Machine Learning and Pattern Recognition

A High Level Overview

Prof. Anderson Rocha

(Main bulk of slides kindly provided by **Prof. Sandra Avila**
and largely based on other materials as well (e.g., Andrew Ng's))

Institute of Computing (IC/Unicamp)

MC886/MO444

House Price Prediction

$ 70 000

https://www.youtube.com/watch?v=IpGxLWOIZy4

House Price Prediction

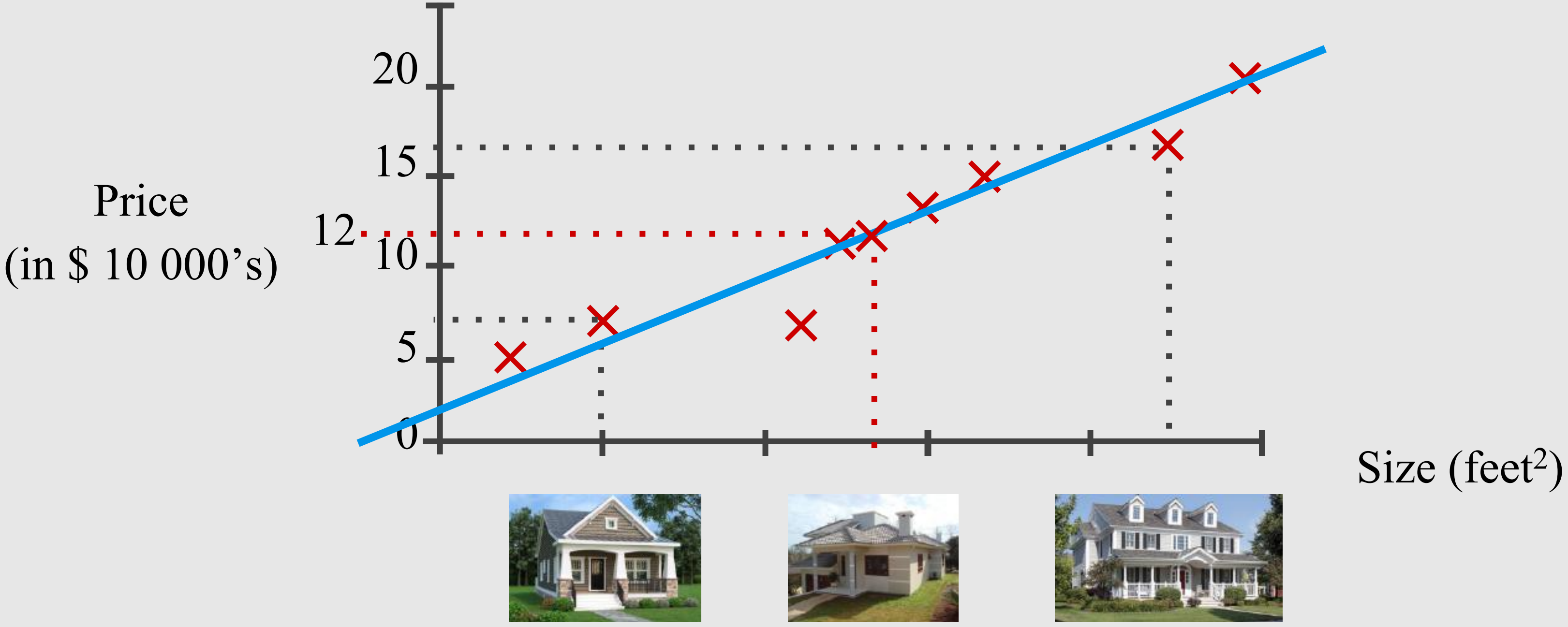
$ 160 000

House Price Prediction

???

House Price Prediction

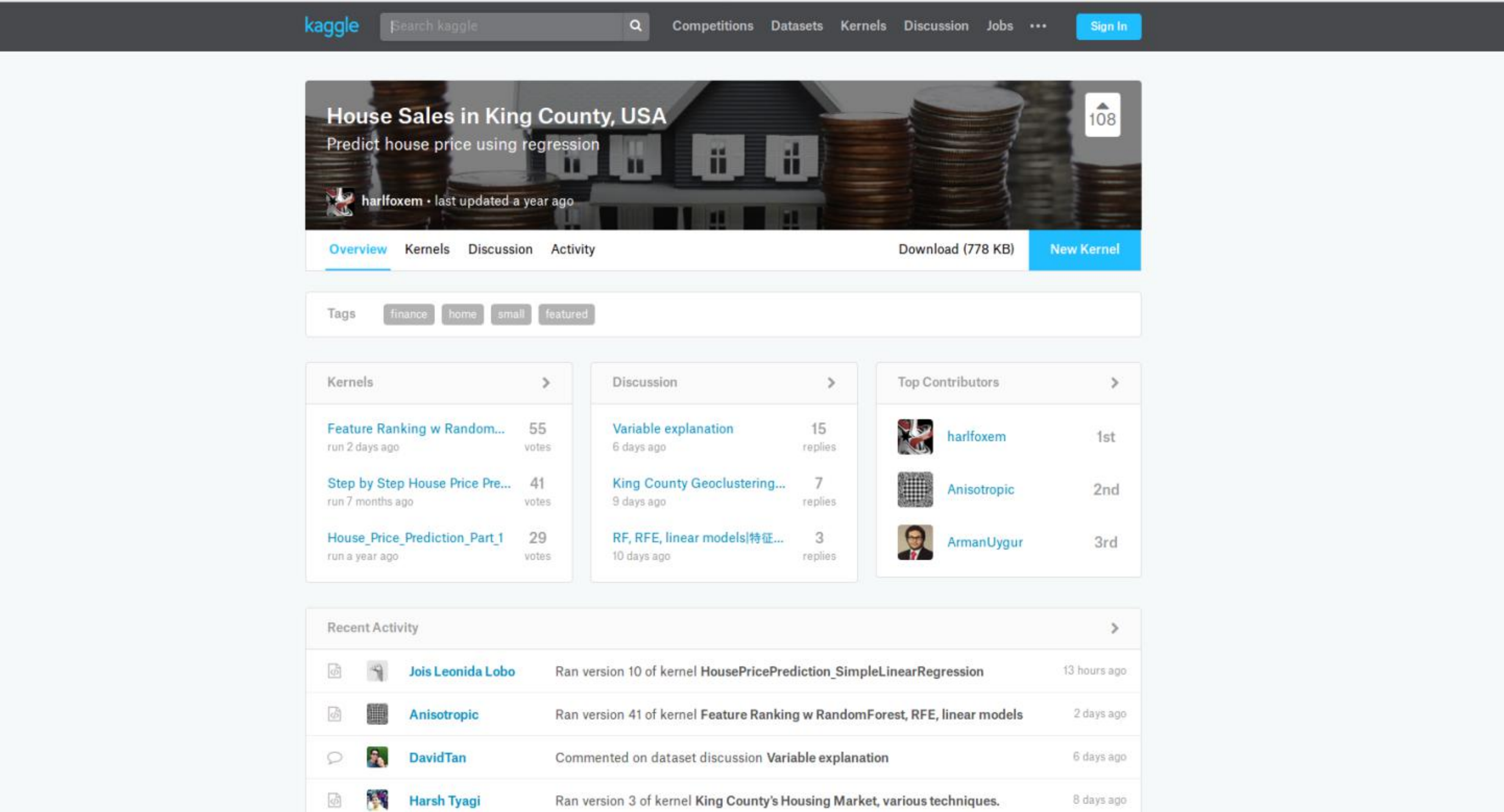
Linear Regression



Today’s Agenda

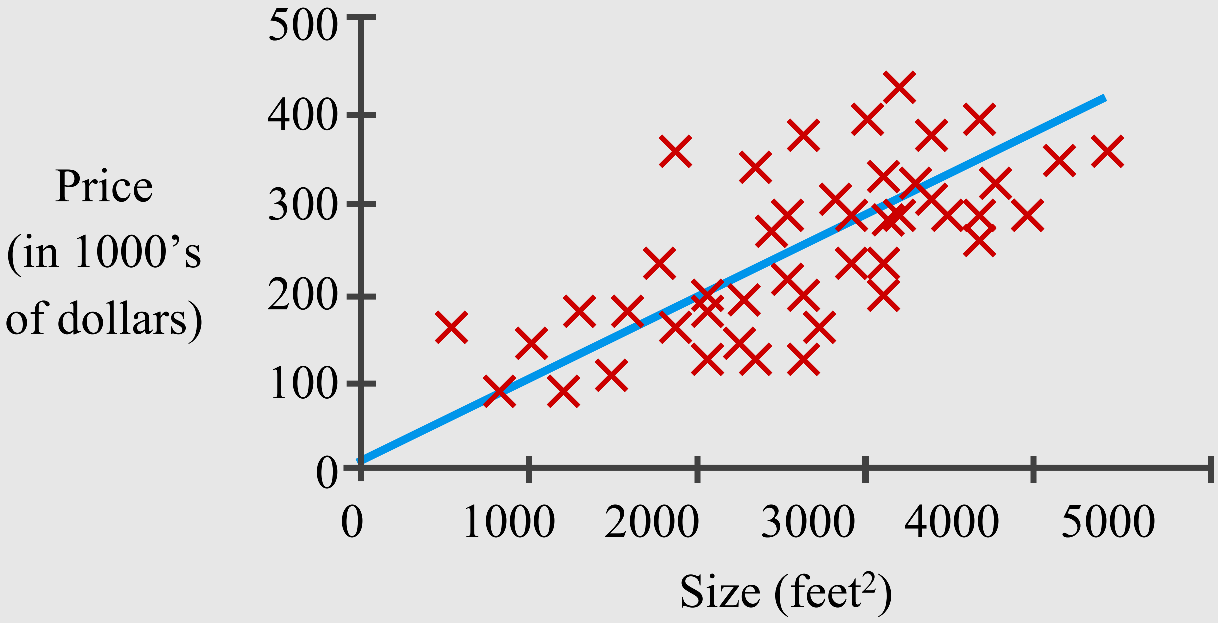
- Linear Regression with One Variable
  - Model Representation
  - Cost Function
  - Gradient Descent
- Linear Regression with Multiple Variables
  - Gradient Descent for Multiple Variables
  - Feature Scaling
  - Learning Rate
  - Features and Polynomial Regression
  - Normal Equation

Model Representation



https://www.kaggle.com/harlfoxem/housesalesprediction

Housing Prices



Supervised Learning

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

Regression Problem

Predict real-valued output

Training set of housing prices

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

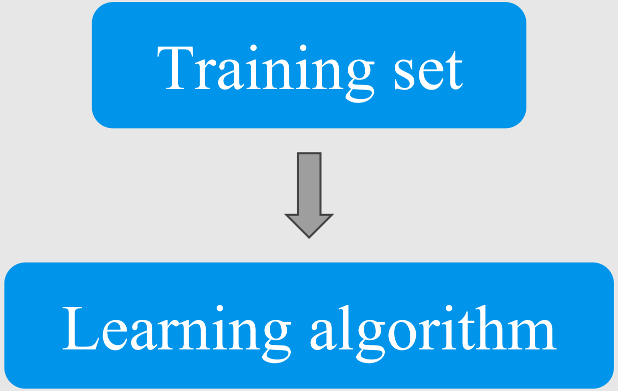
Notation:

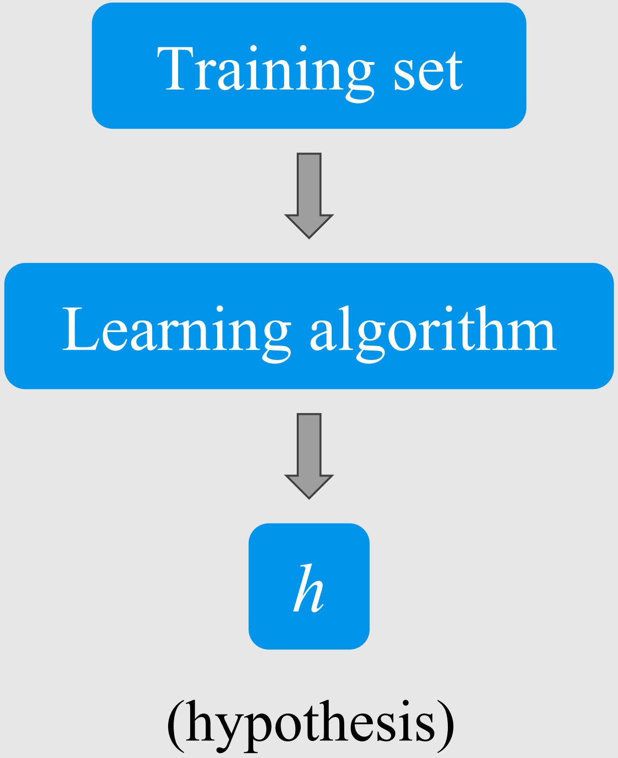
m = Number of training examples

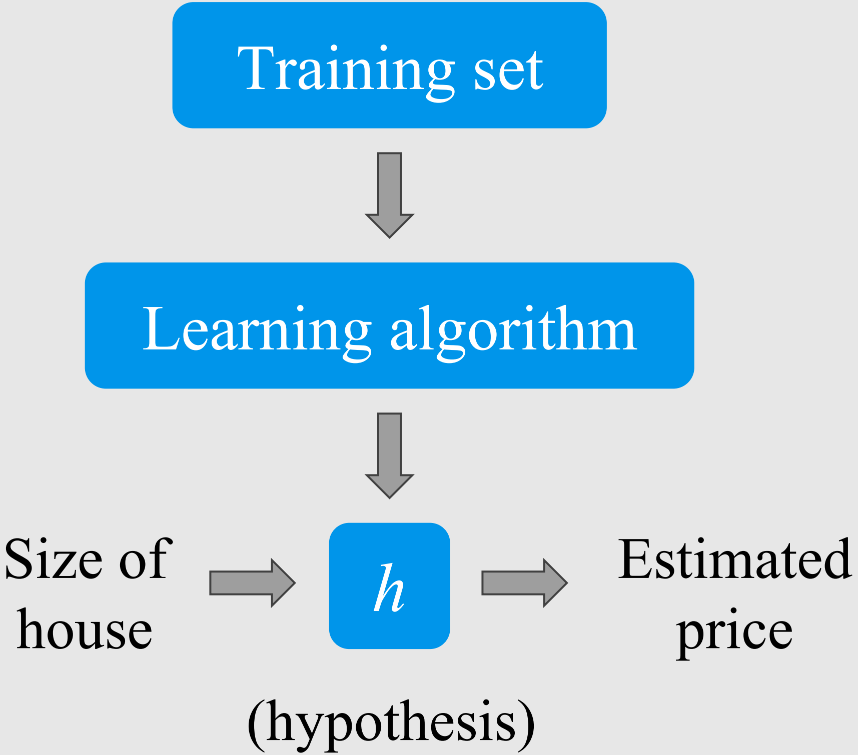
x's = "input" variable / features

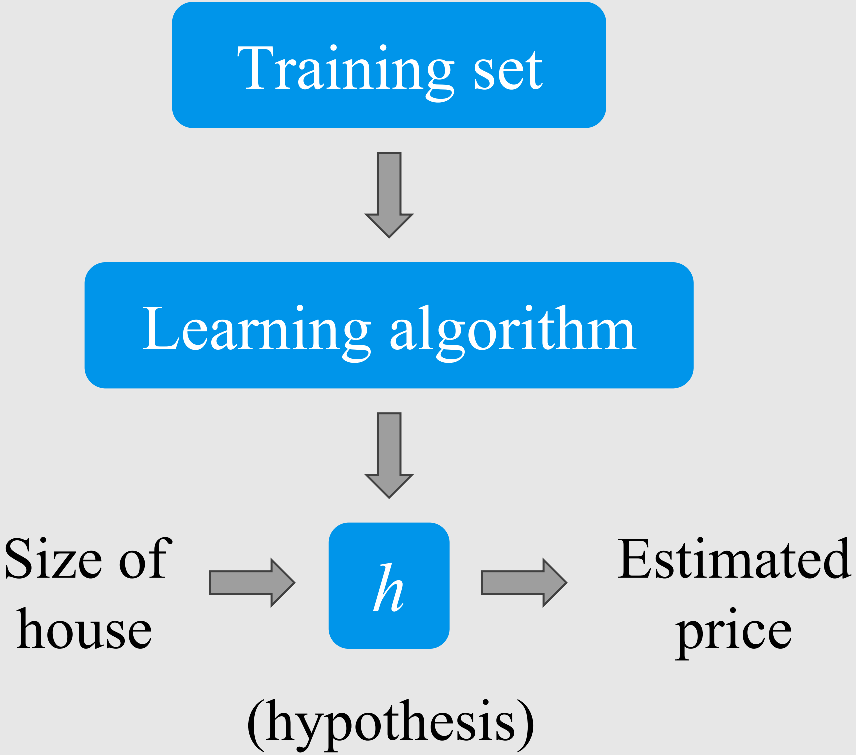
y's = "output" variable / "target" variable

Training set


Training set
Learning algorithm

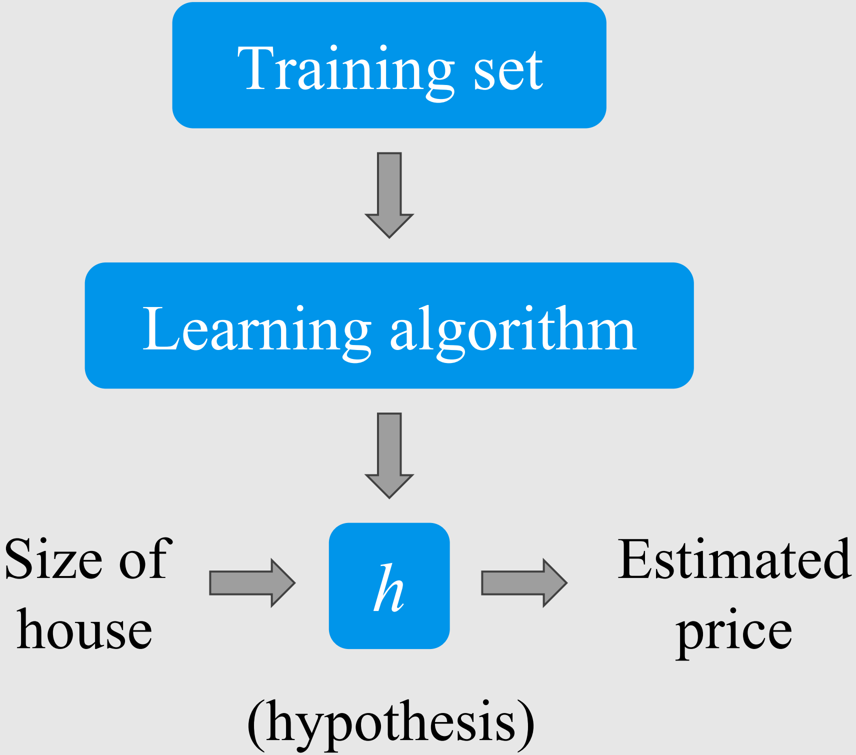

Training set
Learning algorithm
h
(hypothesis)


Training set
Learning algorithm
Size of house
h
Estimated price
(hypothesis)



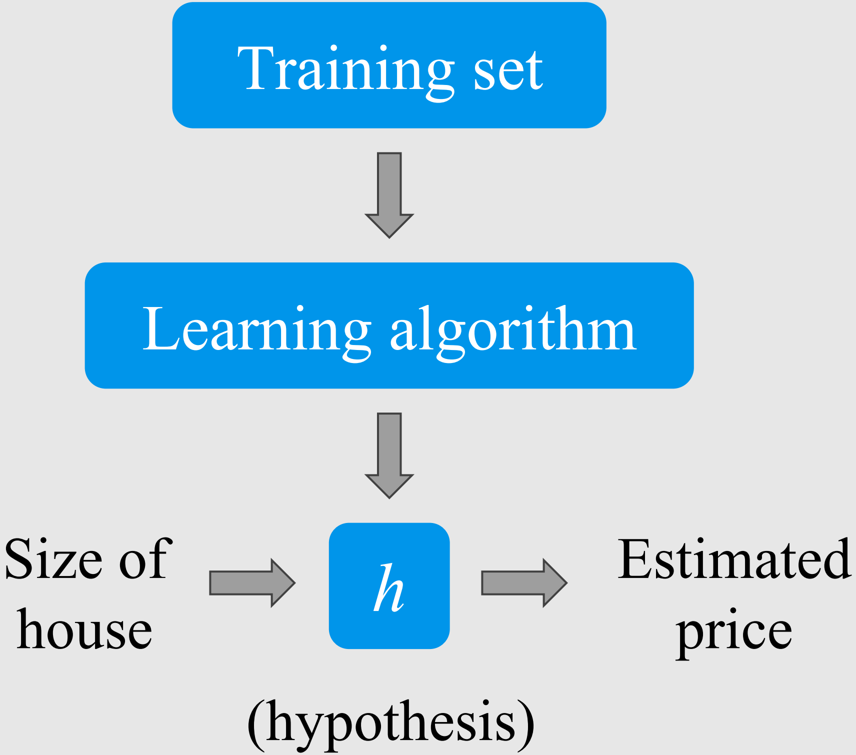
***h* maps *x*’s to *y*’s**

How do we represent h ?



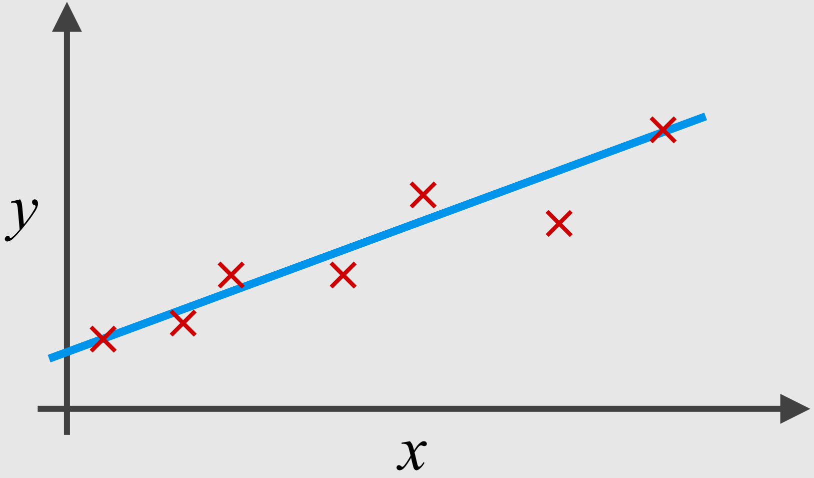
***h* maps *x*'s to *y*'s**

How do we represent h ?

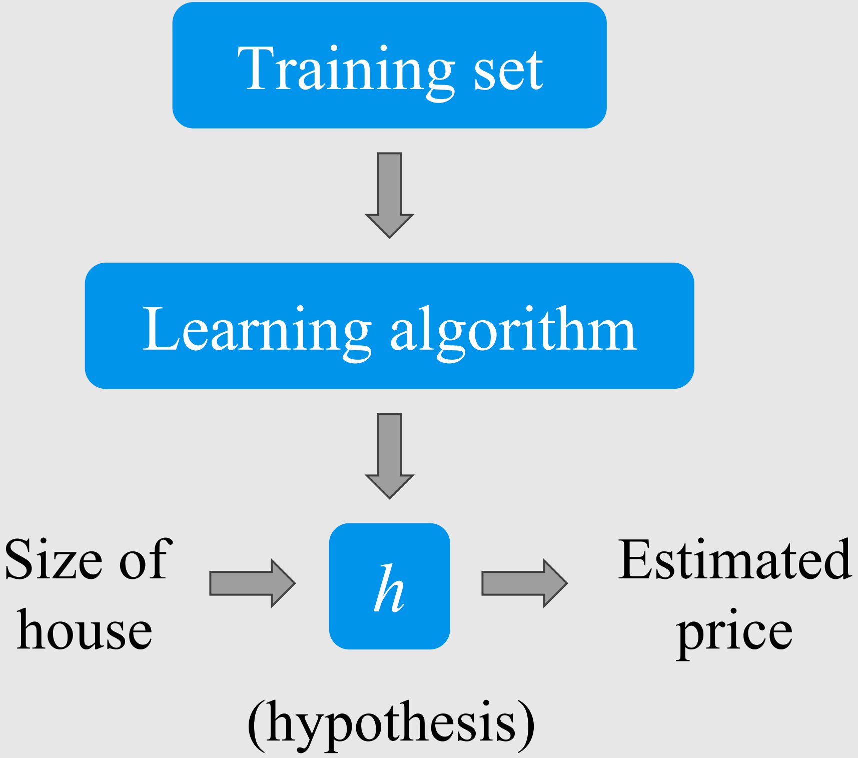


***h* maps *x*'s to *y*'s**

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

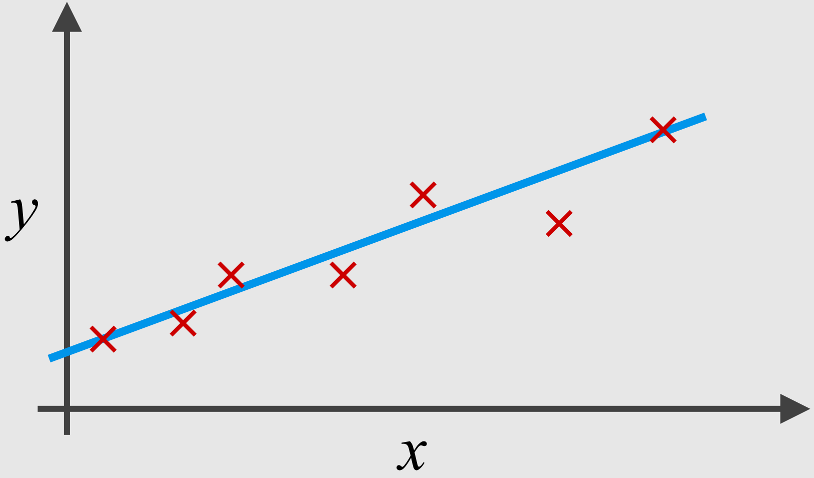


How do we represent *h* ?



***h* maps *x*'s to *y*'s**

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



Linear regression with one variable.
Univariate linear regression.

Cost Function

Training Set

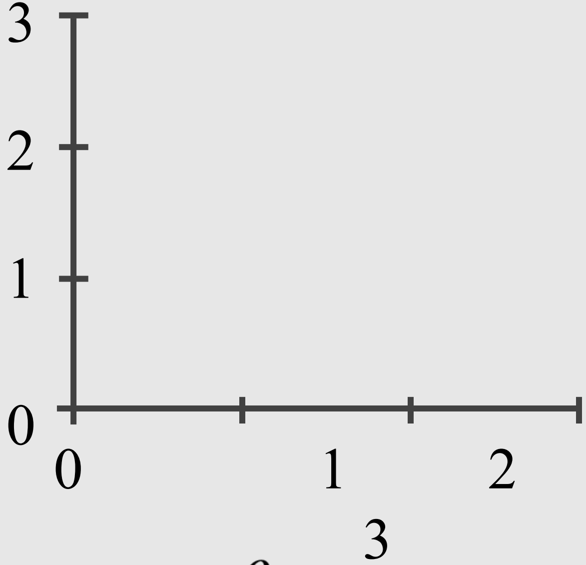
Size in feet² (x)	**Price ($) in 1000's ($y$)**
2104	460
1416	232
1534	315
852	178
...	...

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

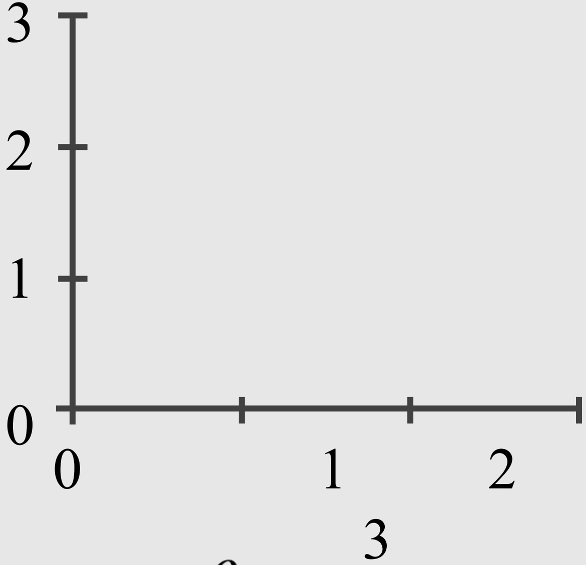
θ_i's: Parameters

How to choose θ_i's ?

