Linear Regression and Gradient Descent

Souptik Barua

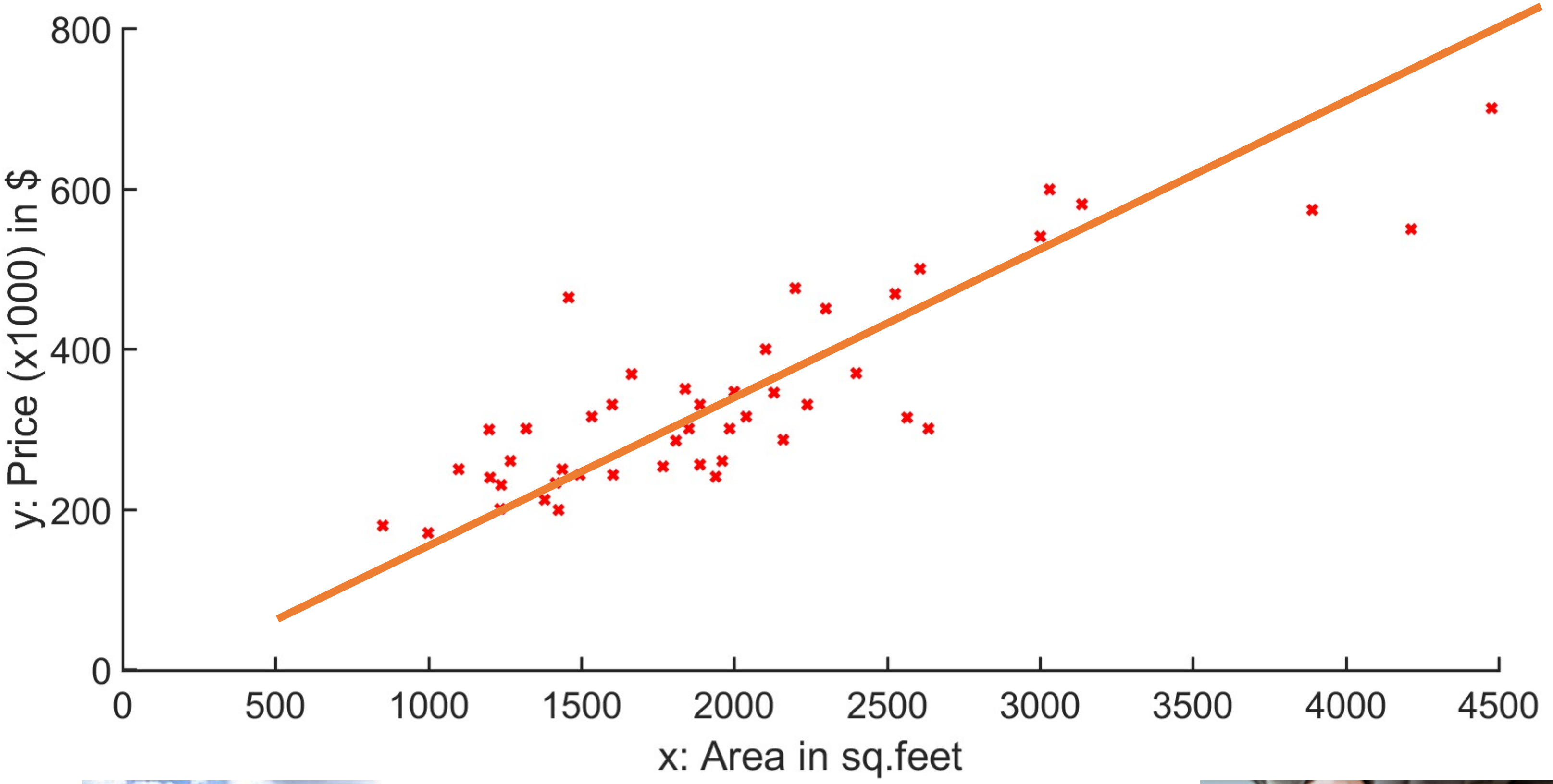
PATHS-UP Course module

Part I: Linear regression

Overview of linear regression

- Linear regression is a statistical analysis technique to model the relationship between a **'response'** variable and several '**predictor**' variables.
- Linear regression is widely used in computational biology research to quantify the relationship between a clinical outcome (response) and several potential explanatory variables (predictor), in two major ways:
  - **Prediction:** Goal is to predict clinical outcome of interest, e.g. the HbA1c level of a diabetes patient given their age, weight, waist circumference, daily carb intake, and time spent in post-meal hyperglycemia.
  - **Identifying key associations**: Goal is to identify which explanatory variables have a statistically significant relationship with the clinical outcome of interest, which improves understanding of the physiology of a disorder or condition. E.g., if the HbA1c is significantly associated with the time spent in post-meal hyperglycemia, it shows the importance of diet in diabetes management.

A real-world linear regression problem- I

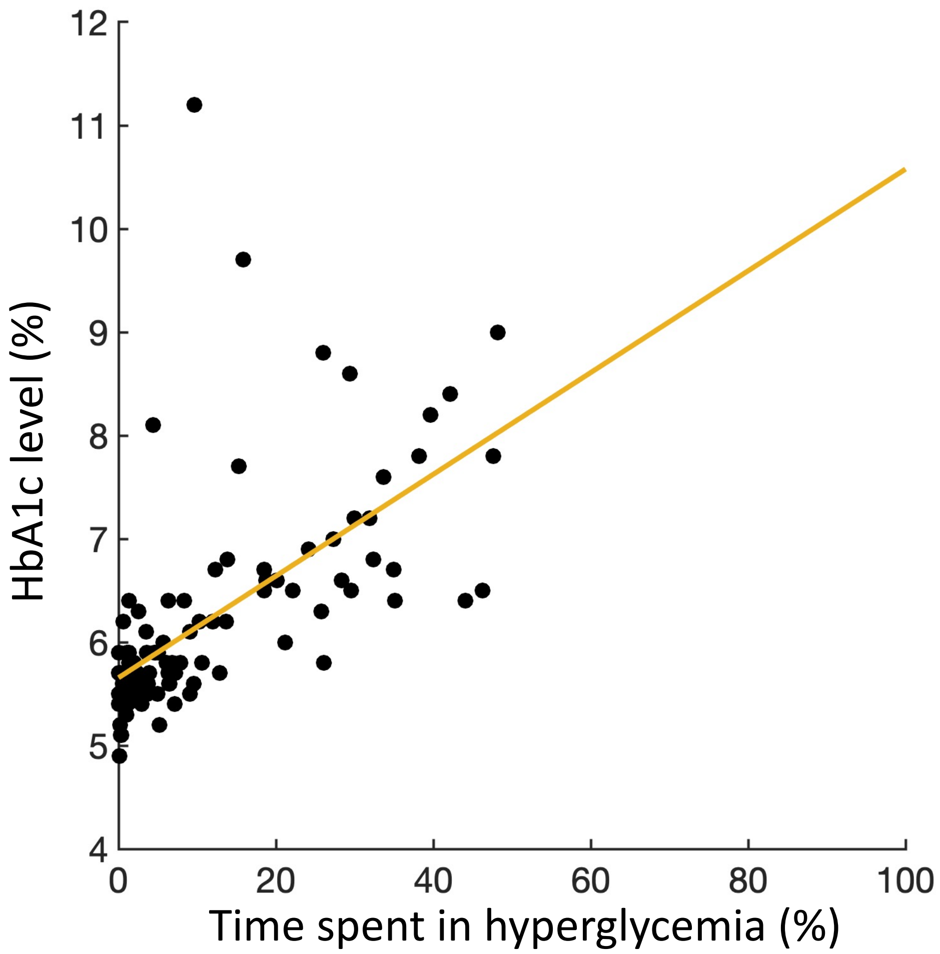


Predicting housing prices given size of house

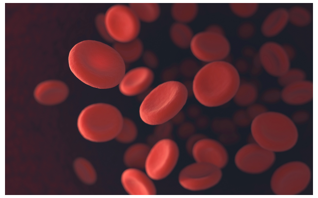
x = 3600 sq.feet

y= ?? dollars

A real-world linear regression problem



Predicting the HbA1c level
given the time spent in hyperglycemia*



x = 25 %

y= ?? %

*hyperglycemia: Exceedingly high blood sugar level,
> 180 mg/dL

Linear $\boxed{\text{Regression}}$

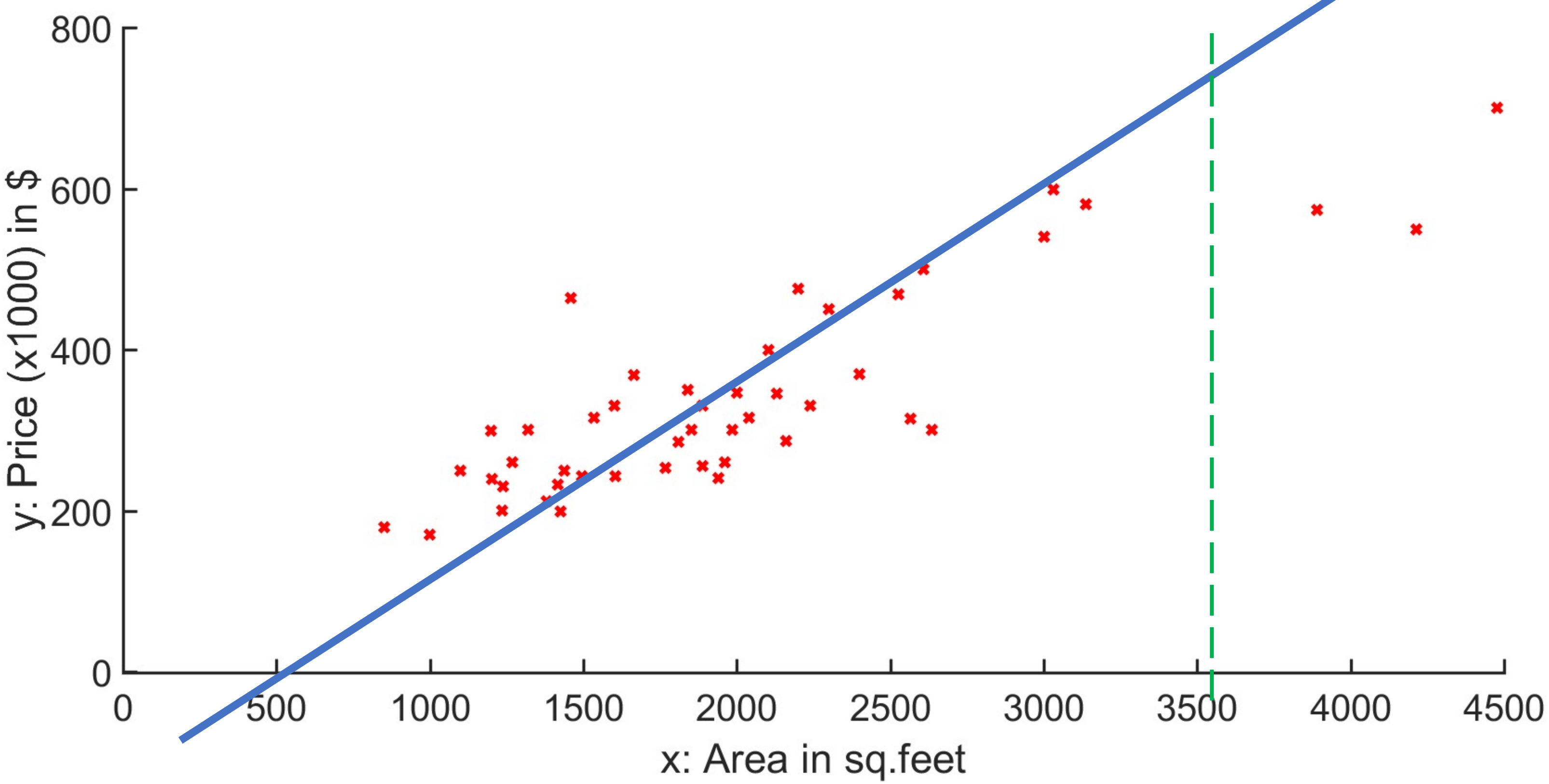
Regression: When the output you are trying to estimate or predict is a continuous-valued number

For e.g: Price of a house, or HbA1c level of an individual with diabetes

Classification: When the output you are trying to estimate or predict is a categorical quantity

For e.g: Cats vs Dogs image classification, or level of risk (High, moderate, or low) for an individual with diabetes

Linear Regression



3600 sq.feet

A real-world linear regression problem

x: Area in square feet	y: Price (x1000) in $
2104	399
1600	329
2400	369
1416	232
3000	539
…	…

Housing prices dataset (Portland, OR)

Notation:
n: Number of examples
x: Input variable/ features/ predictors
y: Output variable/ target/ response

Dataset from https://www.kaggle.com/kennethjohn/housingprice

A real-world linear regression problem

x: Area in square feet	y: Price (x1000) in $
2104	399
1600	329
2400	369
1416	232
3000	539
…	…

Housing prices dataset (Portland, OR)

Notation:
n: Number of examples
x: Input variable/ features/ predictors
y: Output variable/ target/ response

A specific example shall be denoted as:

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

Where i indicates the row number

For example $x^{(1)} = 2104; y^{(1)} = 399$

Dataset from https://www.kaggle.com/kennethjohn/housingprice

Quiz: Notation

Consider the data set shown below. What is $y^{(3)}$?

Area in square feet	Price (x1000) in $
2104	399
1600	329
2400	369
1416	232
…	…

(a) 2400

(b) 1416

(c) 369

(d) 232

Quiz: Notation

Consider the data set shown below. What is $y^{(3)}$?

x: Area in square feet	y: Price (x1000) in $
2104	399
1600	329
2400	369
1416	232
…	…

