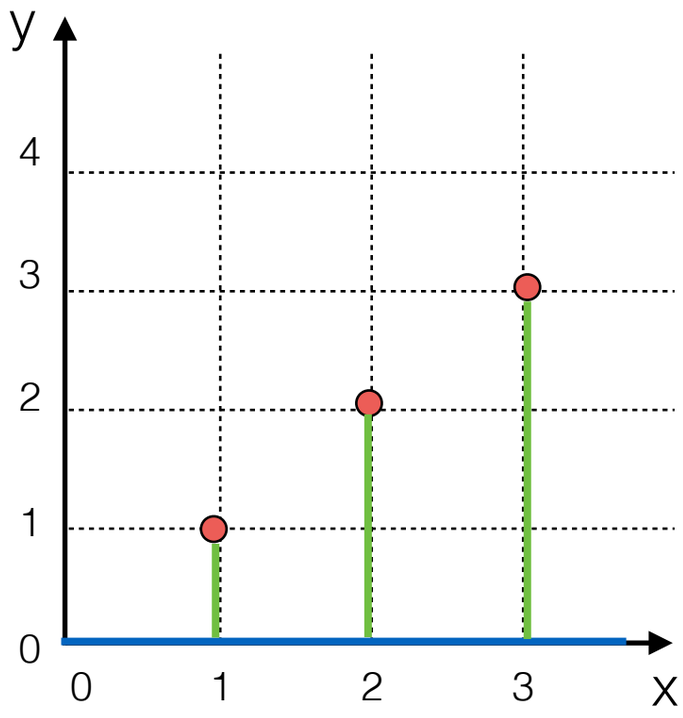


$$\mathrm{J}(0.5) = \frac{1}{2\mathrm{m}}\left[(0.5\text{-}1)^2 + (1\text{-}2)^2 + (1.5\text{-}3)^2\right] = \frac{1}{6}\left[0.25+1+2.25\right] \approx 0.67$$

Cost Function - Intuition

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

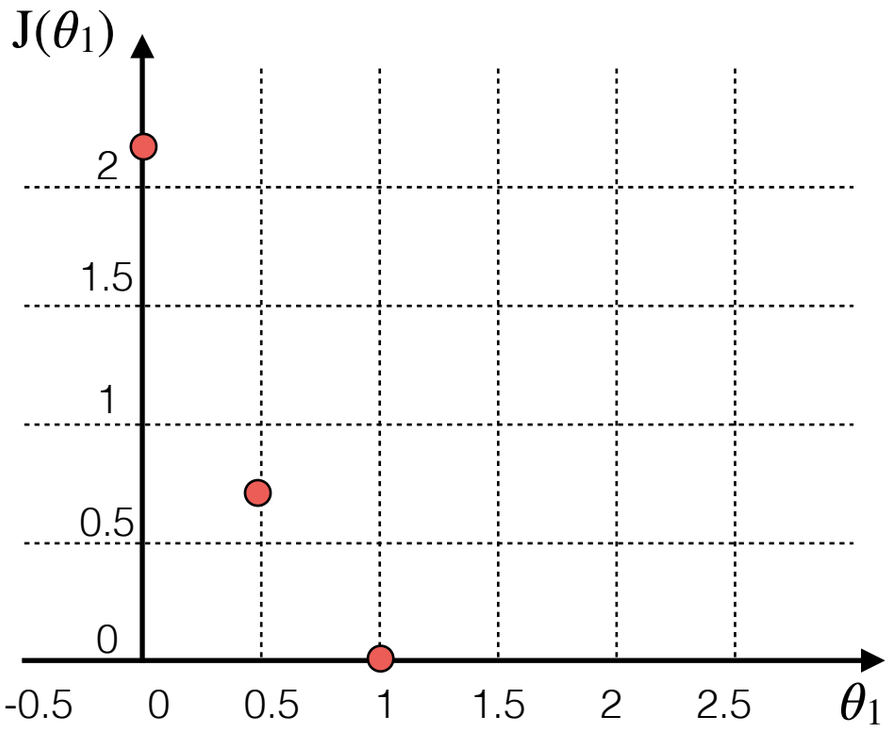
(for fixed θ_1 this is a function of x)



θ_1=0 ?

$J(\theta_1)$

(function of the parameter θ_1)

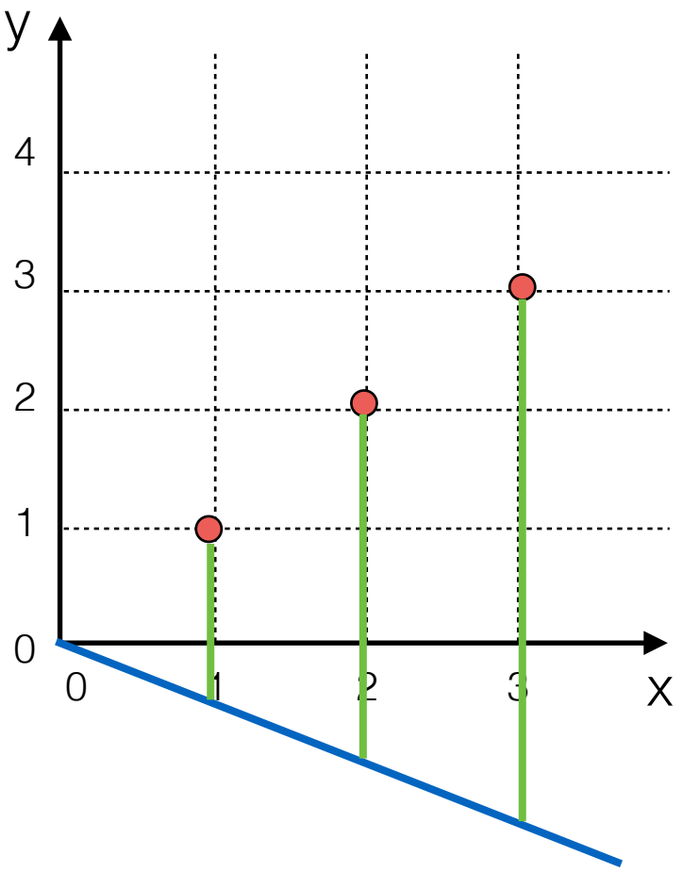


$$J(0) = \frac{1}{2m}\left[(0-1)^2 + (0-2)^2 + (0-3)^2\right] = \frac{1}{6}\left[1+4+9\right] \approx 2.33$$

Cost Function - Intuition

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

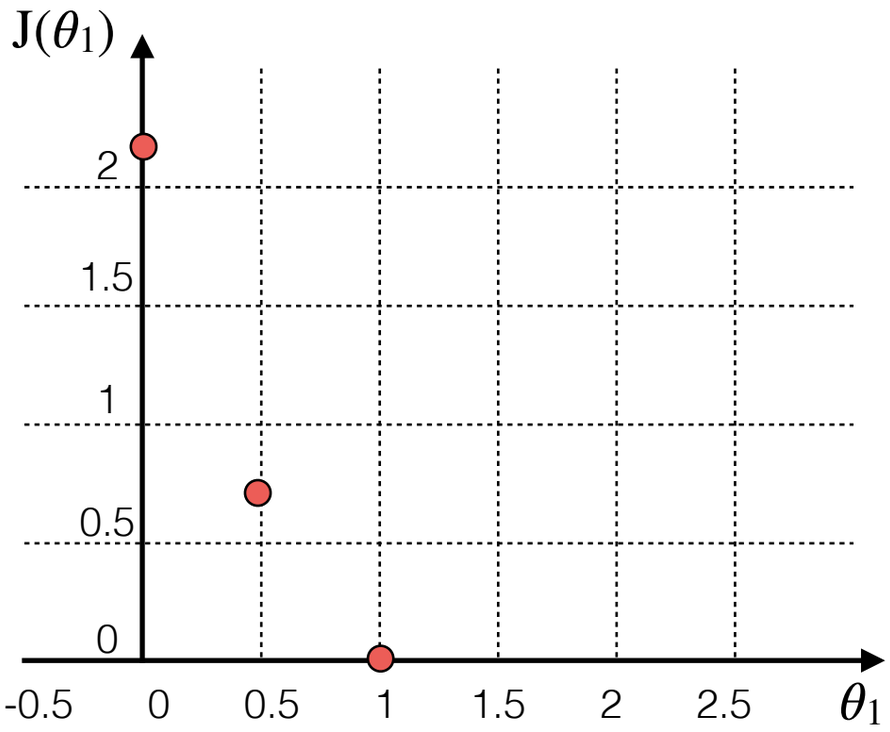
(for fixed θ_1 this is a function of x)



θ_1=-1/2 ?