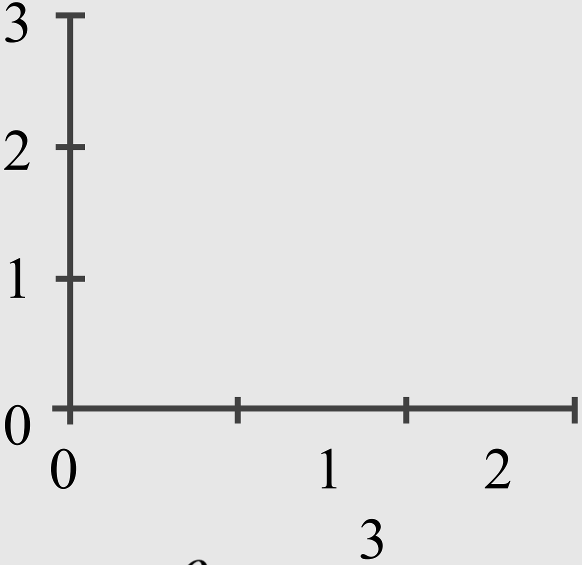
$$h_\theta(x) = \theta_0 + \theta_1 x$$



$\theta_0 = 1.5$
$\theta_1 = 0$

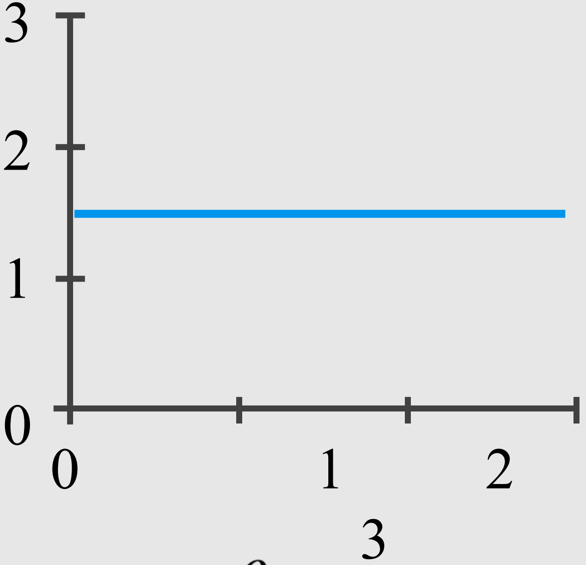


$\theta_0 = 0$
$\theta_1 = 0.5$



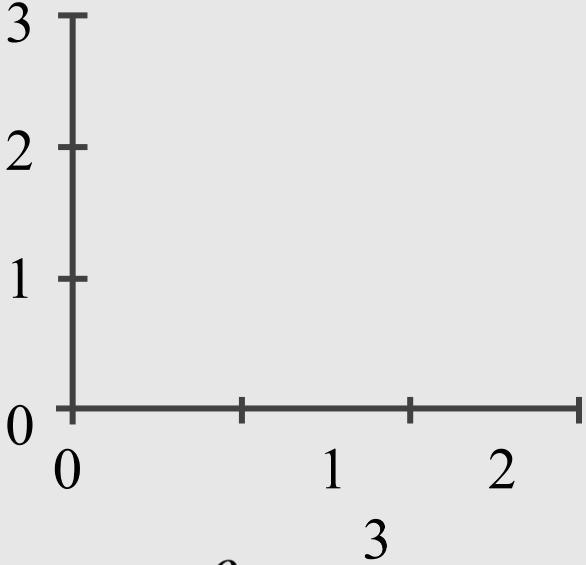
$\theta_0 = 1$
$\theta_1 = 0.5$

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



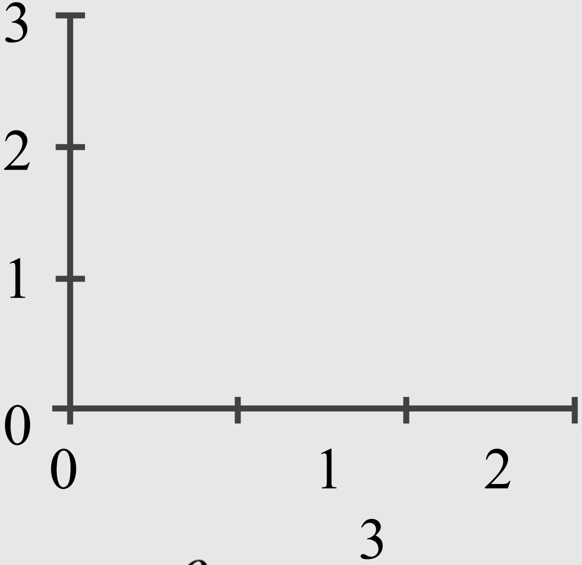
$\theta_0 = 1.5$

$\theta_1 = 0$



$\theta_0 = 0$

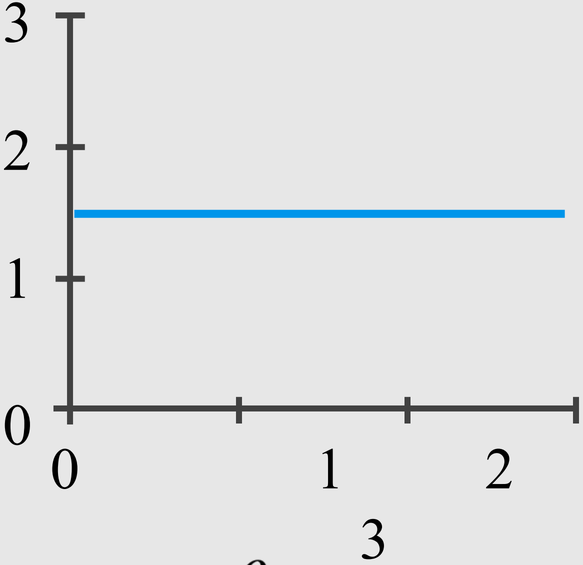
$\theta_1 = 0.5$



$\theta_0 = 1$

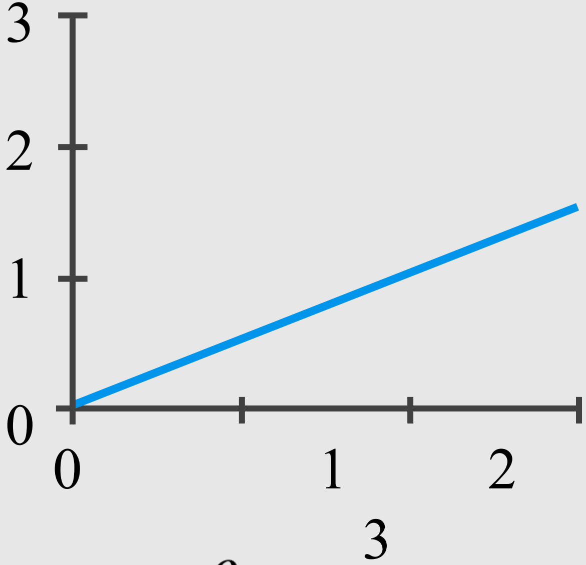
$\theta_1 = 0.5$

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



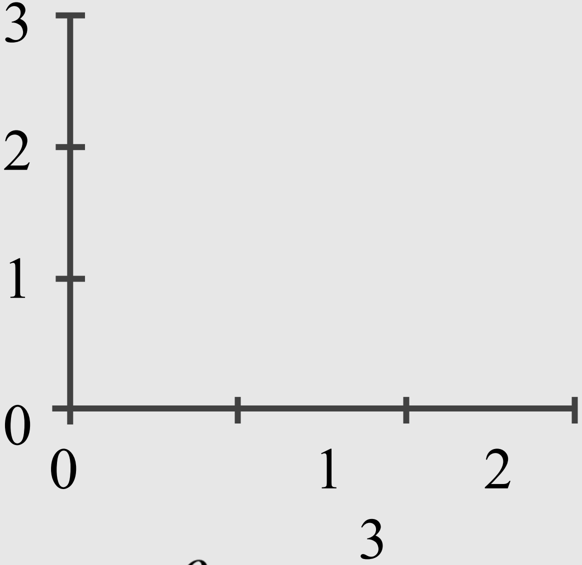
$\theta_0 = 1.5$

$\theta_1 = 0$



$\theta_0 = 0$

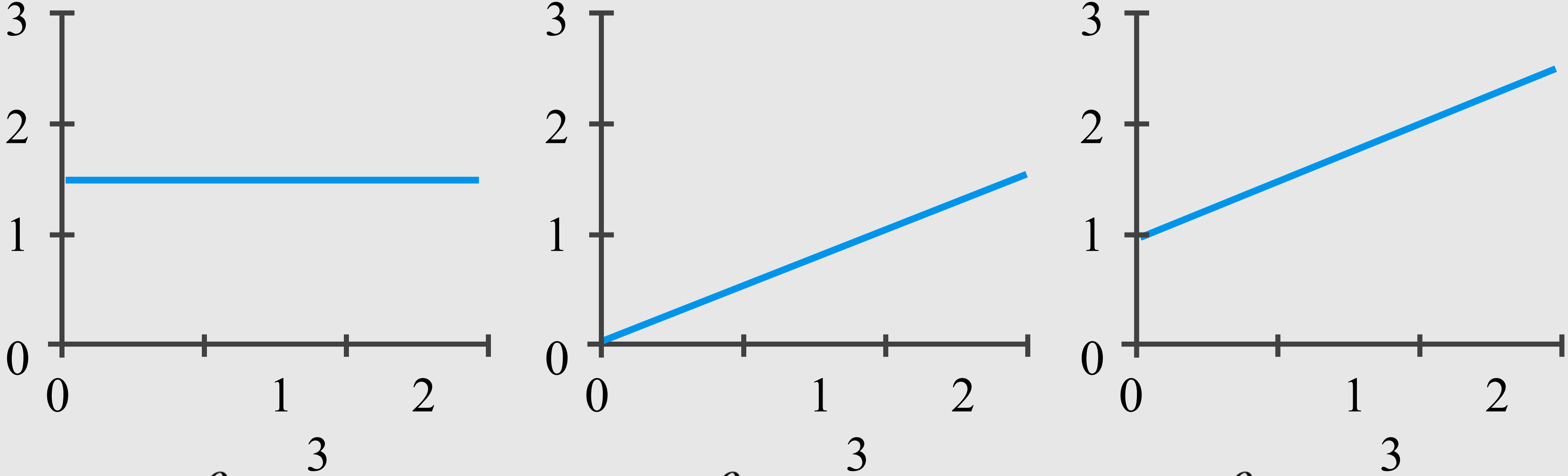
$\theta_1 = 0.5$



$\theta_0 = 1$

$\theta_1 = 0.5$

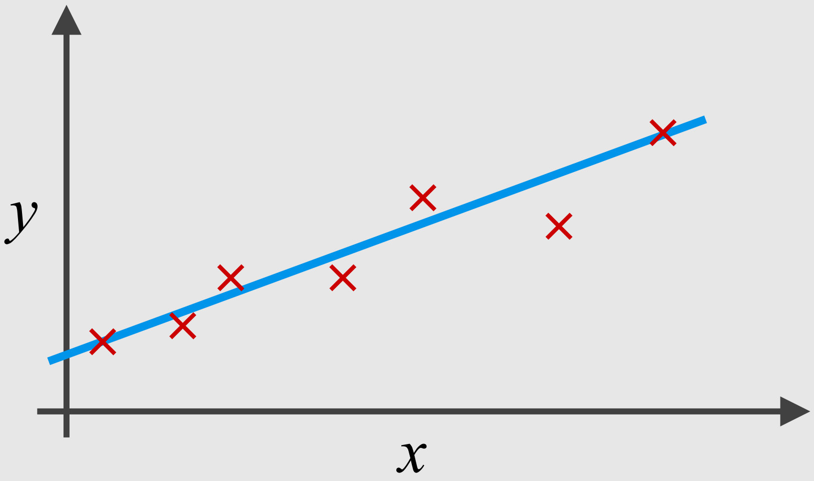
$$h_\theta(x) = \theta_0 + \theta_1 x$$



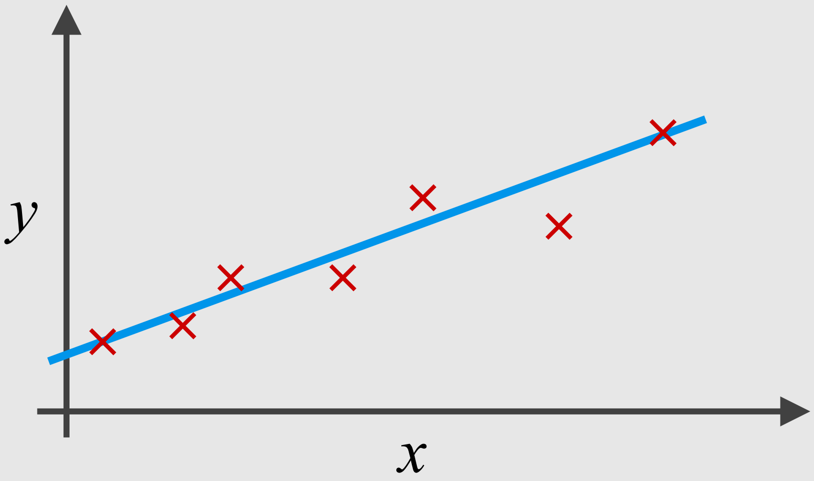
$\theta_0 = 1.5$
$\theta_1 = 0$

$\theta_0 = 0$
$\theta_1 = 0.5$

$\theta_0 = 1$
$\theta_1 = 0.5$

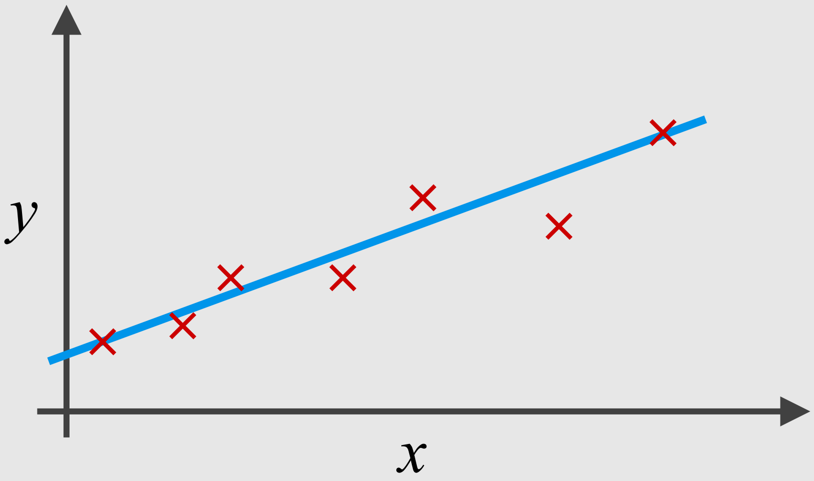


Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ close to y for our training examples (x,y)



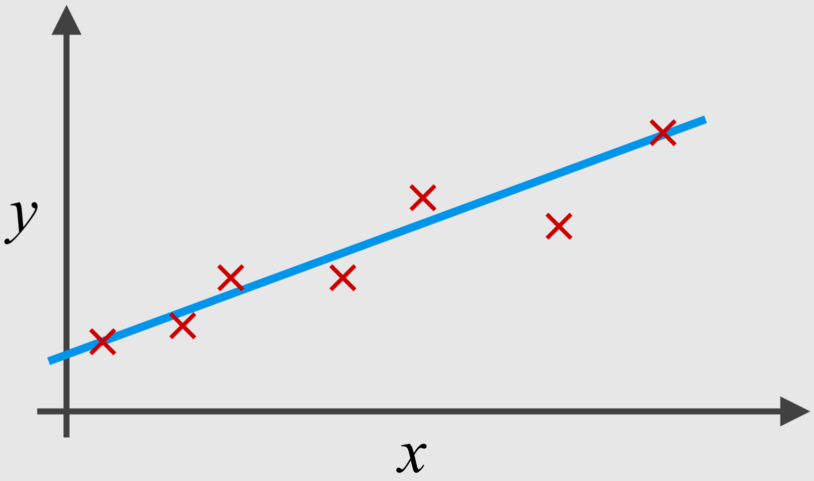
$$\underset{\theta_0,\theta_1}{\text{minimize}}$$

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



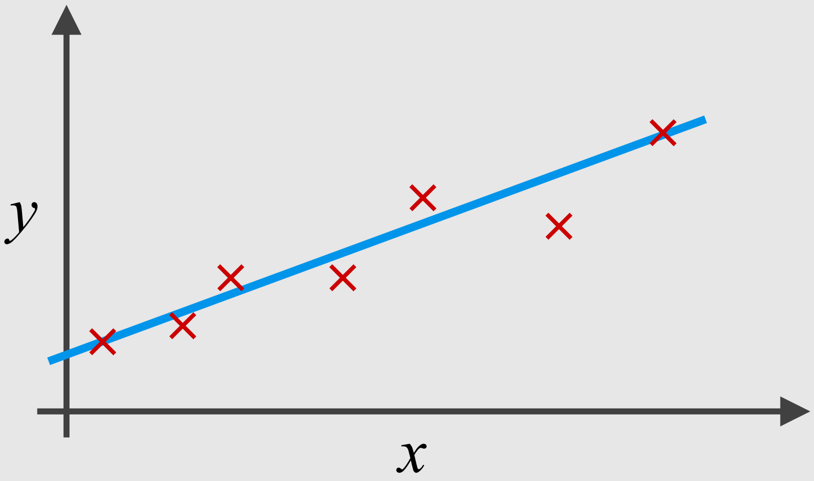
$$\underset{\theta_0,\theta_1}{\text{minimize}} \qquad (h_\theta(x \;) - y \;)^2$$

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



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

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

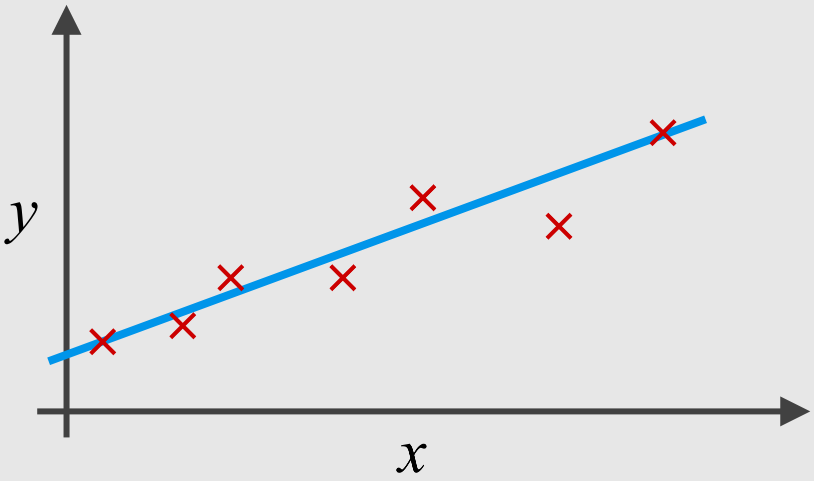


$$\underset{\theta_0,\theta_1}{\text{minimize}} \ \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

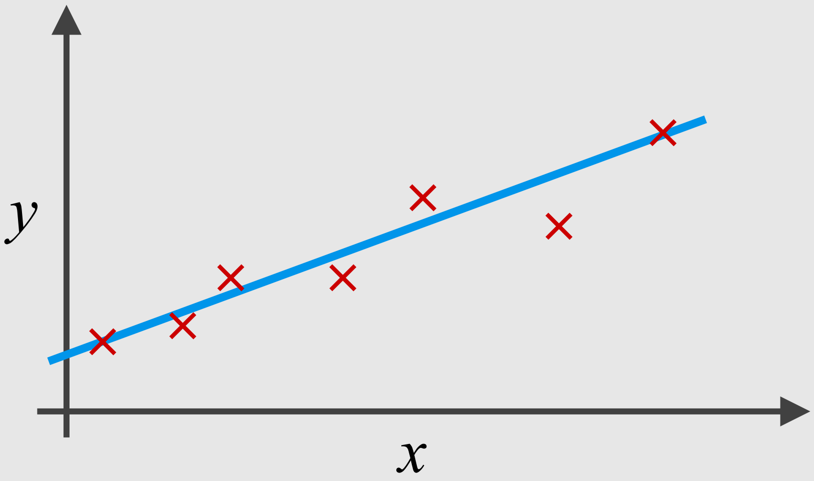
Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ close to y for our training examples (x,y)

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

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



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

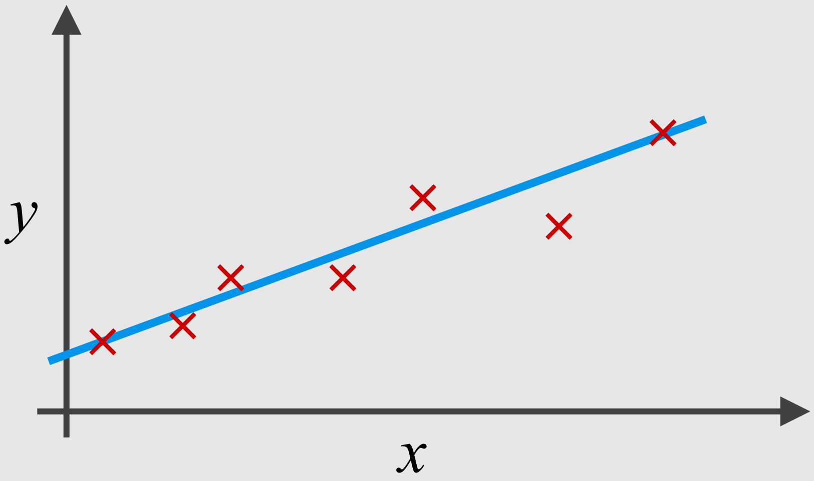


$$\underset{\theta_0,\theta_1}{\text{minimize}} \; \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

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

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

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



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

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

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

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

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

Cost function

(Squared error function)

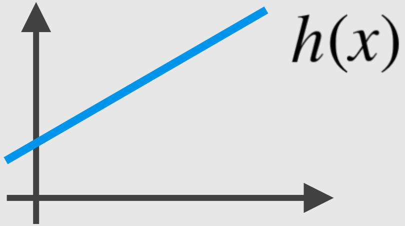
Cost Function
Intuition I

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:

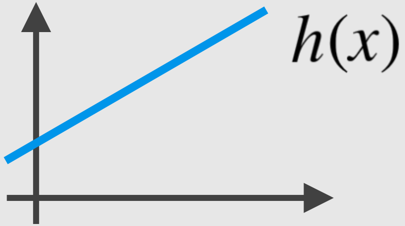
$$\underset{\theta_0,\theta_1}{\text{minimize}}\ J(\theta_0,\theta_1)$$

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$



Cost Function:

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

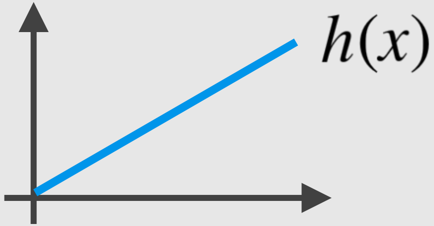
Goal:

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

<u>Simplified</u>

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

$$\theta_1$$



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

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

$h_\theta(x)$

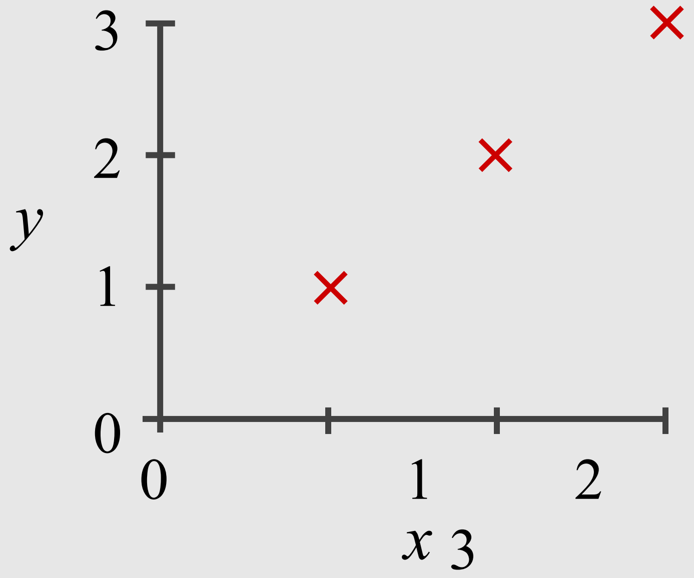
(for fixed θ_1 , this is a function of x)

$J(\theta_1)$

(function of the parameters θ_1)

$h_\theta(x)$

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

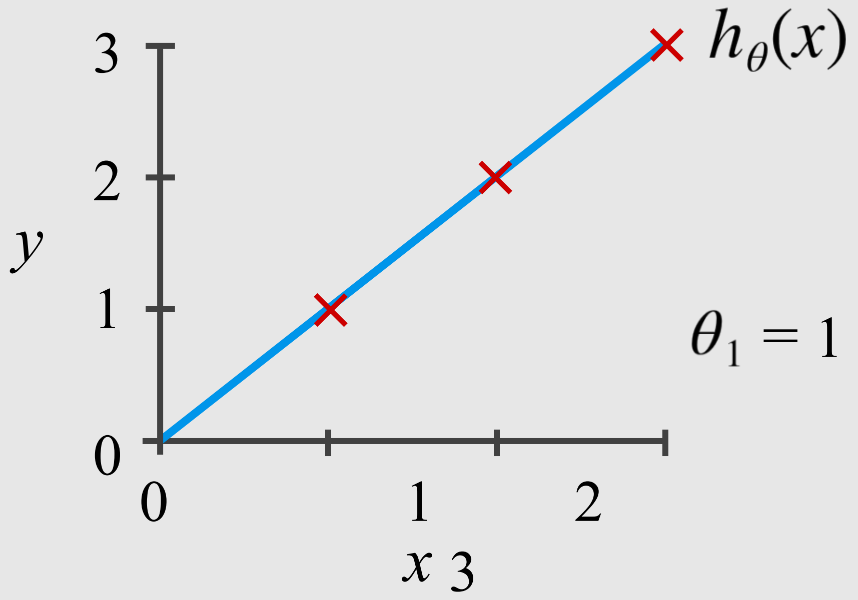


$J(\theta_1)$

(function of the parameters θ_1)

$h_\theta(x)$

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



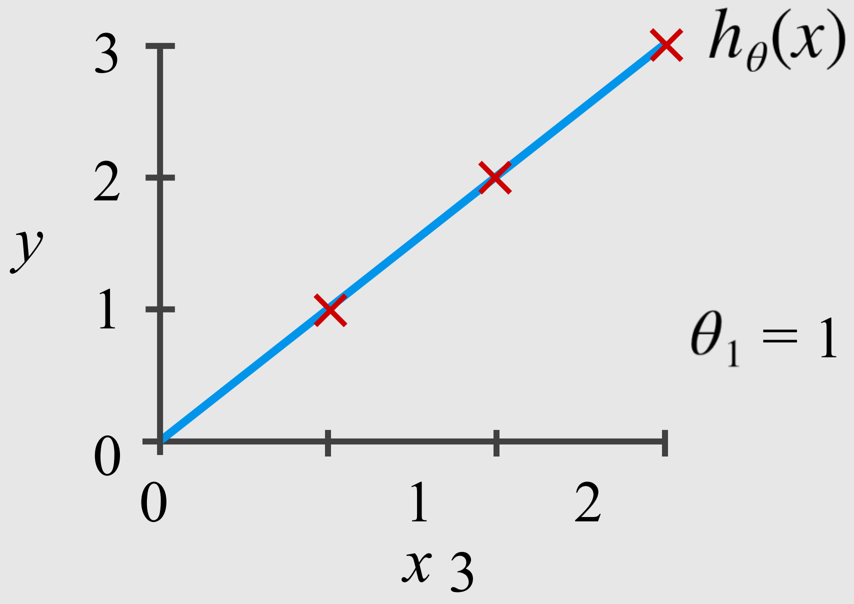
$J(\theta_1) = J(1) = ?$

$J(\theta_1)$

(function of the parameters θ_1)

$h_\theta(x)$

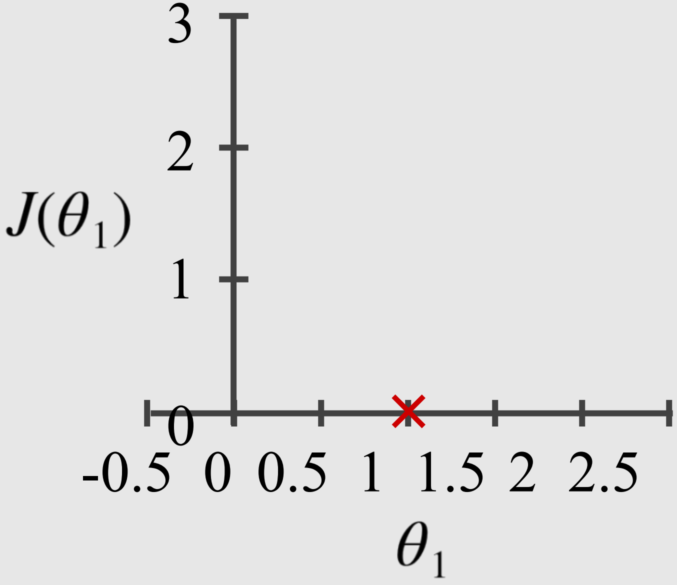
(for fixed θ_1 , this is a function of x)



$J(\theta_1) = J(1) = 0$

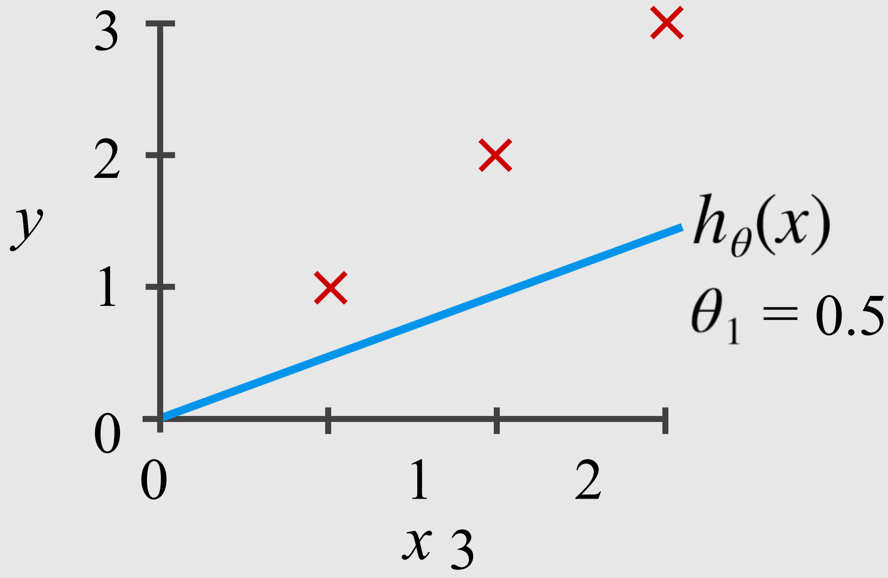
$J(\theta_1)$

(function of the parameters θ_1)



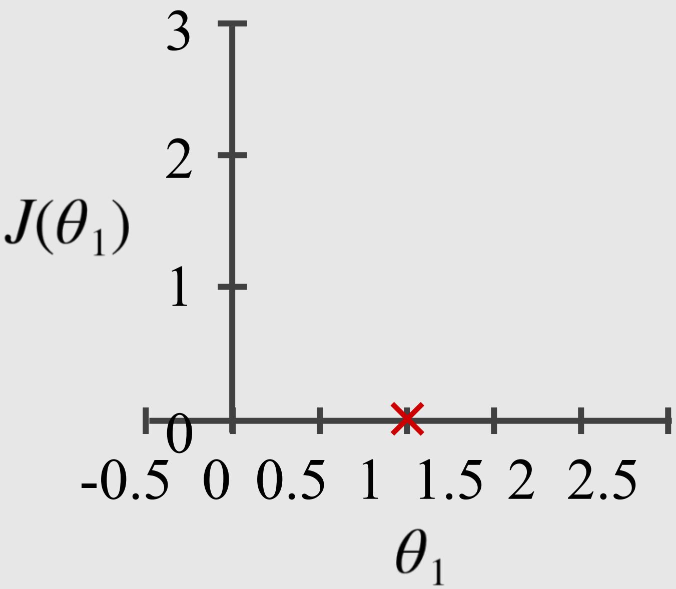
$h_\theta(x)$

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



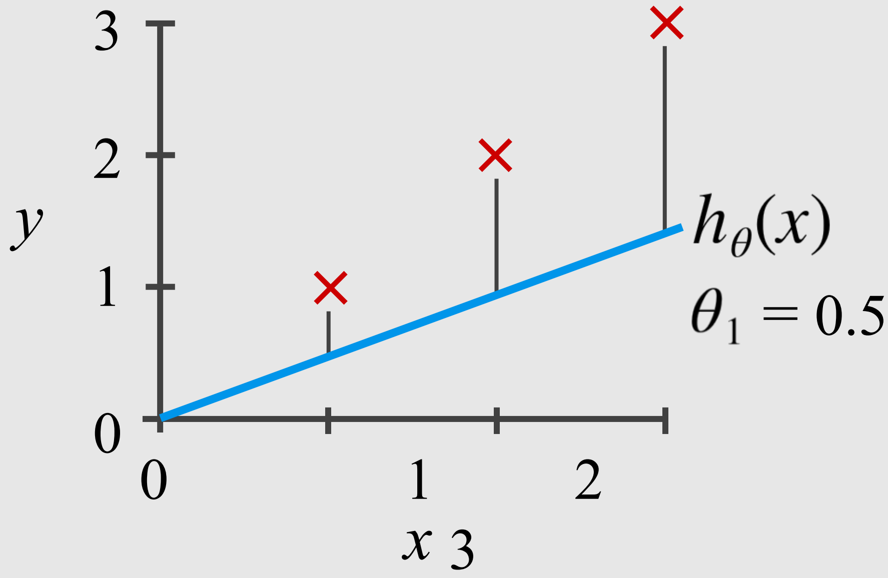
$J(\theta_1)$

(function of the parameters θ_1)



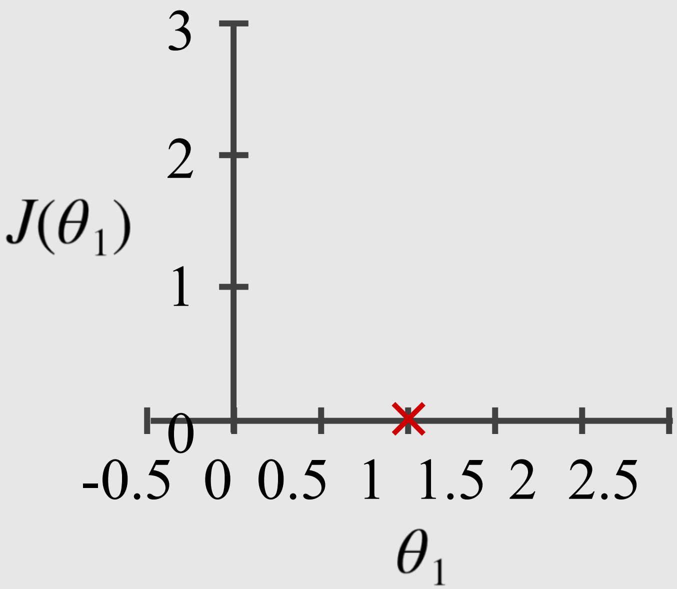
$h_\theta(x)$

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



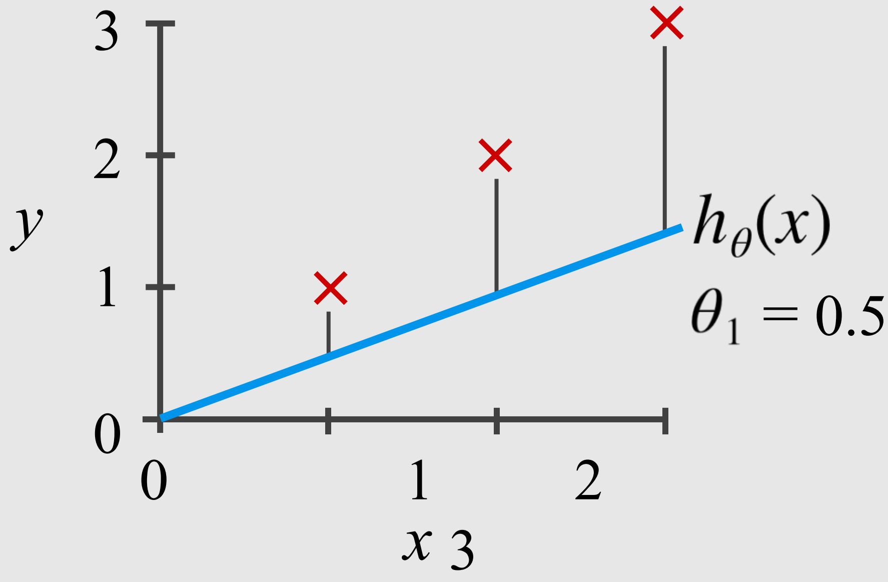
$J(\theta_1)$

(function of the parameters θ_1)