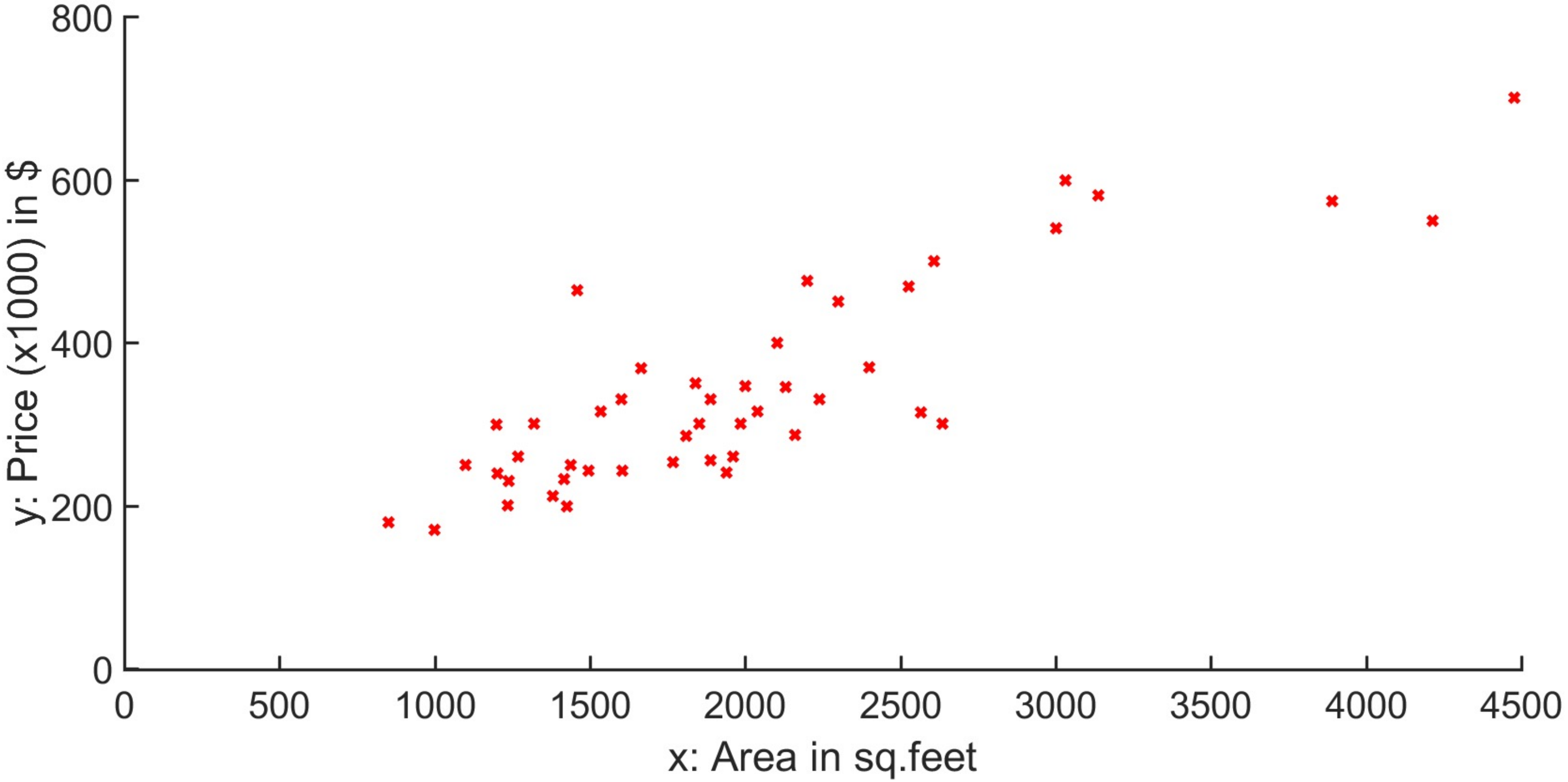
(a) 2400

(b) 1416

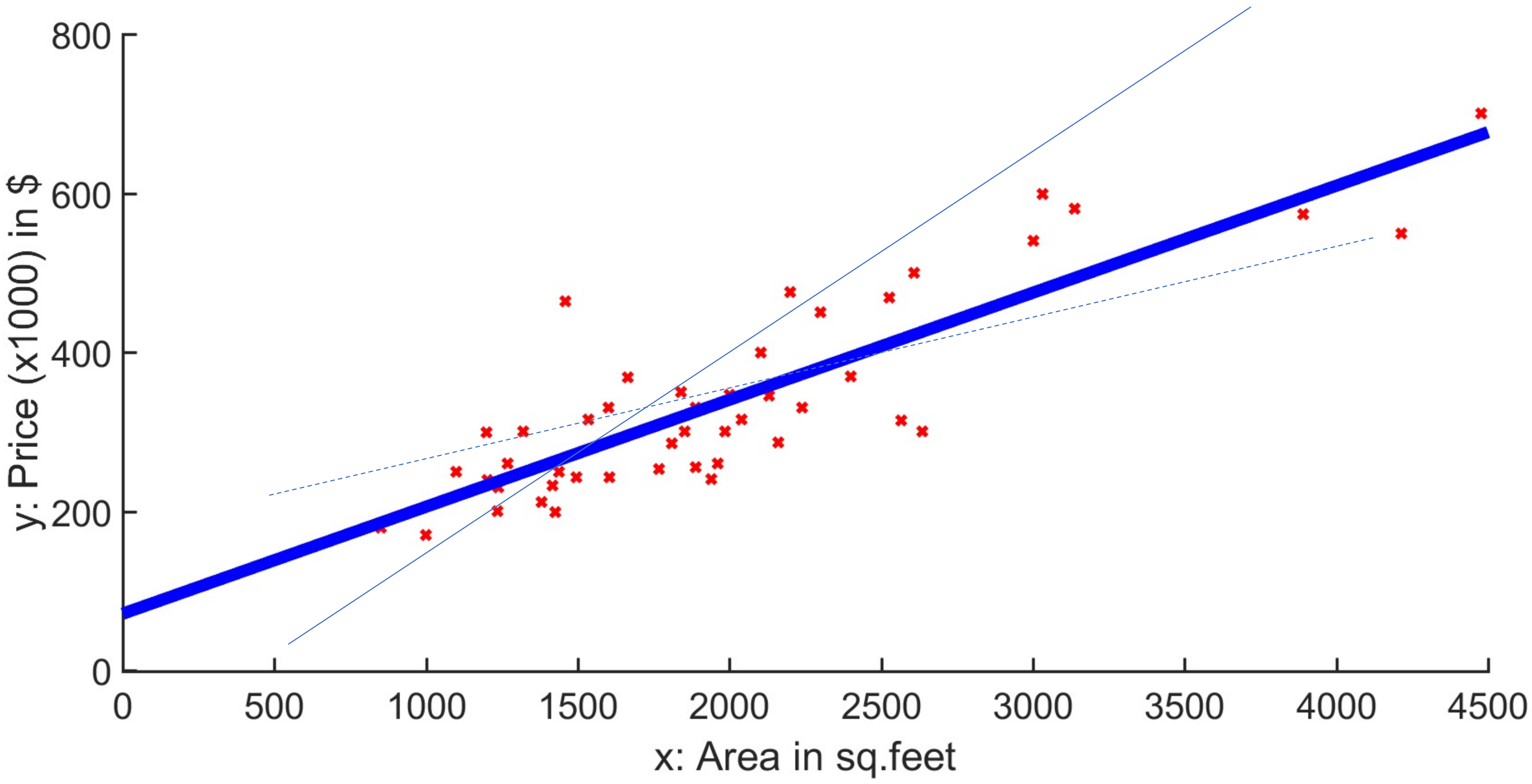
$\boxed{\text{(c) 369}}$

(d) 232

A real-world linear regression problem

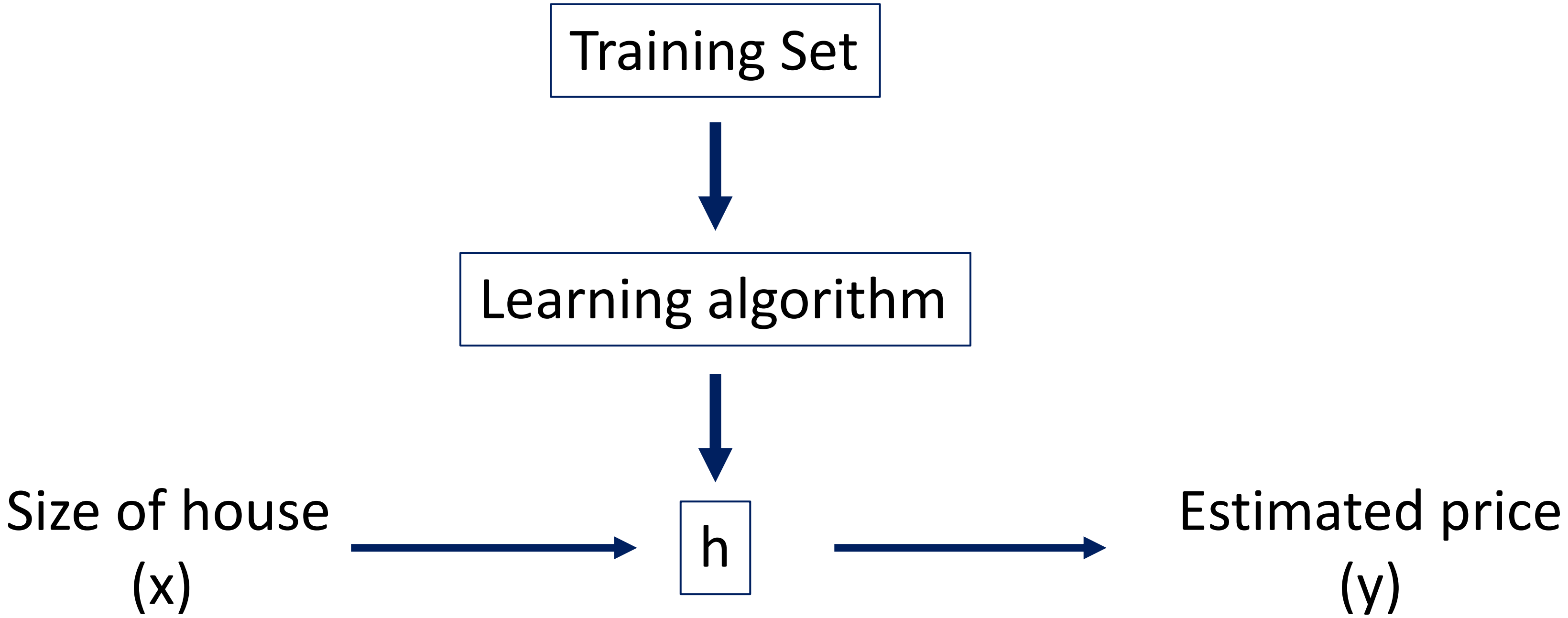


A real-world linear regression problem



Find the line that ‘best’ fits the training data

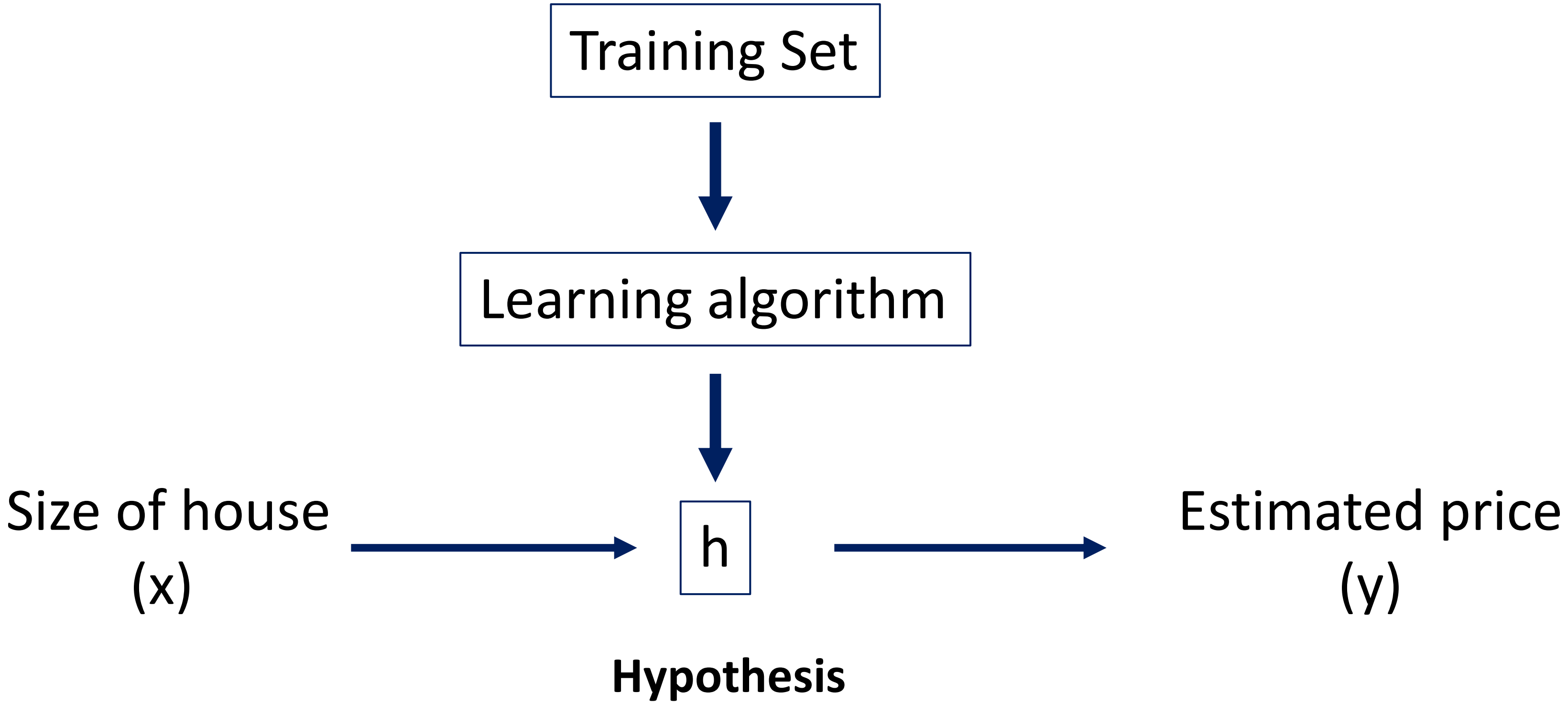
The framework



Hypothesis

The hypothesis maps ‘x’ to ‘y’

The framework



Linear regression hypothesis:

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

The task is to find the appropriate parameters θ_0 and θ_1 such that $y \approx h_\theta(x)$

Solving the linear regression problem

x: Area in square feet	y: Price (x1000) in $
2104	399
1600	329
2400	369
1416	232
3000	539
…	…

Housing prices dataset (Portland, OR)

Notation:
n: Number of examples/observations
x: Input variable/ features/ predictors
y: Output variable/ target/ response

Dataset from https://www.kaggle.com/kennethjohn/housingprice

Solving the linear regression problem

x: Area in square feet	y: Price (x1000) in $
2104	399
1600	329
2400	369
1416	232
3000	539
…	…

Housing prices dataset (Portland, OR)

Hypothesis:

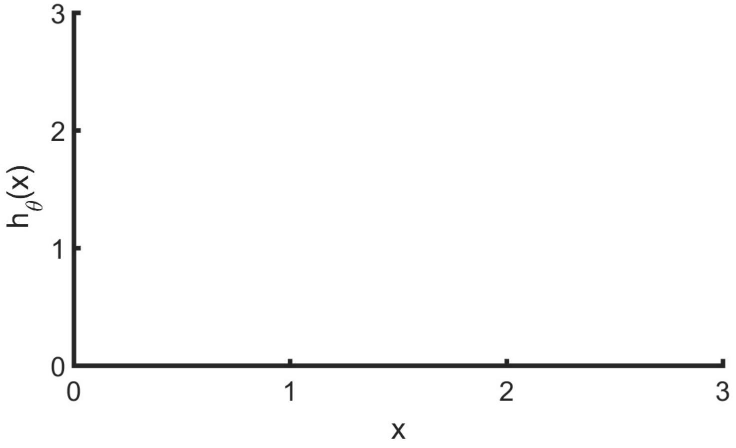
$$\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x}$$

θ refers to the parameters $[\theta_0, \theta_1]$

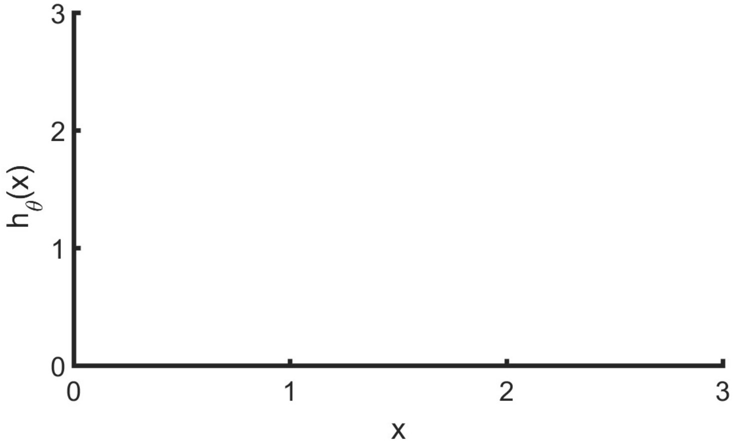
Dataset from https://www.kaggle.com/kennethjohn/housingprice

Different straight lines using different θ

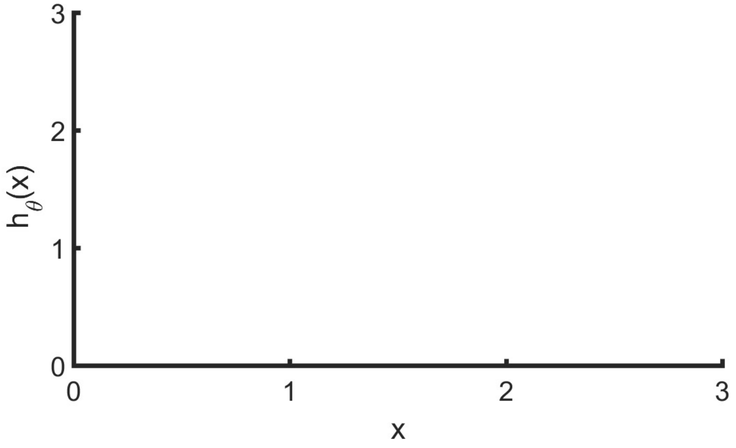
$\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x}$