$J(\theta_1)$

(function of the parameter θ_1)

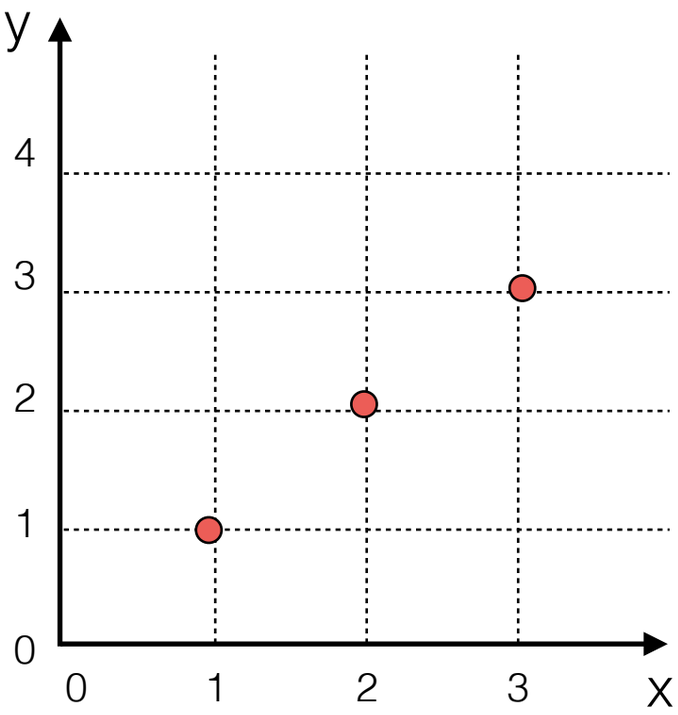


$$J(-0.5) = \frac{1}{2m}\left[(-0.5-1)^2+(-1-2)^2+(-1.5-3)^2\right] \approx 5.25$$

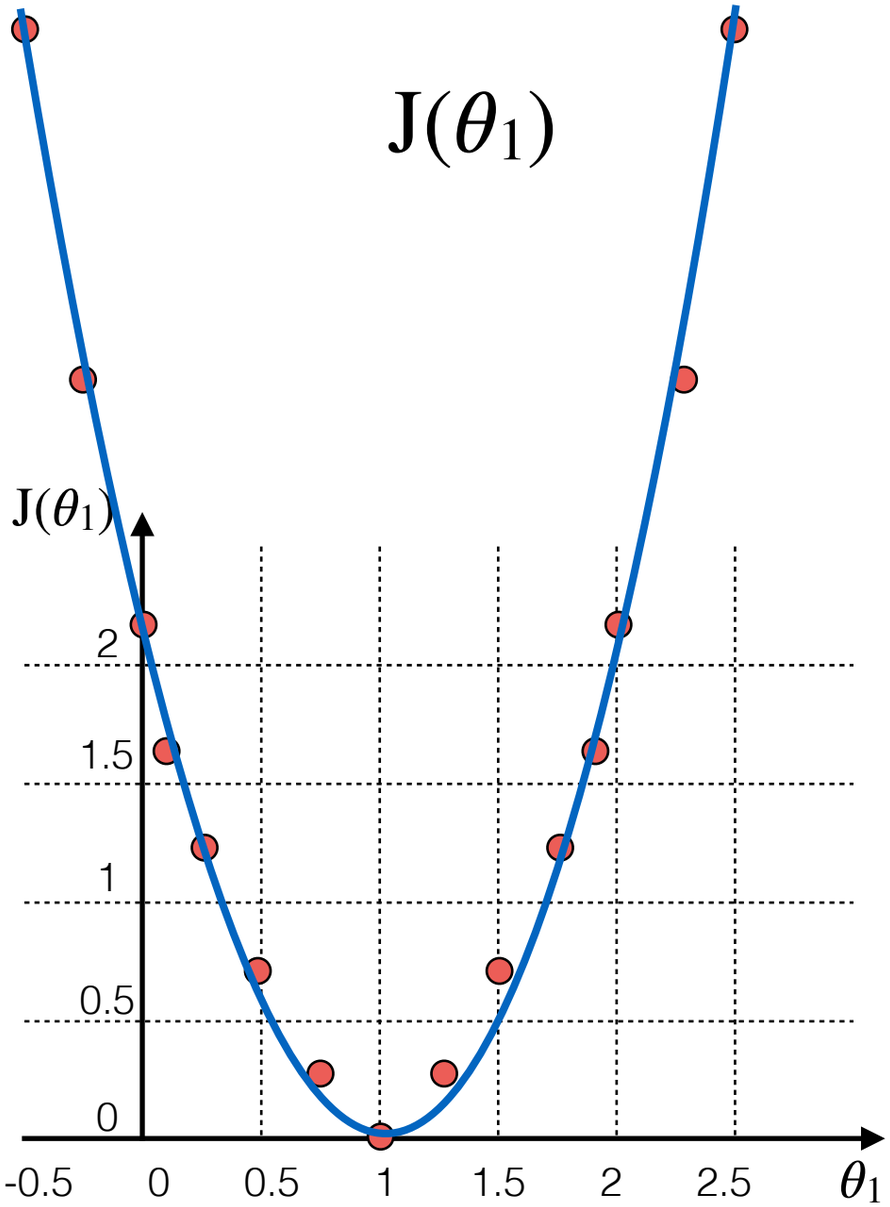
Cost Function - Intuition

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

(for fixed θ_1 this is a function of x)



θ_1= ?