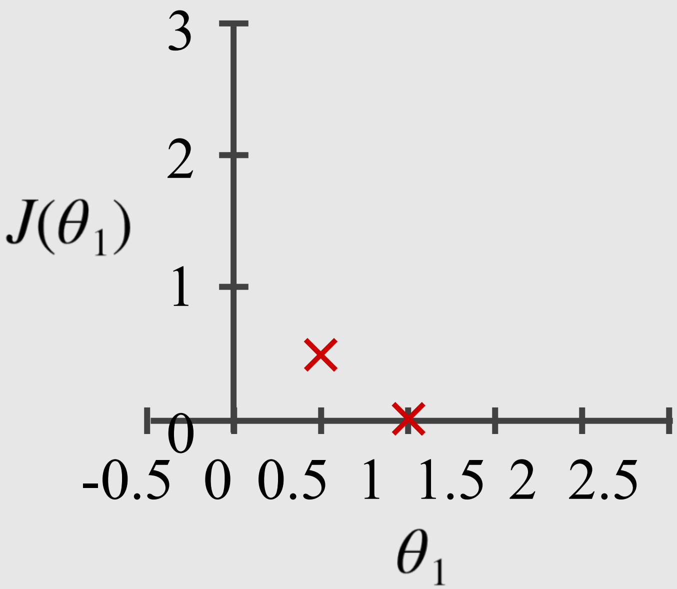
$h_\theta(x)$

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



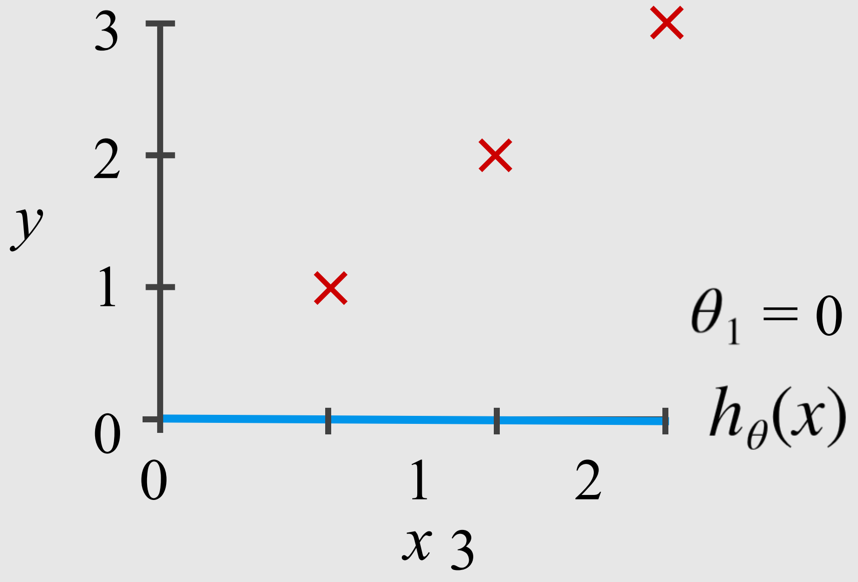
$J(\theta_1)$

(function of the parameters θ_1)



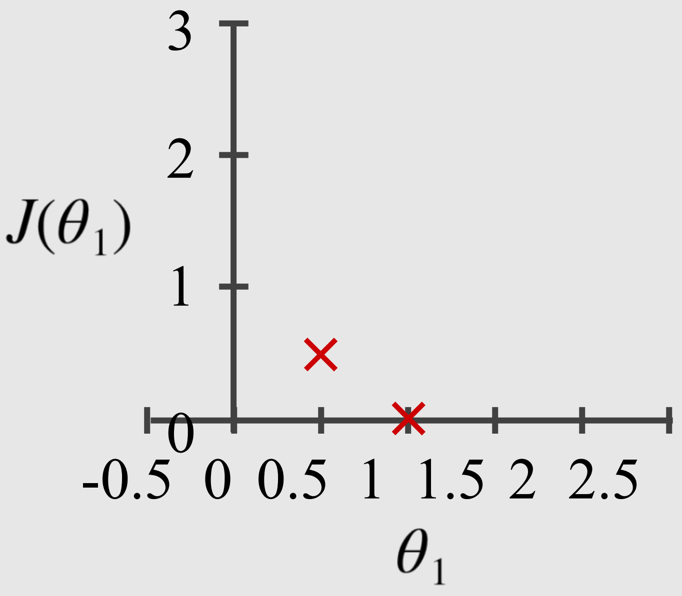
$h_\theta(x)$

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



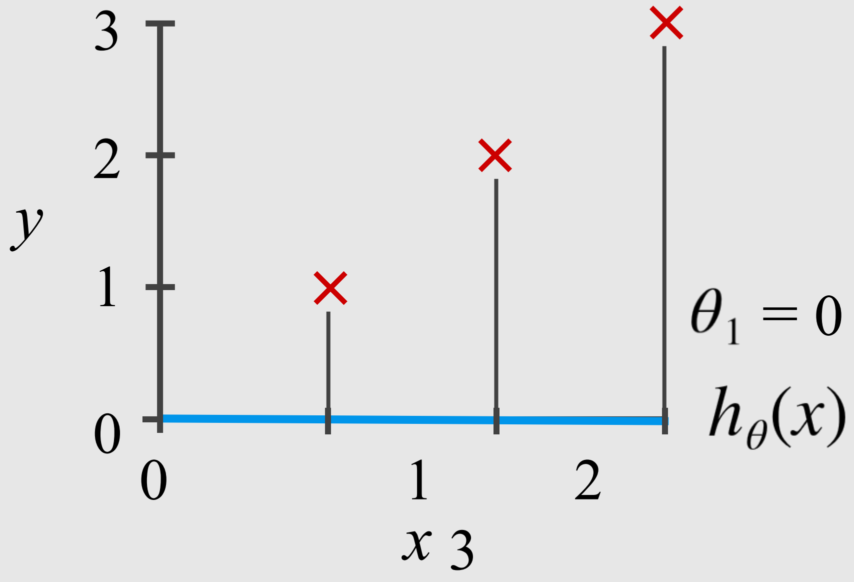
$J(\theta_1)$

(function of the parameters θ_1)



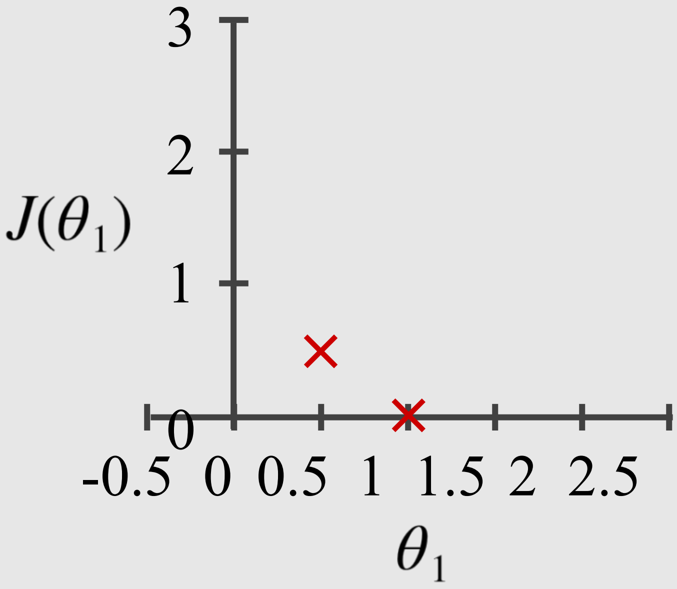
$h_\theta(x)$

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



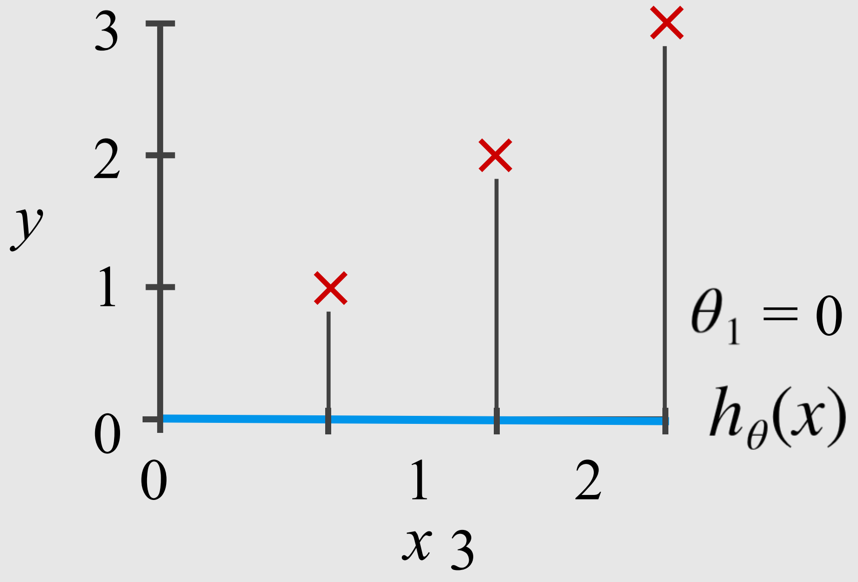
$J(\theta_1)$

(function of the parameters θ_1)



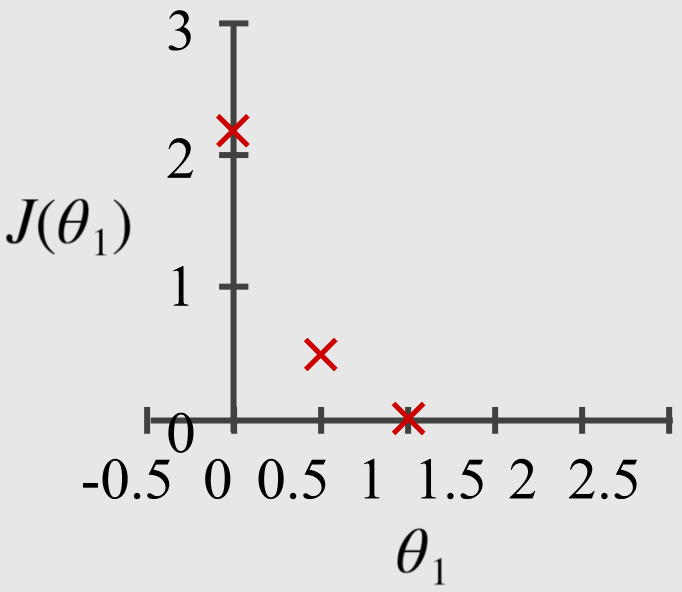
$h_\theta(x)$

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



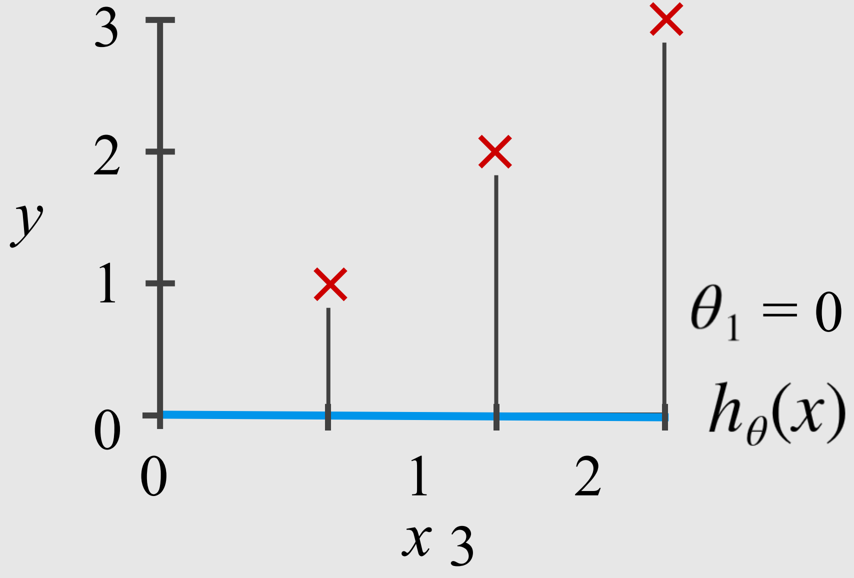
$J(\theta_1)$

(function of the parameters θ_1)



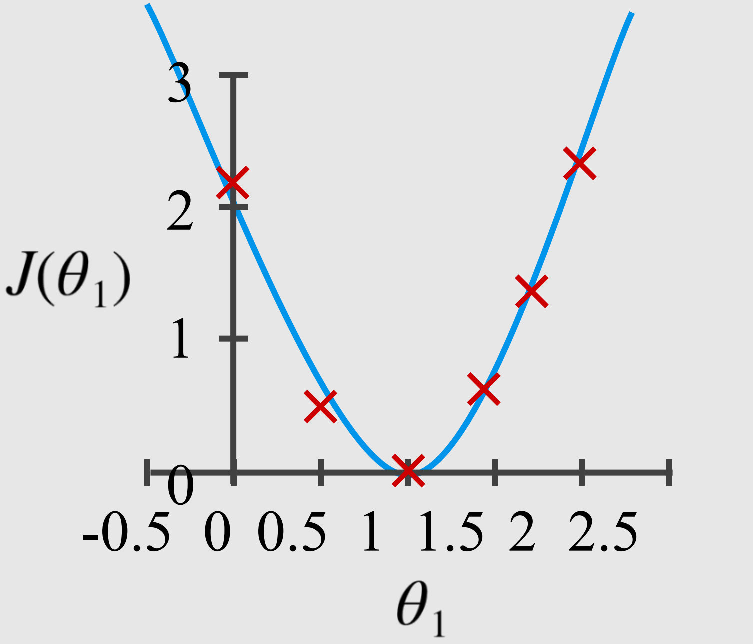
$h_\theta(x)$

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



$J(\theta_1)$

(function of the parameters θ_1)



Cost Function

Intuition II

$h_\theta(x)$

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

$J(\theta_0, \theta_1)$

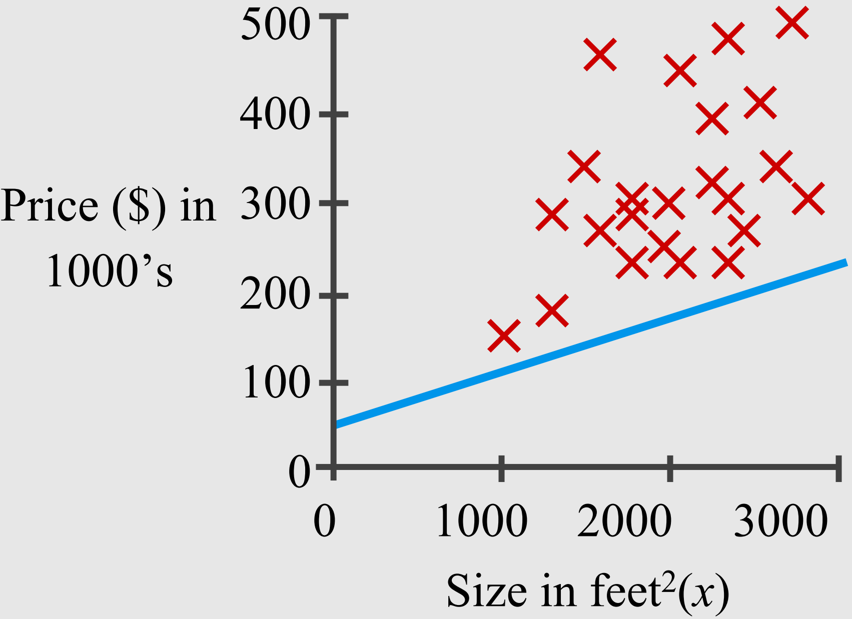
(function of the parameters θ_0, θ_1)

$h_\theta(x)$

(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)



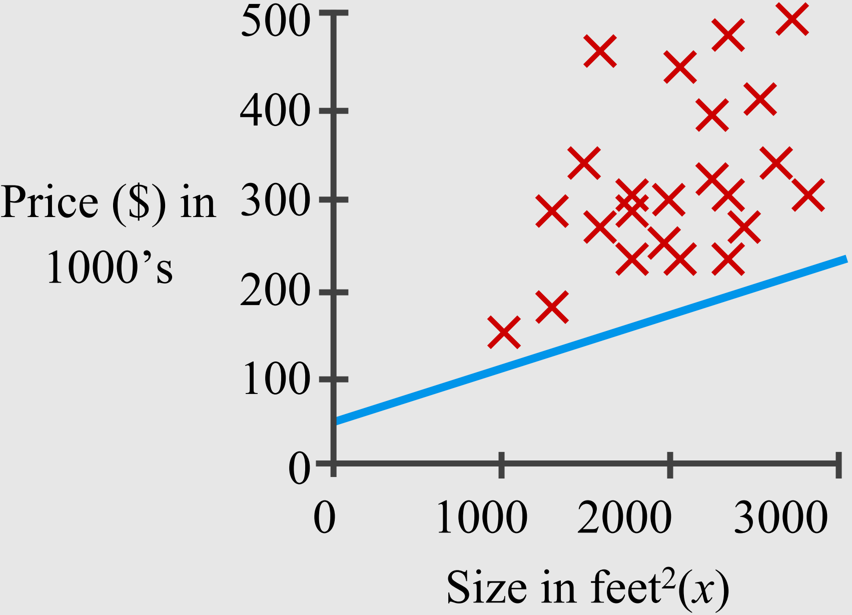
$$h_\theta(x) = 50 + 0.06x$$

$\theta_0 = 50$

$\theta_1 = 0.06$

$h_\theta(x)$

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



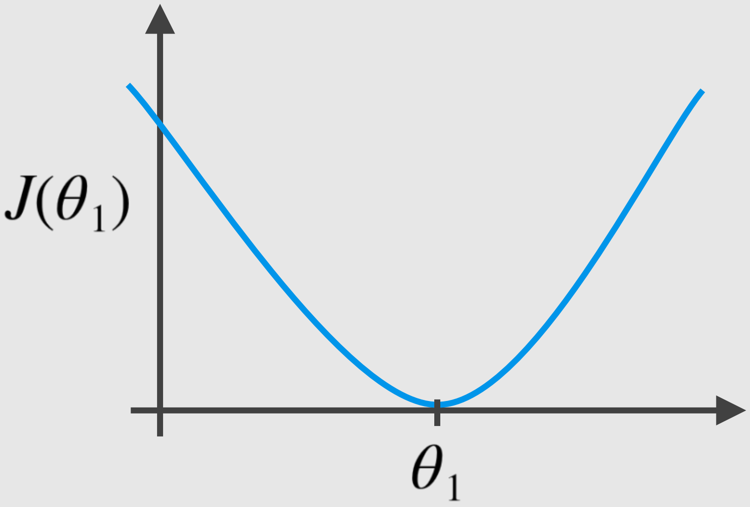
$h_\theta(x) = 50 + 0.06x$

$\theta_0 = 50$

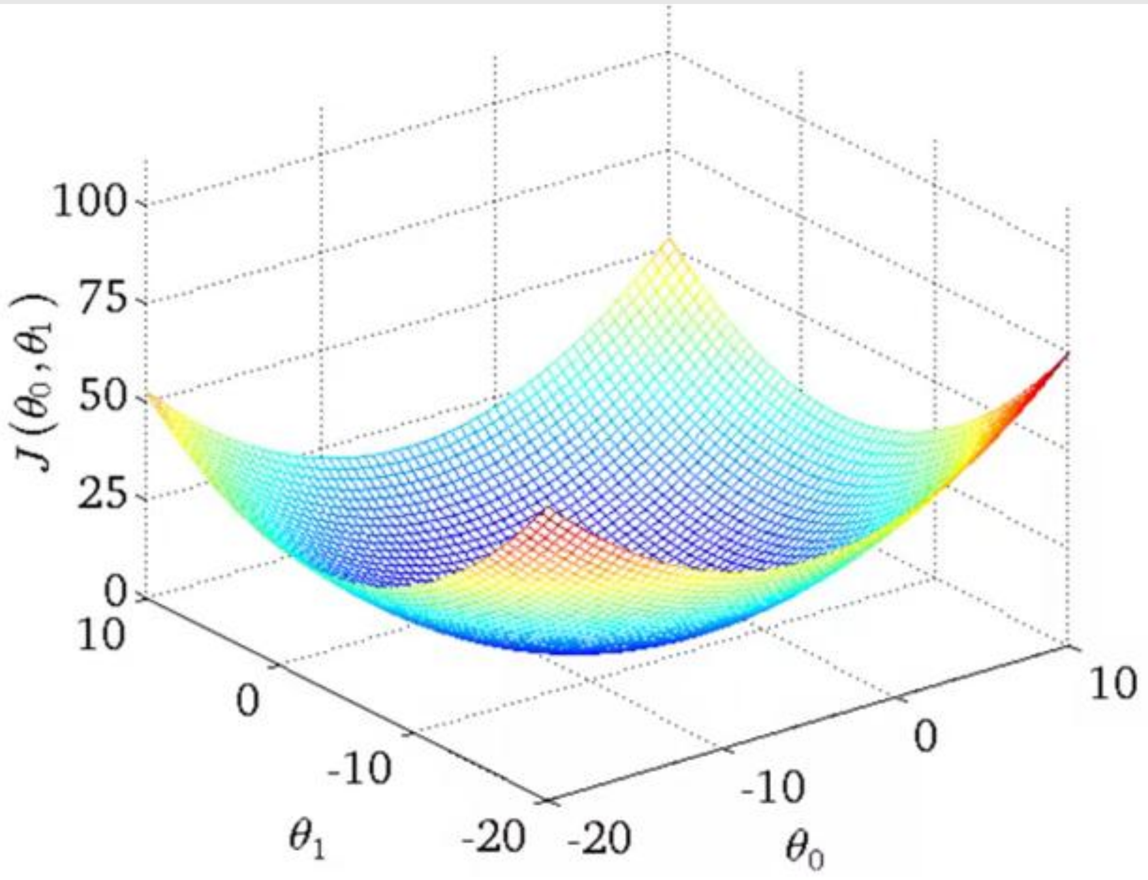
$\theta_1 = 0.06$

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

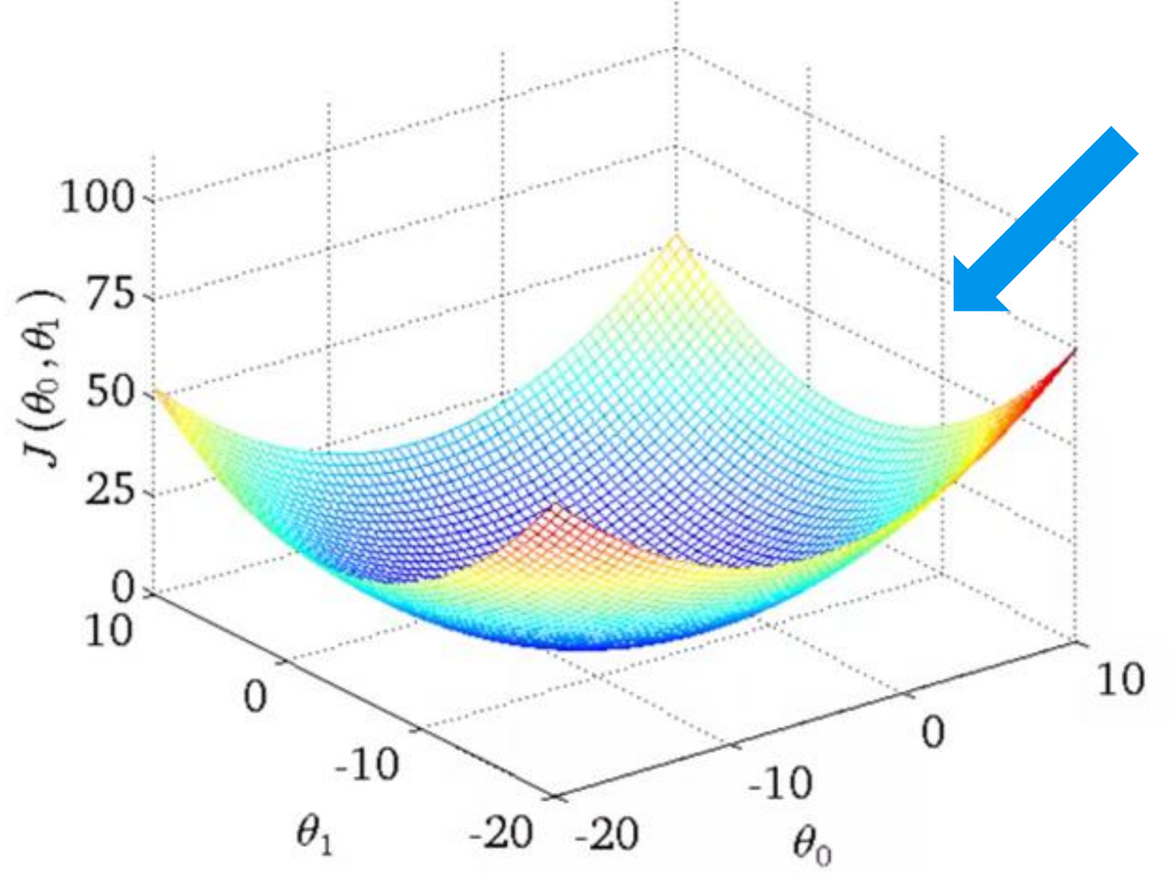


θ_0 and θ_1?



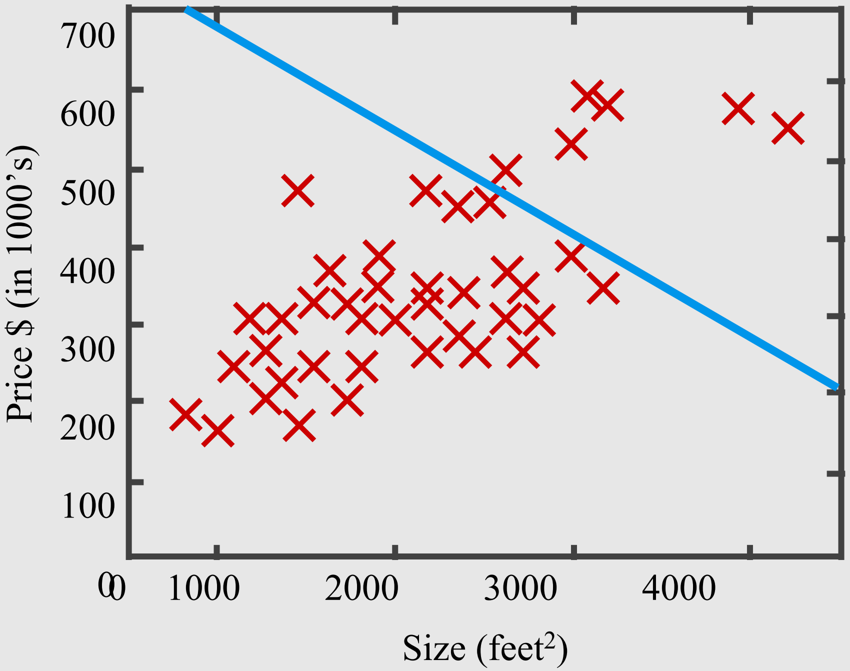
100
75
50
25
0
$J(\theta_0, \theta_1)$
10
0
-10
-20
θ_1
-20
-10
0
10
θ_0

Convex Function



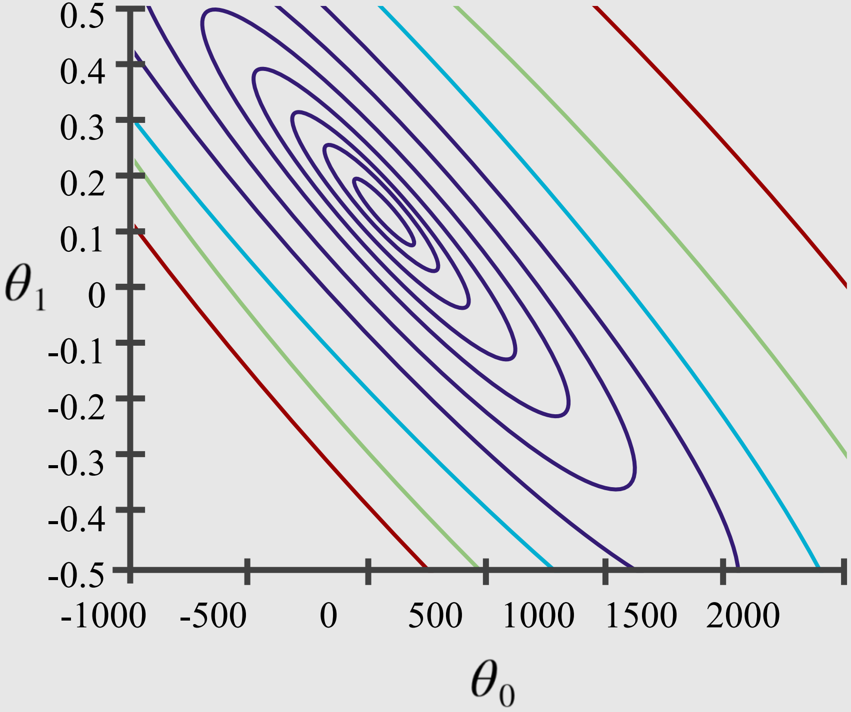
$h_\theta(x)$

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



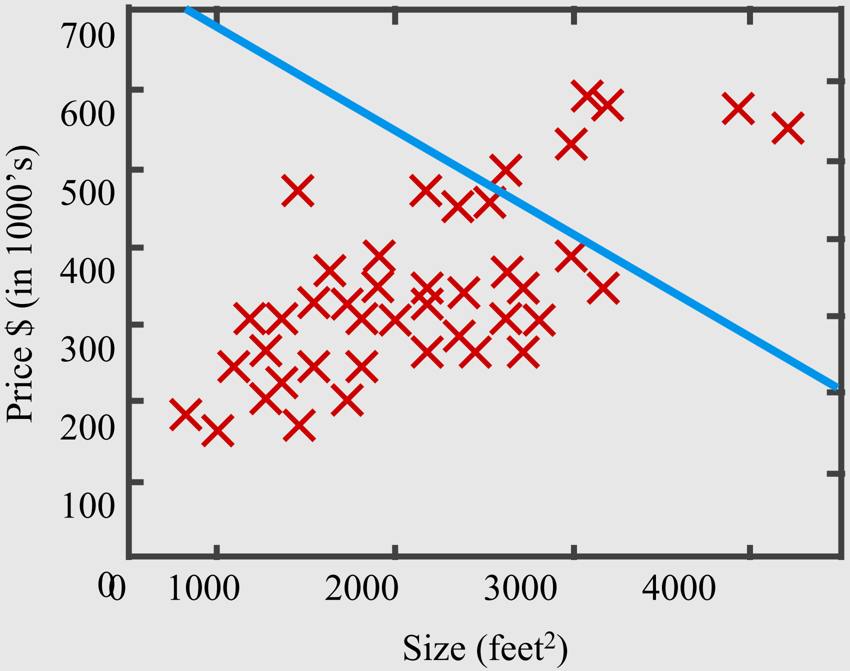
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