$\theta_0 = 1.5, \theta_1 = 0$



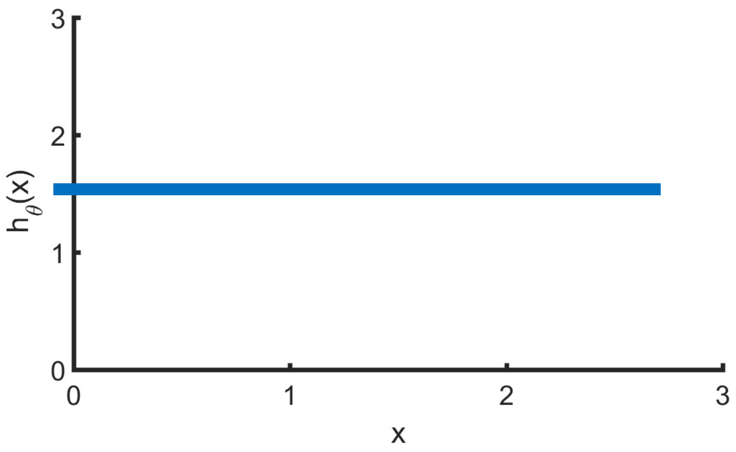
$\theta_0 = 0, \theta_1 = 0.5$



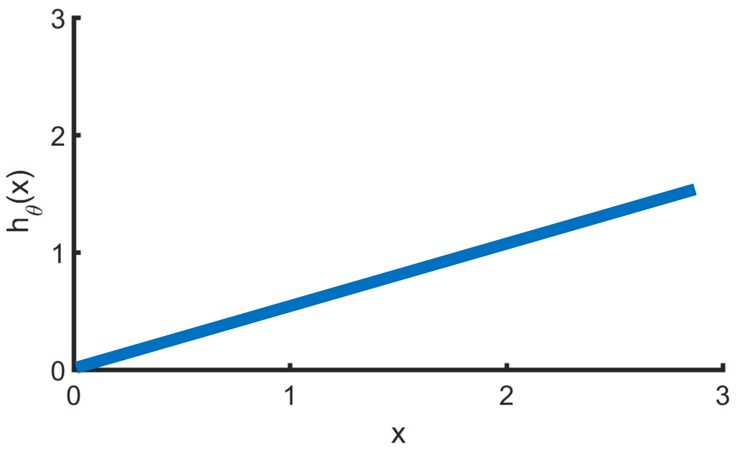
$\theta_0 = 1, \theta_1 = 0.5$

Different straight lines using different θ

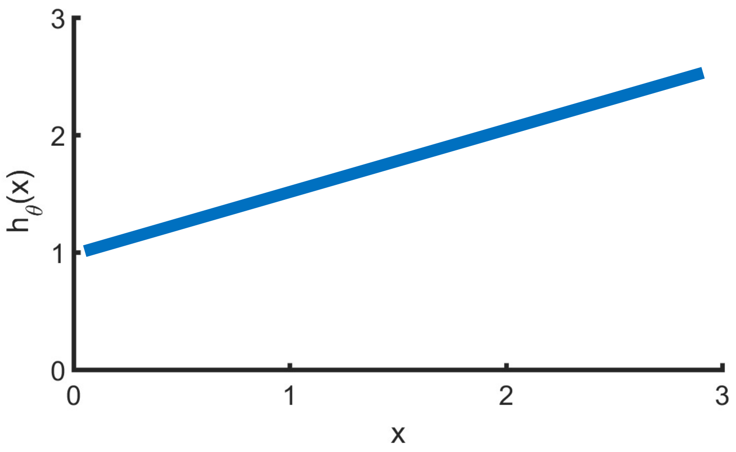
$$h_\theta(x) = \theta_0 + \theta_1 x$$



$\theta_0 = 1.5, \theta_1 = 0$



$\theta_0 = 0, \theta_1 = 0.5$



$\theta_0 = 1, \theta_1 = 0.5$

Quiz: Different straight lines using different θ

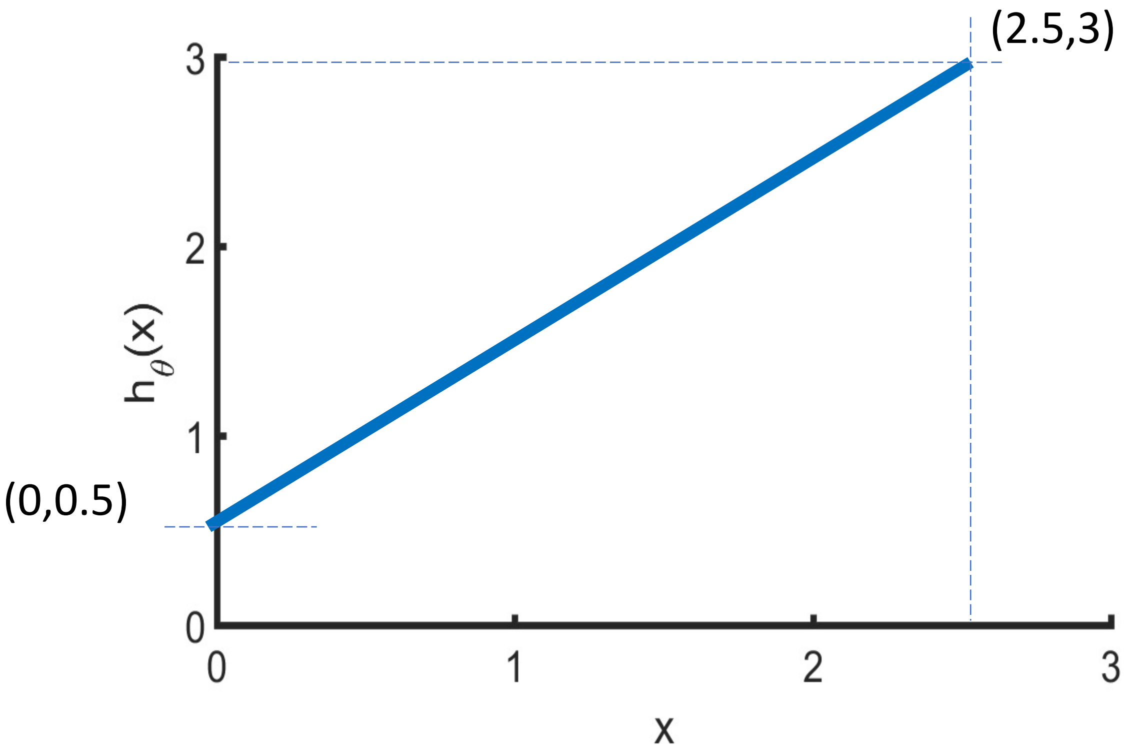
An example plot of $h_\theta(x) = \theta_0 + \theta_1 x$ is shown.
What are the values $\theta_0$and θ_1?

(a) $\theta_0 = 0, \theta_1 = 1$

(b) $\theta_0 = 0.5, \theta_1 = 1$

(c) $\theta_0 = 1, \theta_1 = 0.5$

(d) $\theta_0 = 0.5, \theta_1 = 1.2$



Quiz: Different straight lines using different $\boldsymbol{\theta}$

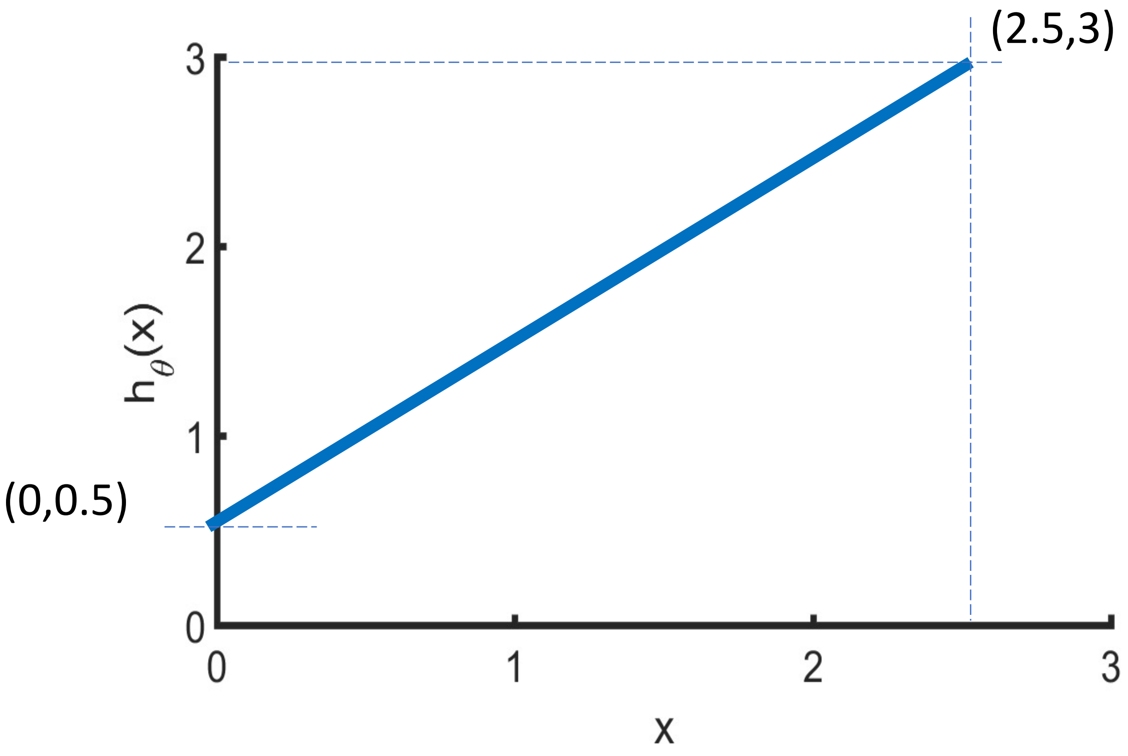
An example plot of $h_\theta(x) = \theta_0 + \theta_1 x$ is shown.
What are the values $\theta_0$and θ_1?

(a) $\theta_0 = 0, \theta_1 = 1$

(b) $\boxed{\theta_0 = 0.5, \theta_1 = 1}$

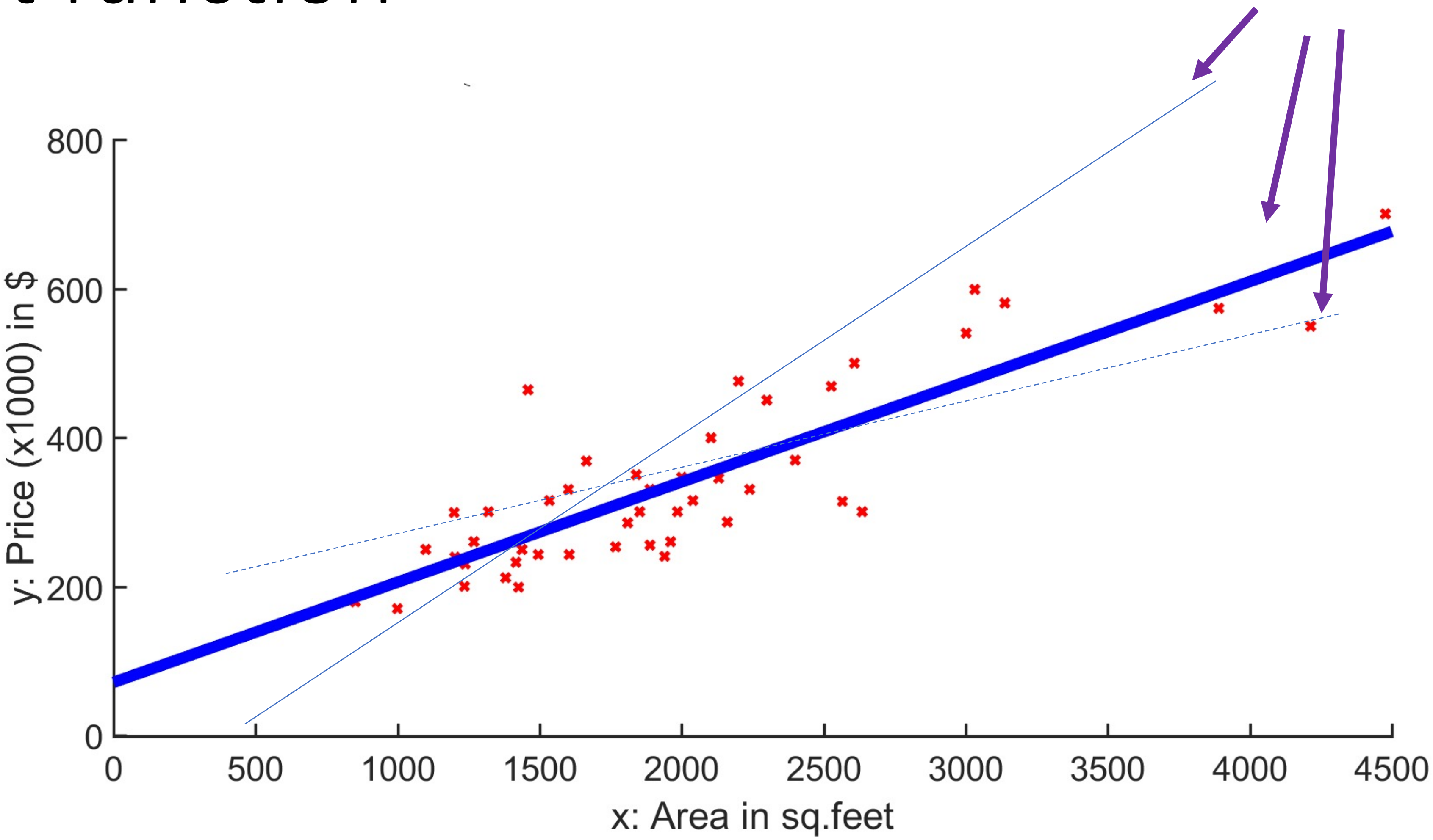
(c) $\theta_0 = 1, \theta_1 = 0.5$

(d) $\theta_0 = 0.5, \theta_1 = 1.2$



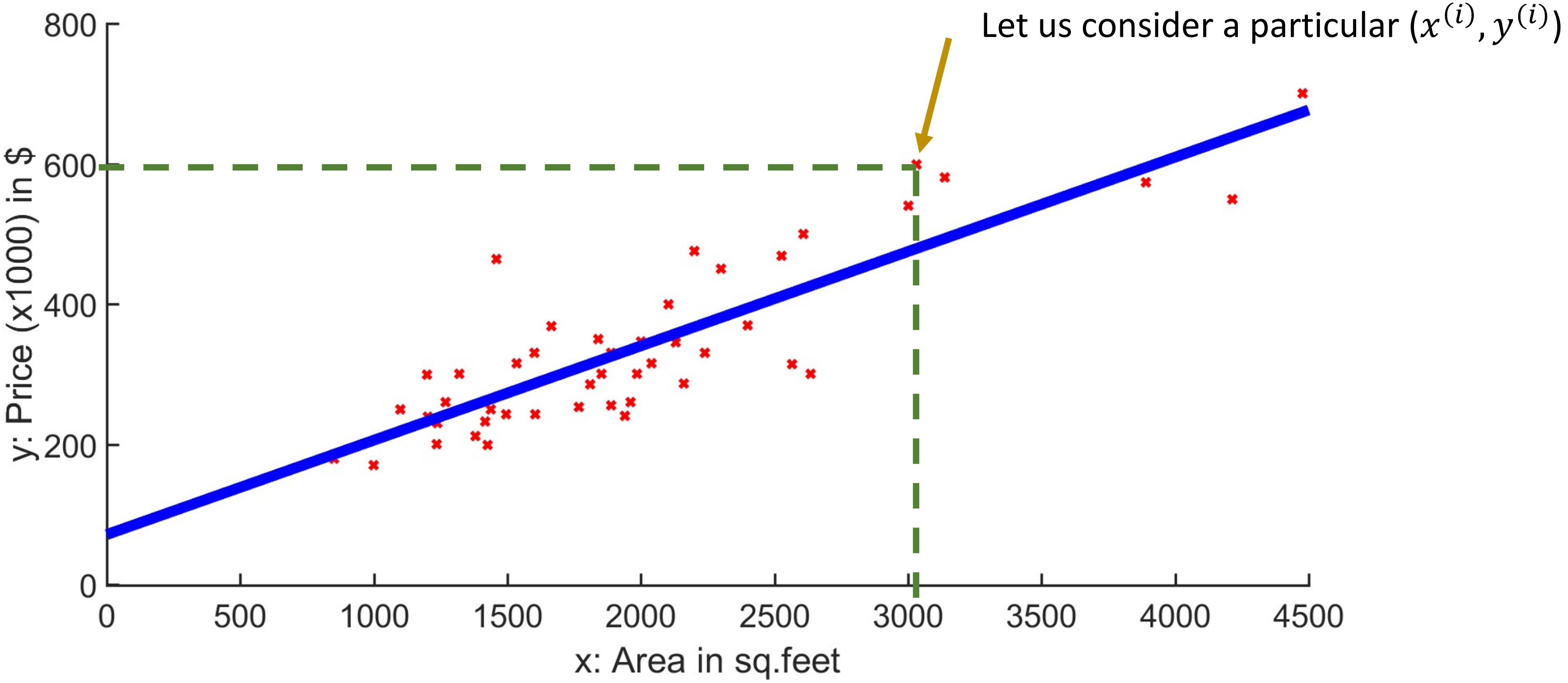
Cost function

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

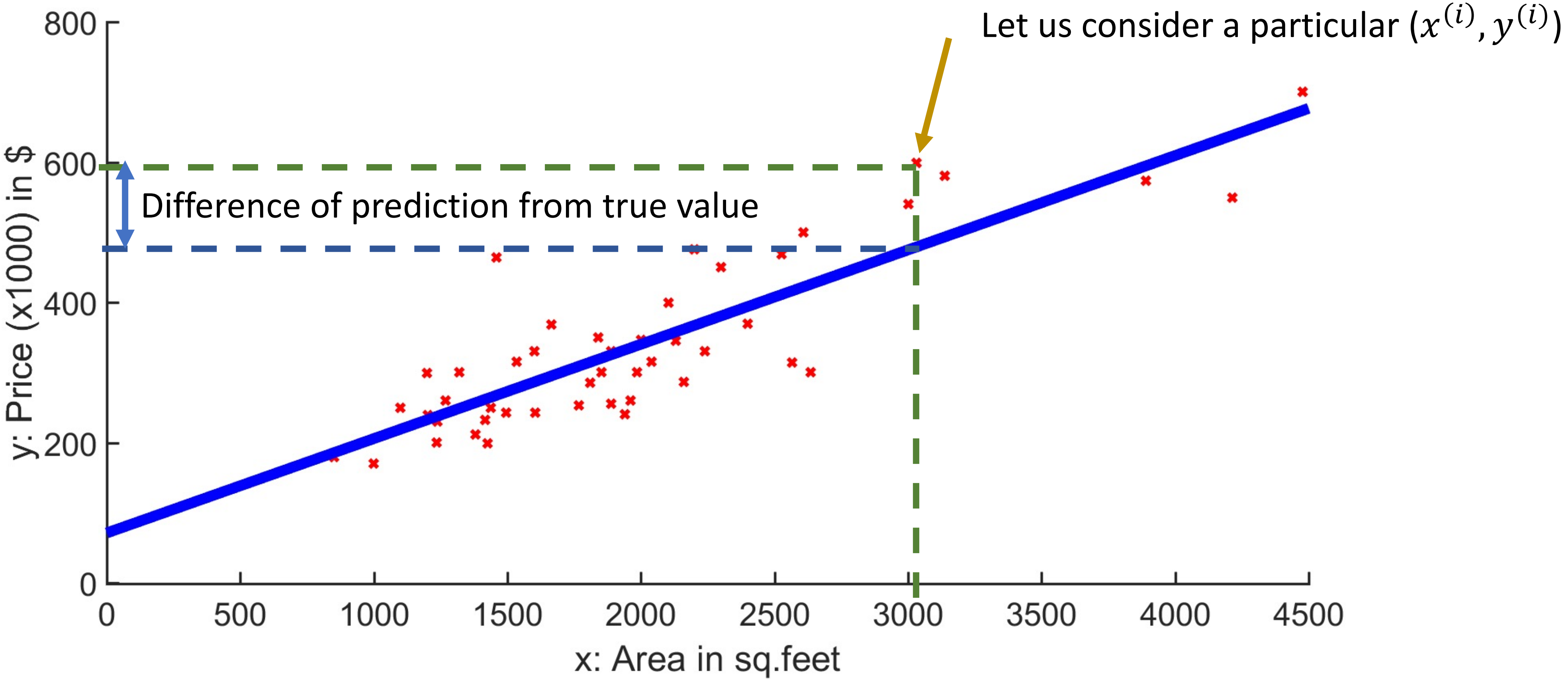


All the lines are hypotheses with different choices of $\theta_0$and θ_1

Cost function



Cost function



Cost function

Cost of prediction for each observation = $(h_\theta(x^{(i)}) - y^{(i)})^2$

Prediction *True value*

Total Cost of predictions for the whole training set = $\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Simply add all the individual costs!