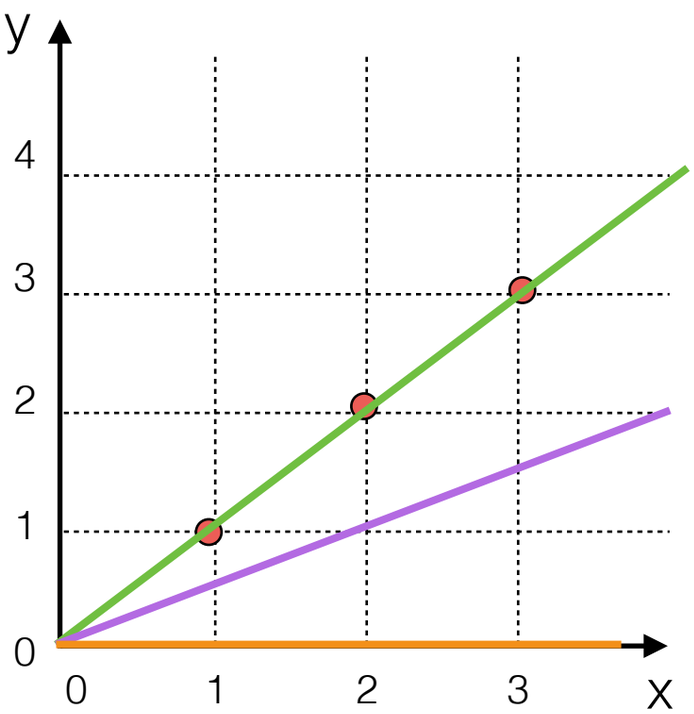


So, for different values, $\mathrm{J}(\theta_1)$ = …

Cost Function - Intuition

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

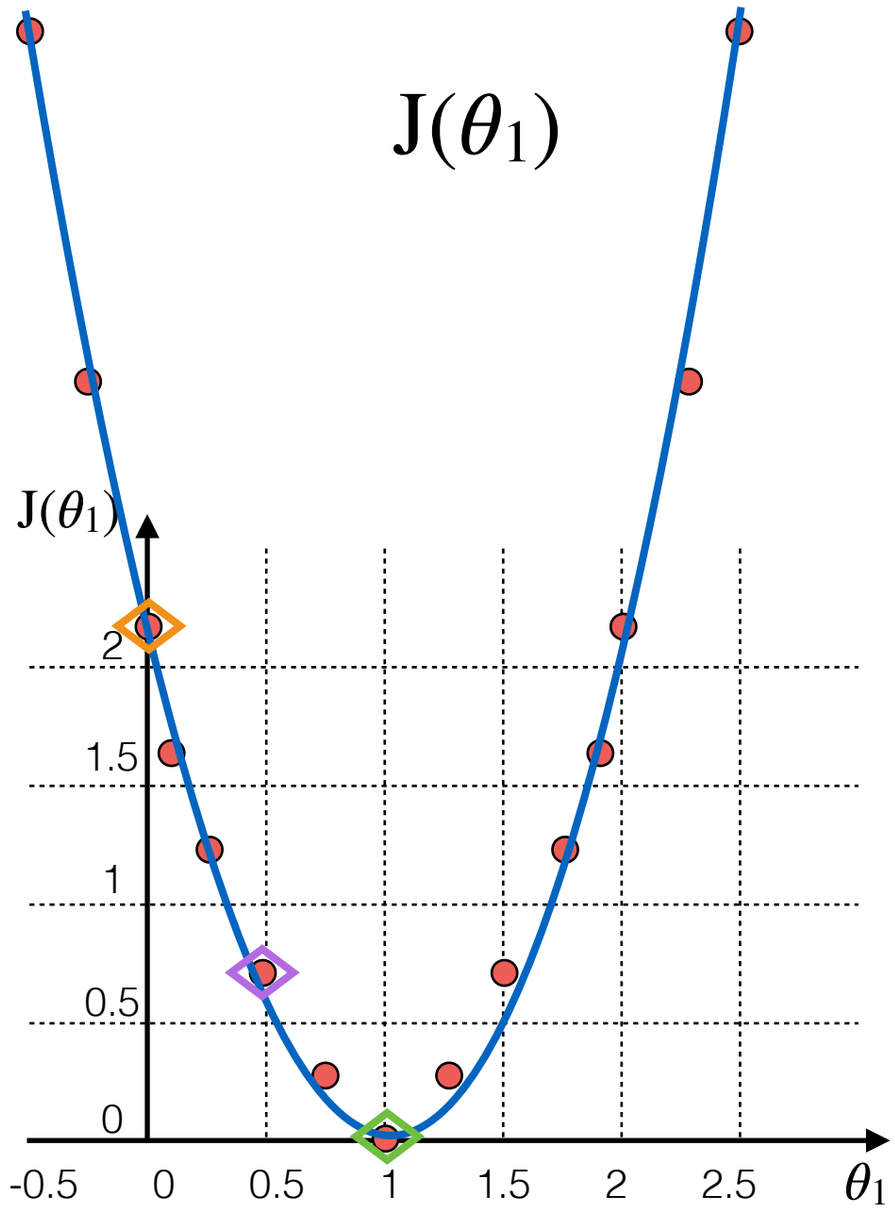
(for fixed θ_1 this is a function of x)



θ_1= 1

θ_1=1/2

θ_1=0

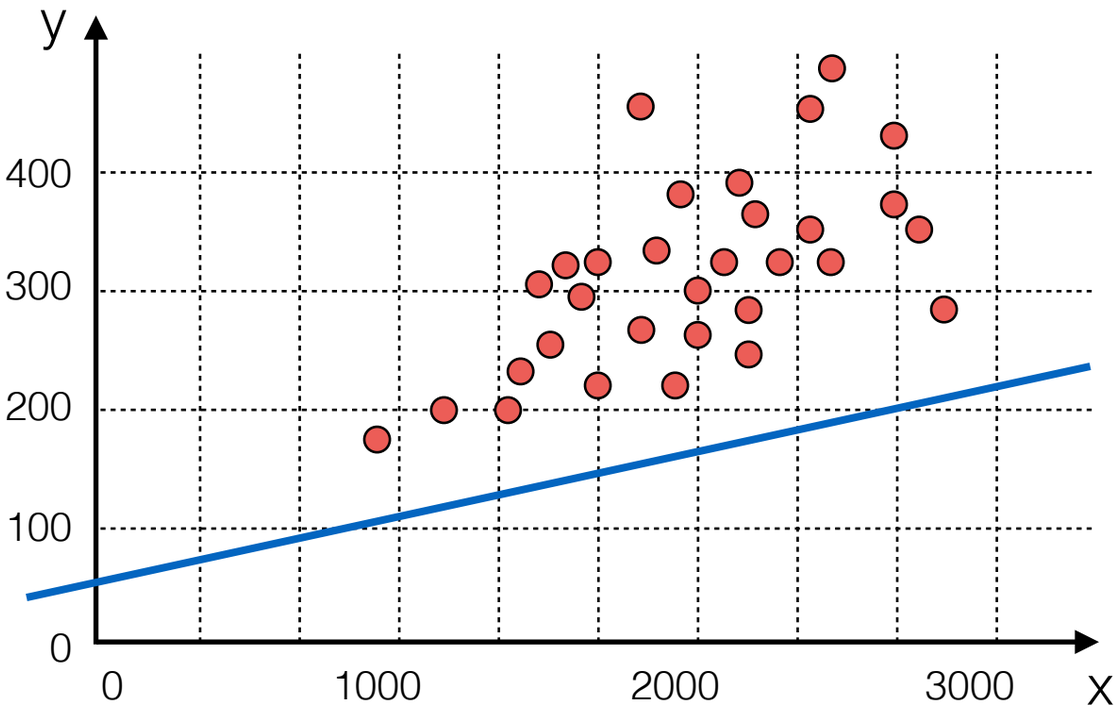


- To recap, each value of θ_1 corresponds to a different hypothesis (i.e. a different line through the data)