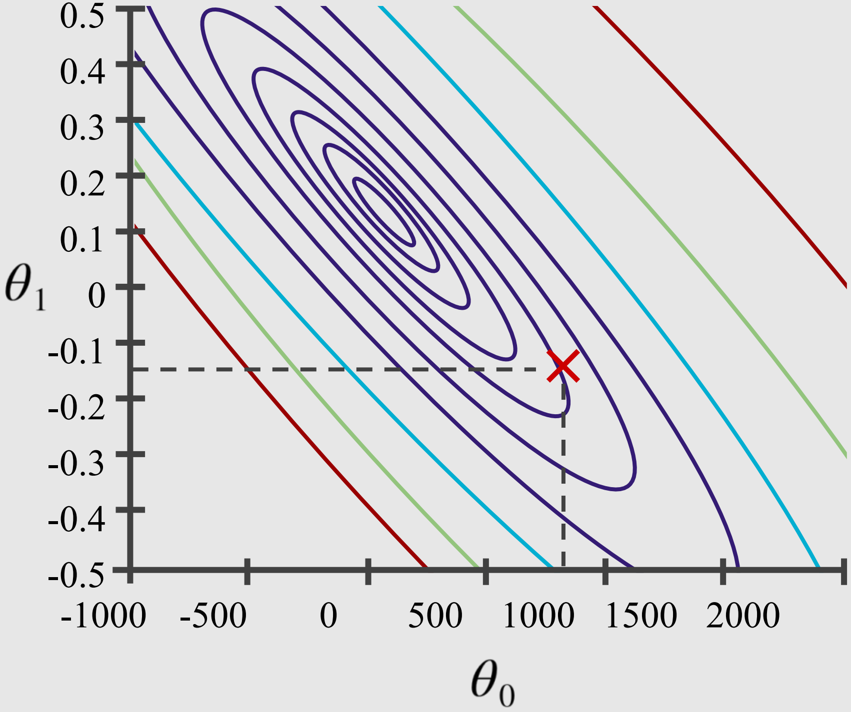
$h_\theta(x)$

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



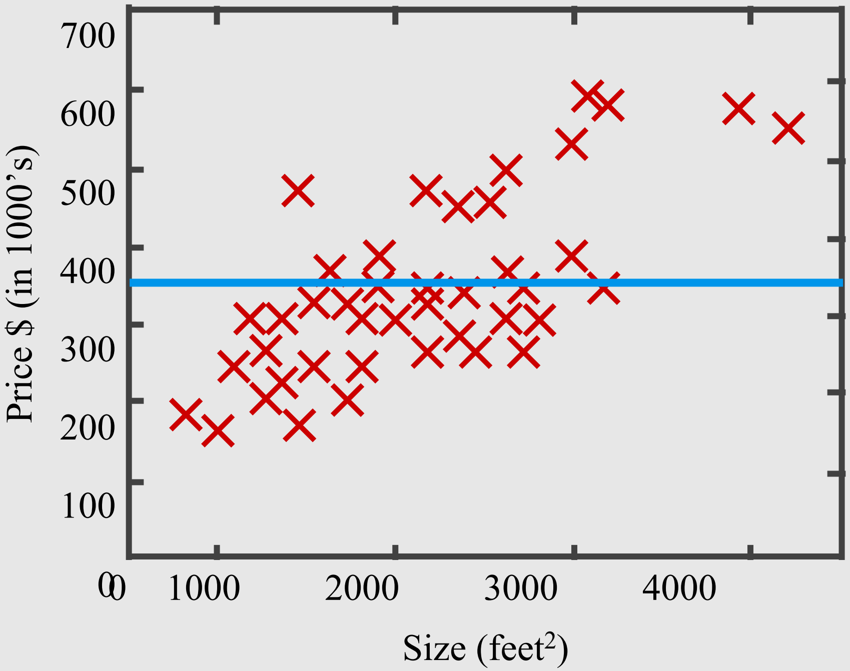
$J(\theta_0,\theta_1)$

(function of the parameters θ_0, θ_1)



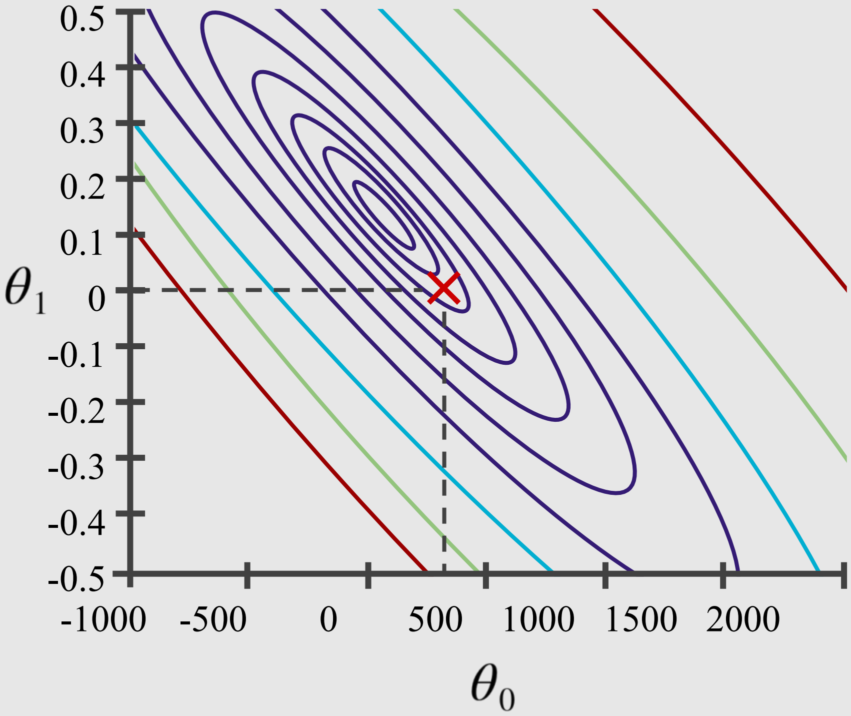
$h_\theta(x)$

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



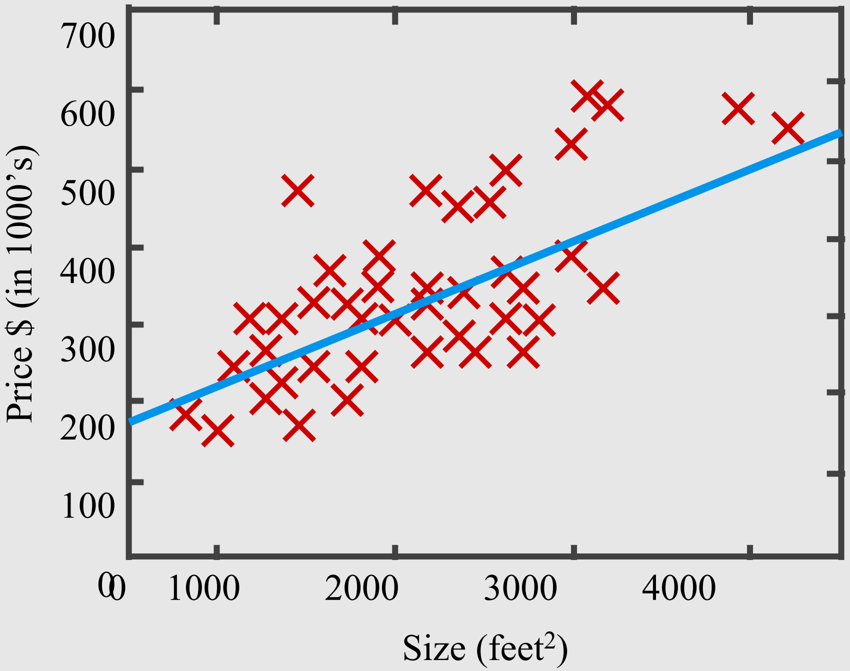
$J(\theta_0,\theta_1)$

(function of the parameters θ_0, θ_1)



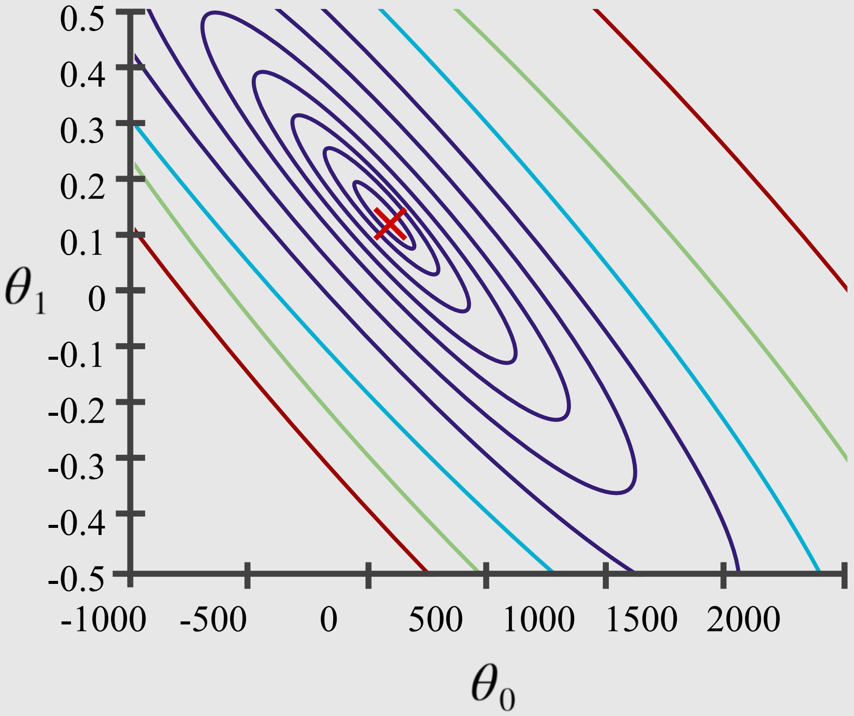
$h_\theta(x)$

(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)



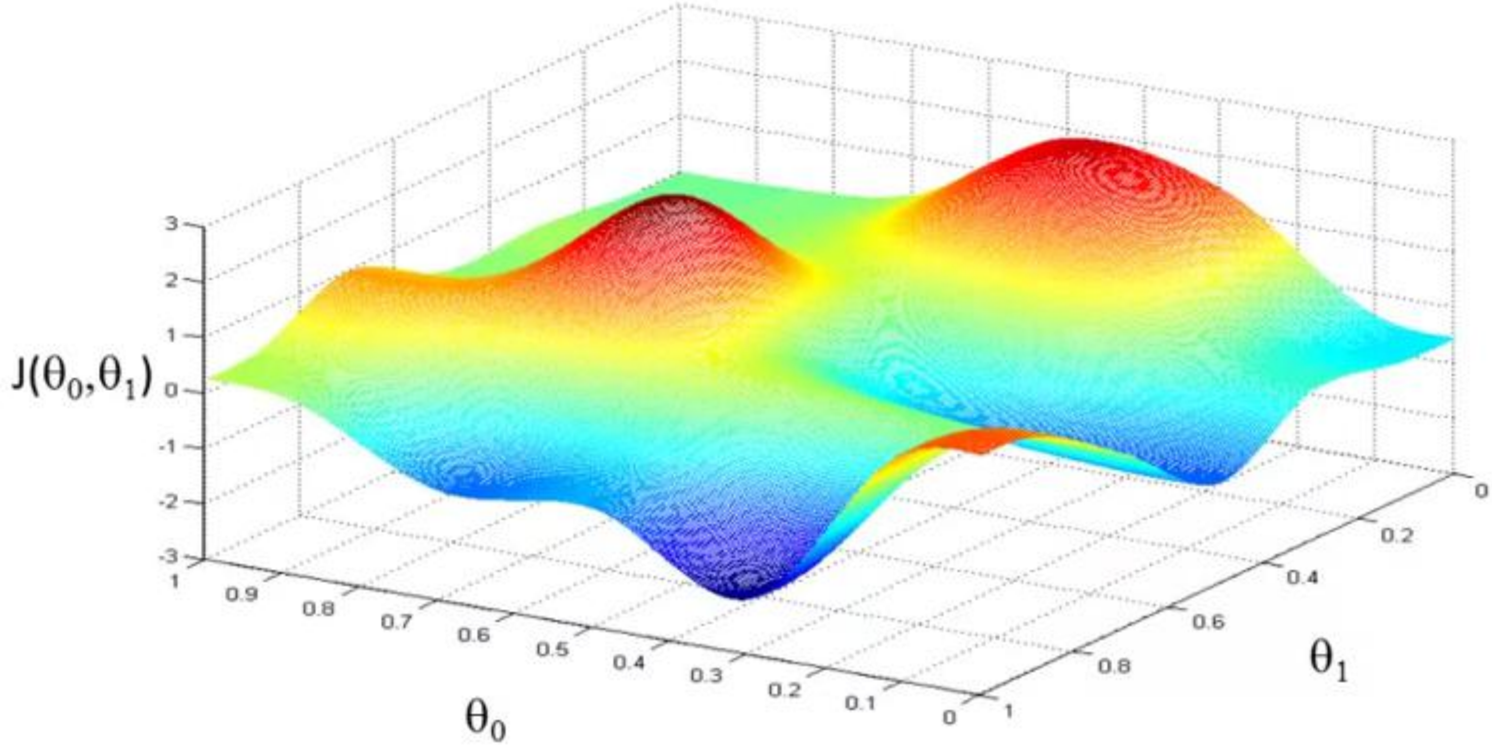
Gradient Descent

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

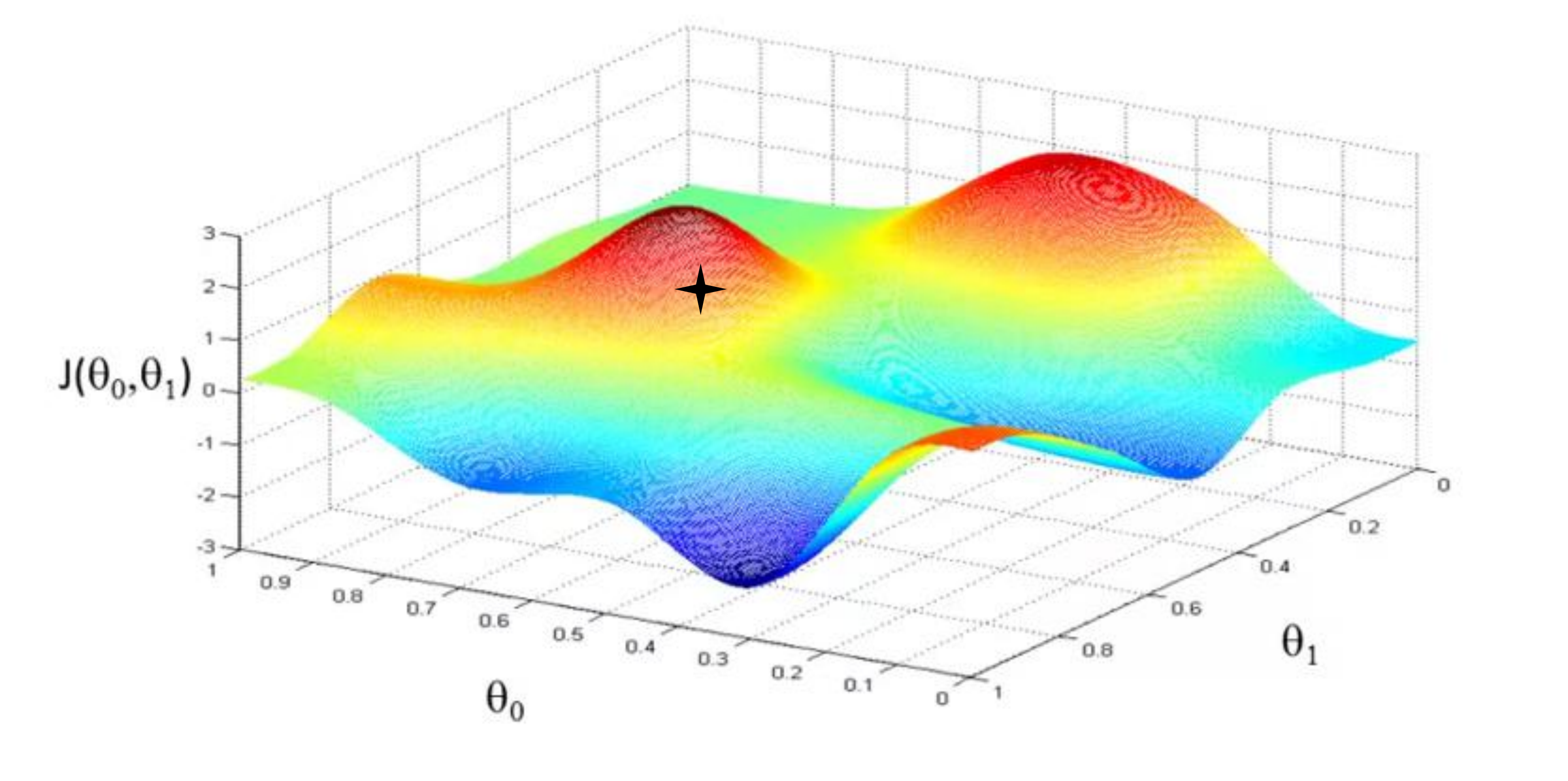
Want $\underset{\theta_0,\theta_1}{\text{minimize}}\, J(\theta_0,\theta_1)$

Outline:

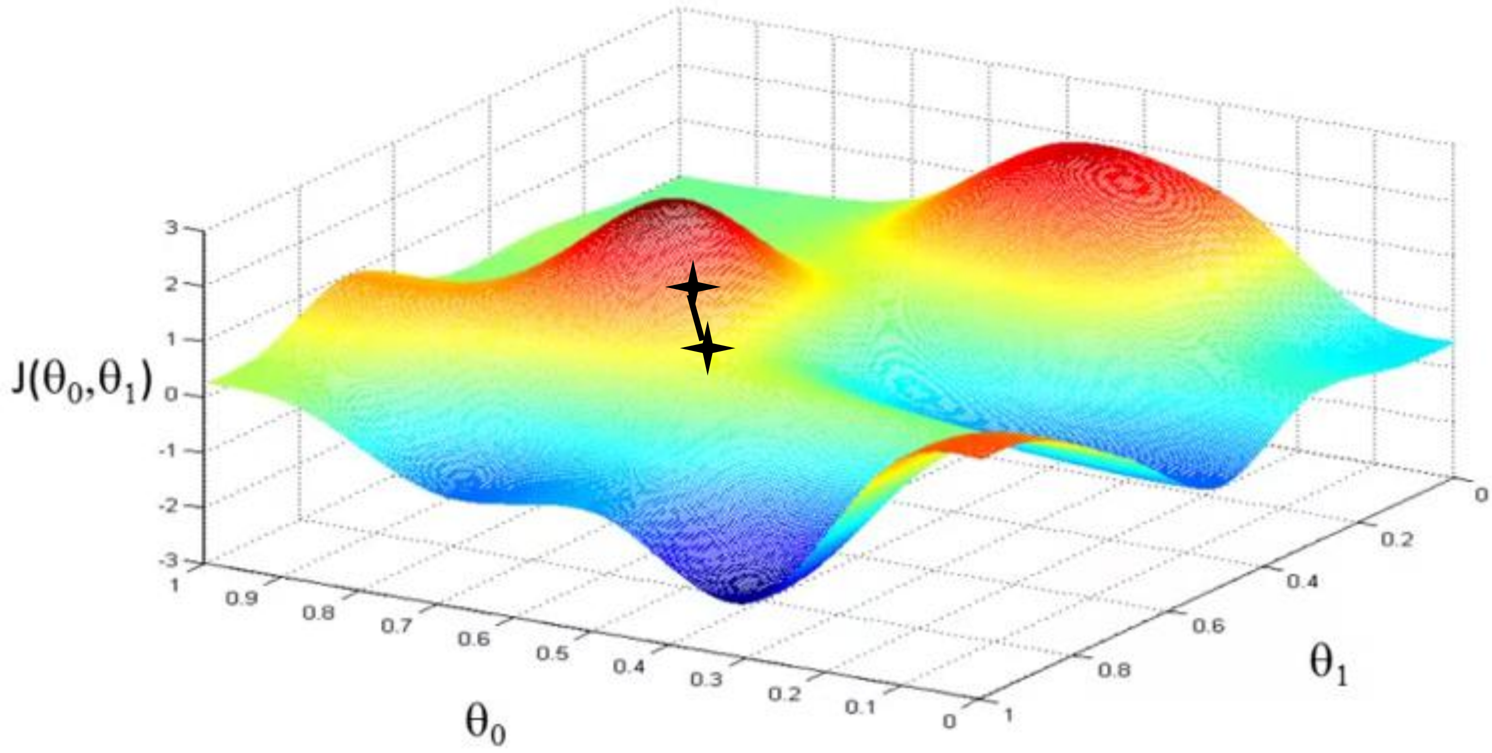
- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0,\theta_1)$ until we hopefully end up at a minimum



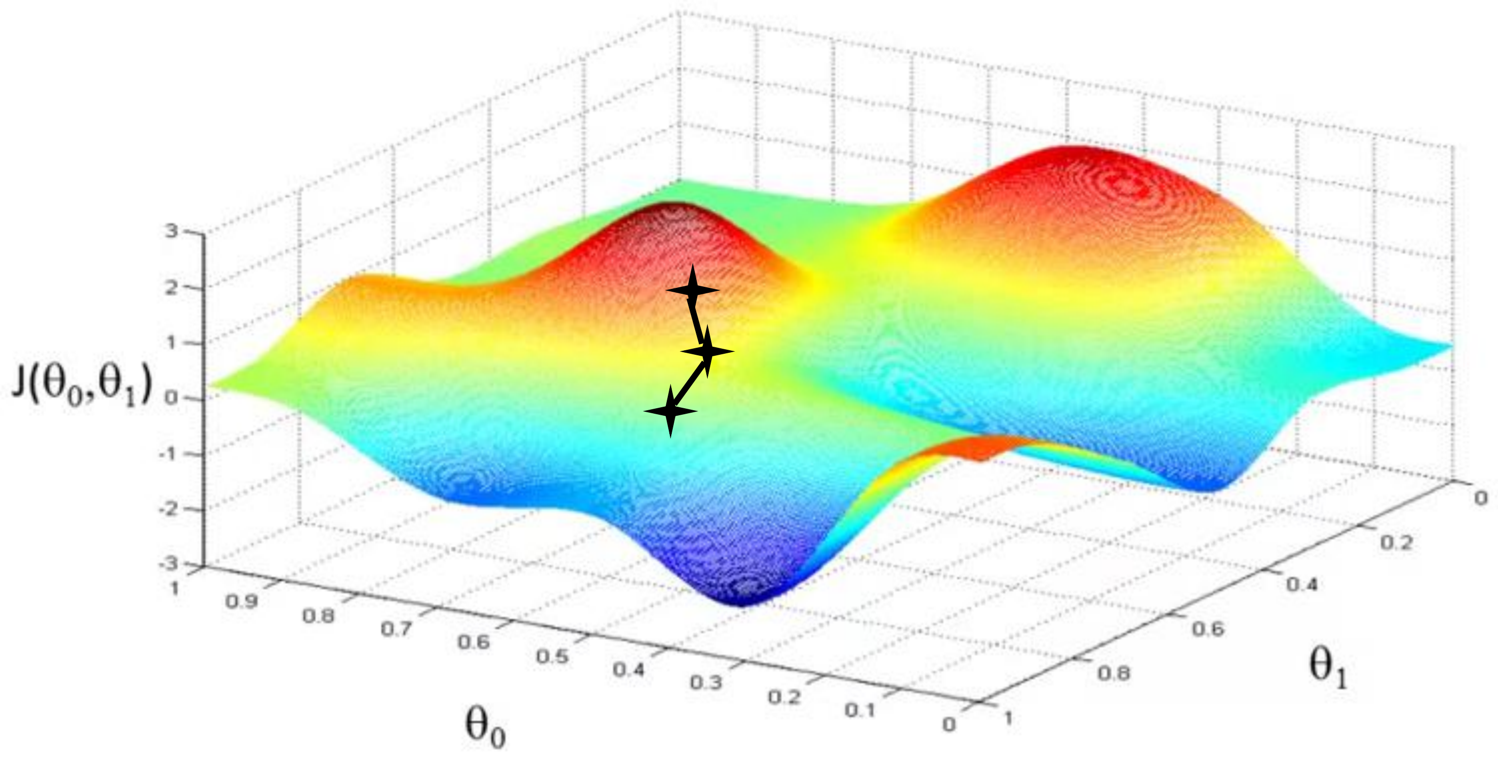
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
θ_0
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_1
0
0.2
0.4
0.6
0.8
1



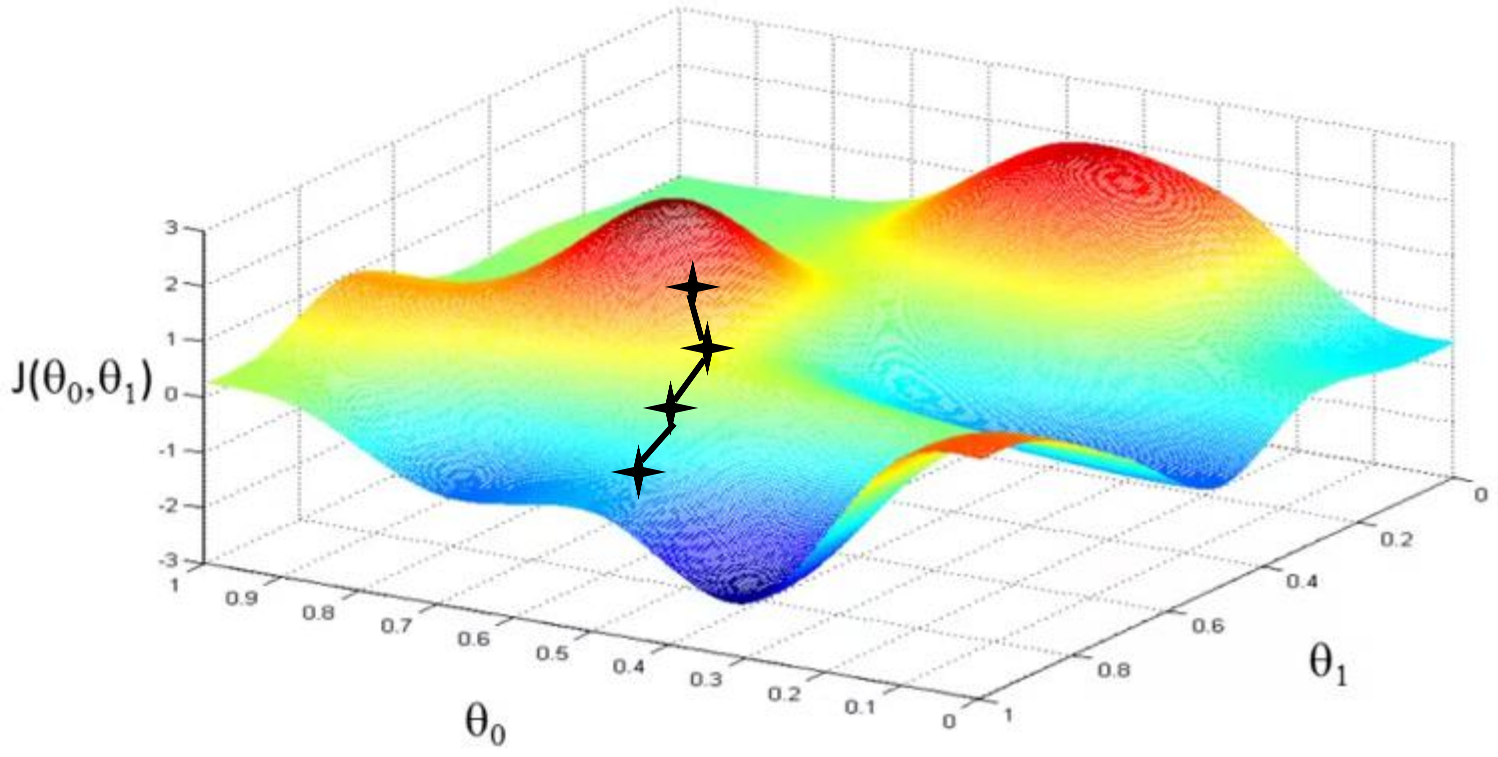
J(θ0,θ1)
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ0
0
0.2
0.4
0.6
0.8
1
θ1



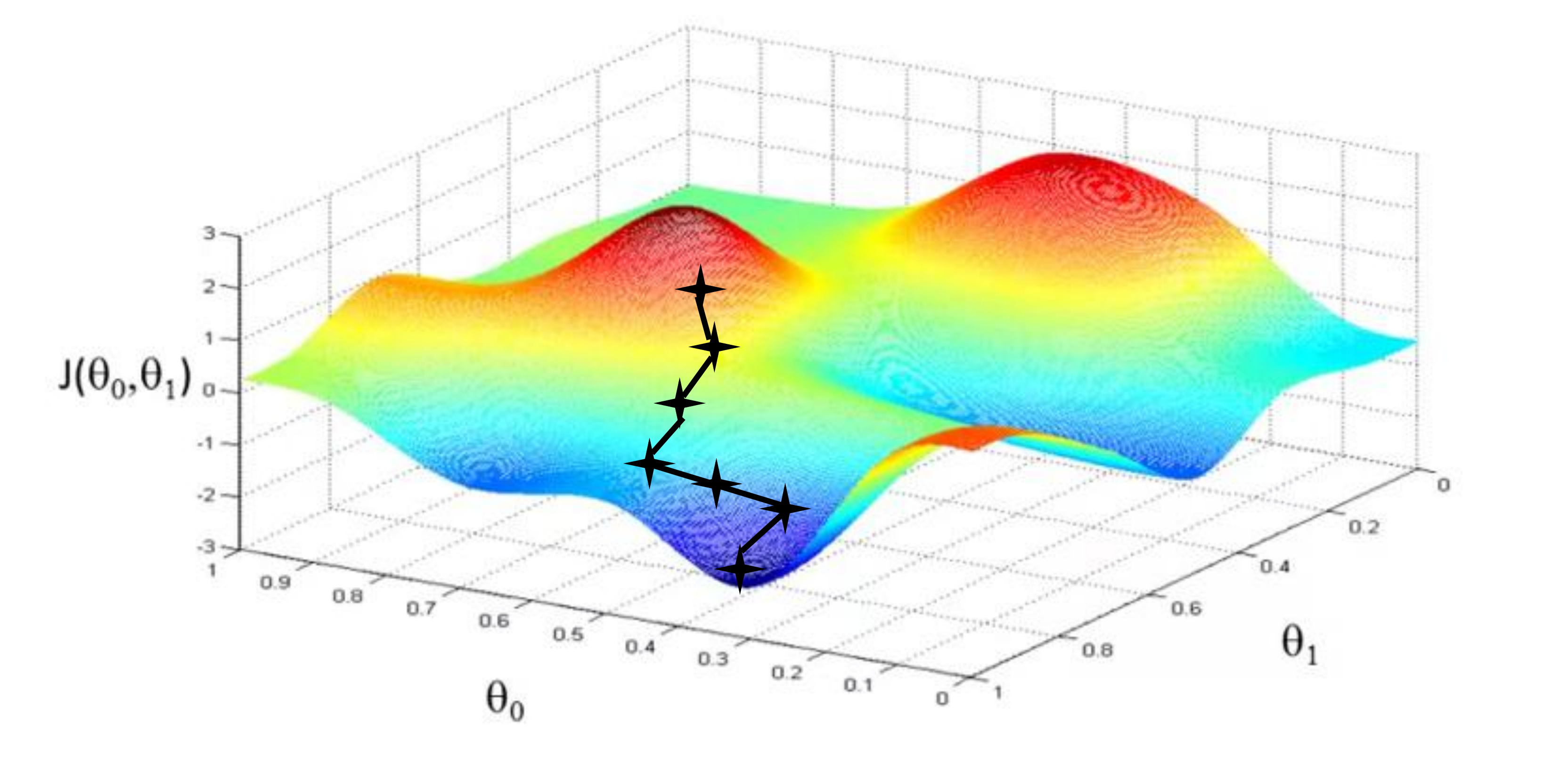
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
θ_0
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_1
0
0.2
0.4
0.6
0.8
1