Cost function

Cost of prediction for each observation = $(h_\theta(x^{(i)}) - y^{(i)})^2$

Total cost of predictions for the whole training set = $\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Linear regression cost function:

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

An averaged form of the total cost

Cost function

Cost of prediction for each observation = $(h_\theta(x^{(i)}) - y^{(i)})^2$

Total cost of predictions for the whole training set = $\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Linear regression cost function:

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

Find the values of $\theta_0$and $\theta_1$that minimize this cost function

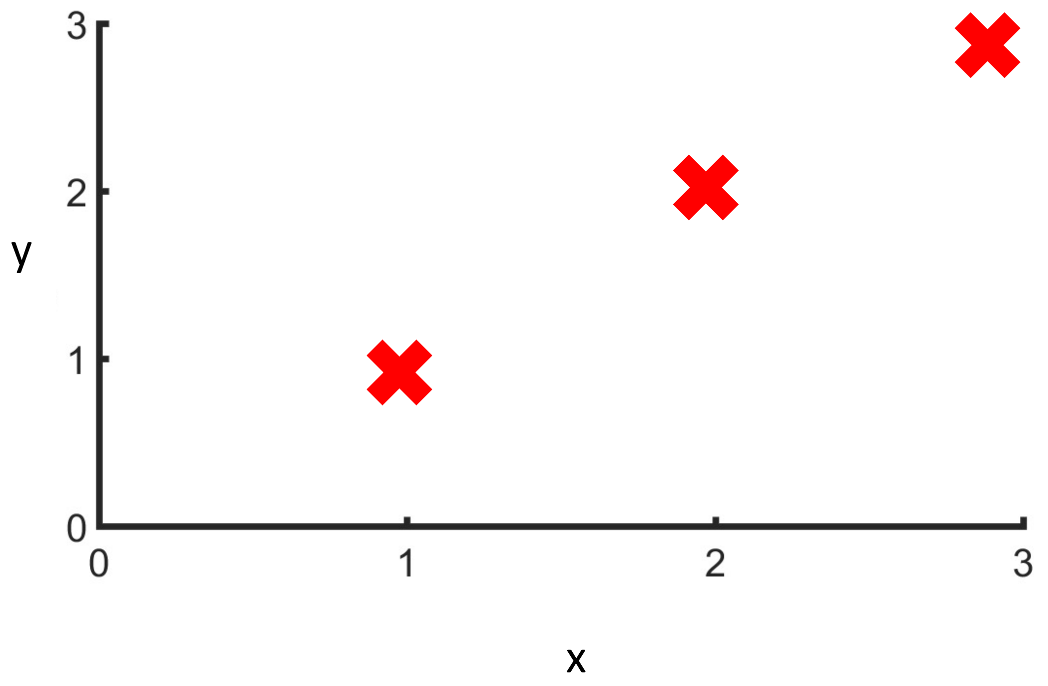
Quiz: Cost function

Suppose we have a training set with three observations as shown. Let $\theta_0 = 0$ for now so:

-- Our hypothesis becomes $h_\theta(x) = \theta_1 x$.

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

What is the value of $J(0)$?



(a) 0

(b) 1/6

(c) 1

(d) 14/6

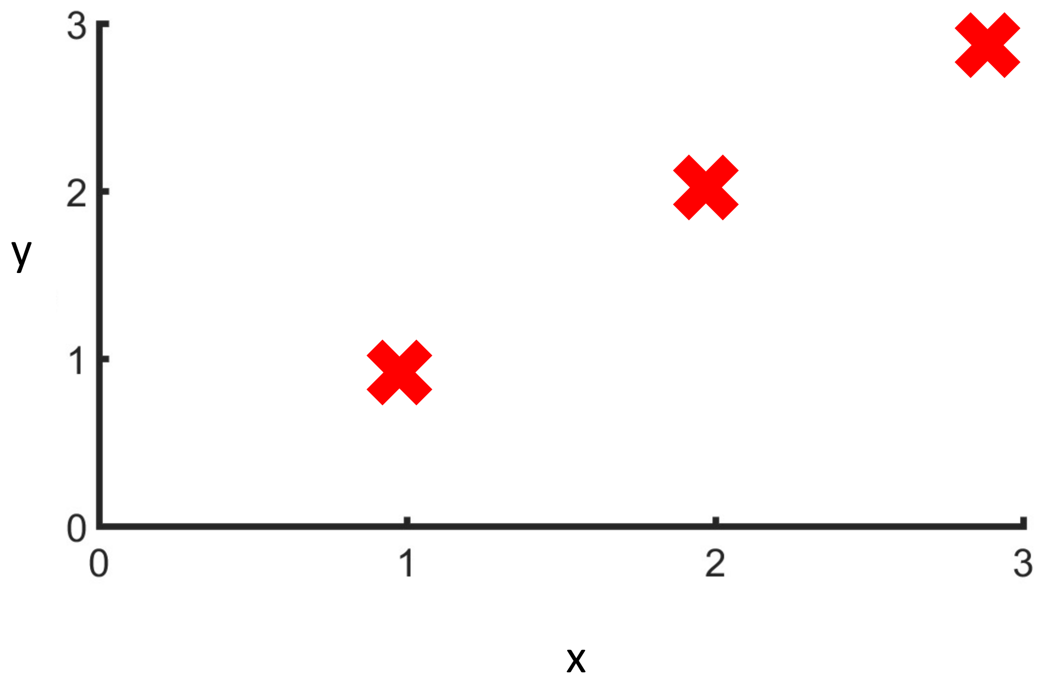
Quiz: Cost function

Suppose we have a training set with three observations as shown. Let $\theta_0 = 0$ for now so:

-- Our hypothesis becomes $h_\theta(x) = \theta_1 x$.

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

What is the value of $J(0)$?



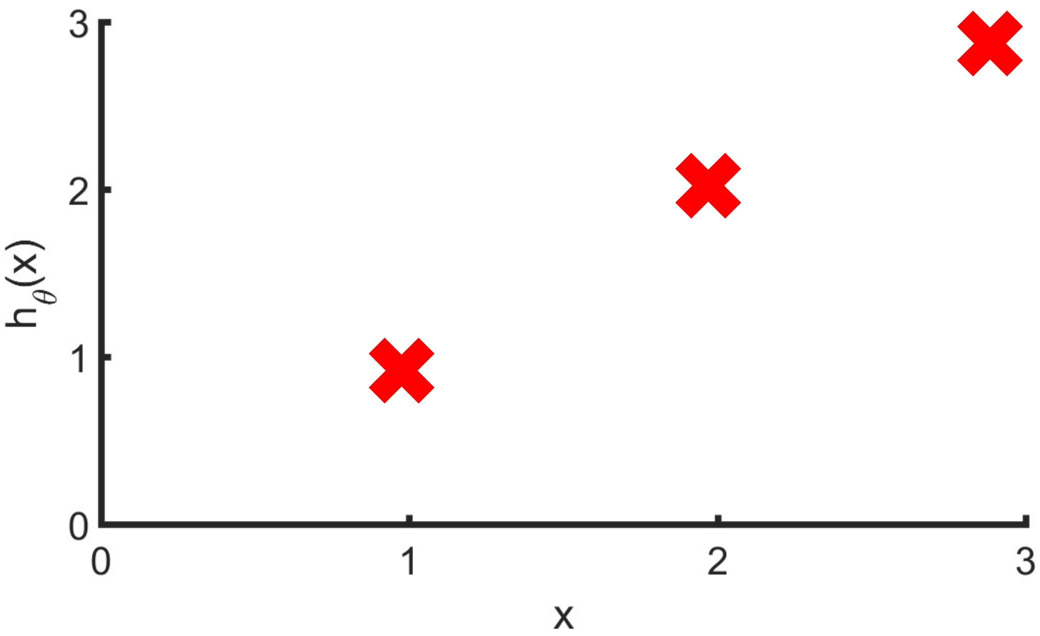
(a) 0

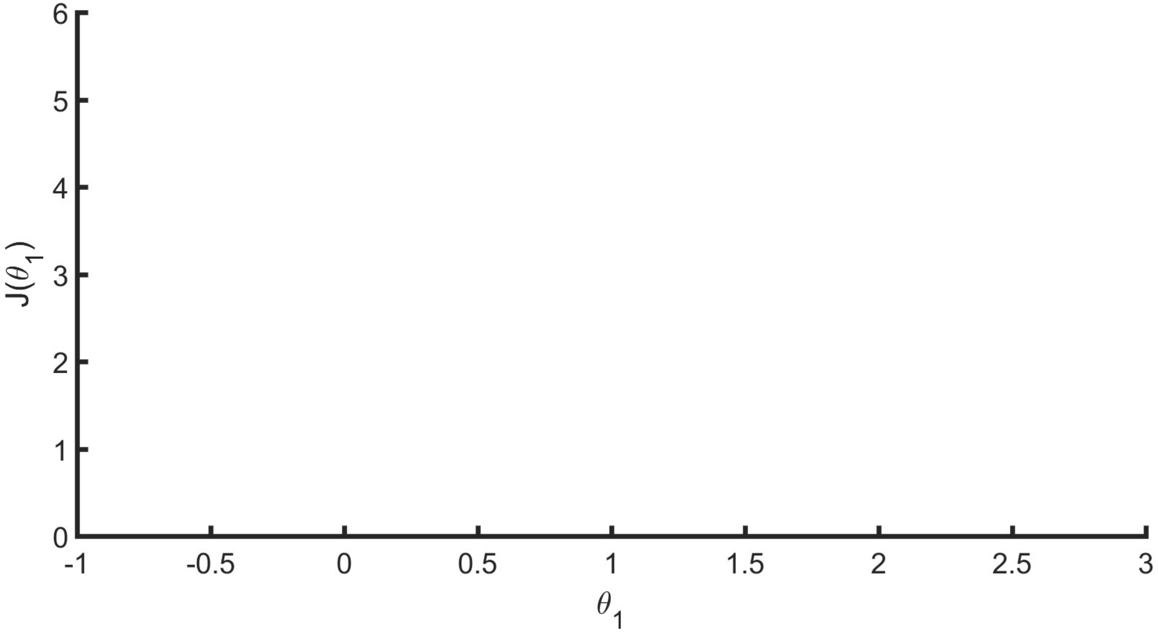
(b) 1/6

(c) 1

(d) 14/6

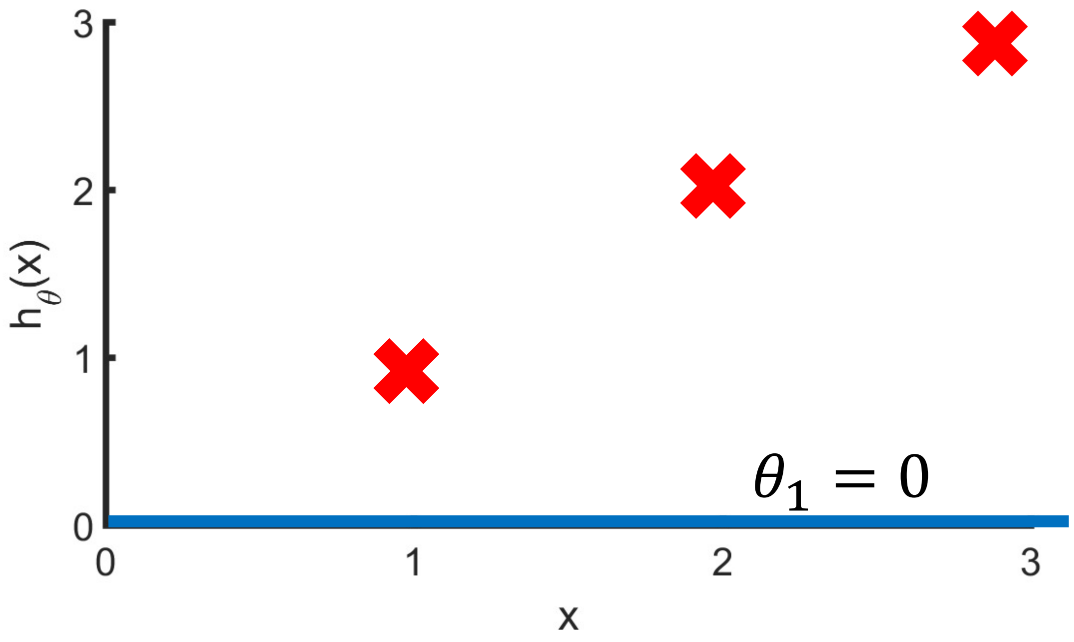
Cost function: Varying θ_1

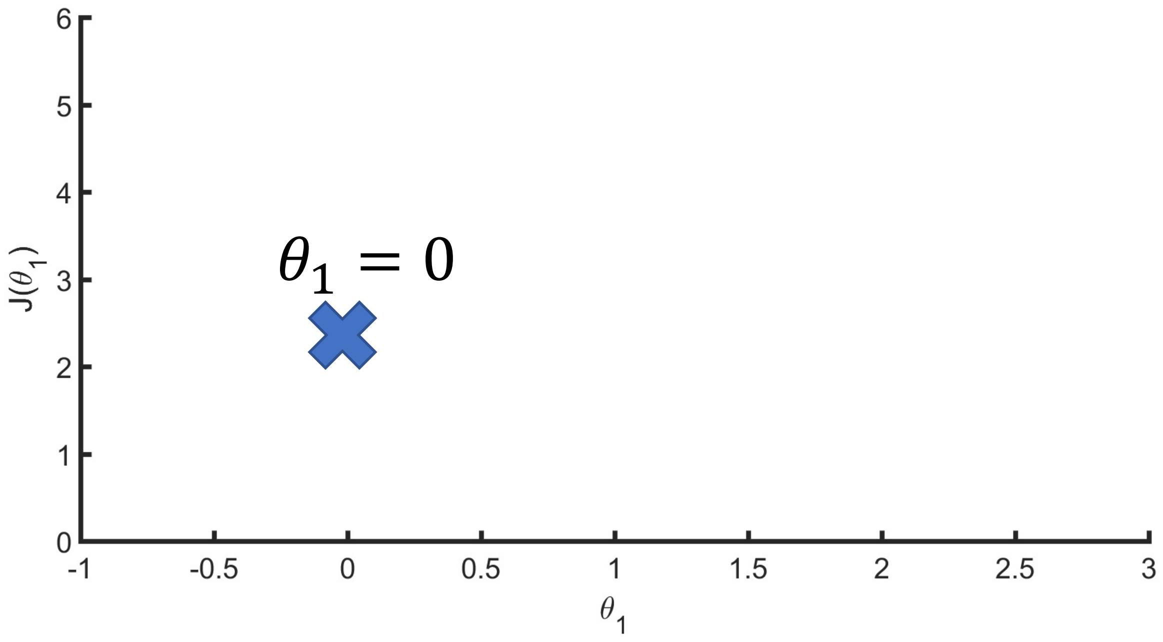




How does the cost function change with different choices of θ_1?

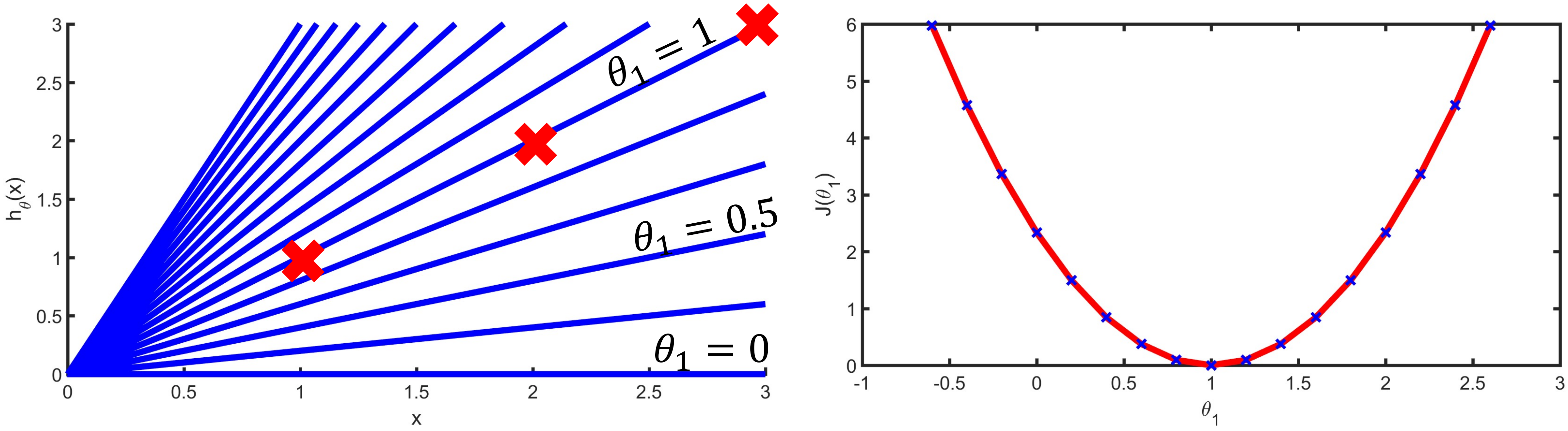
Cost function: Varying θ_1





How does the cost function change with different choices of θ_1?