Cost Function - Intuition

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

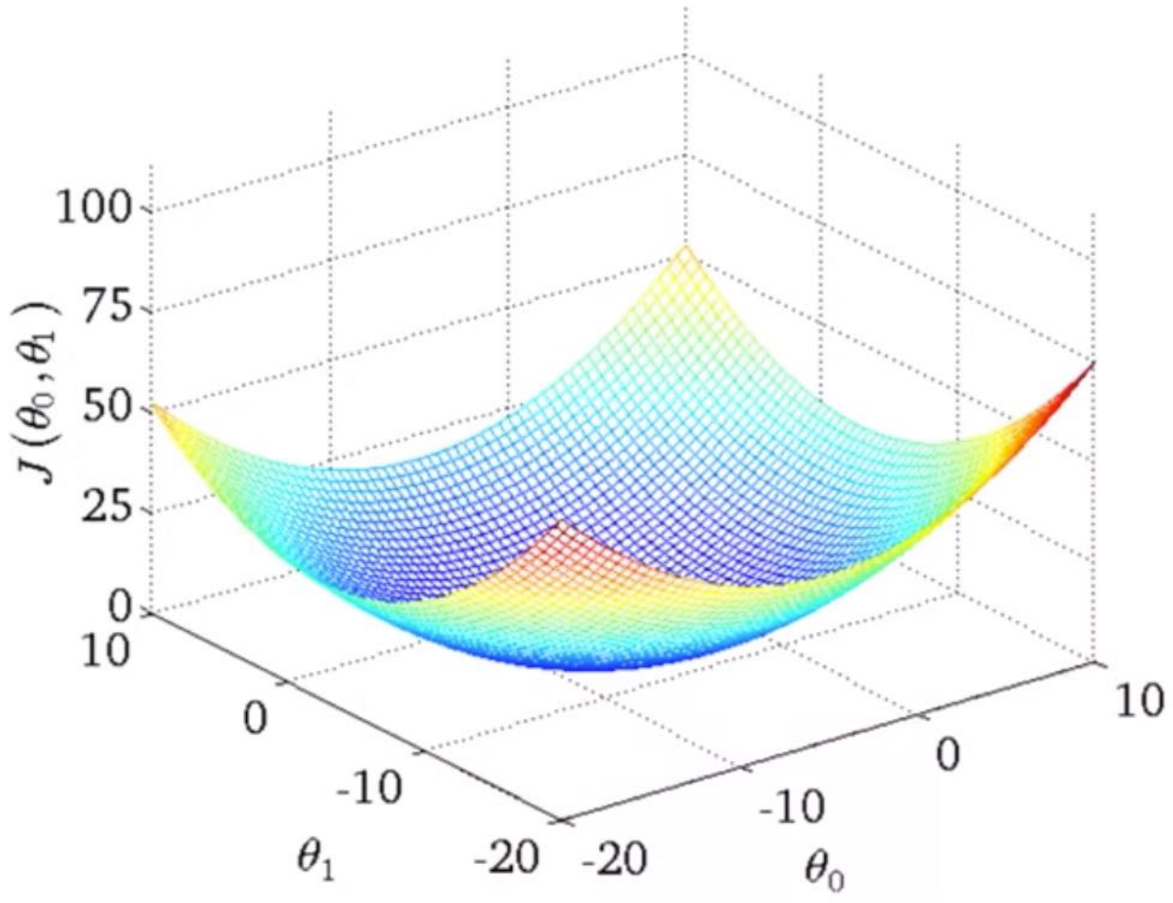
(for fixed θ_0, θ_1 this is a function of x)



$\theta_0 = 50, \theta_1 = 0.06$

$J(\theta_0, \theta_1)$

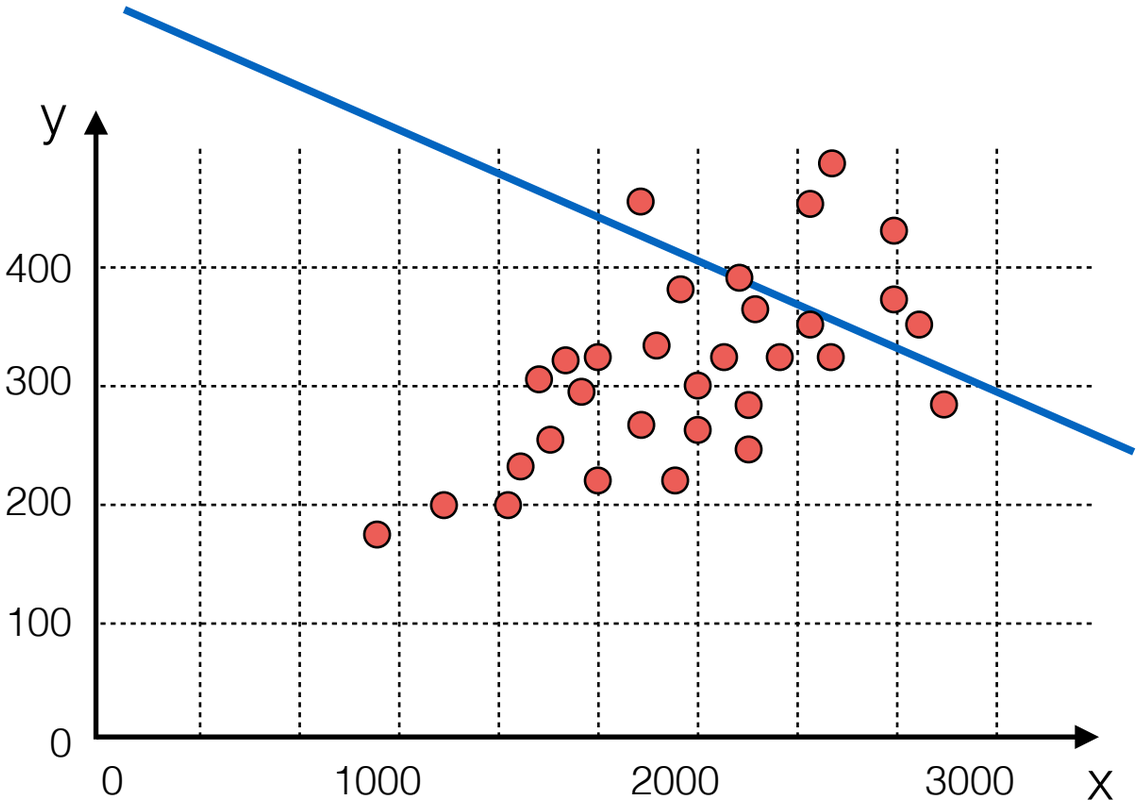
(function of the parameters θ_0, θ_1)



Cost Function - Intuition

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

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

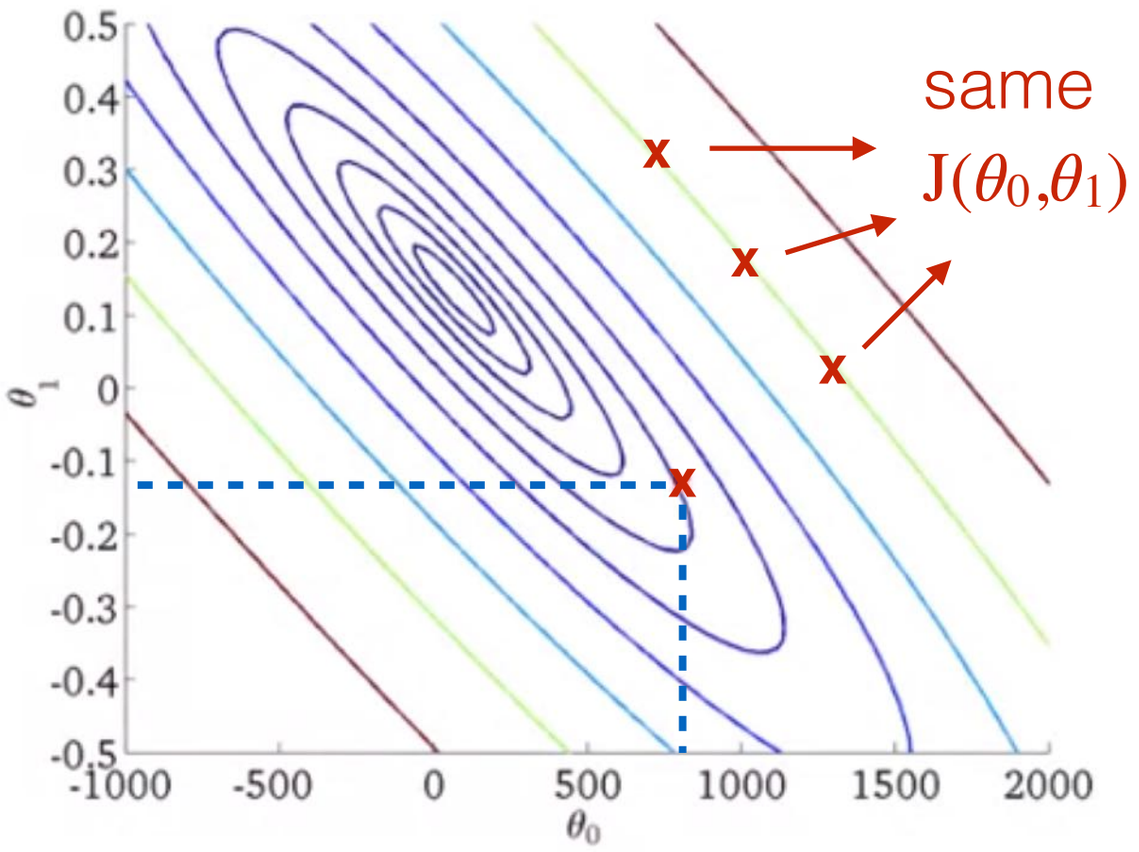


Training data

Current hypothesis

$J(\theta_0,\theta_1)$

(function of the parameters θ_0,θ_1)