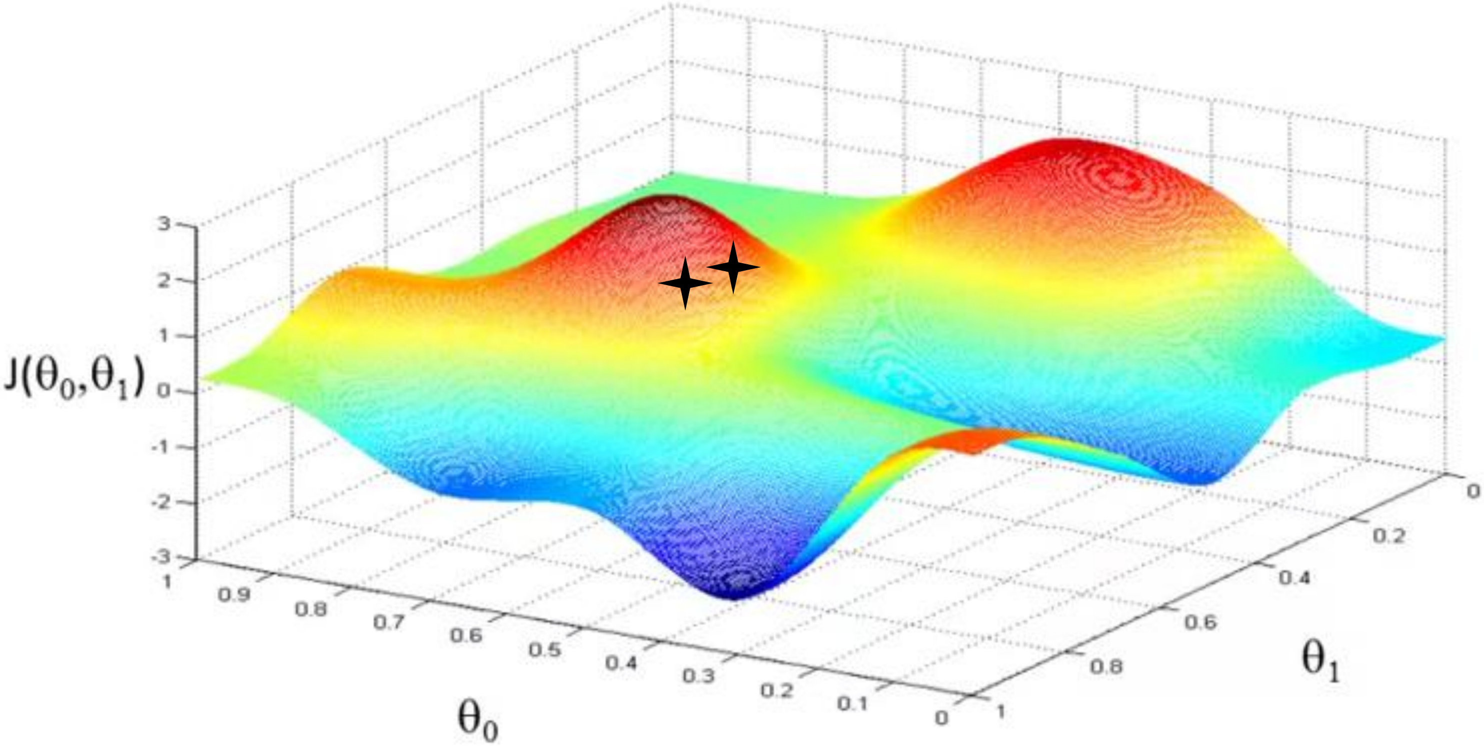
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



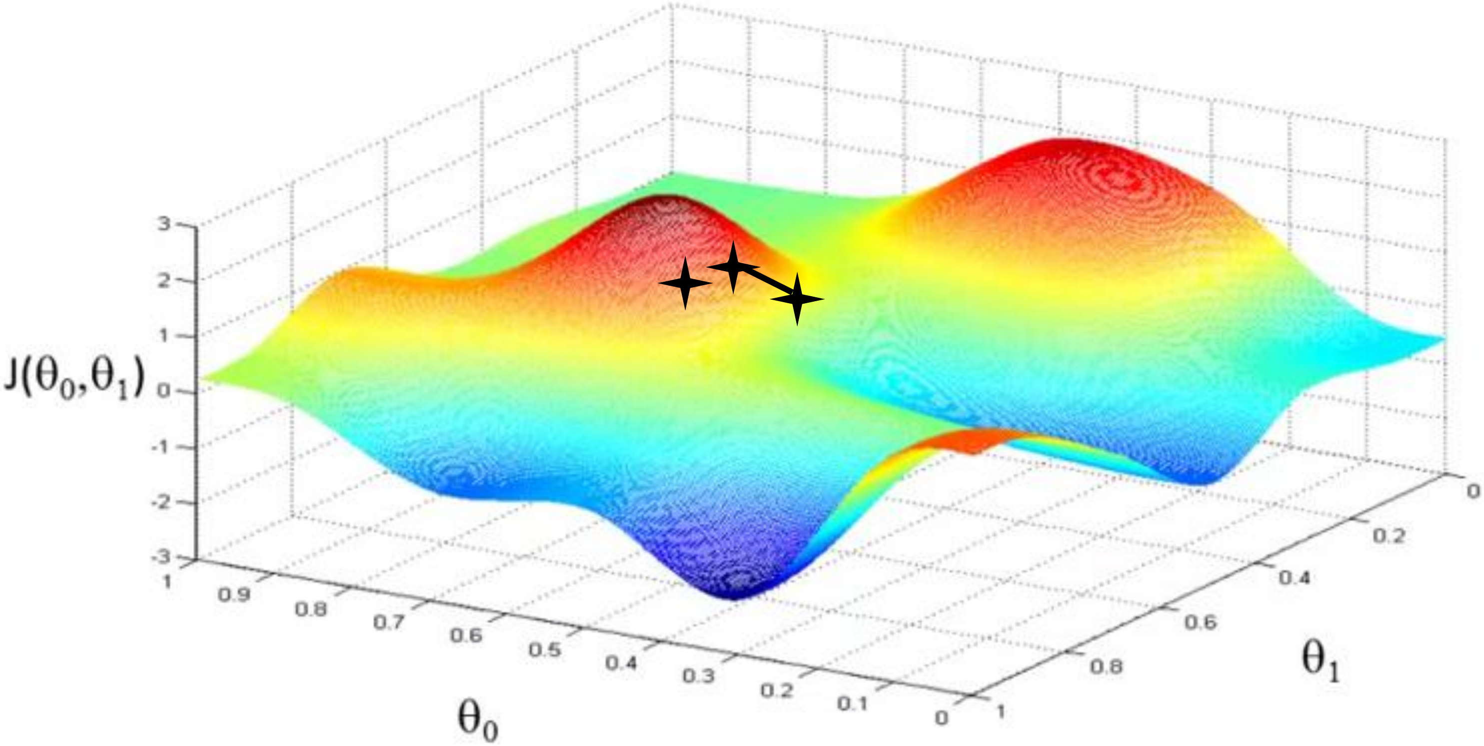
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



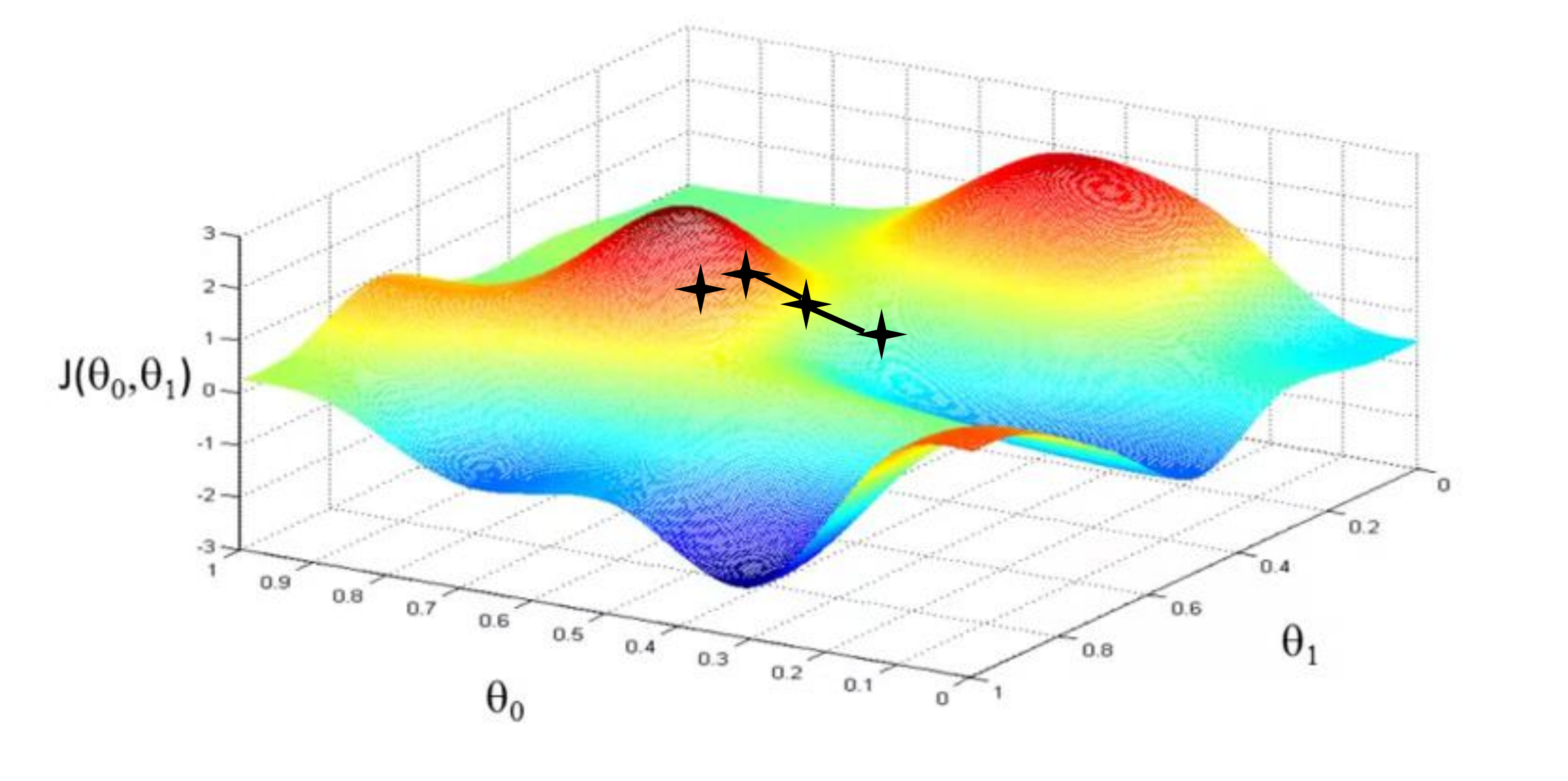
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



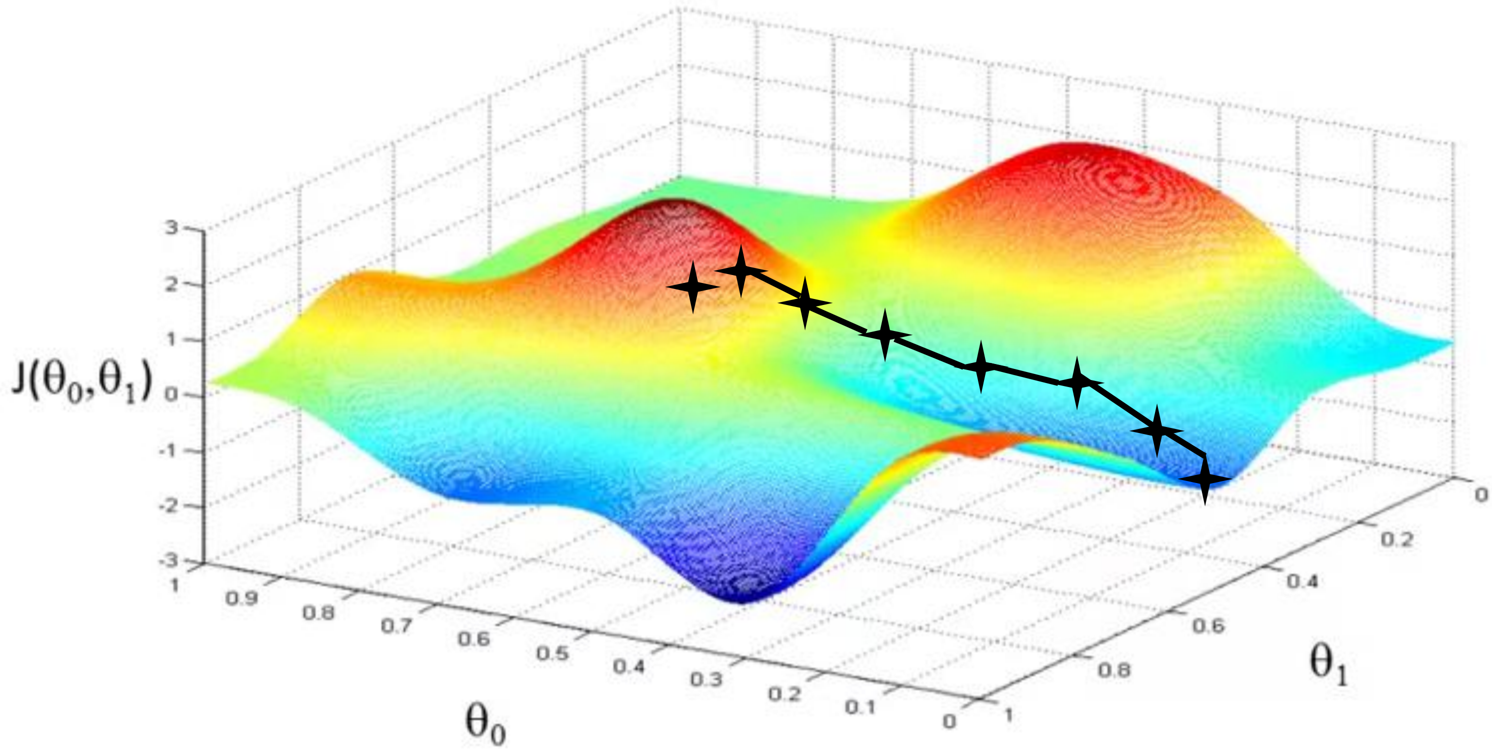
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



$J(\theta_0,\theta_1)$
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



$J(\theta_0,\theta_1)$
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



$J(\theta_0,\theta_1)$
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 algorithm

repeat until convergence {

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

}

(simultaneously update

$j = 0$ and $j = 1$)

Gradient Descent algorithm

repeat until convergence {

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

}

(simultaneously update

$j = 0$ and $j = 1$)

Learning rate

Derivative term

Gradient Descent algorithm

repeat until convergence {

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

}

Correct: Simultaneous update

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

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

$$\theta_0 := \text{temp0}$$

$$\theta_1 := \text{temp1}$$

Gradient Descent algorithm

repeat until convergence {

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

}

Correct: Simultaneous update

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

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

$$\theta_0 := \text{temp0}$$

$$\theta_1 := \text{temp1}$$

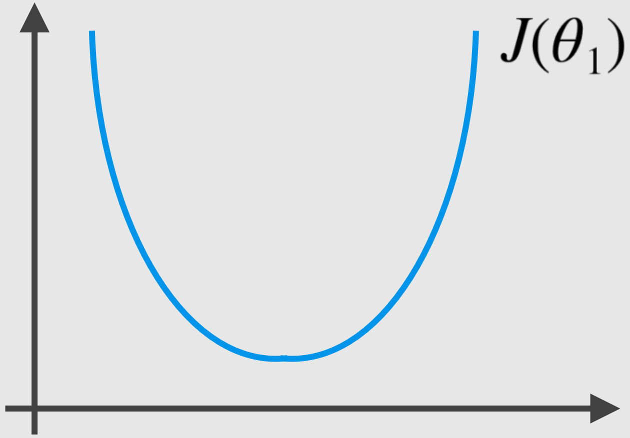
Incorrect

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

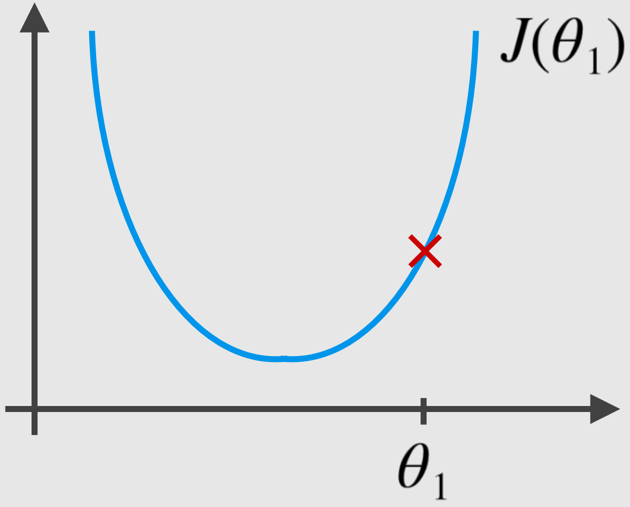
$$\theta_0 := \text{temp0}$$

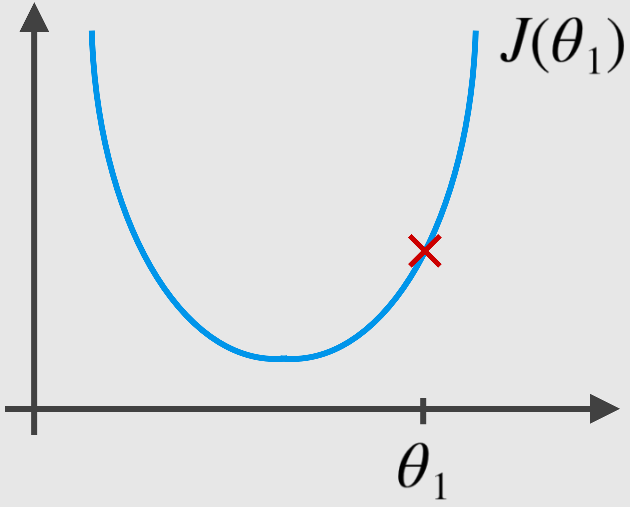
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_1 := \text{temp1}$$


$J(\theta_1)$

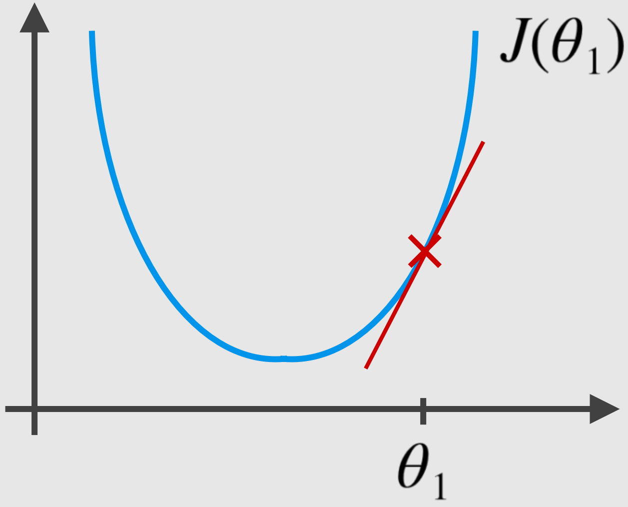
$$\theta_1 \in \mathbb{R}$$





$$\theta_1 \in \mathbb{R}$$

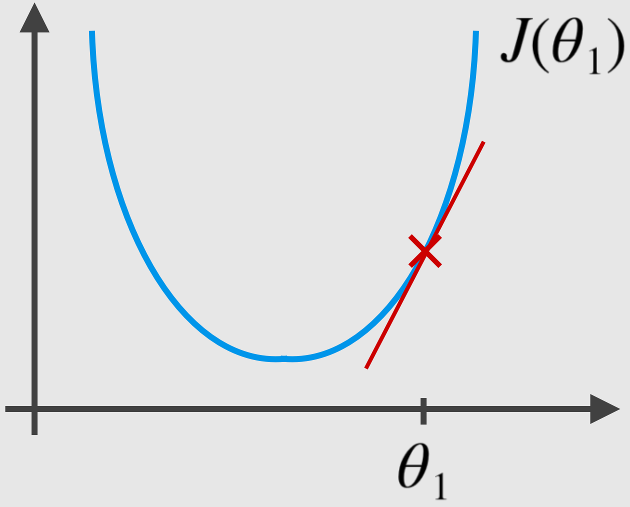
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$



$$\theta_1 \in \mathbb{R}$$

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

$\theta_1 \in \mathbb{R}$

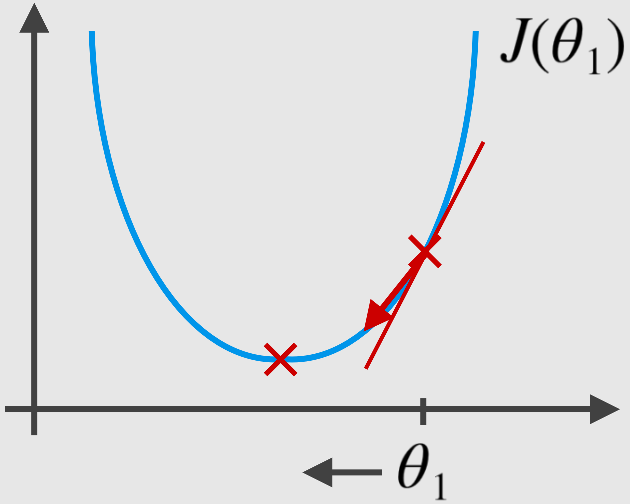


$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)}$$

≥ 0

$$\theta_1 := \theta_1 - \alpha \cdot \text{(positive number)}$$

$\theta_1 \in \mathbb{R}$

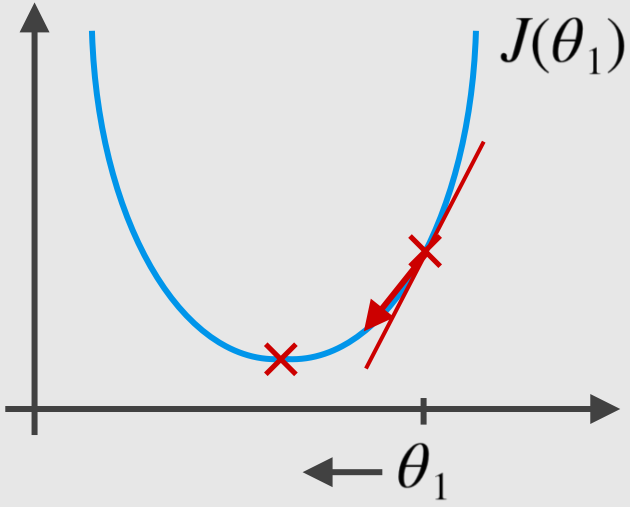


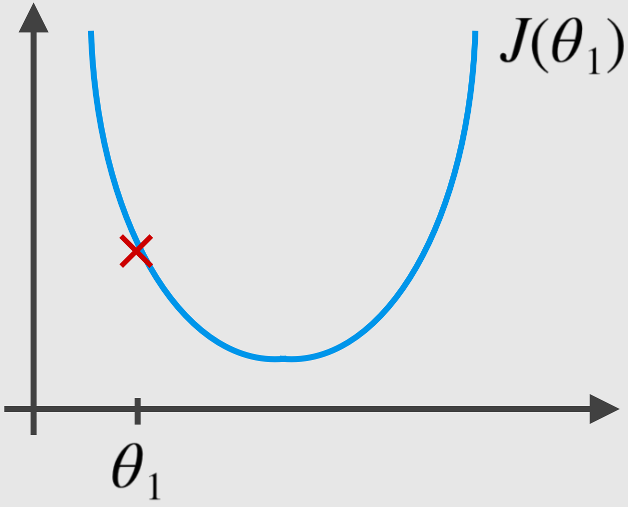
$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)}$$

≥ 0

$$\theta_1 := \theta_1 - \alpha \cdot \text{(positive number)}$$

$$\theta_1 \in \mathbb{R}$$

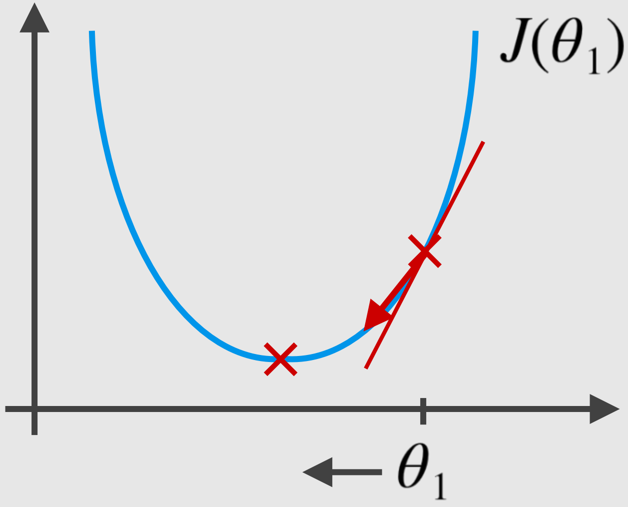




$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)} \quad \geq 0$$

$$\theta_1 := \theta_1 - \alpha \cdot \text{(positive number)}$$

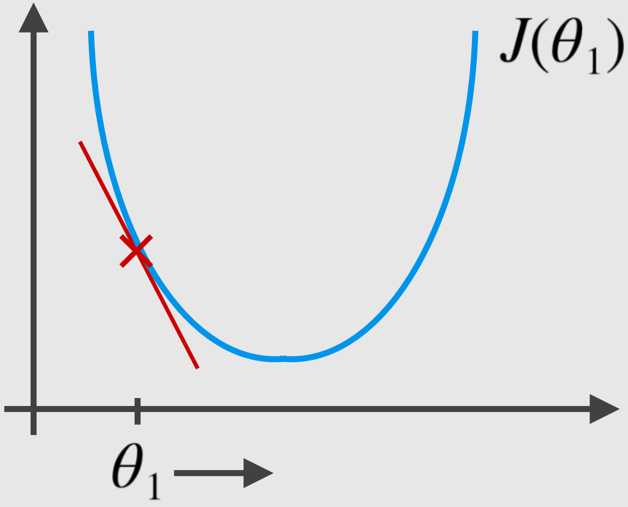
$\theta_1 \in \mathbb{R}$



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

≥ 0

$$\theta_1 := \theta_1 - \alpha \cdot \text{(positive number)}$$

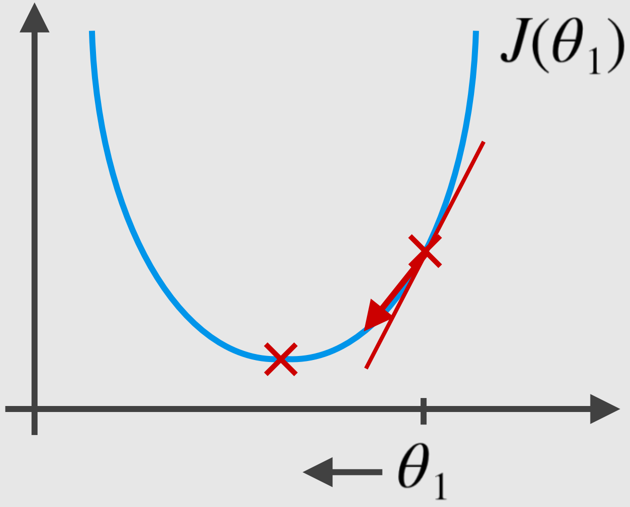


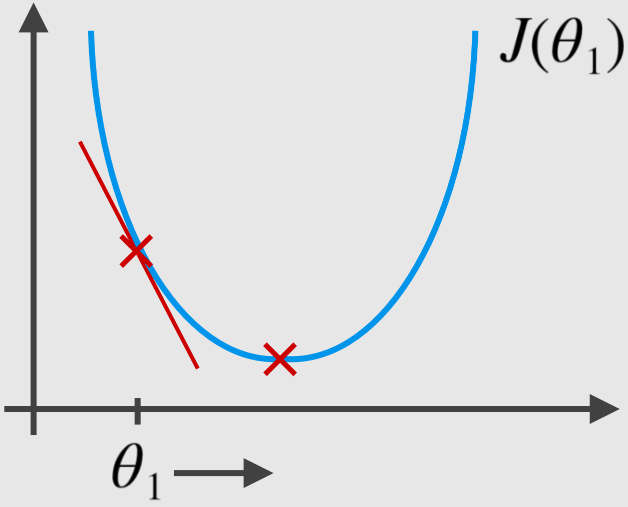
$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)}$$

≤ 0

$$\theta_1 := \theta_1 - \alpha \cdot \text{(negative number)}$$

$\theta_1 \in \mathbb{R}$





$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)} \quad \geq 0$$

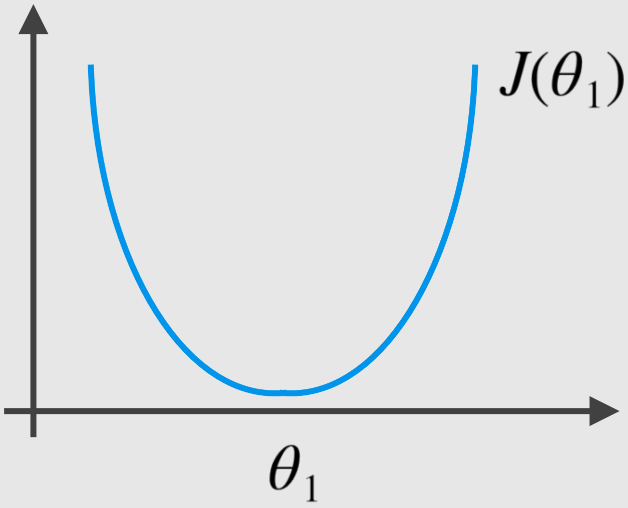
$$\theta_1 := \theta_1 - \alpha \cdot \text{(positive number)}$$

$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)} \quad \leq 0$$

$$\theta_1 := \theta_1 - \alpha \cdot \text{(negative number)}$$

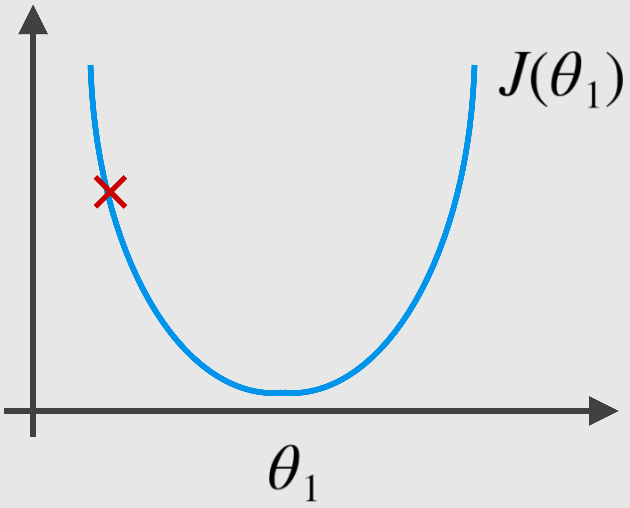
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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

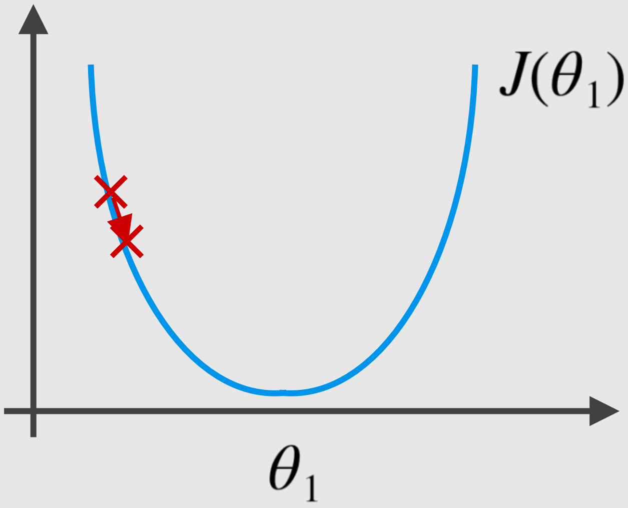


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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



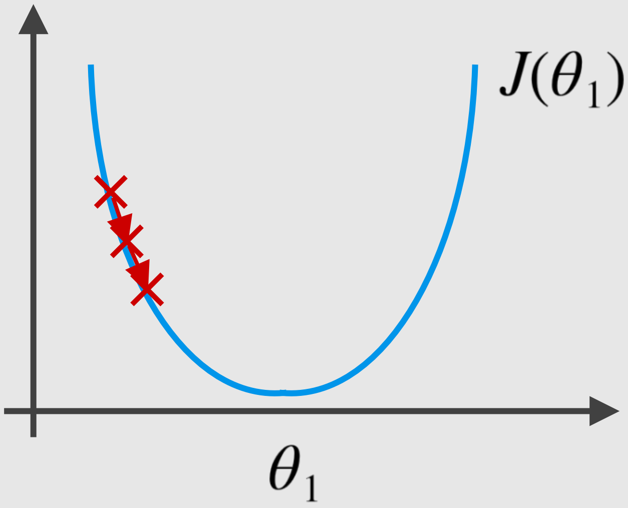
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$



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

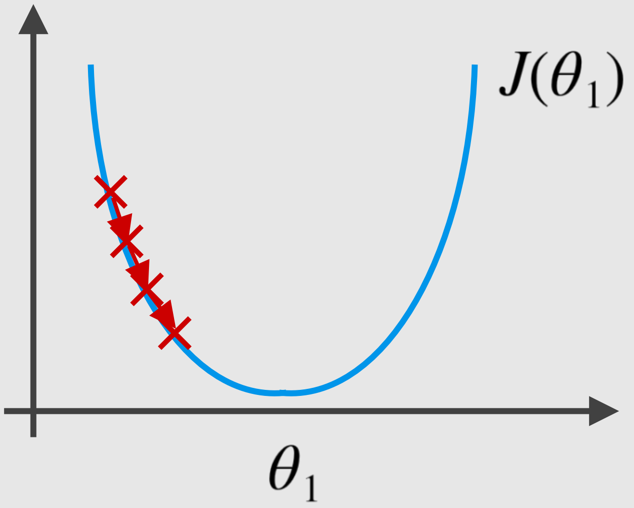
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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