Cost function: Varying θ_1



Plotting the different hypotheses and cost function values as a function of θ_1

Learning outcomes - I

By the end of part I, you now know how to:

-- Formulate a linear regression **hypothesis**: $h_\theta(x) = \theta_0 + \theta_1 x$

-- Understand intuitively what the **parameters** represent: θ_0 and θ_1

-- Understand what a **cost function** is and why the linear regression cost function is formulated as:

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

-- The main **computational goal** to generate our linear regression model:

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

Next class

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

Parameters: θ_0 and θ_1

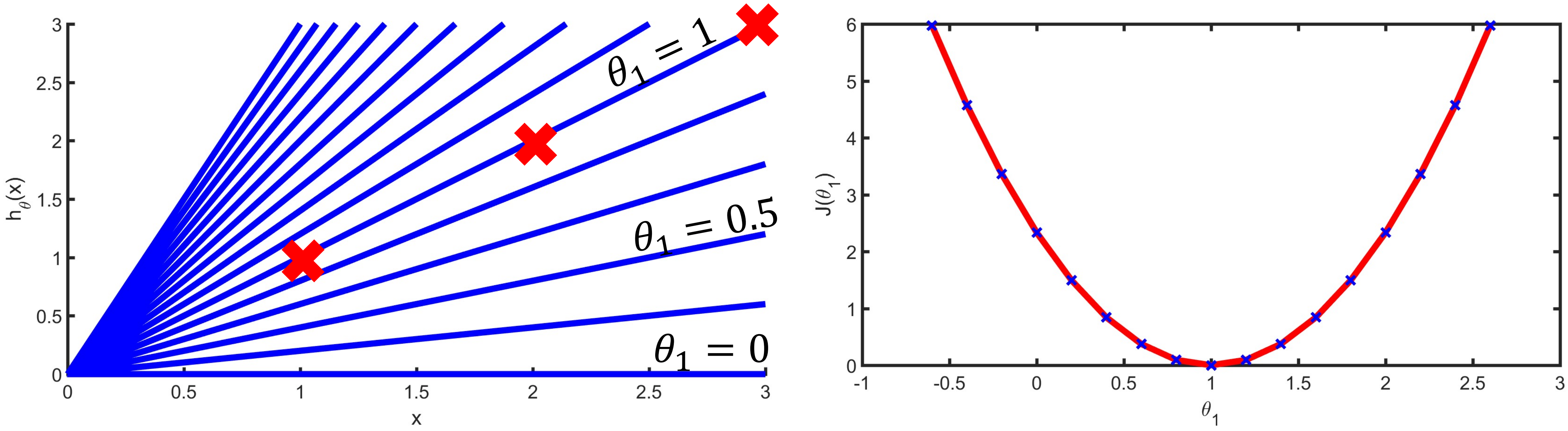
Cost function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$

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

A powerful and standard tool: ***Gradient Descent***

Part-II: Gradient Descent

Cost function: Varying θ_1



Hypothesis

Cost function

Plotting the different hypotheses and cost function values as a function of θ_1

A solution strategy

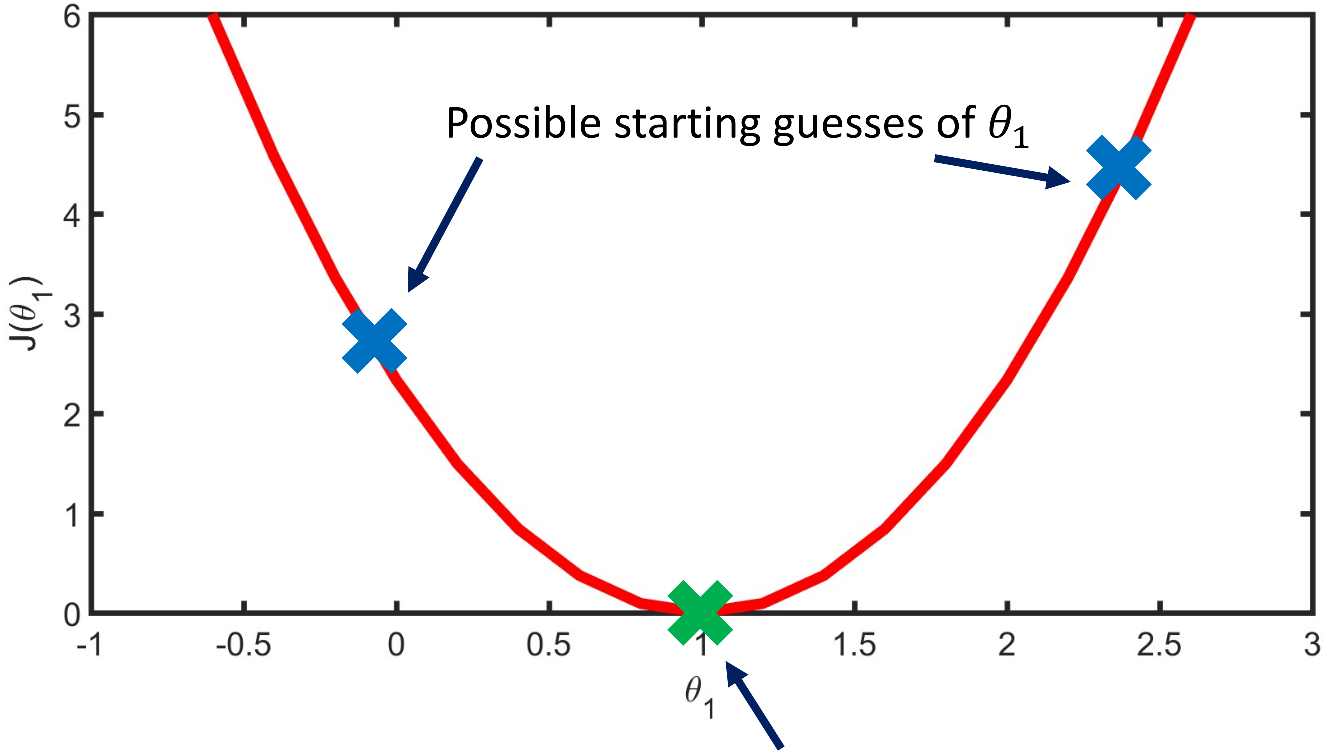
Given a function $J(\theta_1)$:

Step 1: Start with some θ_1

Step 2: Update θ_1 such that it reduces $J(\theta_1)$

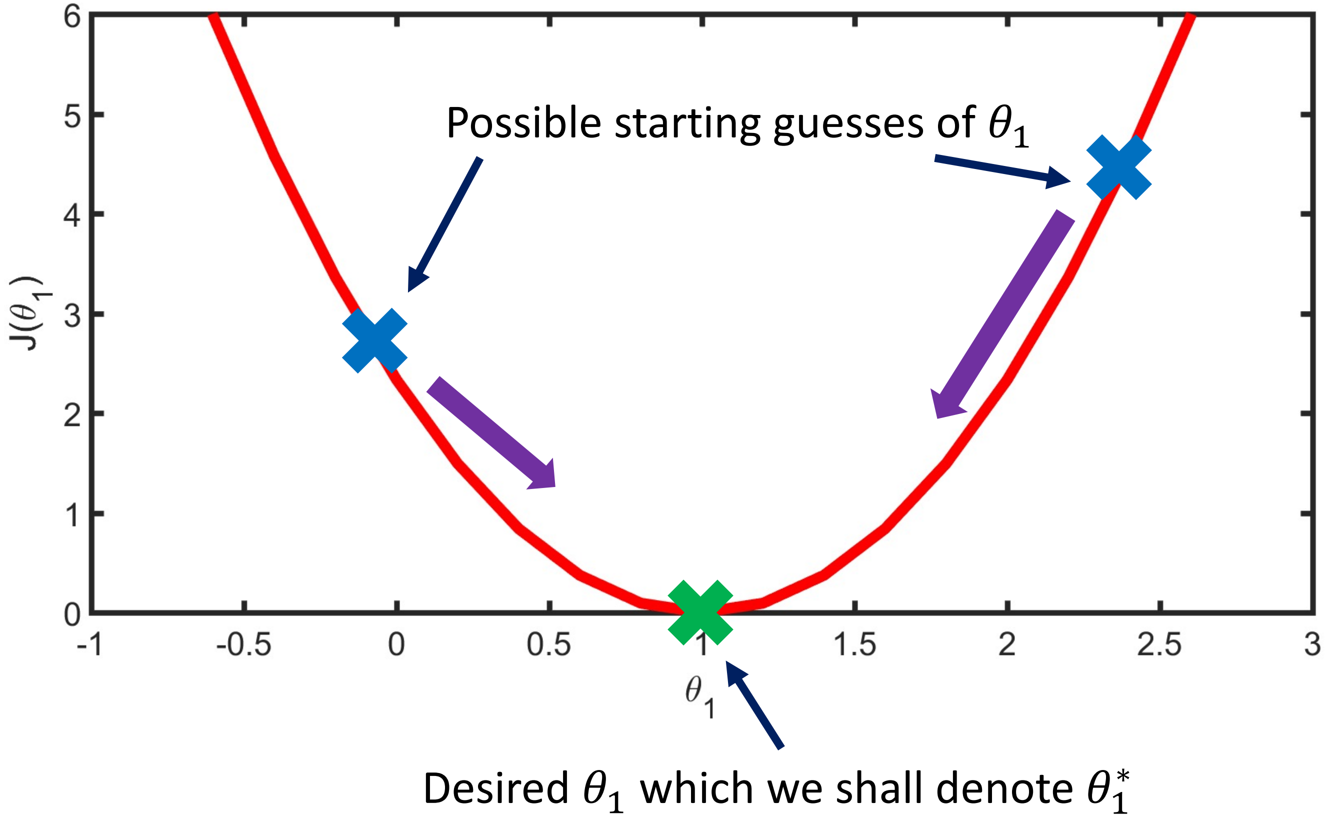
Step 3: Keep repeating step 2 until we hopefully reach the minimum value of $J(\theta_1)$

Gradient $\boxed{\text{descent}}$

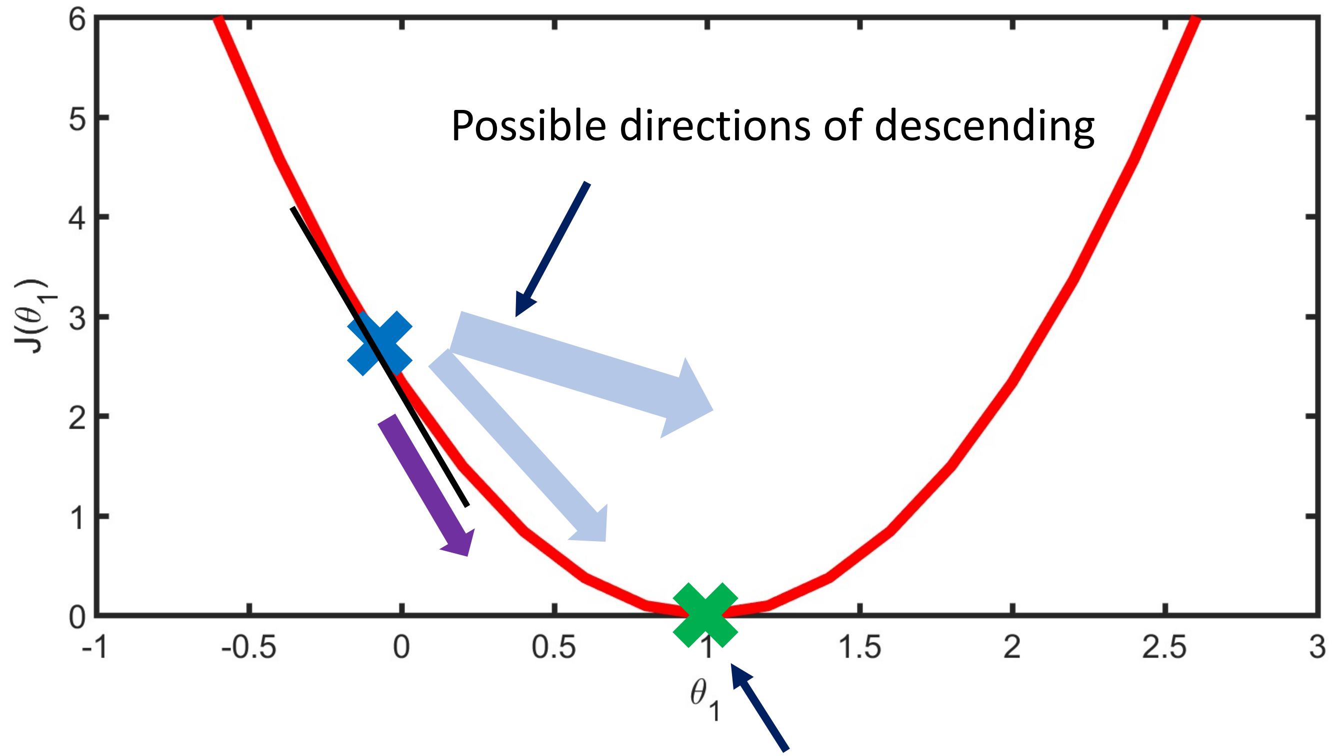


Desired θ_1 which we shall denote θ_1^*

Gradient descent



Gradient descent



Desired θ_1 which we shall denote θ_1^*

A solution strategy

Given a function $J(\theta_1)$:

Step 1: Start with some θ_1

Step 2: Update θ_1 such that it reduces $J(\theta_1)$

Step 3: Keep repeating step 2 until we hopefully reach the minimum value of $J(\theta_1)$

Gradient descent algorithm

$$\boxed{\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial \mathrm{J}(\theta_1)}{\partial \theta_1}}$$

Iteration continues until θ_1 does not change much.

-- [k] : denotes the iteration number. k = 0,1,2,3,…

-- $\theta_1^{[0]}$is our starting value for θ_1

-- α: Learning rate; a positive number

-- $\frac{\partial \mathrm{J}(\theta_0,\theta_1)}{\partial \theta_1}$: Gradient term

Gradient descent algorithm

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial \mathrm{J}(\theta_1)}{\partial \theta_1}$$

Your current guess of θ_1

Gradient descent algorithm

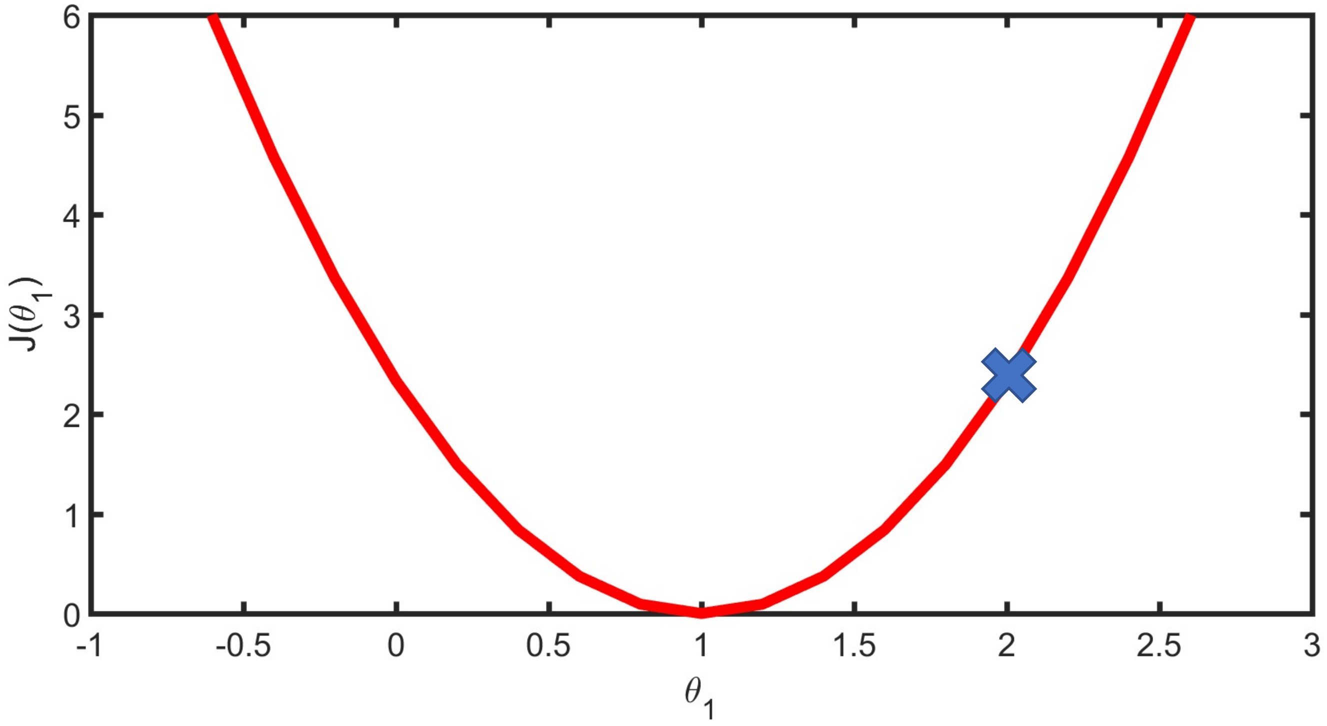
$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

Your current guess of θ_1

$\frac{\partial J(\theta_1)}{\partial \theta_1}$: *which direction* to descend?