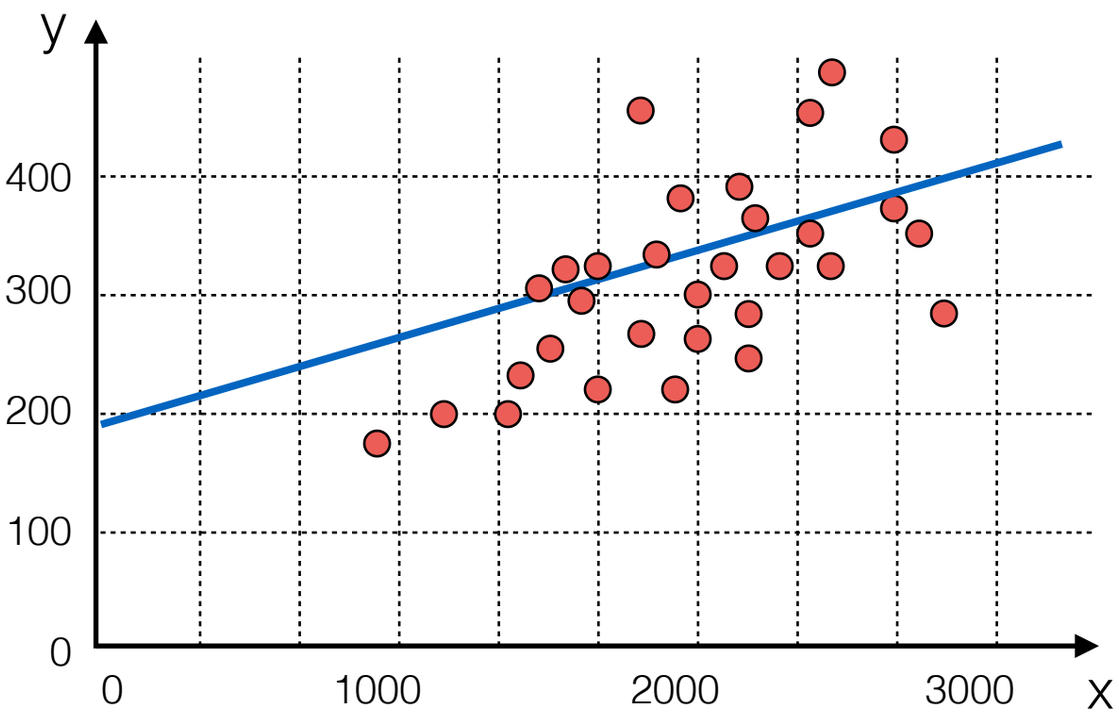


Contour plot

Cost Function - Intuition

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

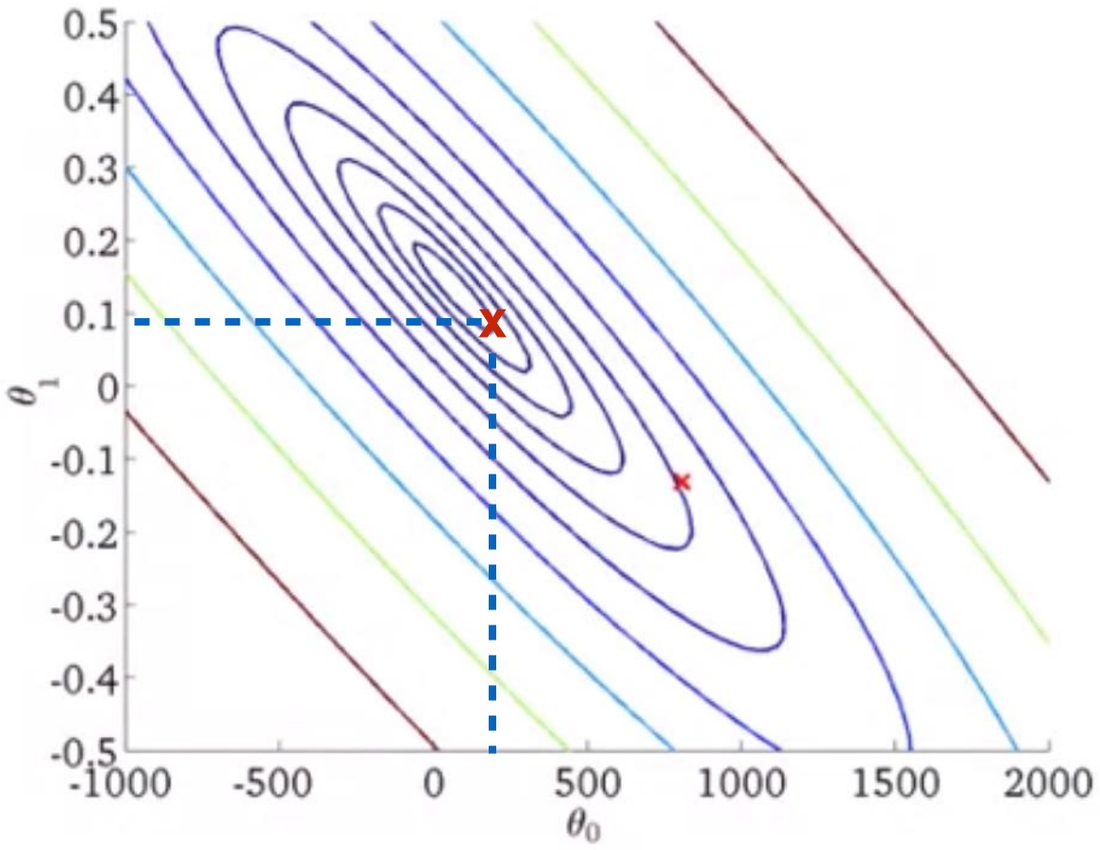
(for fixed θ_0,θ_1 this is a function of x)



- Training data
- Current hypothesis

$J(\theta_0,\theta_1)$

(function of the parameters θ_0,θ_1)



Contour plot

Adapted from slide by A. Ng

Parameter Learning

- What we really want is an efficient algorithm for automatically finding the value of θ_0 and θ_1

- In other words we need an efficient algorithm that minimizes the cost function $\mathbf{J}$

Gradient Descent

- You have a cost function $J(\theta_0, \ldots, \theta_n)$

- Objective: $\underset{\theta_0, \ldots, \theta_n}{\text{minimize}}\ J(\theta_0, \ldots, \theta_n)$