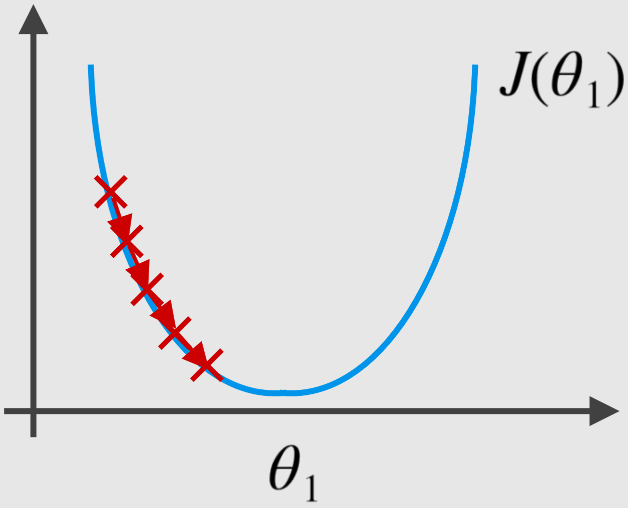
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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



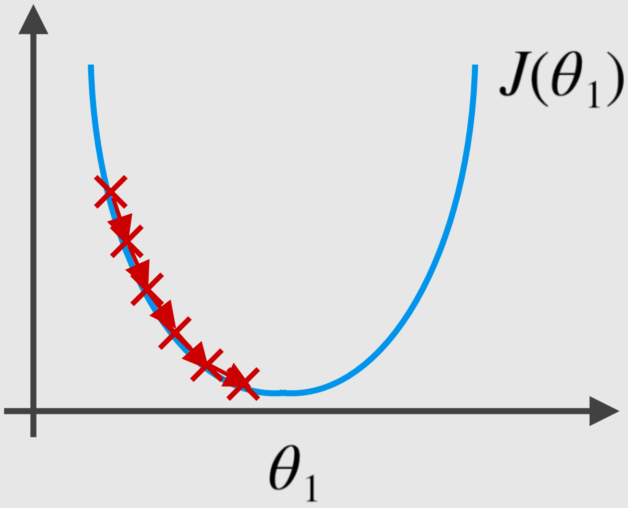
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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



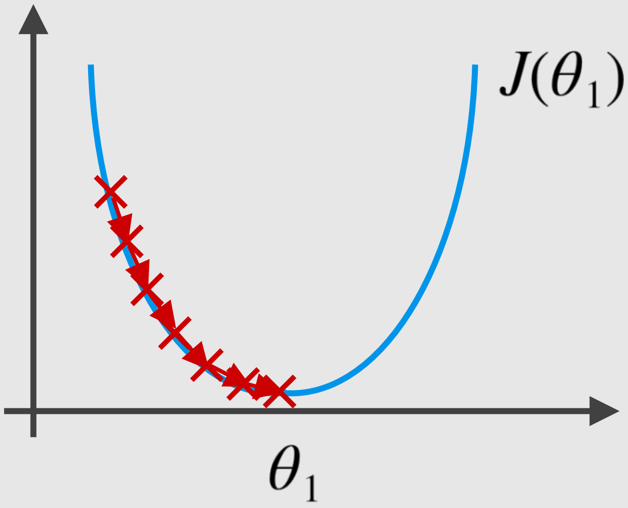
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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



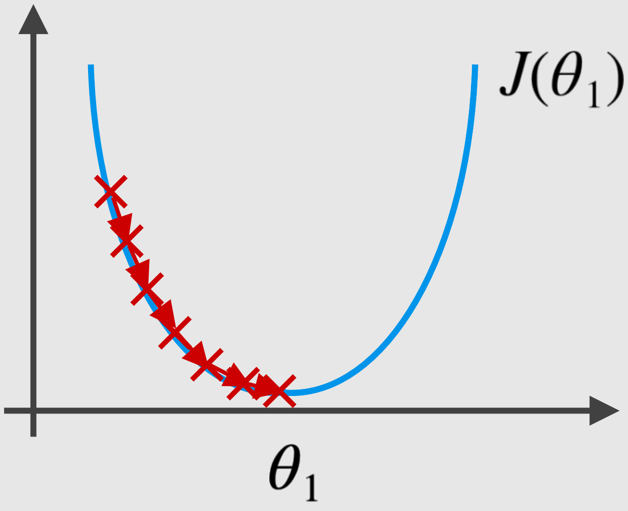
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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

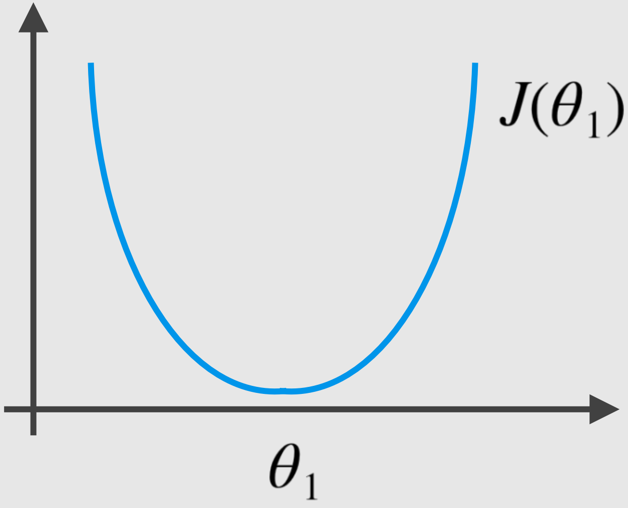


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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

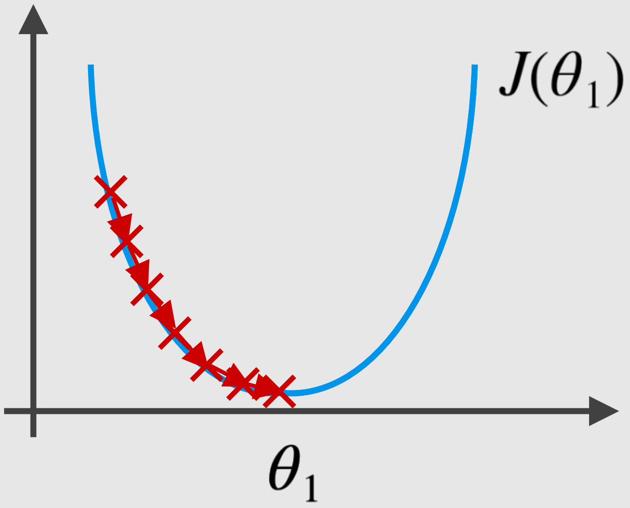


If α is too large, gradient descent can be ...

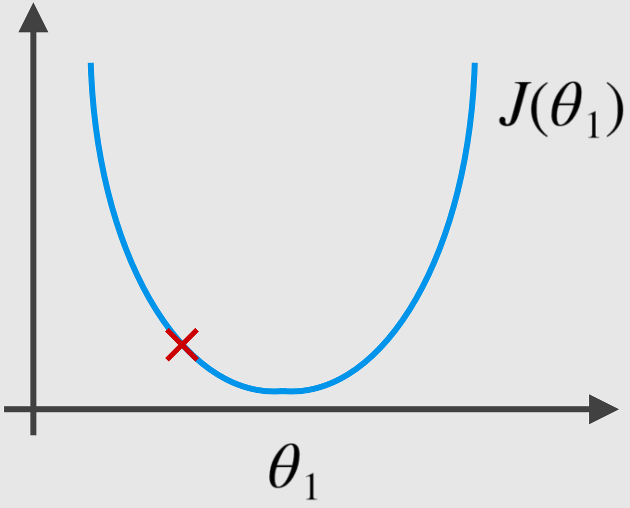


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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

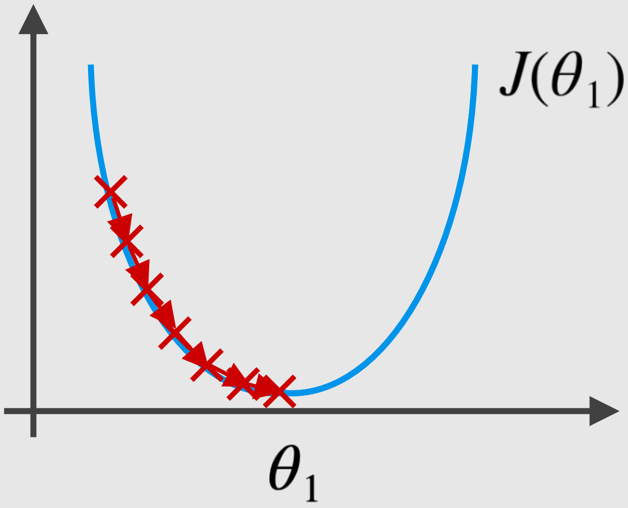


If α is too large, gradient descent can be ...

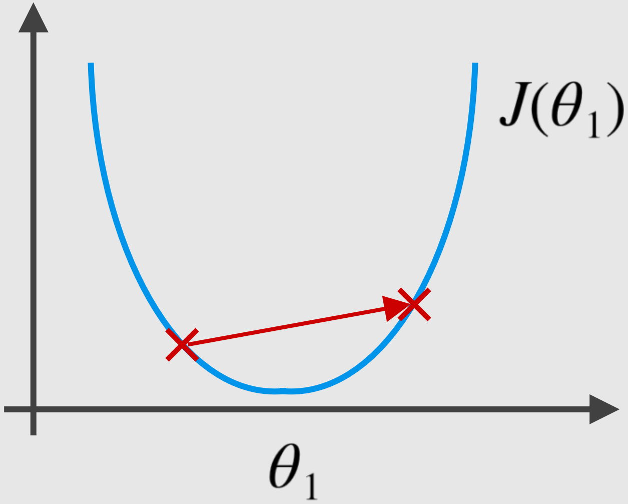


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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

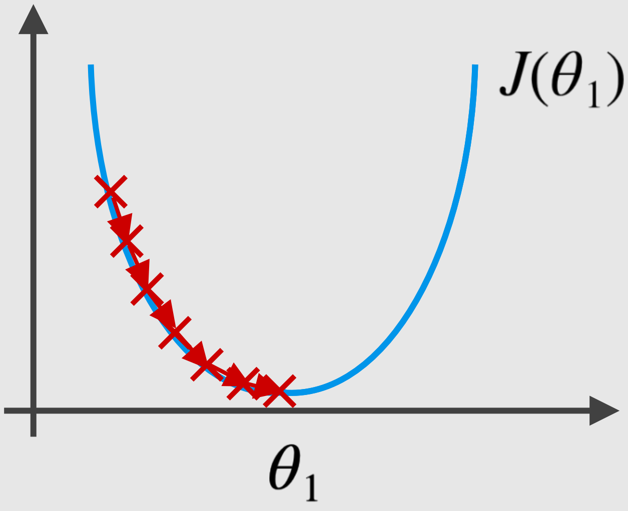


If α is too large, gradient descent can be overshoot the minimum. It may fail to converge, or even diverge.

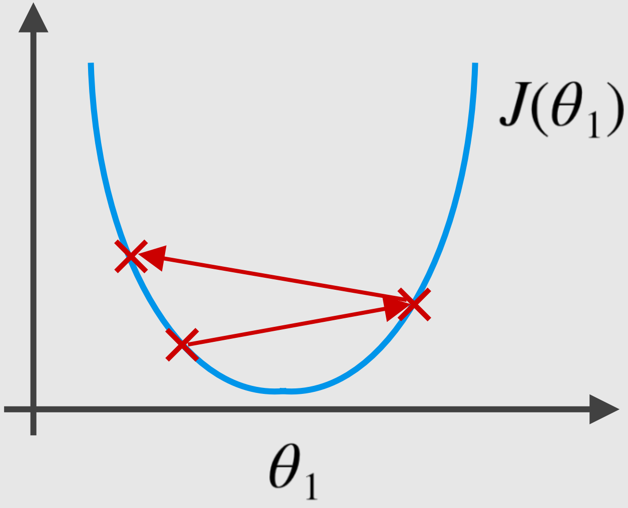


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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

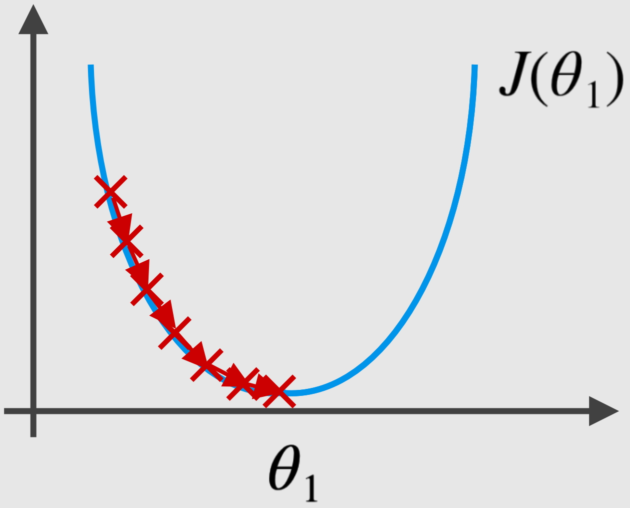


If α is too large, gradient descent can be overshoot the minimum. It may fail to converge, or even diverge.

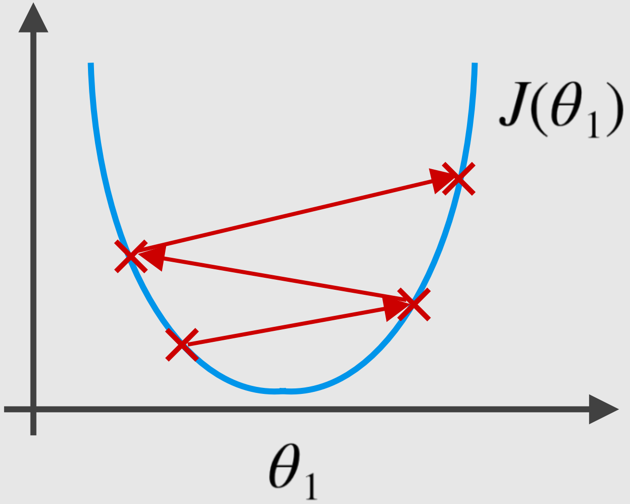


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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

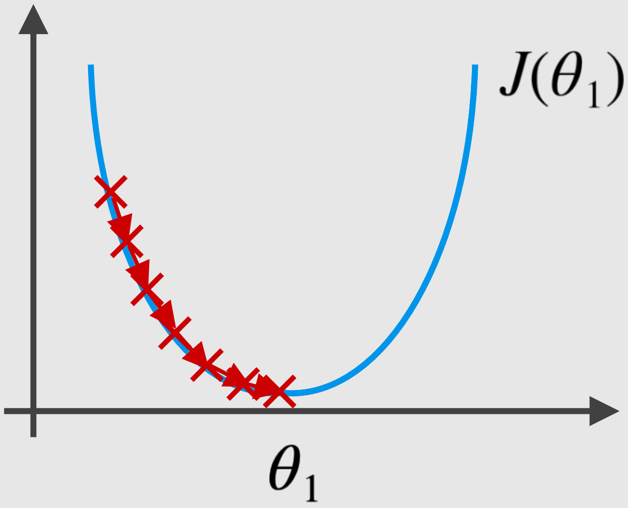


If α is too large, gradient descent can be overshoot the minimum. It may fail to converge, or even diverge.

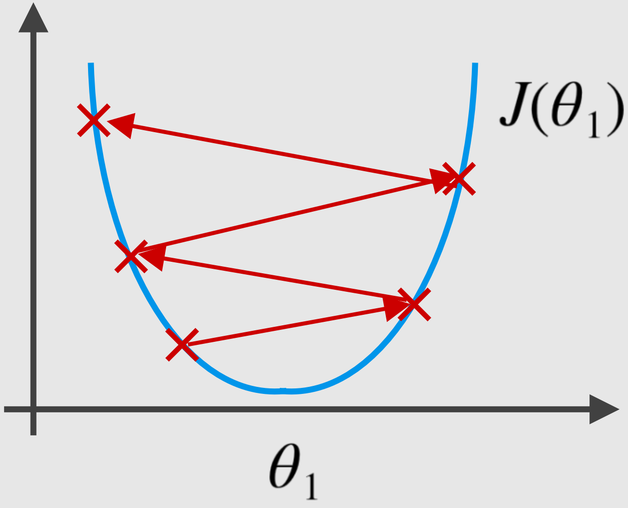


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

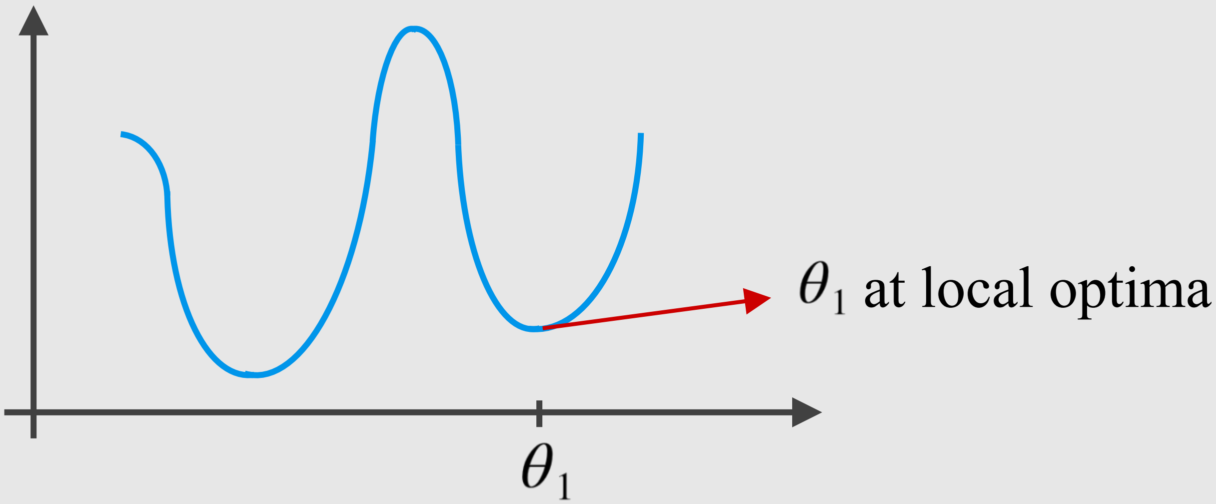
If α is too small, gradient descent can be slow.



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

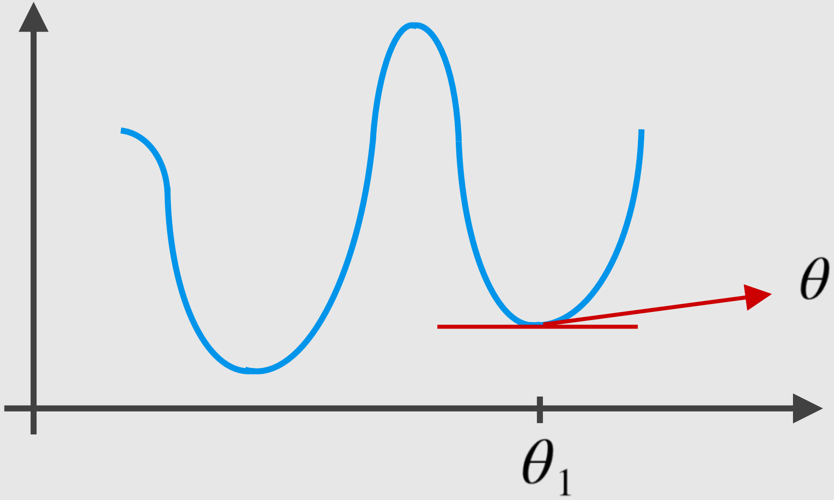


What will one step of gradient descent $\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$ do?



What will one step of gradient descent $\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$ do?

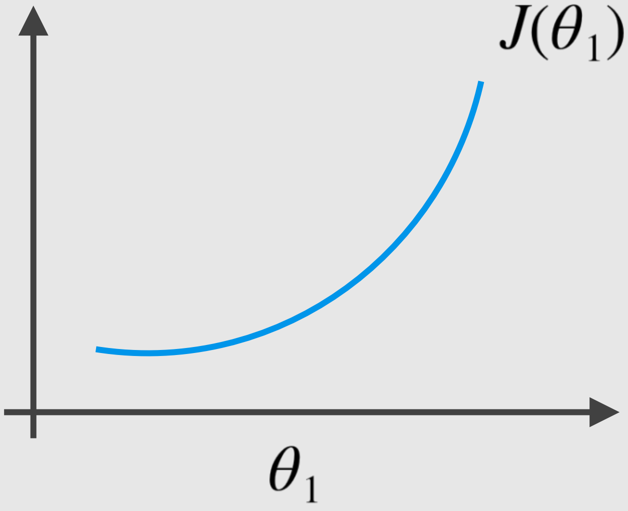
$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)}$$



θ_1 at local optima $= 0$

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

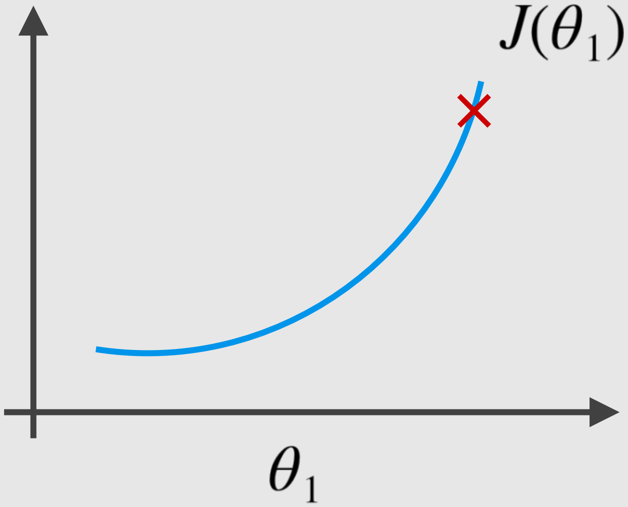
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$



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

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

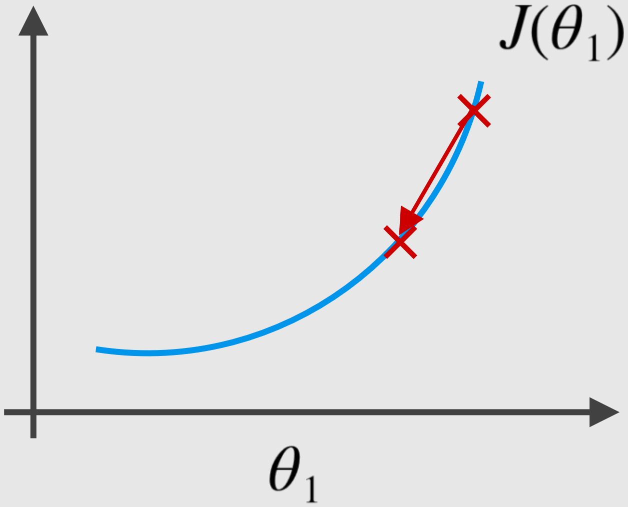
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$



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

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

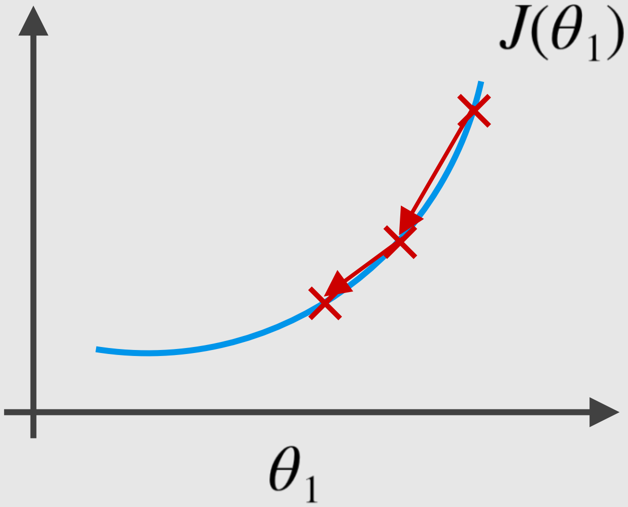
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$



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

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

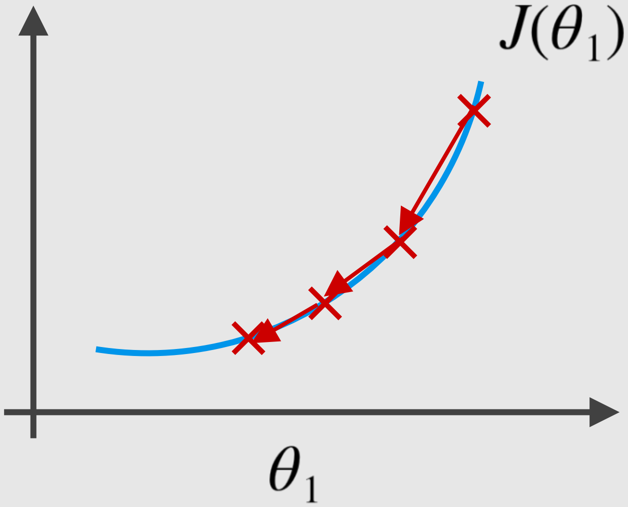
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$



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

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

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

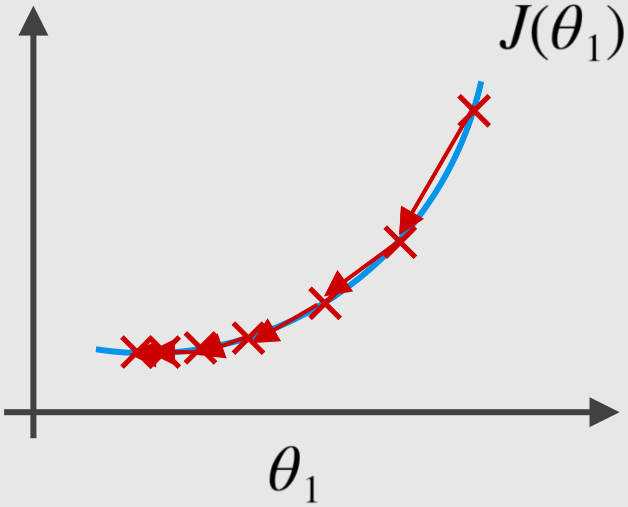


As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.

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.



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 $j = 1$)

}

Linear Regression Model

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

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

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 $j = 1$)

}

Linear Regression Model

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

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

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

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

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

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

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

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

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

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

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

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

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

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

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

Gradient Descent algorithm

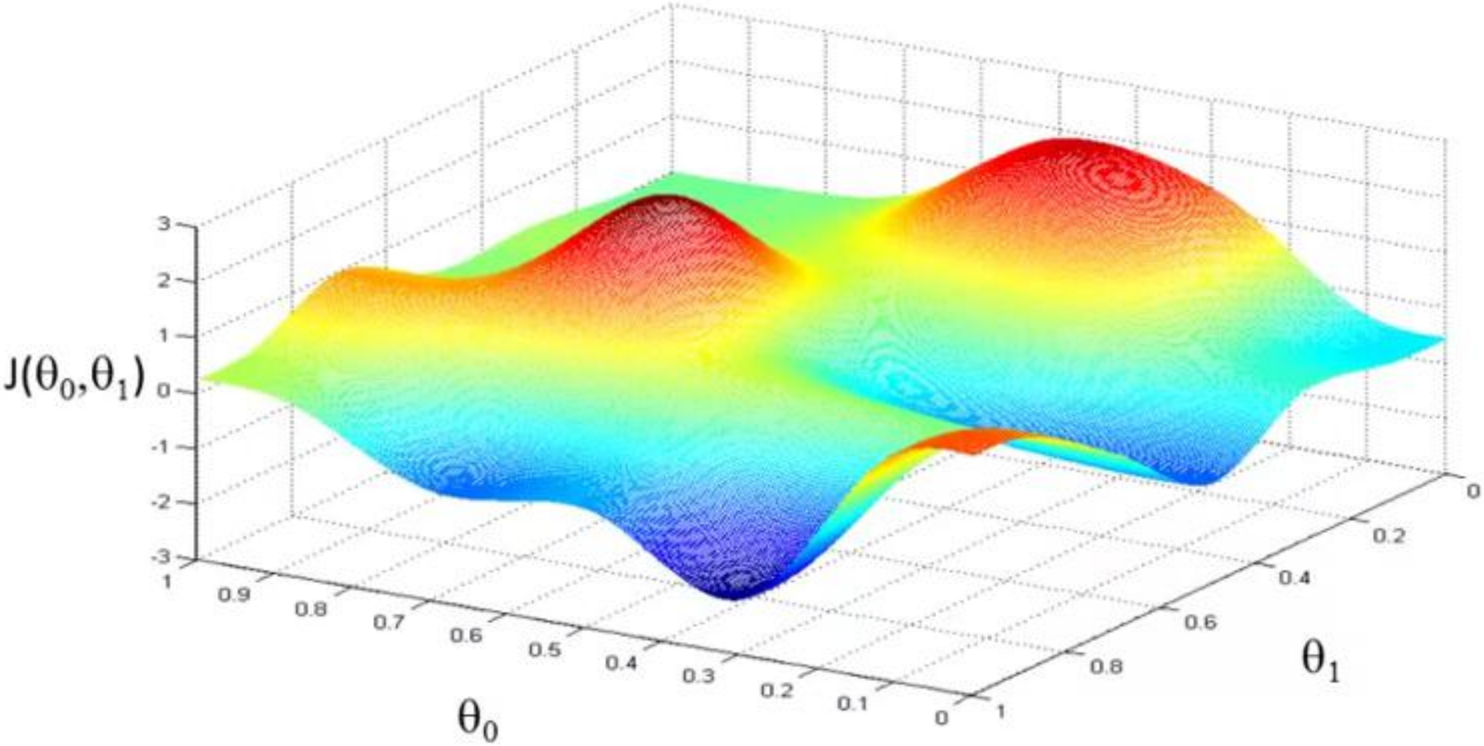
repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