α : *how fast* to descend?

Gradient descent intuition

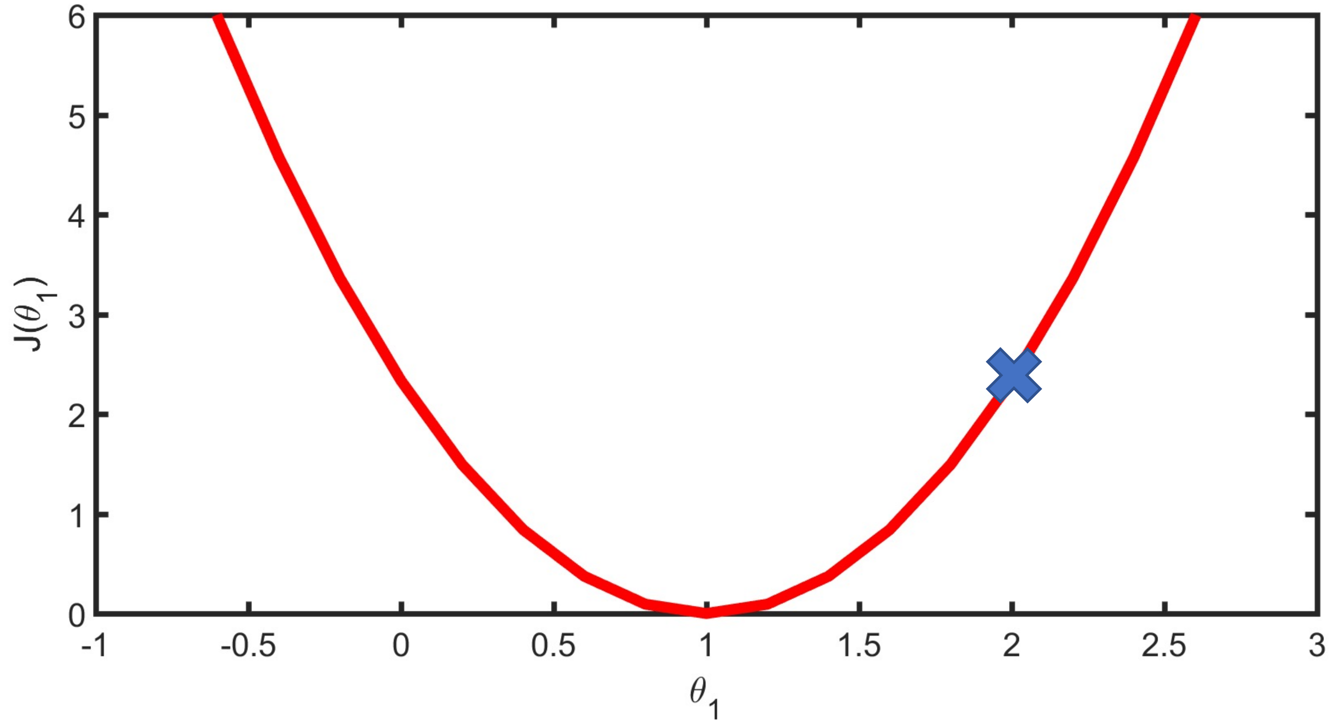


Update rule :

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

Start with some $\theta_1^{[0]}$

Gradient descent intuition

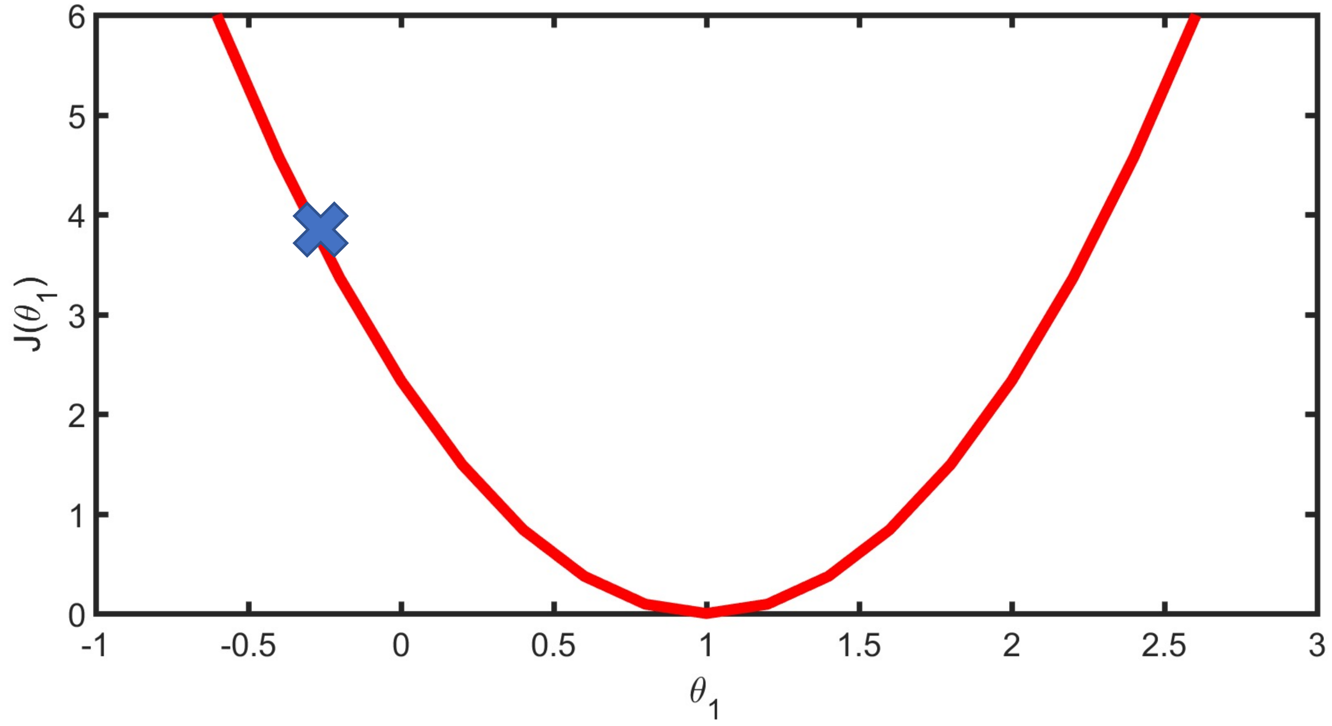


Update rule :

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

Start with some $\theta_1^{[0]}$

Gradient descent intuition

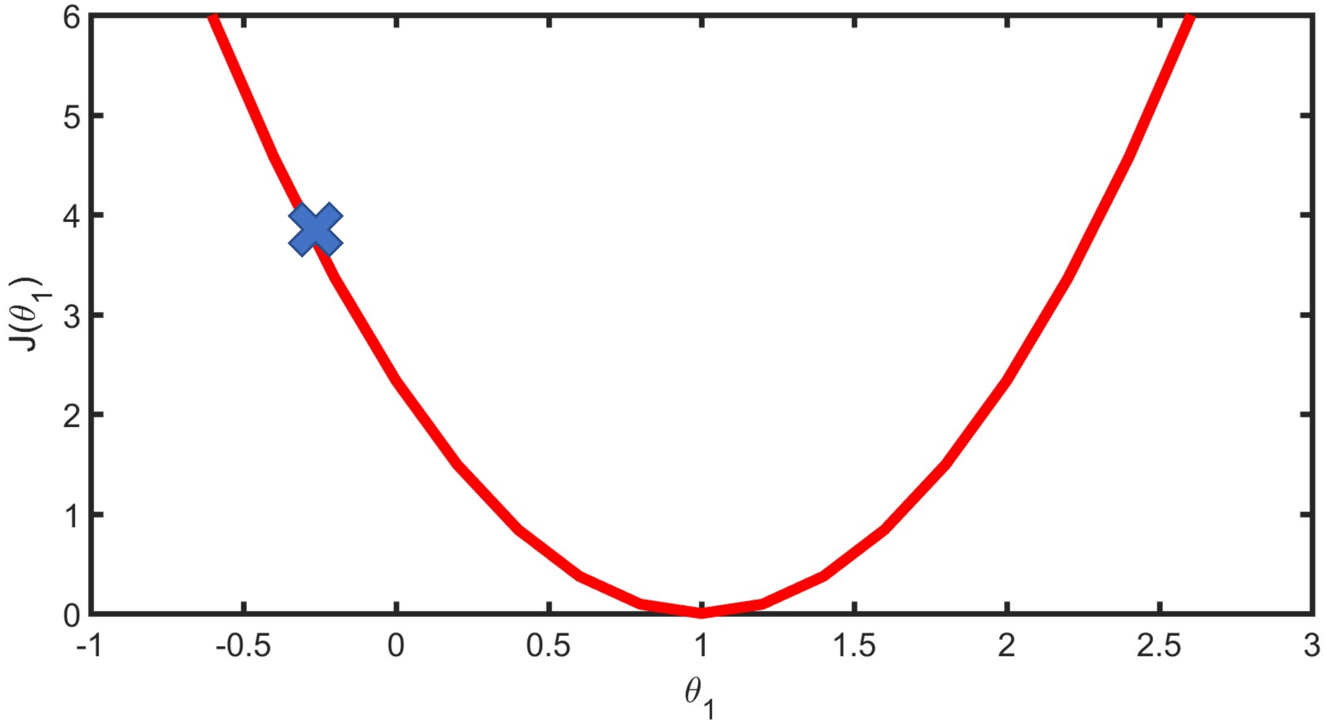


Update rule :

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

What if we started with this $\theta_1^{[0]}$ instead?

Gradient descent intuition



Update rule :

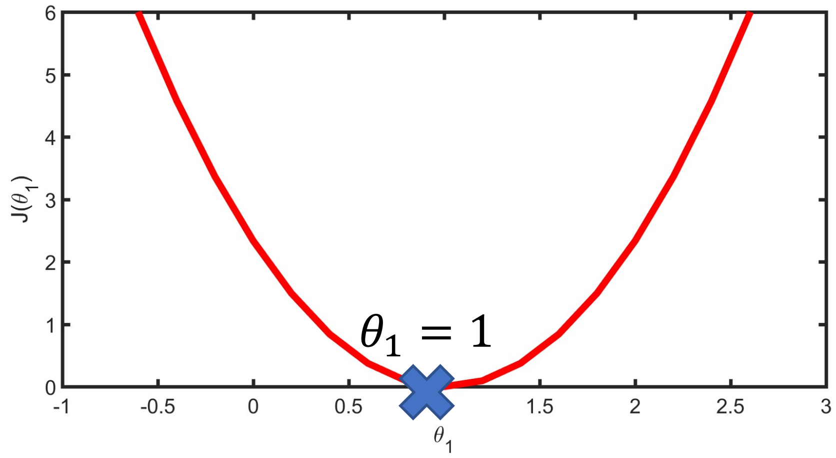
$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

What if we started with this $\theta_1^{[0]}$ instead?

Quiz: Gradient descent intuition

Suppose you're really lucky and your initial guess of $\theta_1^{[0]} = 1$ results in the minimum value of $J(\theta_1)$.
Assume $\alpha > 0$.
What happens if you apply a gradient descent update here?

$$\theta_1^{[1]} = \theta_1^{[0]} - \alpha \frac{\partial \mathrm{J}(\theta_1)}{\partial \theta_1}$$



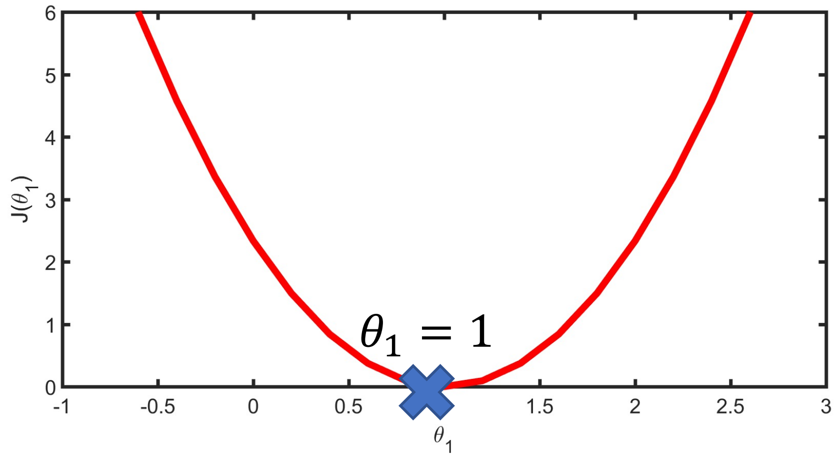
$\theta_1 = 1$ results in the minimum $J(\theta_1)$

(A) No change in θ_1
(B) A random change in θ_1
(C) Increase θ_1
(D) Decrease θ_1

Quiz: Gradient descent intuition

Suppose you're really lucky and your initial guess of $\theta_1^{[0]} = 1$ results in the minimum value of $J(\theta_1)$.
Assume $\alpha > 0$.
What happens if you apply a gradient descent update here?

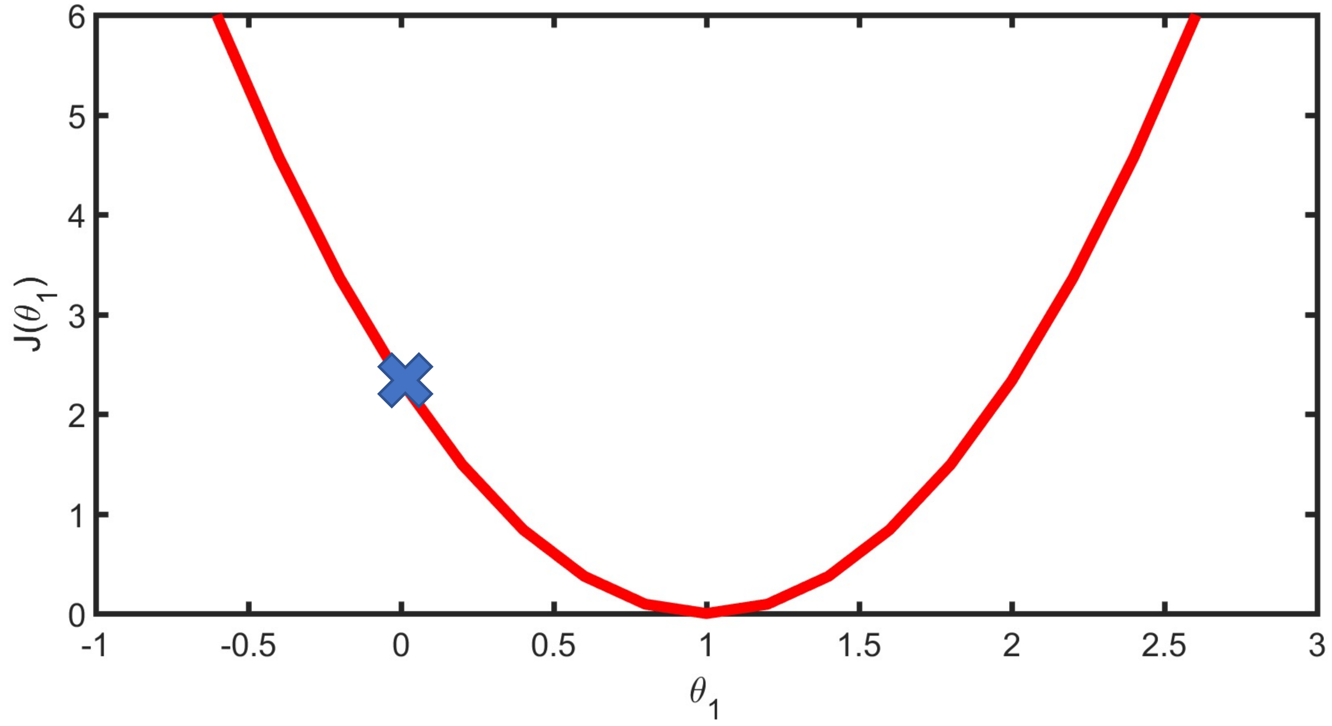
$$\theta_1^{[1]} = \theta_1^{[0]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$



$\theta_1 = 1$ results in the minimum $J(\theta_1)$

(A) No change in θ_1
(B) A random change in θ_1
(C) Increase θ_1
(D) Decrease θ_1