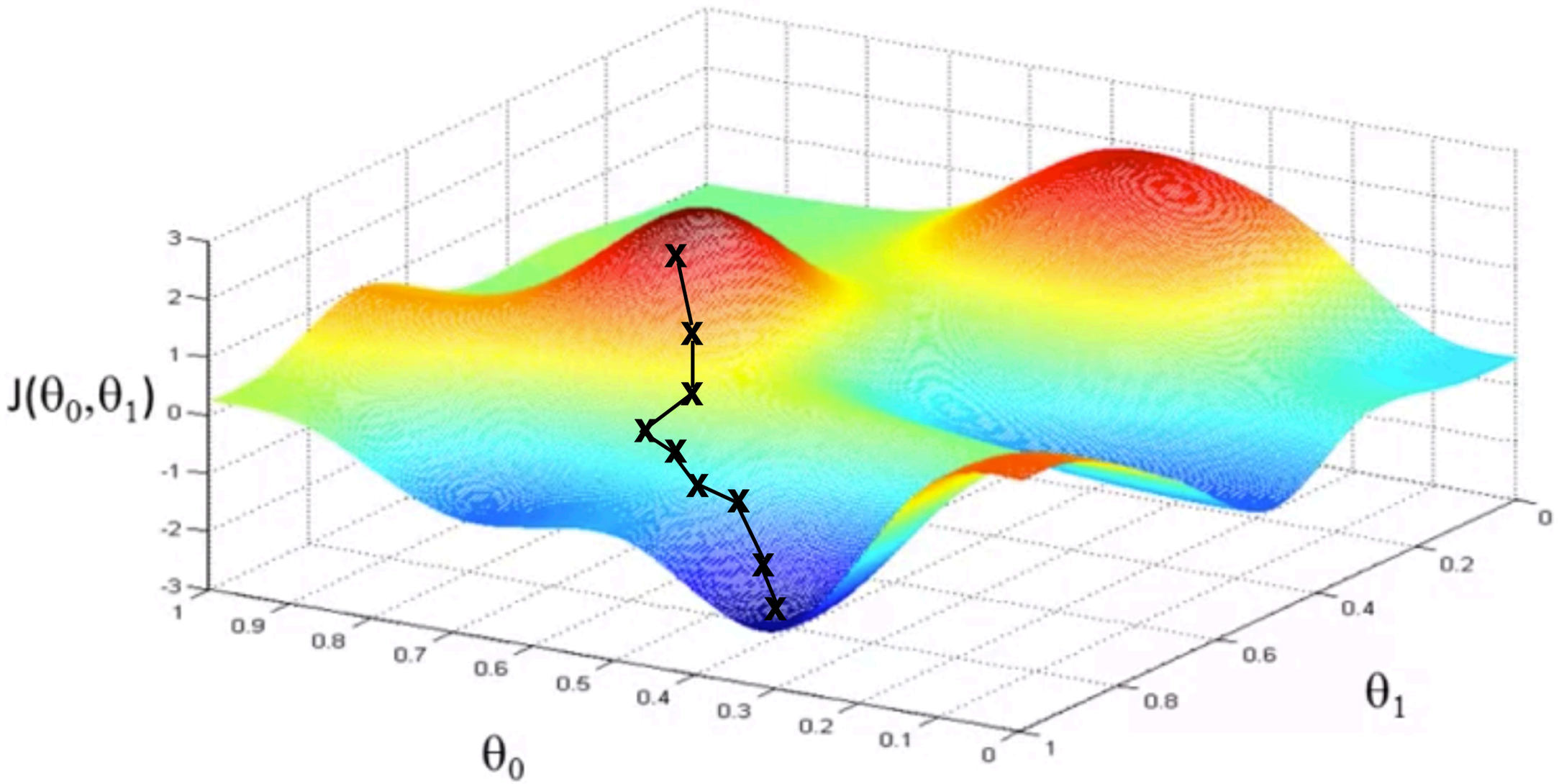
- Outline:
  - Start with some configuration of parameters θ_i's
  - Keep changing θ_i's parameters to reduce the cost function J until we (hopefully) end up at a minimum

Gradient Descent - Intuition

- "Fog in the mountains" analogy:
  - You are stuck in the mountains and visibility is low
  - Look at the steepness of the hill at you current position and proceed in the direction with the steepest descent

Gradient Descent - Intuition

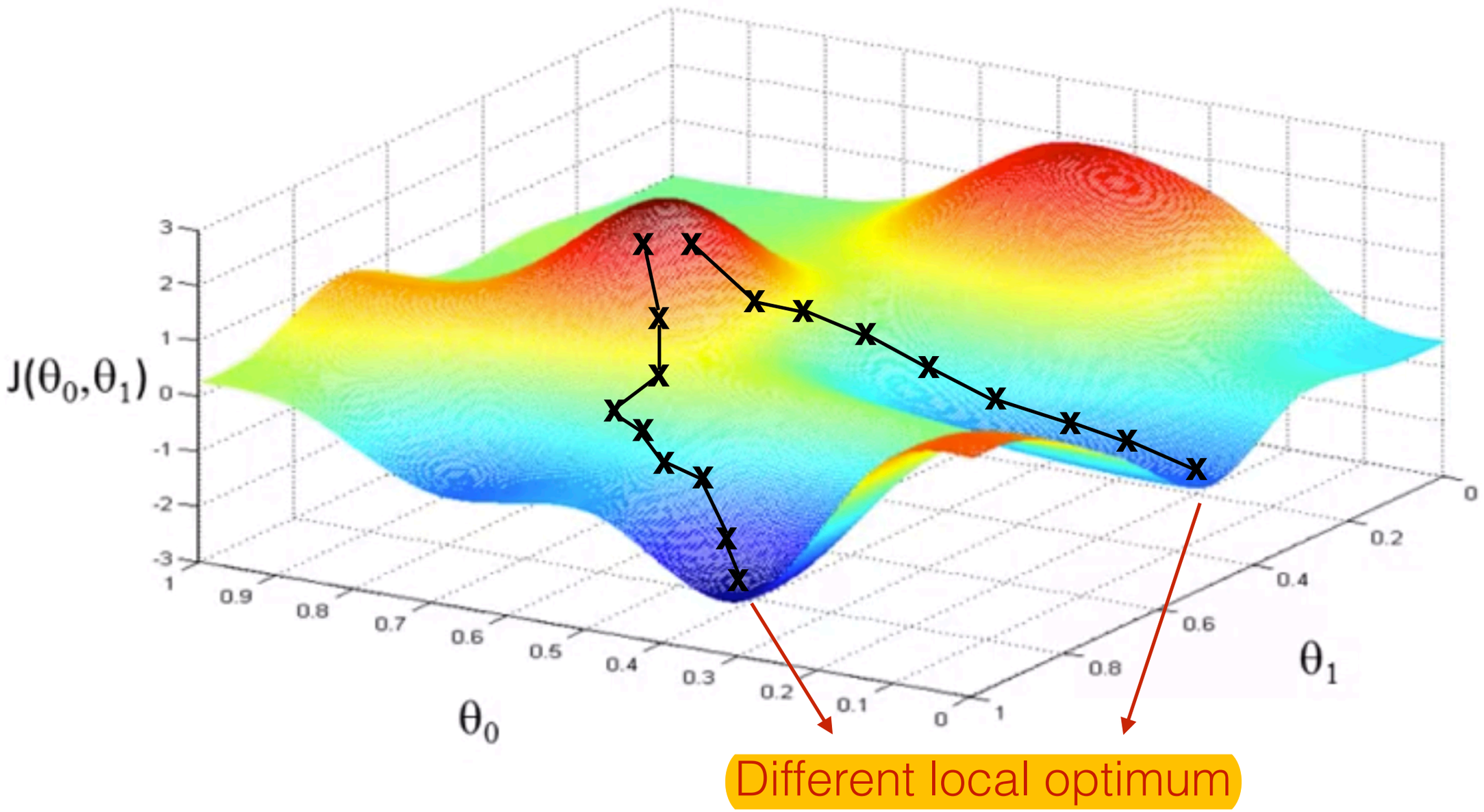
- Let's start with a visualization



Adapted from slide by A. Ng

Gradient Descent - Intuition

- Let's start with a visualization



Adapted from slide by A. Ng

Gradient Descent

- Given $\mathbf{J}(\theta_0, \ldots, \theta_n)$, we take a small step in the direction of the negative gradient, so that:

$$\boldsymbol{\theta}_{j+1} = \boldsymbol{\theta}_j - \eta \nabla \mathbf{J}(\boldsymbol{\theta}_j) \qquad \text{where} \qquad \boldsymbol{\theta}=\{\theta_0, \ldots, \theta_n\}$$

  - The parameter $\eta > 0$ is known as the *learning rate*
  - After each update, the gradient is re-evaluated for the new weight vector $\boldsymbol{\theta}_{j+1}$, and the process repeated
  - Note: the cost function is computed on the entire training set in order to evaluate $\nabla \mathbf{J}$ (i.e. *batch* method)

Gradient Descent

- A simple implementation (using our example):

repeat until convergence{

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

}

Note: simultaneously update θ_0 and θ_1!