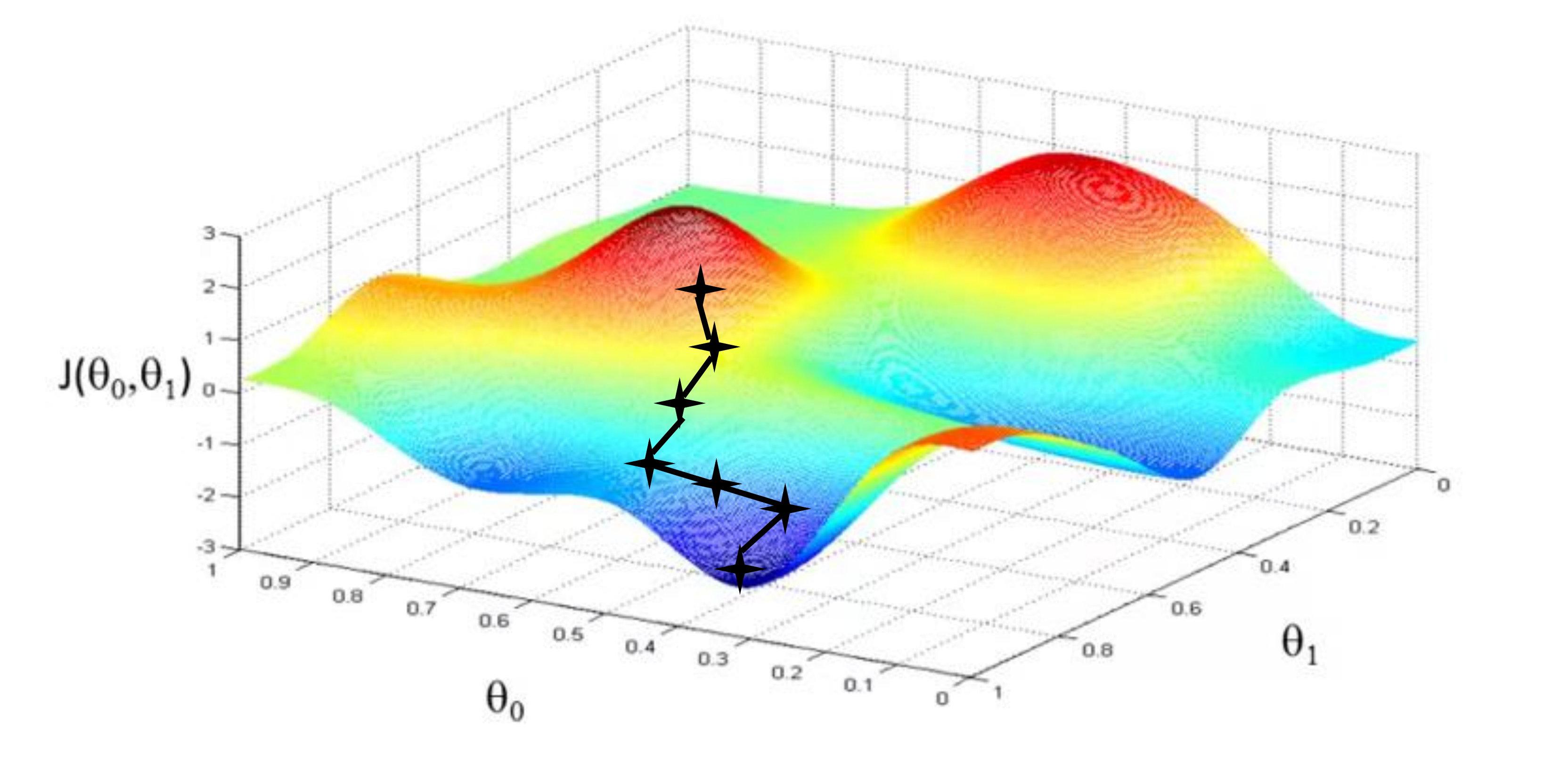
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})\, x^{(i)}$$

update θ_0 and θ_1 simultaneously

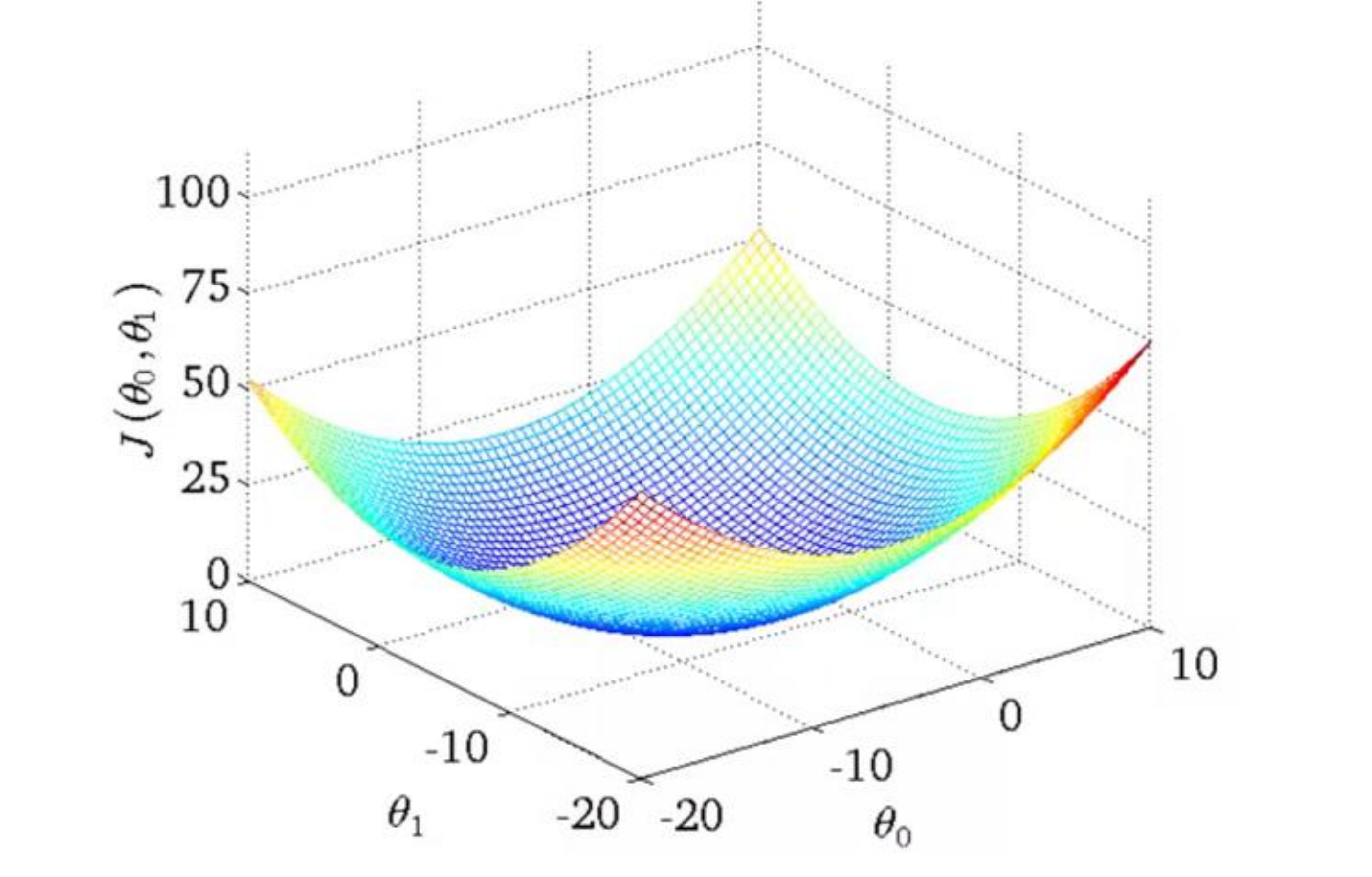
}



3
2
1
0
-1
-2
-3
J(θ0,θ1)
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



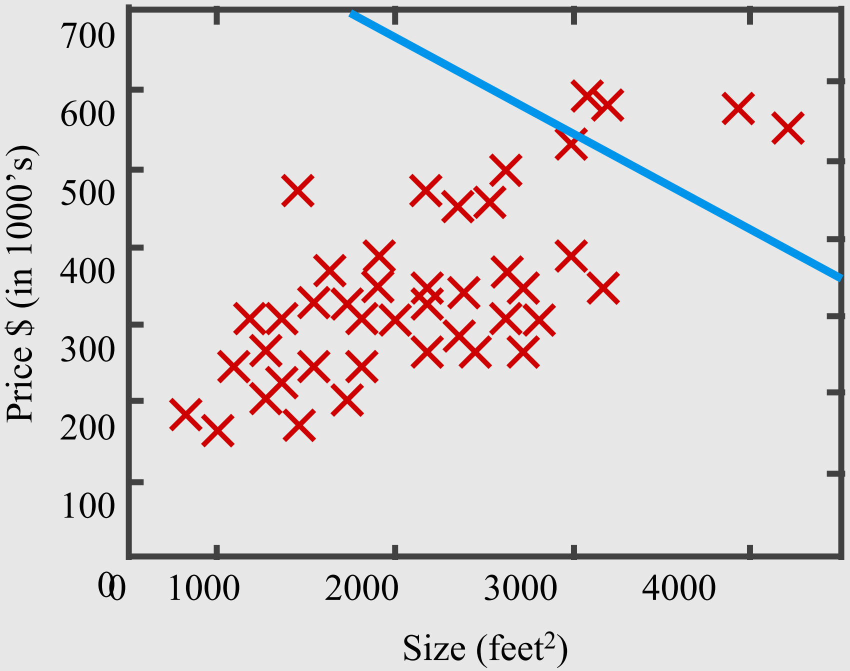
J(θ0,θ1)
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ0
0
0.2
0.4
0.6
0.8
1
θ1



$J(\theta_0, \theta_1)$
100
75
50
25
0
10
0
-10
-20
θ_1
-20
-10
0
10
θ_0

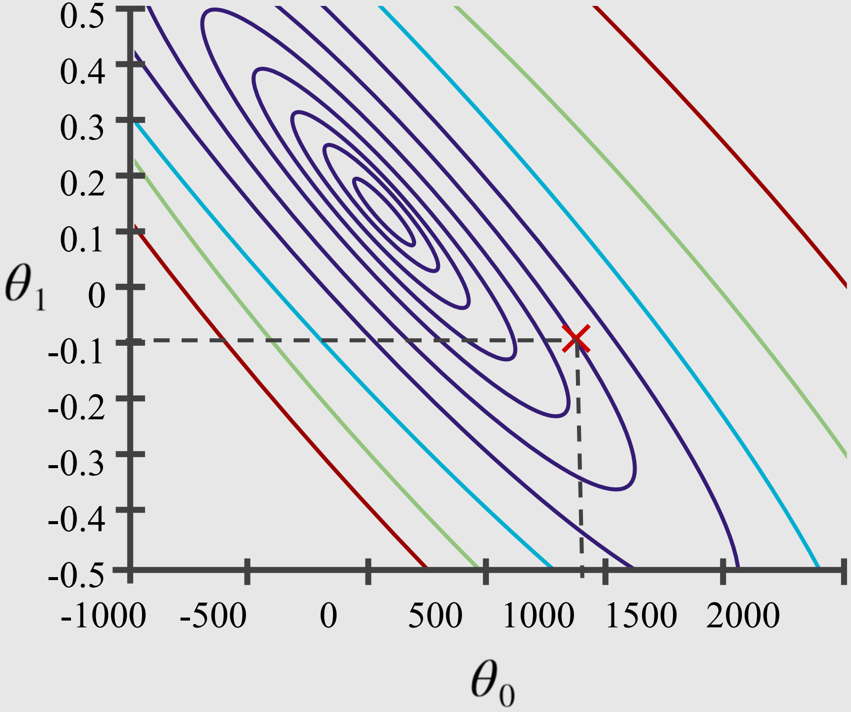
$h_\theta(x)$

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



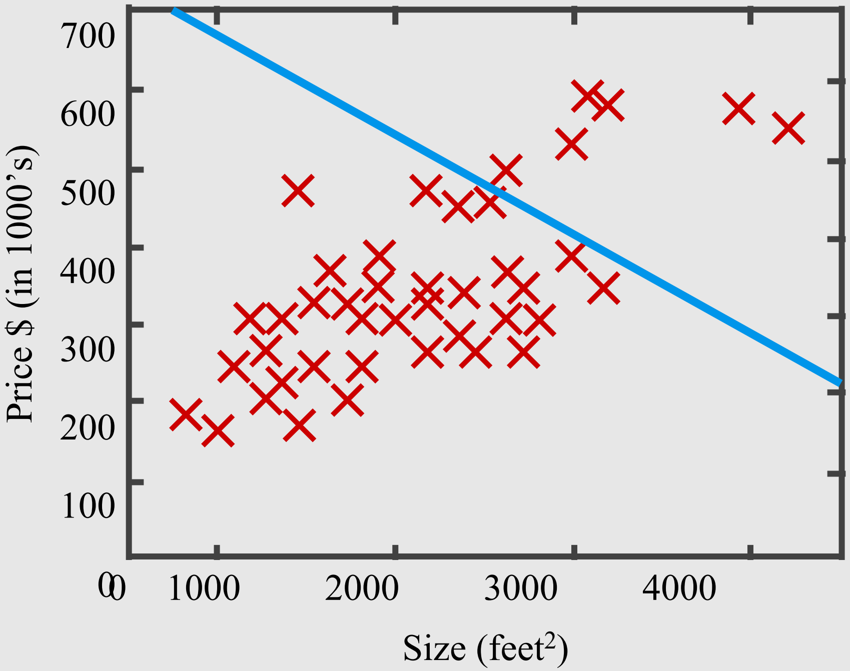
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



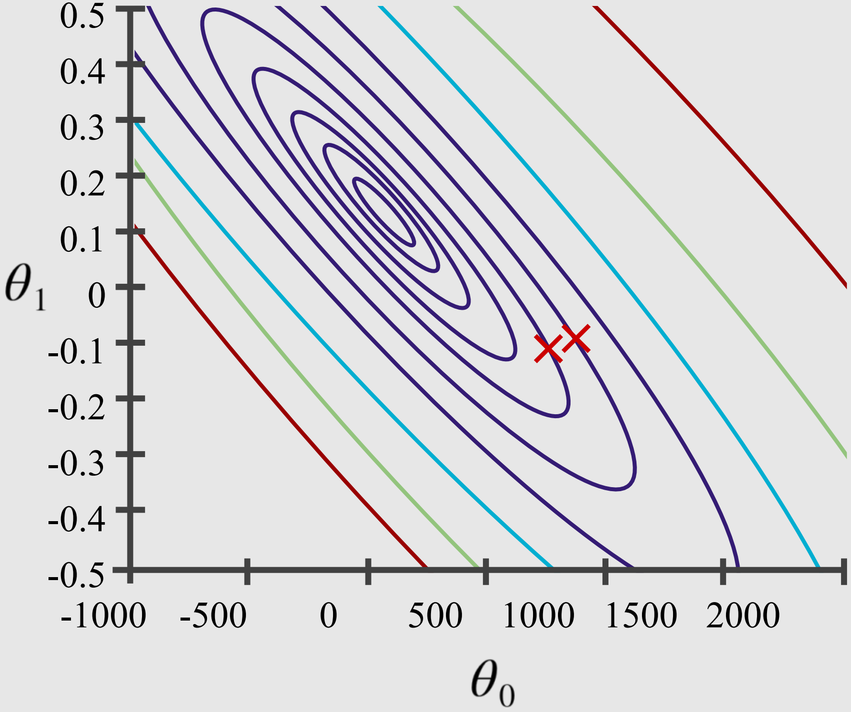
$h_\theta(x)$

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



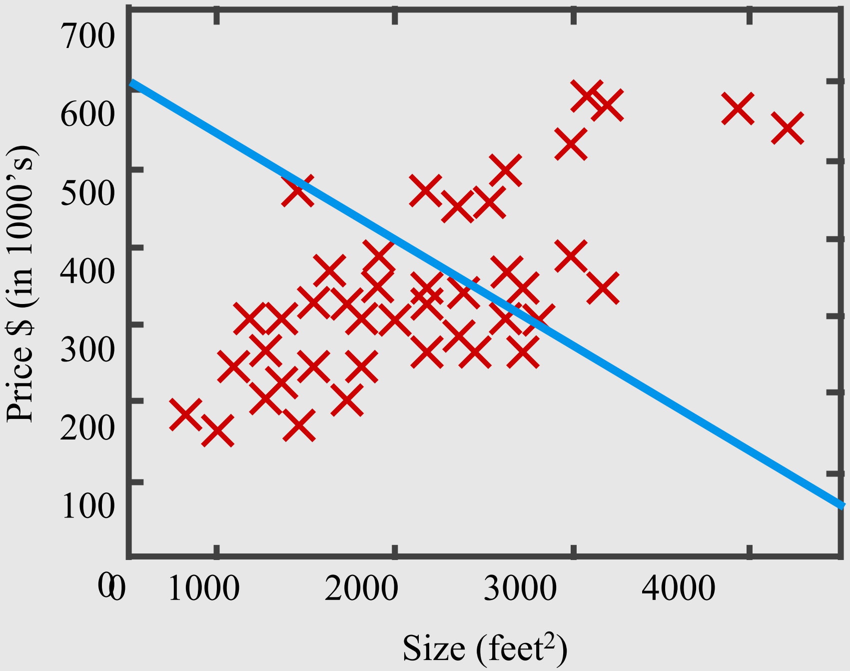
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



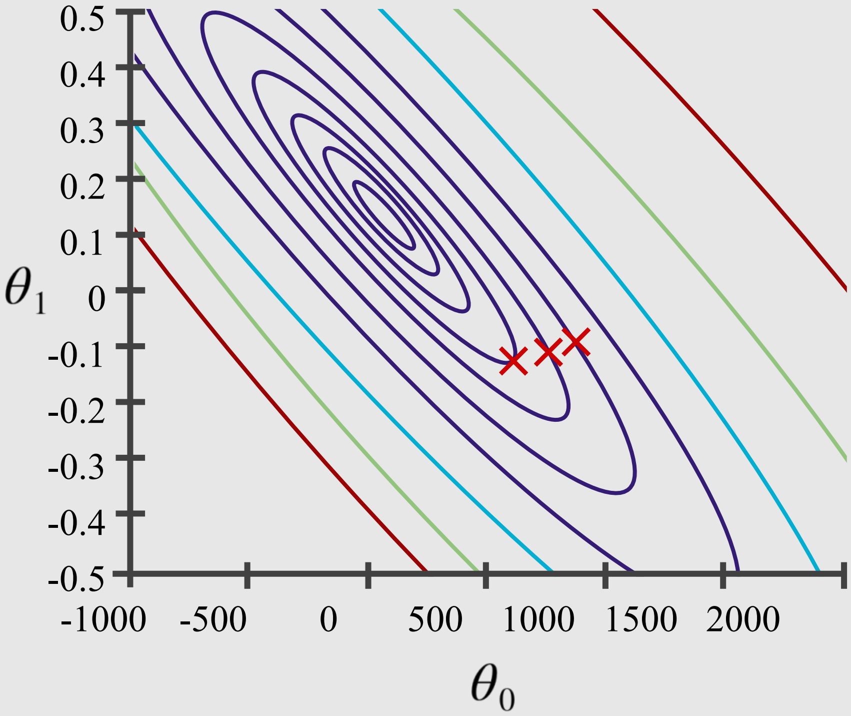
$h_\theta(x)$

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



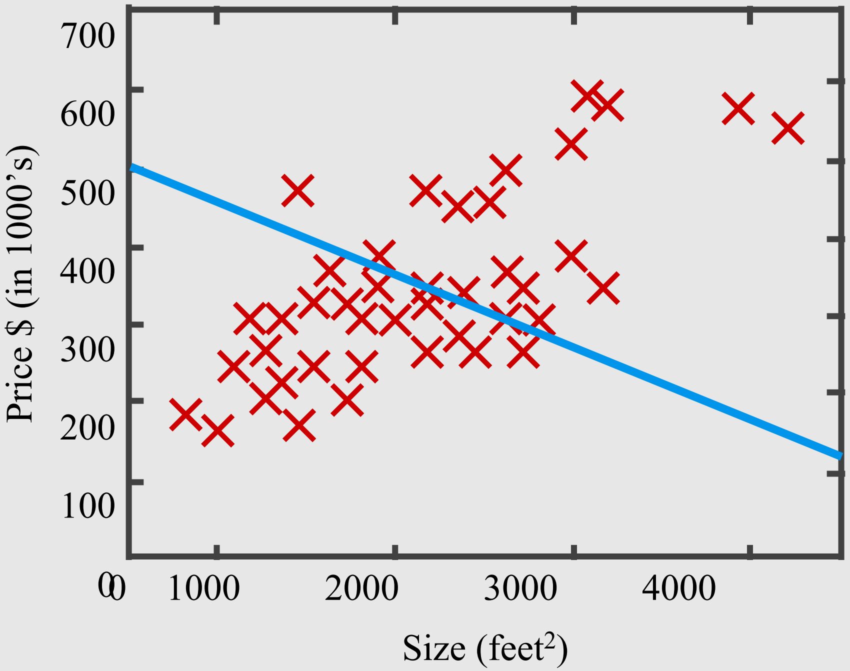
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



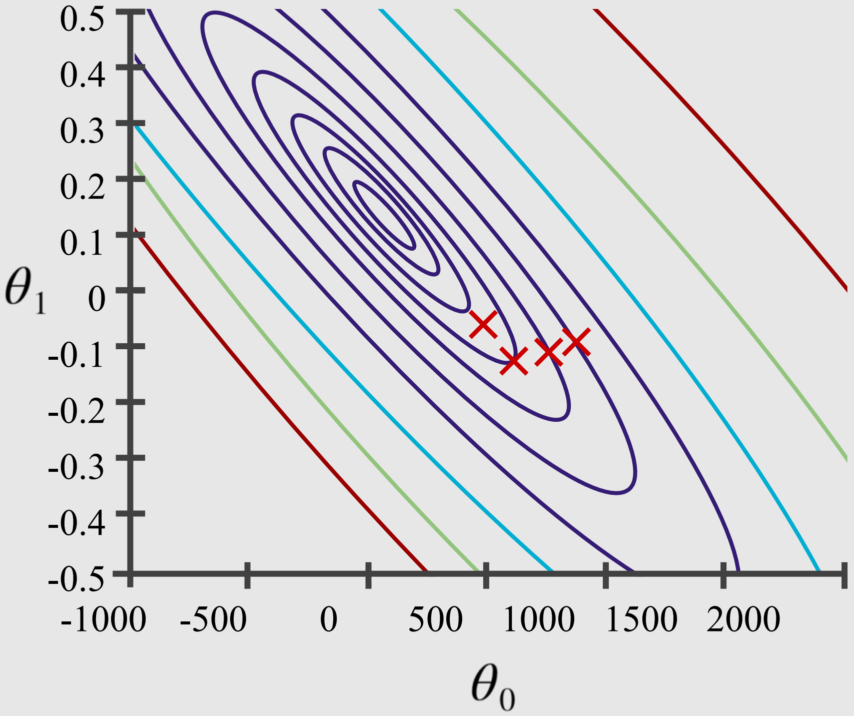
$h_\theta(x)$

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



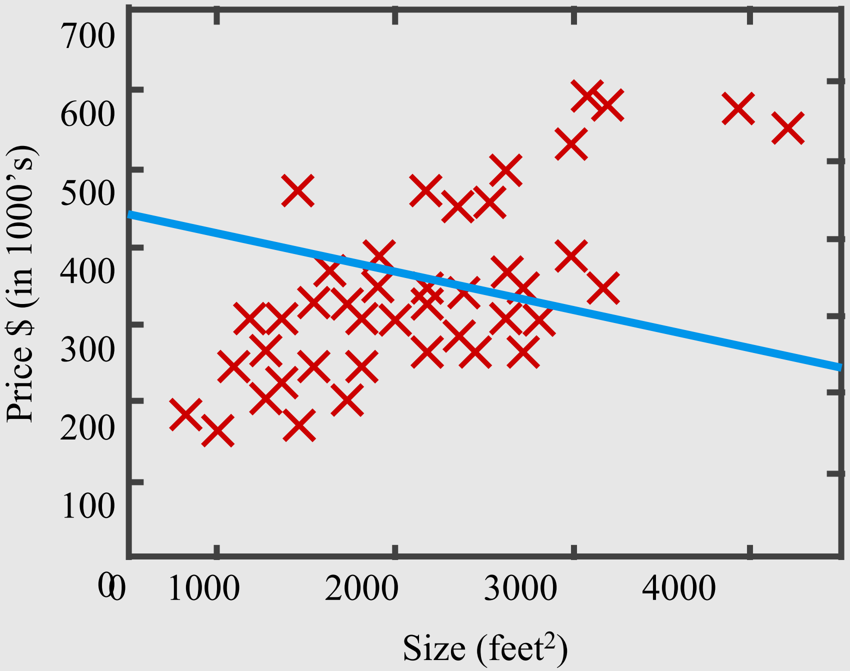
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



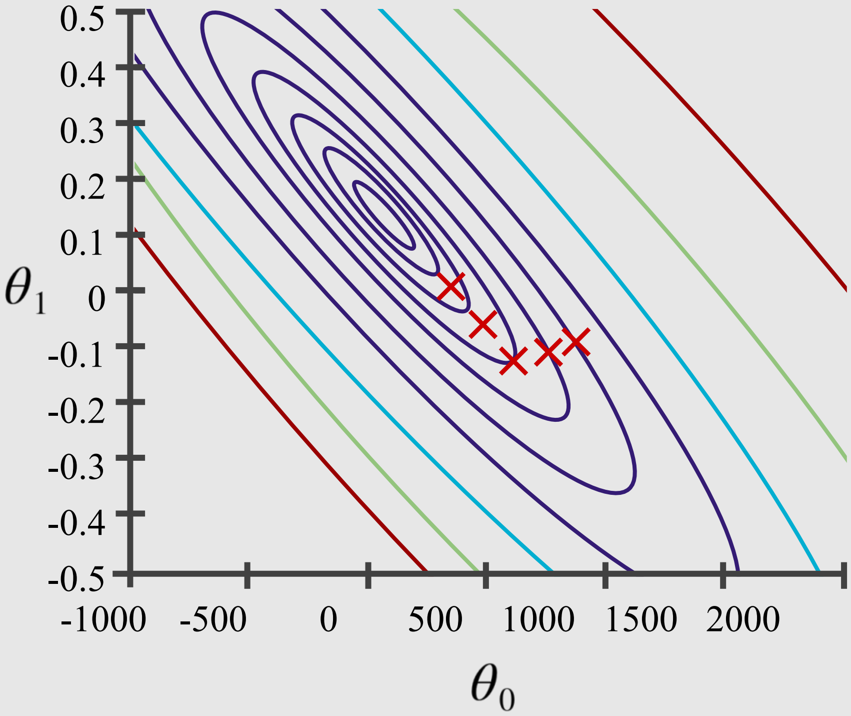
$h_\theta(x)$

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



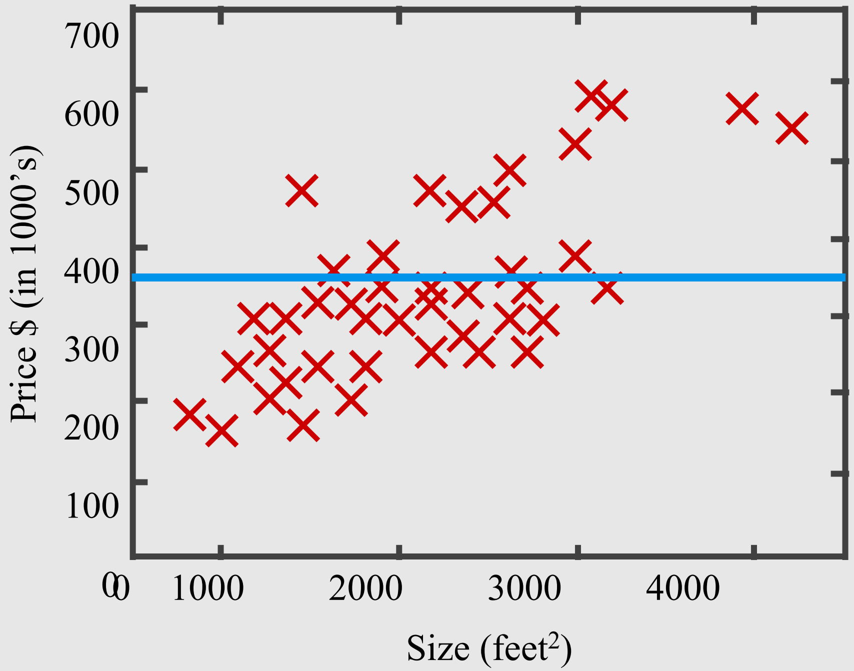
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



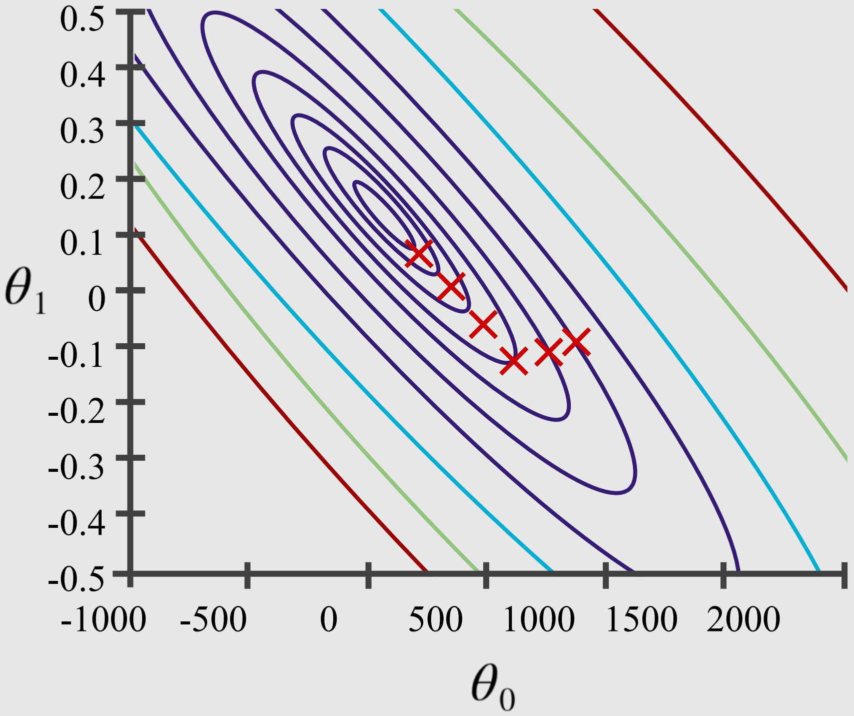
$h_\theta(x)$

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



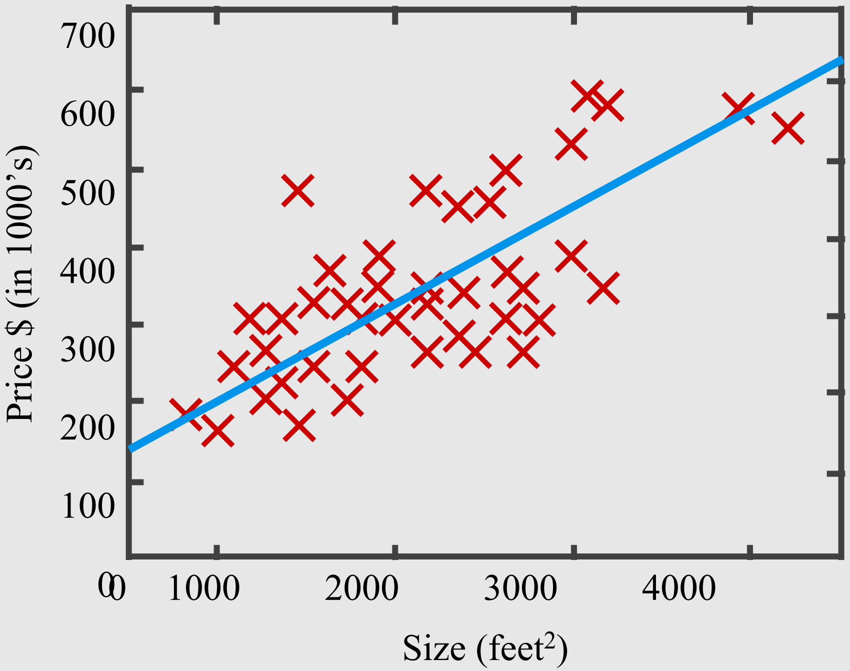
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