Gradient descent intuition

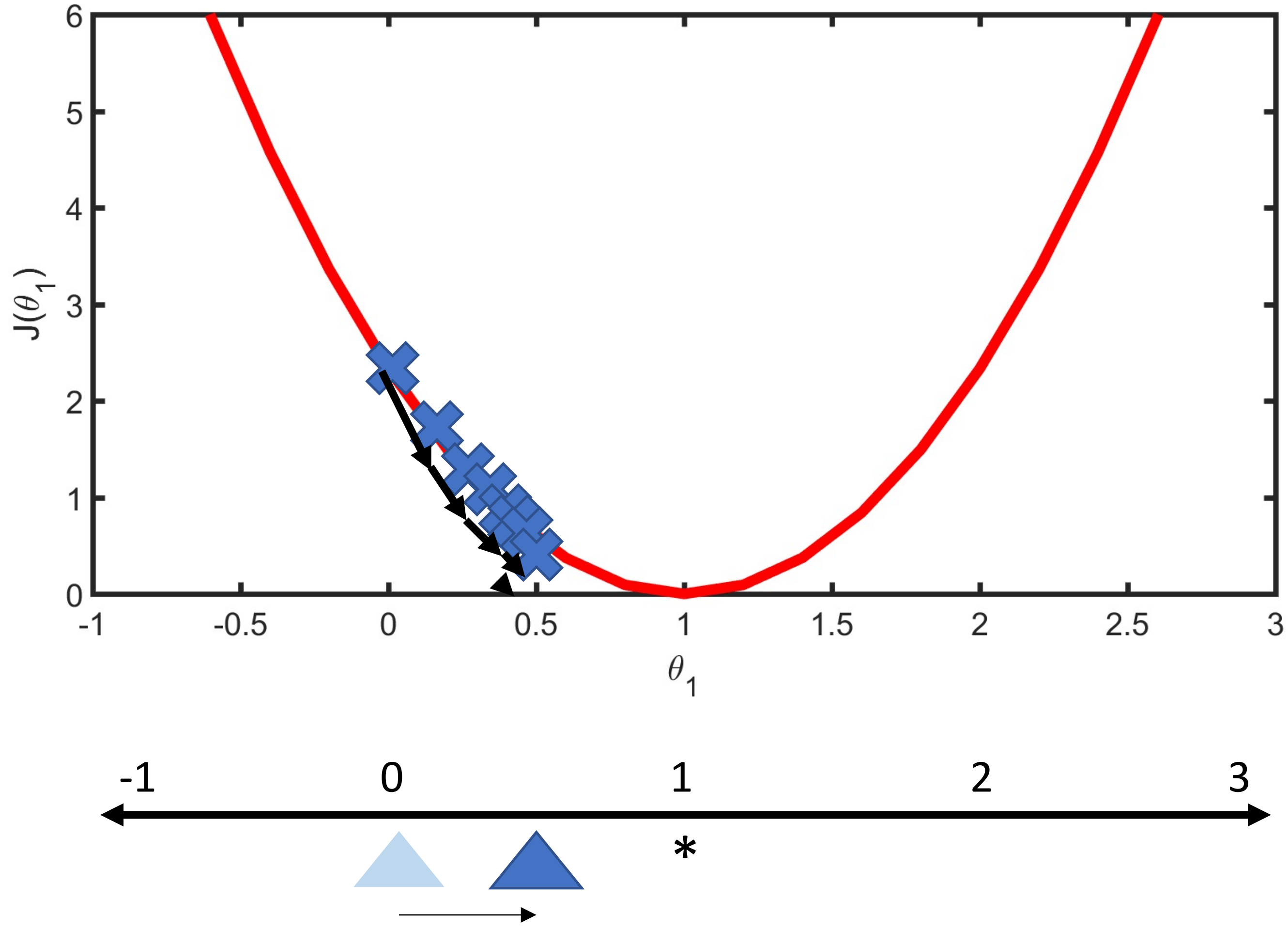


Update rule :

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

What is the role of α?

Gradient descent intuition



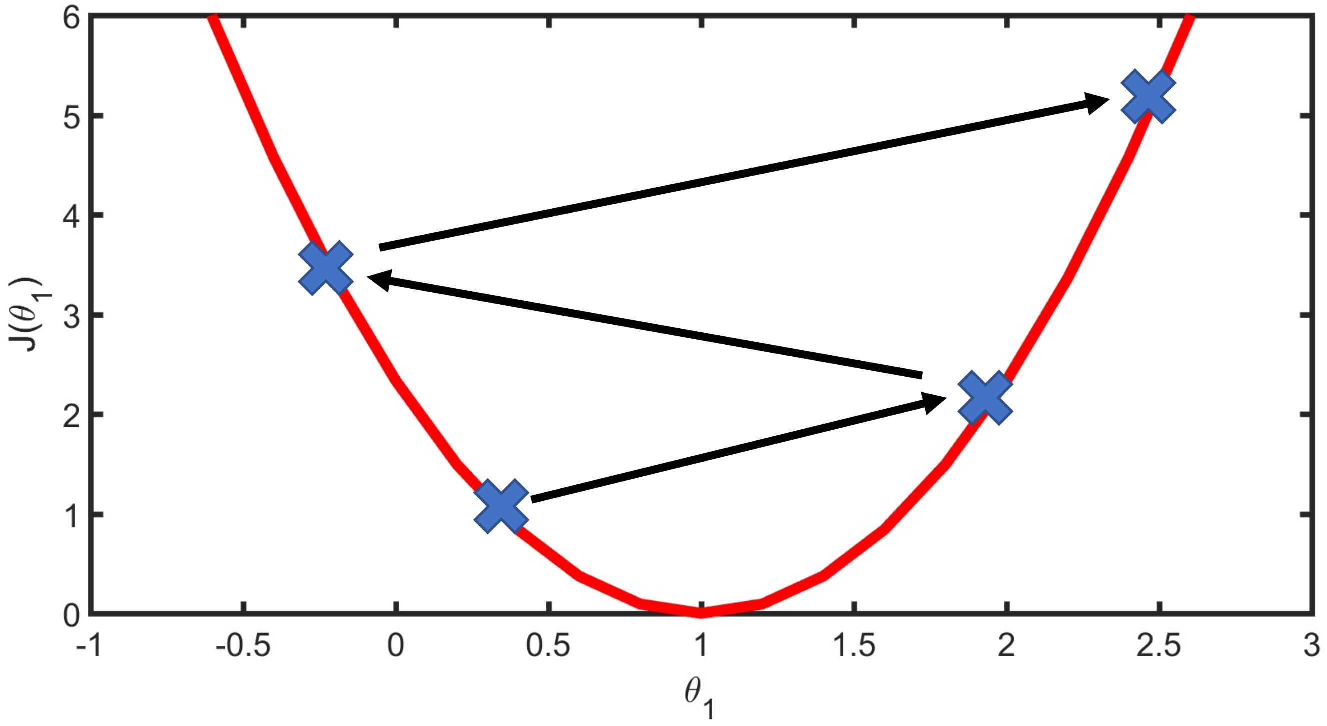
Update rule :

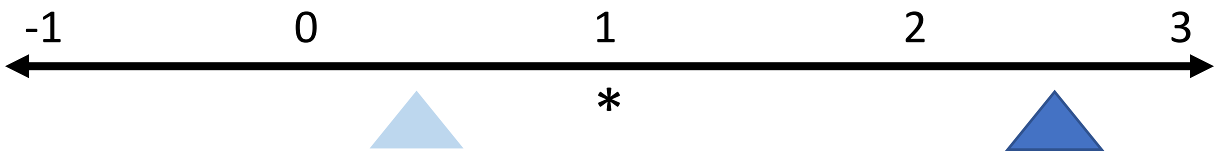
$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

When α is too **small;**

Convergence to θ_1^* can be prohibitively slow

Gradient descent intuition



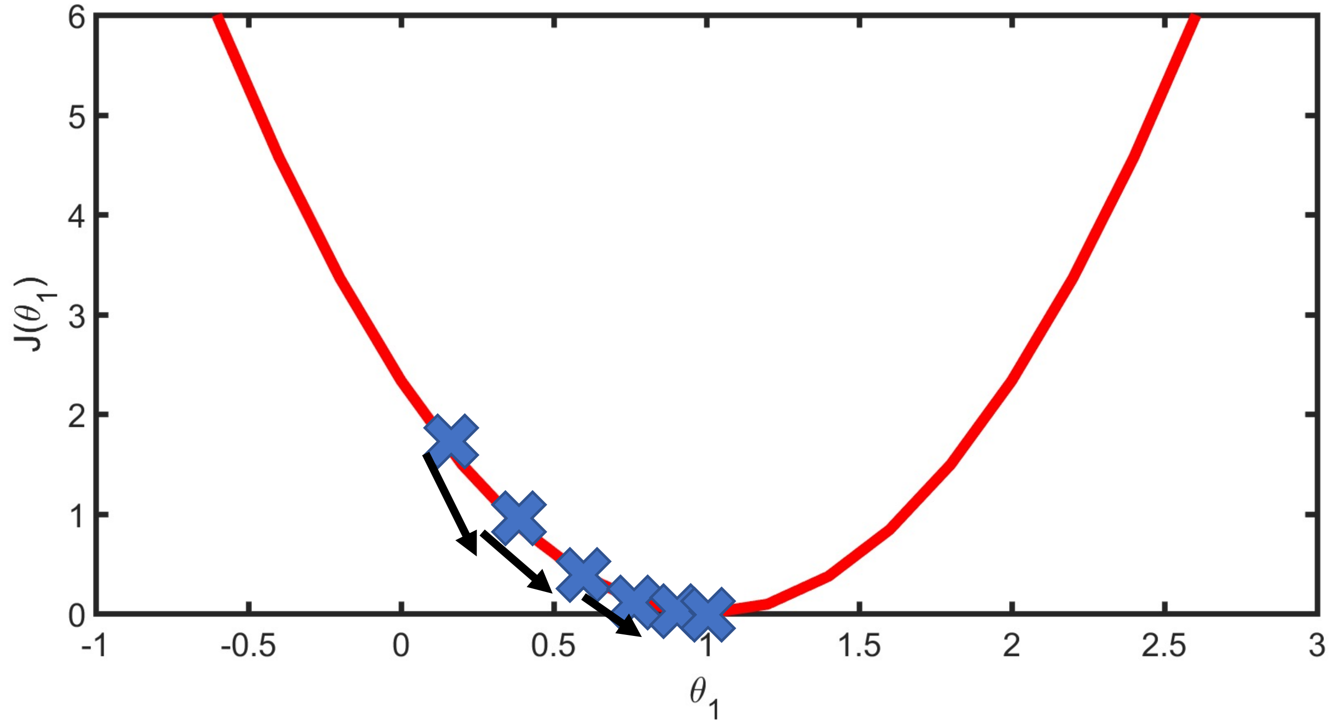


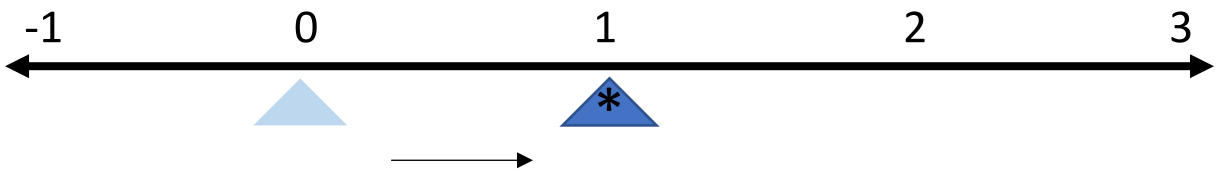
Update rule :

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

When α is too **large;**
Divergence may occur instead of convergence

Gradient descent intuition



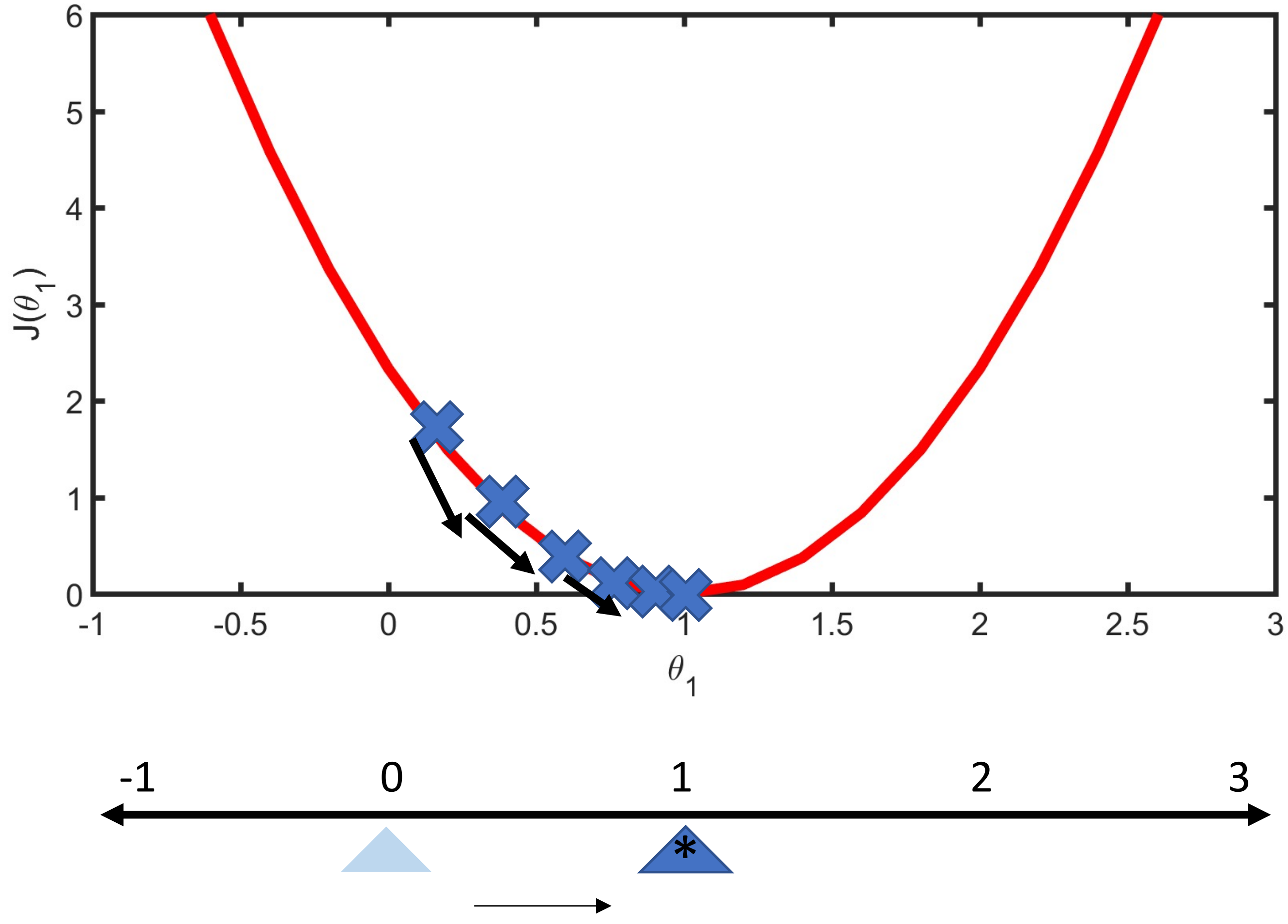


Update rule :

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial \mathrm{J}(\theta_1)}{\partial \theta_1}$$

A good choice of α leads to convergence to θ_1^* in a reasonable amount of time

Gradient descent intuition



Update rule :

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial J(\theta_1)}{\partial \theta_1}$$

-- For a linear regression cost function, a fixed α works because the gradient value itself tapers off

A solution strategy

Given a function $J(\theta_0, \theta_1)$:

Step 1: Start with some θ_0, θ_1

Step 2: Update θ_0, θ_1 such that it reduces $J(\theta_0, \theta_1)$

Step 3: Keep repeating step 2 until we hopefully reach the minimum value of $J(\theta_0, \theta_1)$

An iterative approach to finding the best θ_0, θ_1

Gradient descent algorithm

$$\theta_0^{[k+1]} = \theta_0^{[k]} - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$$

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial \mathrm{J}(\theta_0, \theta_1)}{\partial \theta_1}$$

Gradient descent algorithm

$$\theta_0^{[k+1]} = \theta_0^{[k]} - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$$

$$\theta_1^{[k+1]} = \theta_1^{[k]} - \alpha \frac{\partial \mathrm{J}(\theta_0, \theta_1)}{\partial \theta_1}$$

Update θ_0 and θ_1 simultaneously!

-- If you do sequentially, that's a different algorithm called coordinate descent

-- Parallel computation

Quiz: Gradient descent algorithm

Let $\theta_0^{[0]} = 1, \theta_1^{[0]} = 2$. For a different problem, it turns out that the update rule should be:

$\theta_j^{[1]} = \theta_j^{[0]} + \sqrt{\theta_0 \theta_1}$ for $j = 0,1$. What are $\theta_0^{[1]}, \theta_1^{[1]}$?

(A) $\theta_0^{[1]} = 1, \theta_1^{[1]} = 2$

(B) $\theta_0^{[1]} = 1 + \sqrt{2}, \theta_1^{[1]} = 2 + \sqrt{2}$

(C) $\theta_0^{[1]} = 2 + \sqrt{2}, \theta_1^{[1]} = 1 + \sqrt{2}$

(D) $\theta_0^{[1]} = 1 + \sqrt{2}, \theta_1^{[1]} = 2 + \sqrt{2(1 + \sqrt{2})}$

Quiz: Gradient descent algorithm

Let $\theta_0^{[0]} = 1, \theta_1^{[0]} = 2$. For a different problem, it turns out that the update rule should be:

$\theta_j^{[1]} = \theta_j^{[0]} + \sqrt{\theta_0 \theta_1}$ for $j = 0,1$. What are $\theta_0^{[1]}, \theta_1^{[1]}$?