Correct:

$$\text{tmp0} := \theta_0 - \eta \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1);$$

$$\text{tmp1} := \theta_1 - \eta \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1);$$

$$\theta_0 := \text{tmp0}; \quad \theta_1 := \text{tmp1};$$

Incorrect:

$$\theta_0 := \theta_0 - \eta \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1);$$

$$\theta_1 := \theta_1 - \eta \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1);$$

Gradient Descent

- A simple implementation (using our example):

repeat until convergence{

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

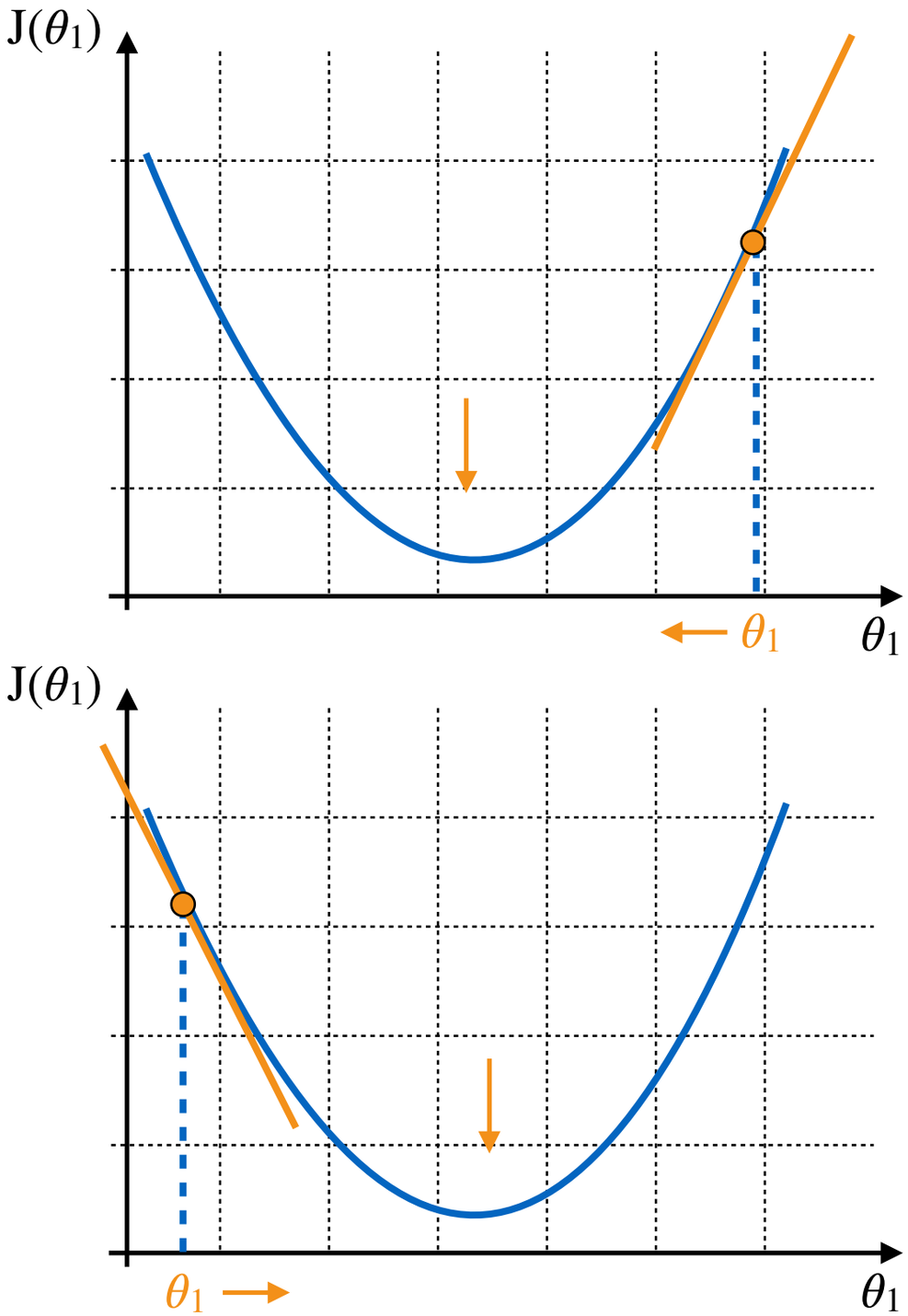
}

learning rate *derivative term*

Simplified version *(like we did in a previous lecture):*

$$\underset{\theta_1}{\text{minimize}}\ J(\theta_1) \quad \text{where} \quad \theta_1 \in \boldsymbol{R}$$

Gradient Descent



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

≥ 0

then we are decreasing θ_1

(in the "right direction")

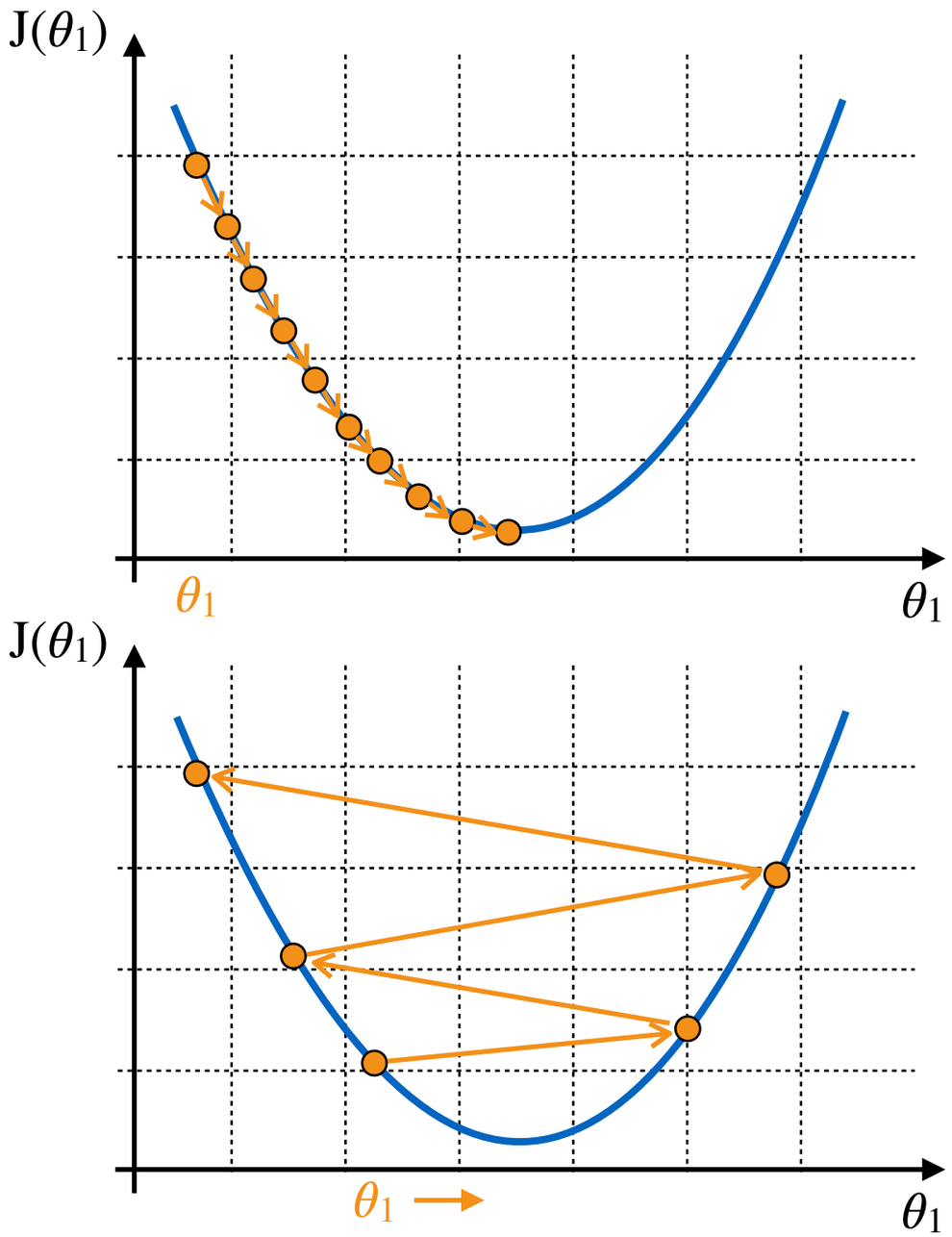
La parte della Derivata indica la DIREZIONE