(A) $\theta_0^{[1]} = 1, \theta_1^{[1]} = 2$

(B) $\theta_0^{[1]} = 1 + \sqrt{2}, \theta_1^{[1]} = 2 + \sqrt{2}$

(C) $\theta_0^{[1]} = 2 + \sqrt{2}, \theta_1^{[1]} = 1 + \sqrt{2}$

(D) $\theta_0^{[1]} = 1 + \sqrt{2}, \theta_1^{[1]} = 2 + \sqrt{2(1 + \sqrt{2})}$

Gradient descent for general cost functions

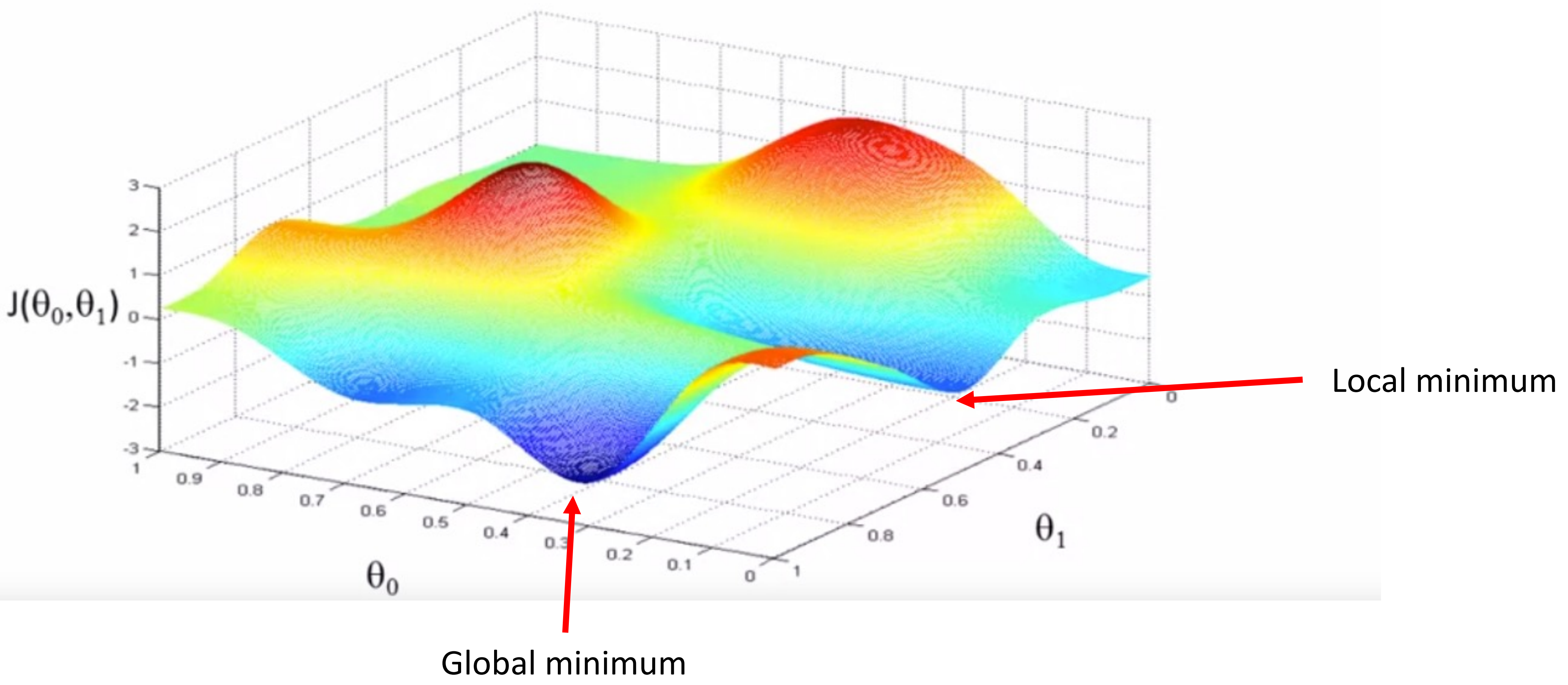


Image courtesy Dr. Andrew Ng, Stanford University Machine learning

Gradient descent for general cost functions

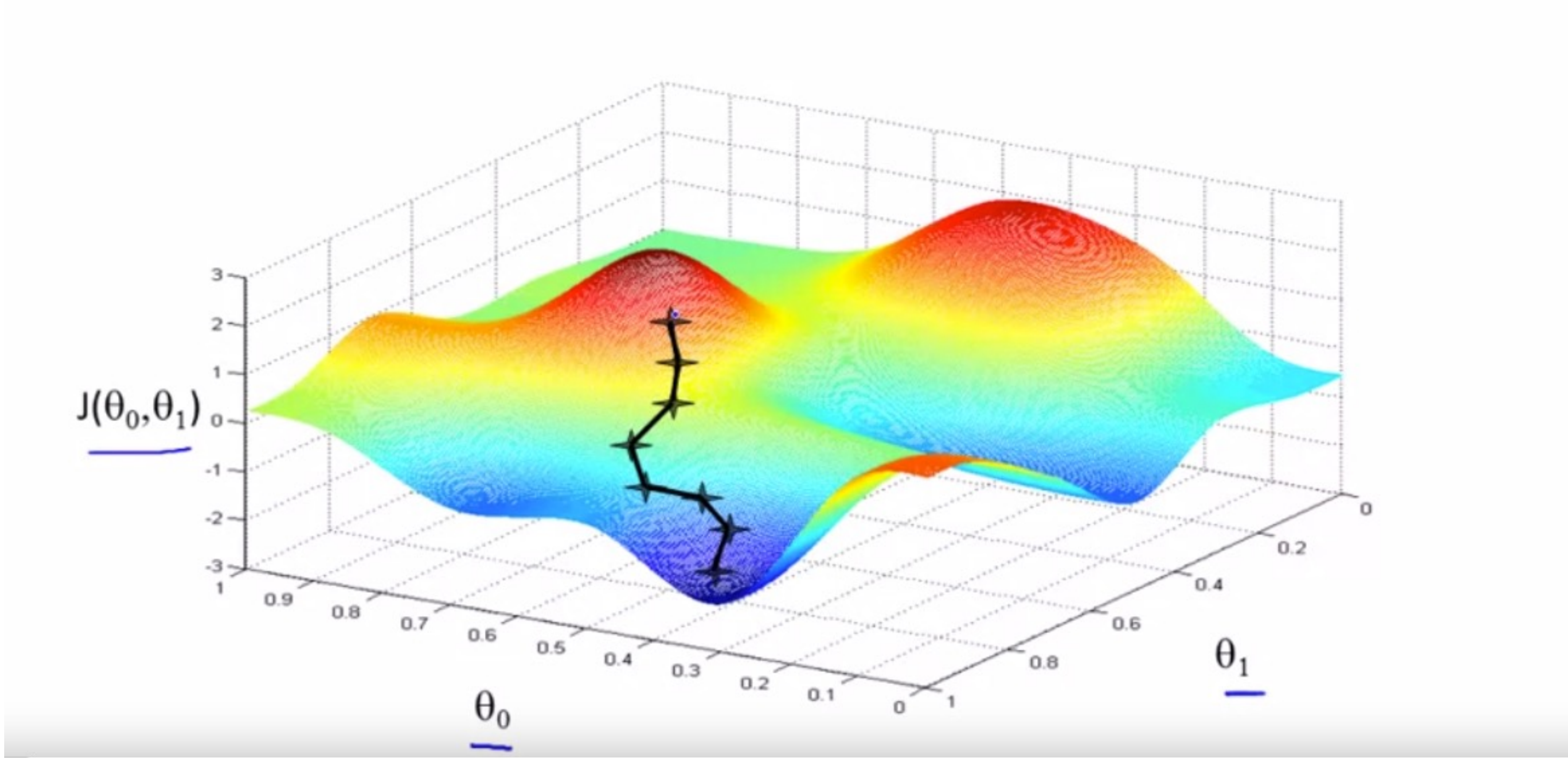


Image courtesy Dr. Andrew Ng, Stanford University Machine learning

Gradient descent for general cost functions

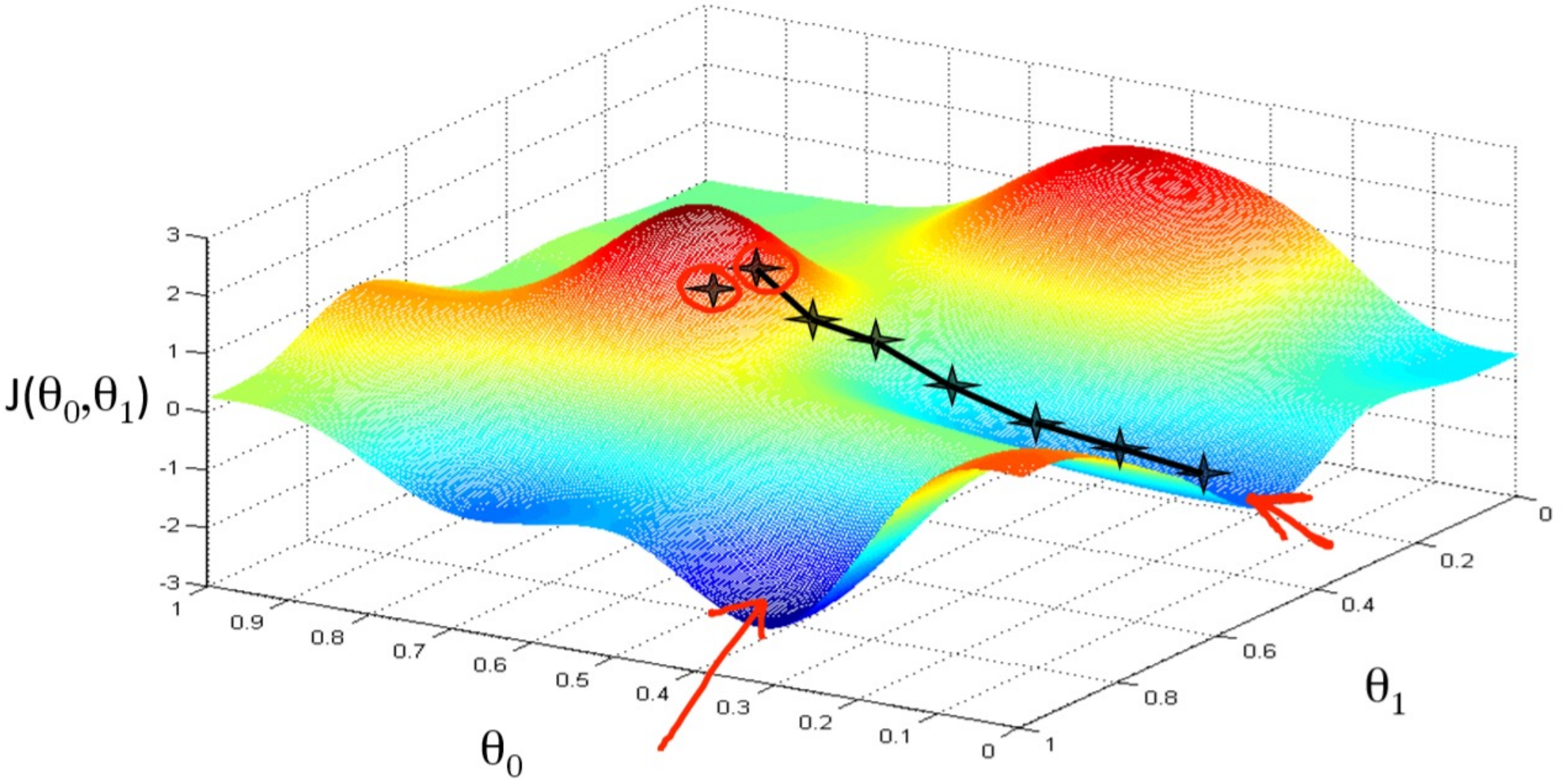


Image courtesy Dr. Andrew Ng, Stanford University Machine learning

Designing a gradient descent framework

Given a function $J(\theta_0, \theta_1)$:

Step 1: Start with some θ_0, θ_1

It's an art. Try multiple random initializations.

Step 2: Update θ_0, θ_1 such that it reduces $J(\theta_0, \theta_1)$

Computing the derivative is important!
A varying α can help!

Step 3: Keep repeating step 2 until we hopefully reach the minimum value of $J(\theta_0, \theta_1)$

Either fix maximum iterations or set a threshold on % change per iteration

Gradient descent for general cost functions

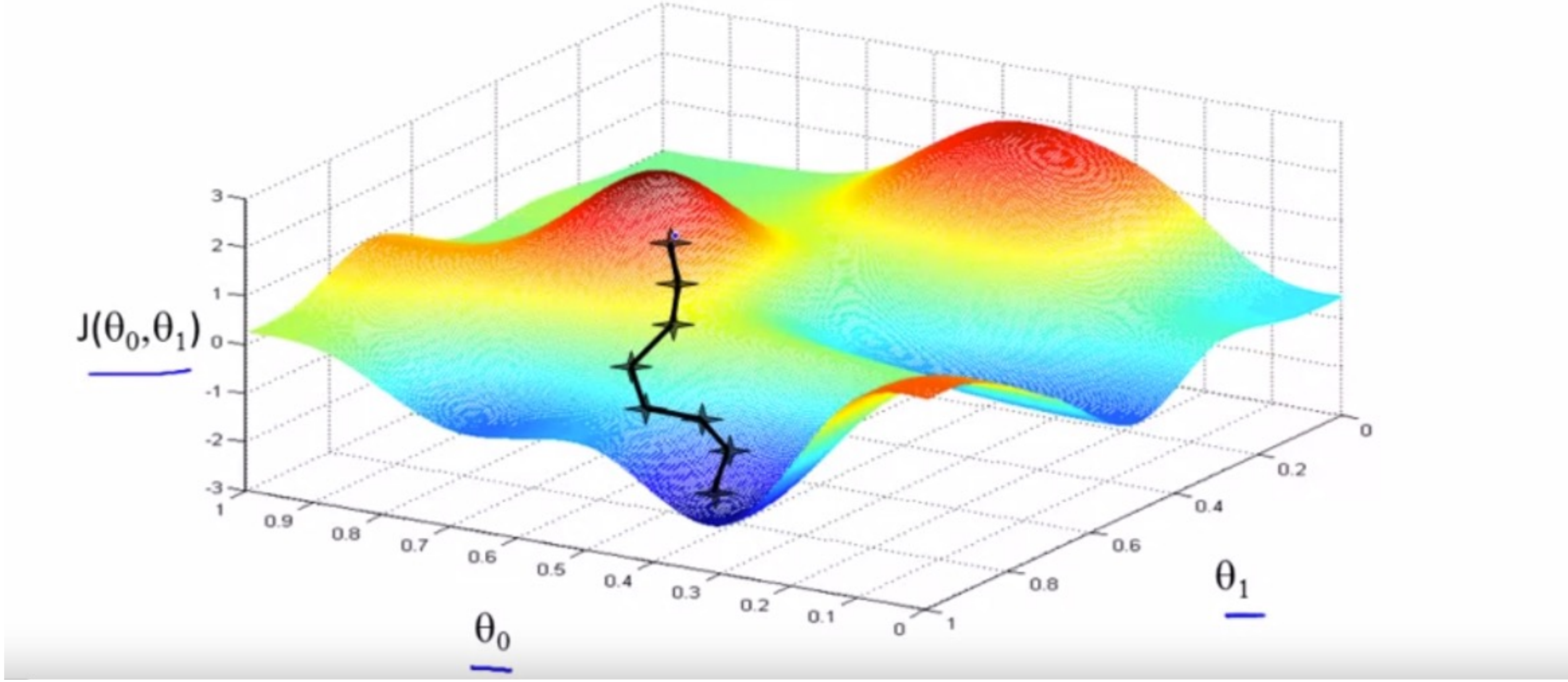


Image courtesy Dr. Andrew Ng, Stanford University Machine learning

Quiz: Gradient descent for general cost functions

Which of the following statements are true regarding gradient descent?

1. To make gradient descent converge, we must always decrease α with time

2. Gradient descent is guaranteed to find the global minimum for any cost function $J(\theta_0, \theta_1)$

3. Gradient descent can converge if α is fixed (But α shouldn't be too large, else there is divergence)

4. For the linear regression cost function, local and global minimum are one and the same

(A) 1 and 4

(B) 3 and 4

(C) 1 and 2

(D) 4

Quiz: Gradient descent for general cost functions

Which of the following statements are true regarding gradient descent?

1. To make gradient descent converge, we must always decrease α with time

2. Gradient descent is guaranteed to find the global minimum for any cost function $J(\theta_0, \theta_1)$

3. Gradient descent can converge if α is fixed (But α shouldn't be too large, else there is divergence)

4. For the linear regression cost function, local and global minimum are one and the same

(A) 1 and 4

$\boxed{\text{(B) 3 and 4}}$

(C) 1 and 2

(D) 4

The Normal equation

There is a direct, mathematical way to obtain the 'best' $\theta = [\theta_0, \theta_1]$ using one equation

$$\boxed{\theta^* = (x^T x)^{-1} x^T y}$$

x: Area in square feet	y: Price (x1000) in $
2104	399
1600	329
2400	369
1416	232
3000	539
…	…

The Normal equation

There is a direct, mathematical way to obtain the 'best' $\theta = [\theta_0, \theta_1]$ using one equation

$$\boxed{\theta^* = (x^T x)^{-1} x^T y}$$

Why, then, don't we use that instead of gradient descent?

-- Computational cost when number of predictors is very large
-- Sparse data (when most of the predictors are zero)
-- Collinear predictors

Learning outcomes - II

By now, you know:

-- Gradient descent is an iterative algorithm that is a standard tool for minimizing cost functions

-- Involves choosing a starting guess for θ, updating θ using a learning rate α and a gradient term $\frac{\partial J(\theta)}{\partial \theta}$, and continuing to update until we reach the solution

-- Why gradient descent converges to a local minimum of the cost function if α isn't too large

-- Effects of a too small or too large α

-- Gradient descent for general cost functions

-- The normal equation is an alternative approach to finding the linear regression model, albeit with a high memory requirement