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

≤ 0

Gradient Descent

- Let's take a look now at the learning rate term η



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

- If η is too small, gradient descent can be slow

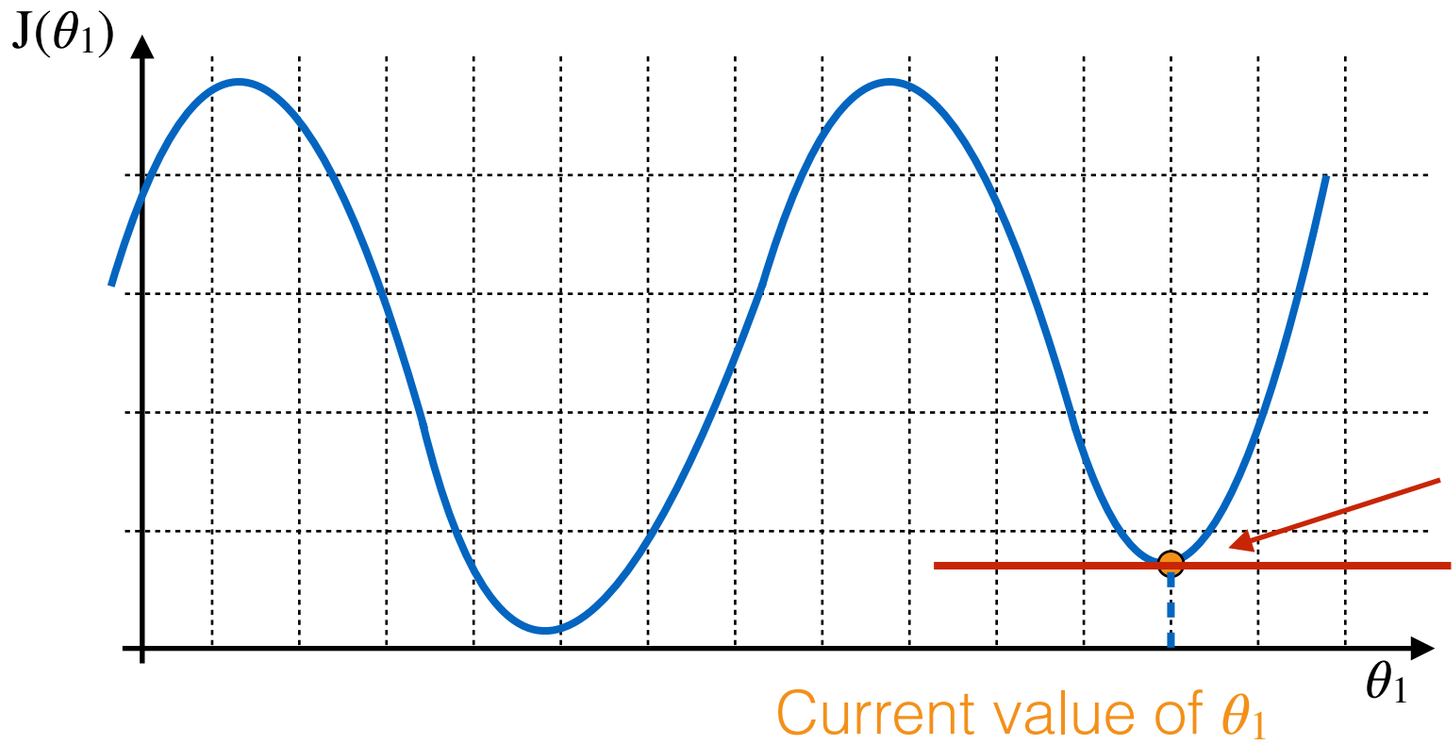
Il parametro controlla l'ampiezza del passo iterativo

- If η is too large, it can overshoot the minimum

  (it may fail to converge or even diverge)

Gradient Descent

- Q: What if your parameter θ_1 is already at a local minimum?

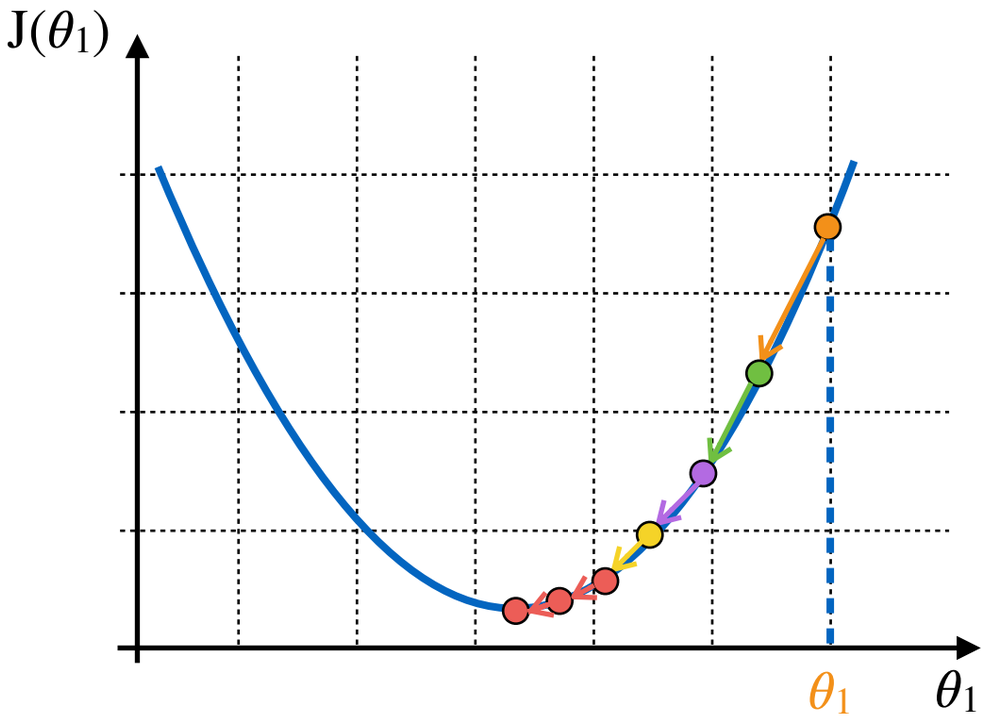


$$\theta_1 := \theta_1 - \eta \boxed{\frac{\mathrm{d}}{\mathrm{d}\theta_1} \mathrm{J}(\theta_1)} = 0$$

- A: one step of gradient descent does not change θ_1

Gradient Descent

- Gradient descent can converge to a local minimum even with a fixed learning rate η



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

- As we approach a local minimum, gradient descent will "naturally" take smaller steps

Linear Regression & Gradient Descent

Linear Regression Model

- Hypothesis:

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

- Cost Function:

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

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

Gradient Descent

```
repeat until convergence{
```

$$\theta_\mathrm{j} := \theta_\mathrm{j} - \eta \frac{\partial}{\partial \theta_\mathrm{j}} \mathrm{J}(\theta_0, \theta_1)$$

(for j = 0 and j = 1)

```
}
```

Contact

- **Office:** Torre Archimede, room 6CD3
- **Office hours** (ricevimento)**:** Friday 9:00-11:00

✉ lamberto.ballan@unipd.it

🏠 http://www.lambertoballan.net

🏠 http://vimp.math.unipd.it

@ twitter.com/lambertoballan