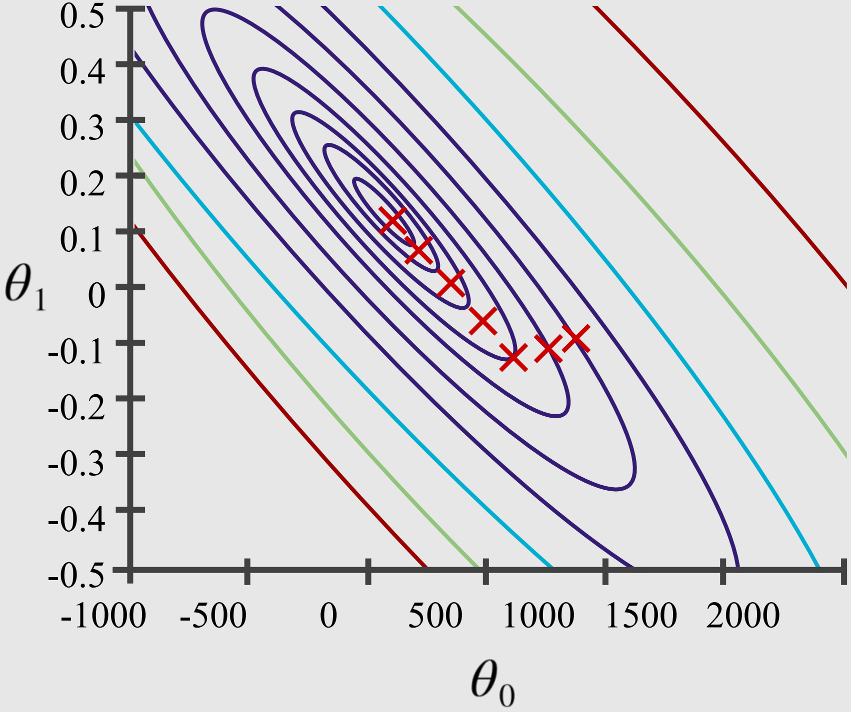
$h_\theta(x)$

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

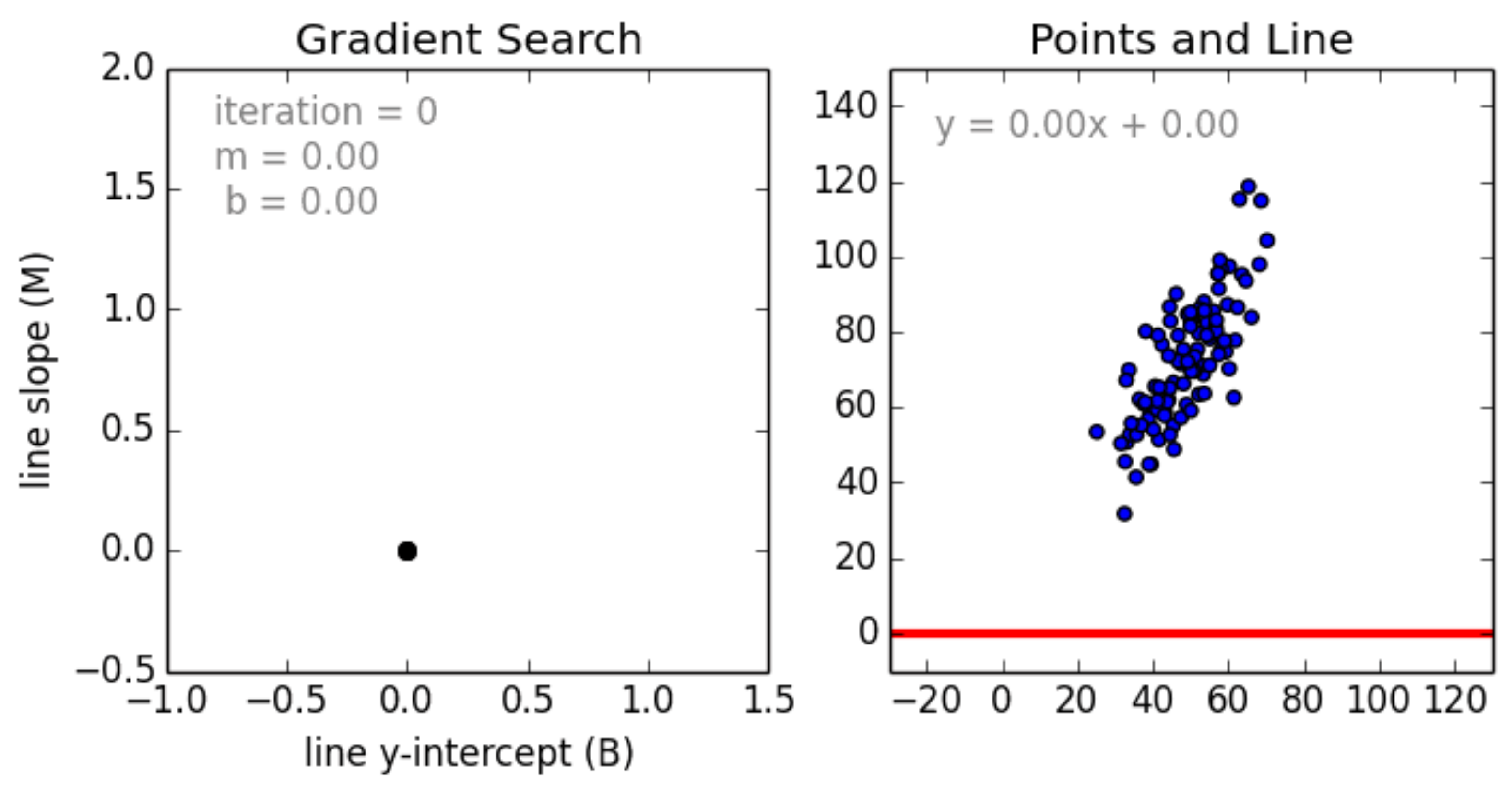


$J(\theta_0,\theta_1)$

(function of the parameters θ_0, θ_1)

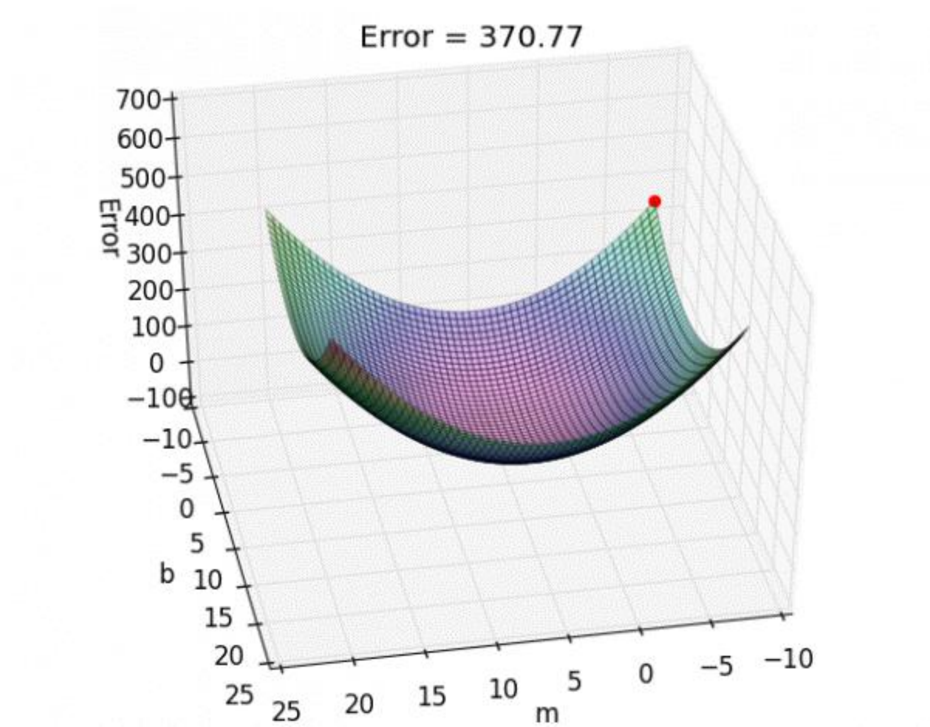


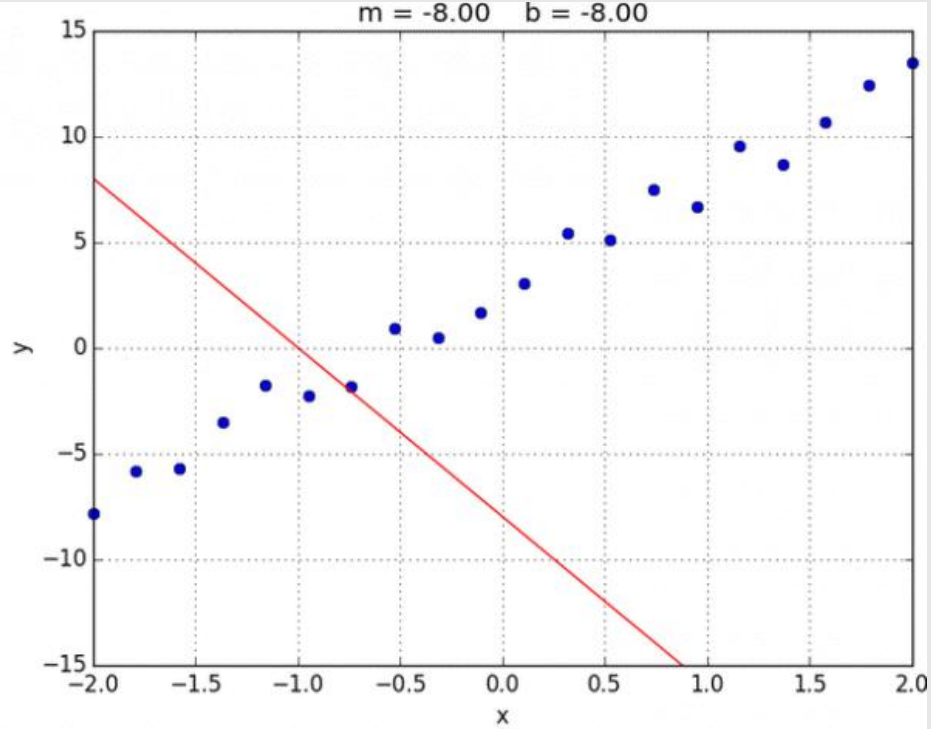
$$h_\theta(x) = \theta_0 + \theta_1 x \quad \Rightarrow \quad y = b + mx$$



Credit: https://github.com/mattnedrich/GradientDescentExample/raw/master/gradient_descent_example.gif

$$y = b + mx$$





Credit: https://alykhantejani.github.io/a-brief-introduction-to-gradient-descent/

“Batch” Gradient Descent

“Batch”: Each step of gradient descent uses **all the training examples**.

“Batch” Gradient Descent

“Batch”: Each step of gradient descent uses **all the training examples**.

- Stochastic Gradient Descent
- Mini-batch Gradient Descent

"Batch" Gradient Descent

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

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

update θ_0 and θ_1 simultaneously

}

Stochastic Gradient Descent

Each step of gradient descent uses **one training example**.

repeat until convergence {

for $i \coloneqq 1, \ldots, m$ {

$$\theta_0 := \theta_0 - \alpha(h_\theta(x^{(i)}) - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

}

}

Mini-batch Gradient Descent

Each step of gradient descent uses ***b* training examples**.

Say $b = 10$, $m = 1000$.
repeat until convergence {

for $i \coloneqq 1, 11, 21\ldots, 991$ {

$$\theta_0 := \theta_0 - \alpha \frac{1}{10} \sum_{i=k}^{i+9} (h_\theta(x^{(k)}) - y^{(k)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{10} \sum_{i=k}^{i+9} (h_\theta(x^{(k)}) - y^{(k)})\, x^{(k)}$$

} }

Linear Regression with multiple variables

Multiple ~~Variables~~ Features

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

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

Multiple ~~Variables~~ Features

Size in feet^2 x_1	**Number of bedrooms** x_2	**Number of floors** x_3	**Age of home (years)** x_4	**Price ($) in 1000's** y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	2	36	178
...	...	...	...	...

Notation:

n = number of features

$x^{(i)}$ = input (features) of i^{th} training example

$x_j^{(i)}$ = value of features j in i^{th} training example

Hypothesis

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

Hypothesis

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

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

Hypothesis

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

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$$h_\theta(x) = 80 + 0.1 x_1 + 10 x_2 + 3 x_3 - 2 x_4$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

For convenience of notation, define $x_0 = 1$.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

For convenience of notation, define $x_0 = 1$.

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

For convenience of notation, define $x_0 = 1$.

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$\begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$h_\theta(x) = \theta^T x$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

For convenience of notation, define $x_0 = 1$.

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$\begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$h_\theta(x) = \theta^T x$$

Multivariate linear regression.

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

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

Gradient Descent:

repeat {

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

} (simultaneously update for every $j = 0, 1, \ldots, n$)

Gradient Descent

Previously ($n = 1$):

repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

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

(simultaneously update θ_0, θ_1)

}

Gradient Descent

Previously ($n = 1$):

repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

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

(simultaneously update θ_0, θ_1)

}

New Algorithm ($n \geq 1$):

repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, 1, \ldots, n$)

}

Gradient Descent

Previously ($n = 1$):

repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

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

(simultaneously update θ_0, θ_1)

}

New Algorithm ($n \geq 1$):

repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, 1, \ldots, n$)

}

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

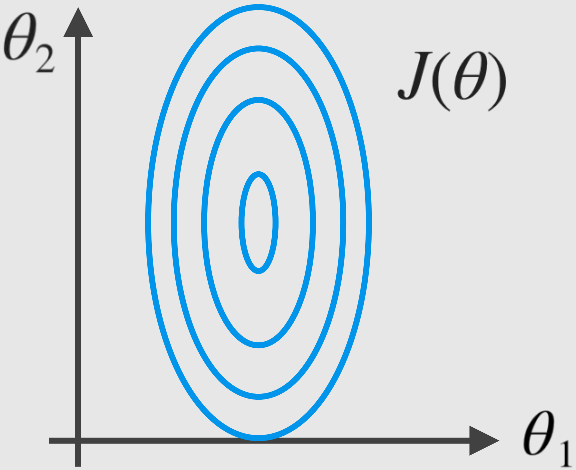
...

Feature Scaling

Feature Scaling

Idea: Make sure features are on similar scale.

E.g. x_1= size (0–2000 feet^2)
x_2= number of bedrooms (1–5)

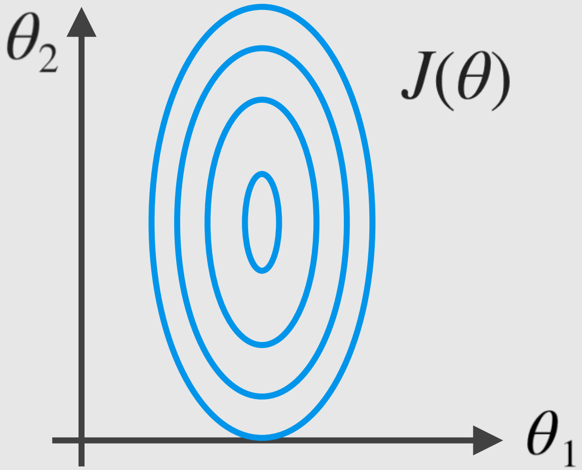


Feature Scaling

Idea: Make sure features are on similar scale.

E.g. x_1 = size (0–2000 feet^2)

x_2 = number of bedrooms (1–5)



$$x_1 = \frac{\text{size (feet}^2)}{2000}$$

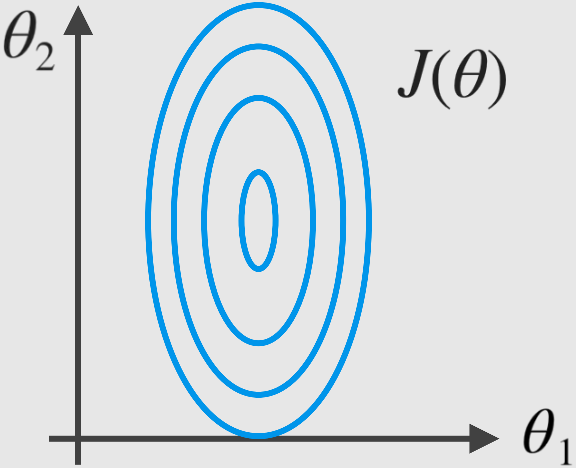
$$x_2 = \frac{\text{number of bedrooms}}{5}$$

Feature Scaling

Idea: Make sure features are on similar scale.

E.g. x_1= size (0–2000 feet^2)

x_2= number of bedrooms (1–5)



$$x_1 = \frac{\text{size (feet}^2)}{2000}$$

$$x_2 = \frac{\text{number of bedrooms}}{5}$$

Feature Scaling

Get every feature into approximately a $-1 \leq x_i \leq 1$ range.

Mean Normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean (do not apply to $x_0 = 1$).

E.g. $x_1 = \dfrac{\text{size} - 1000}{2000}$ ➡ $-0.5 \le x_1 \le 0.5$

$x_2 = \dfrac{\#\text{bedrooms} - 2.5}{5}$ ➡ $-0.5 \le x_2 \le 0.5$

Mean Normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean (do not apply to $x_0 = 1$).

E.g. $x_1 = \dfrac{\text{size} - 1000}{2000}$ ➡ $-0.5 \le x_1 \le 0.5$

$x_2 = \dfrac{\#\text{bedrooms} - 2.5}{5}$ ➡ $-0.5 \le x_2 \le 0.5$

$$x_1 = \frac{x_1 - \mu_1}{s_1} \qquad x_2 = \frac{x_2 - \mu_2}{s_2}$$

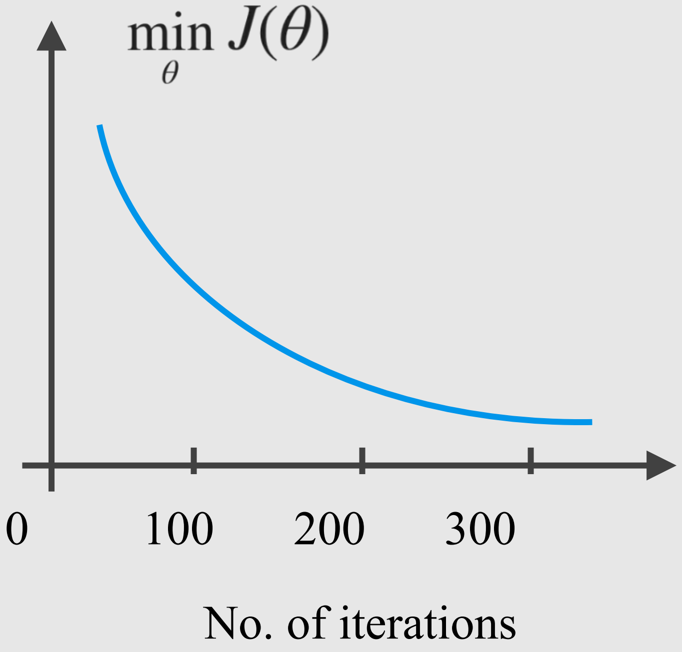
Learning Rate

Gradient Descent

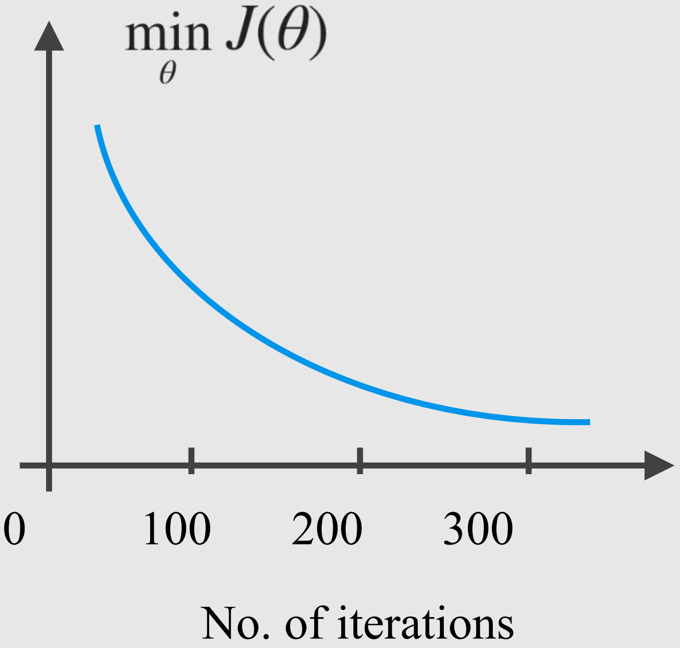
$$\theta_j := \theta_{\mathrm{j}} - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- "Debugging" : How to make sure gradient descent is working correctly.
- How to choose learning rate α.

Making sure gradient descent is working correctly.



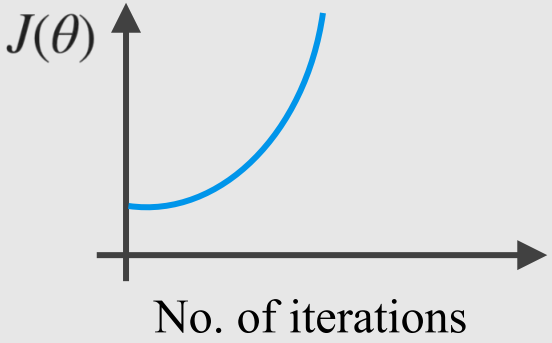
Making sure gradient descent is working correctly.



Example automatic convergence test:

Declare convergence if $J(\theta)$ decreases by less than 10^{-3} in one iteration.

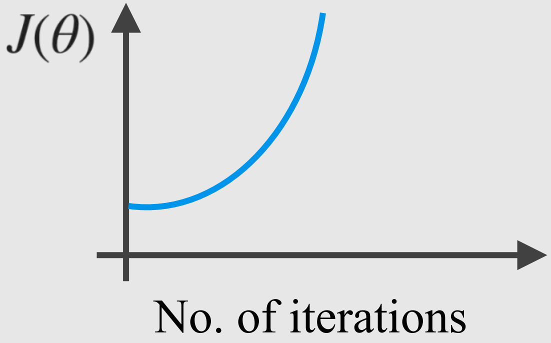
Making sure gradient descent is working correctly.



Gradient descent not working.

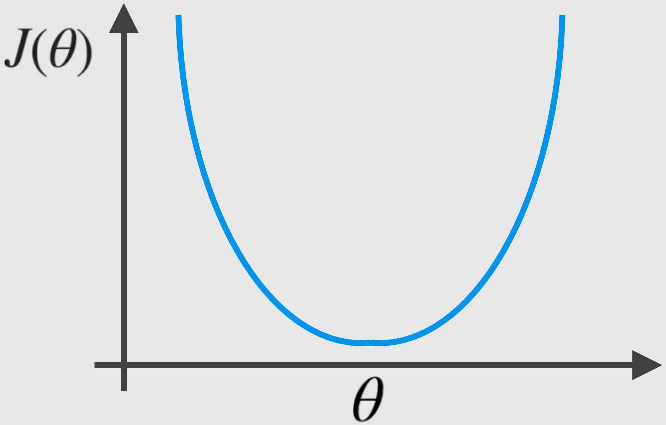
Use smaller α.

Making sure gradient descent is working correctly.

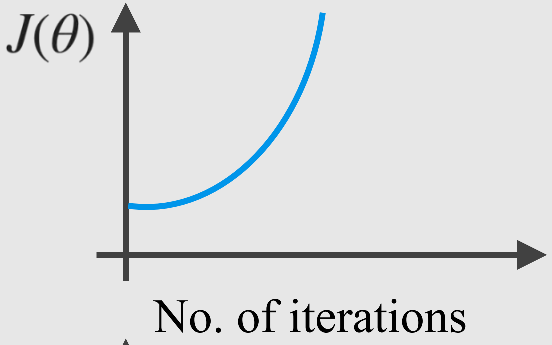


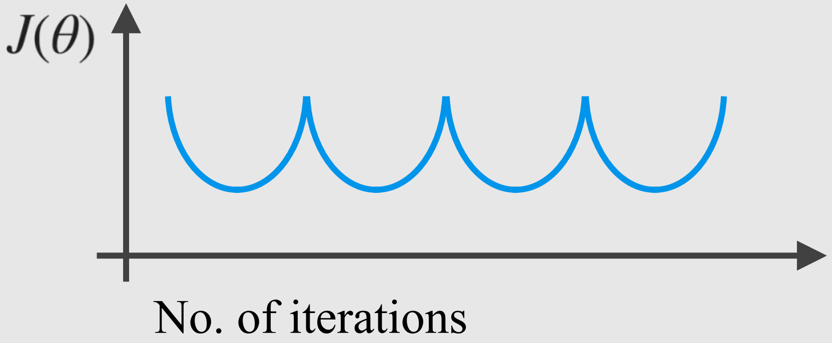
Gradient descent not working.

Use smaller α.



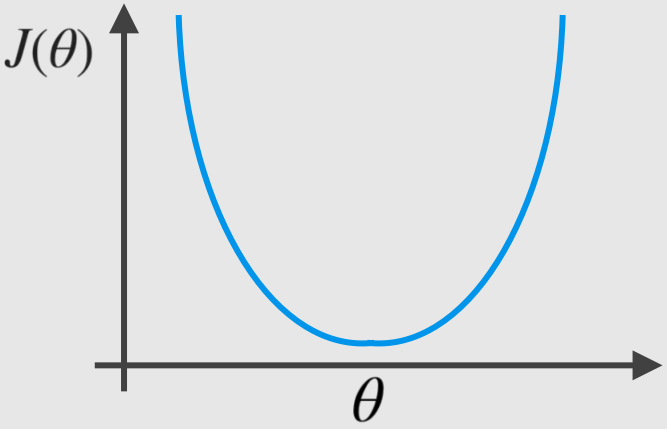
Making sure gradient descent is working correctly.



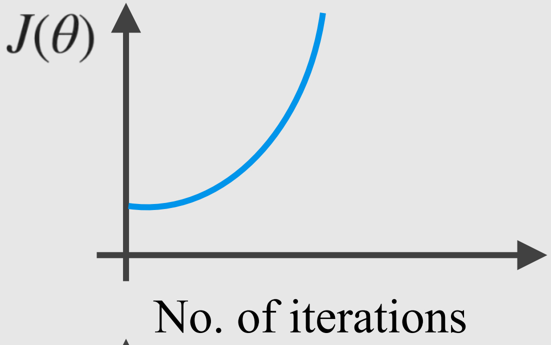


Gradient descent not working.

Use smaller α.

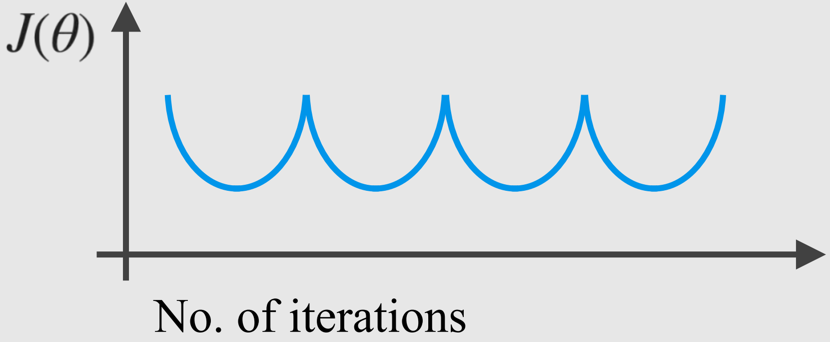


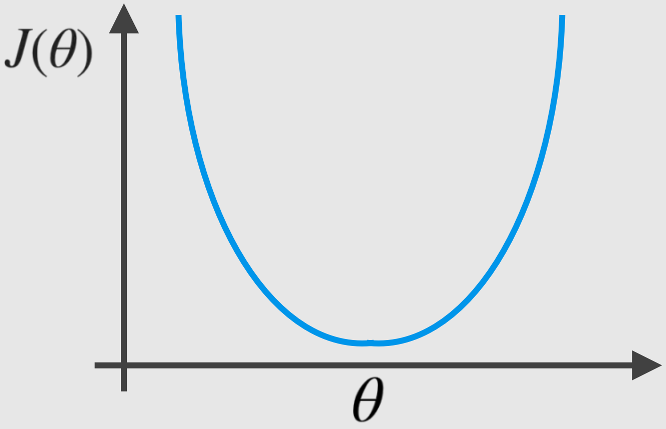
Making sure gradient descent is working correctly.



Gradient descent not working.

Use smaller α.





- For sufficiently small α, $J(\theta)$ should decrease on every iteration.
- But if α is too small, gradient descent can be slow to converge.

Summary

- If α is too small: slow convergence.
- If α is too large: $J(\theta)$ may not decrease on every iteration; may not converge.

To choose α, try

$$\ldots, 0.001, \ldots, 0.01, \ldots, 0.1, \ldots, 1, \ldots$$

Features and Polynomial Regression

Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \text{frontage} + \theta_2 \times \text{depth}$$

Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \underbrace{\text{frontage}}_{x_1} + \theta_2 \times \underbrace{\text{depth}}_{x_2}$$

Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \underbrace{\text{frontage}}_{x_1} + \theta_2 \times \underbrace{\text{depth}}_{x_2}$$

Area x = frontage $\times$ depth

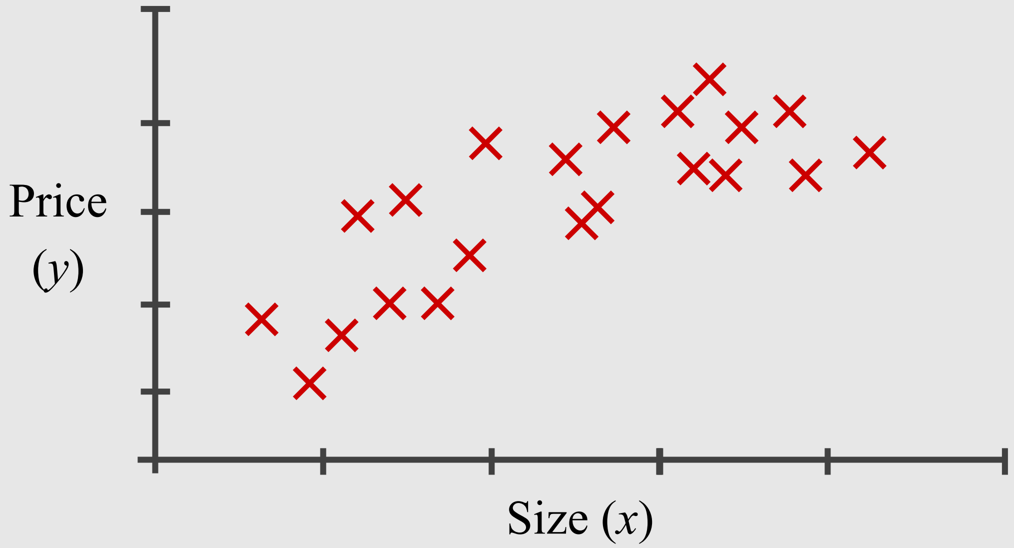
Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \underset{\substack{\Big\downarrow \\ x_1}}{\text{frontage}} + \theta_2 \times \underset{\substack{\Big\downarrow \\ x_2}}{\text{depth}}$$

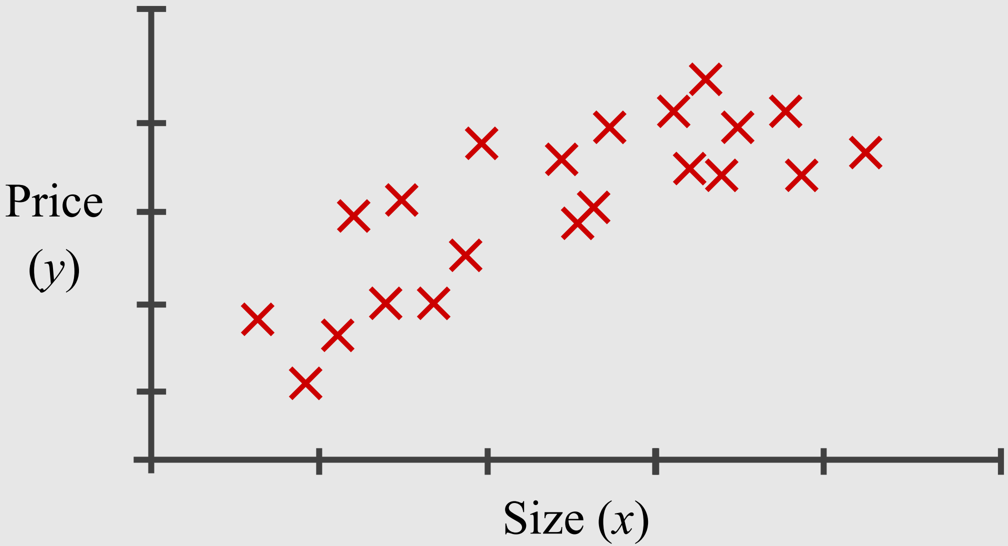
Area x = frontage $\times$ depth

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

Polynomial Regression

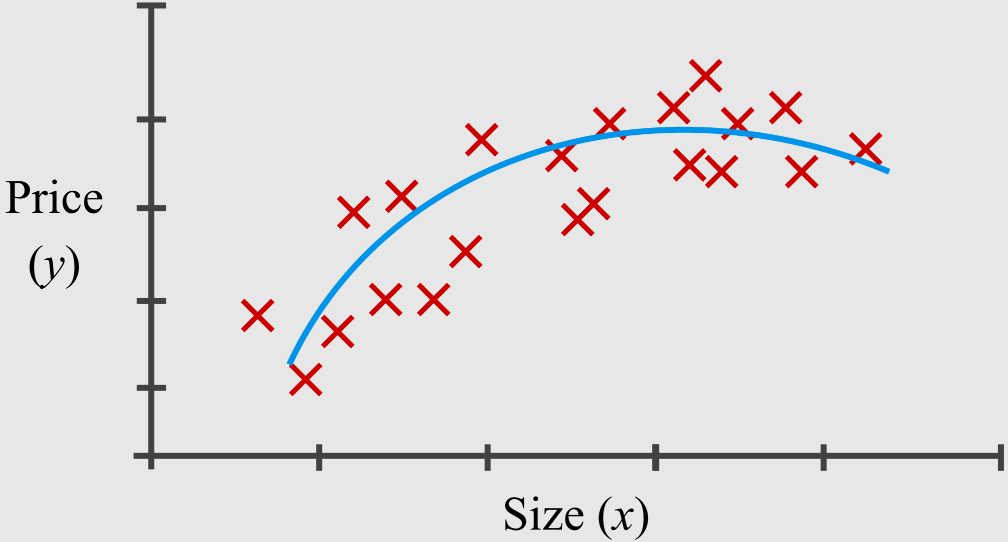


Polynomial Regression



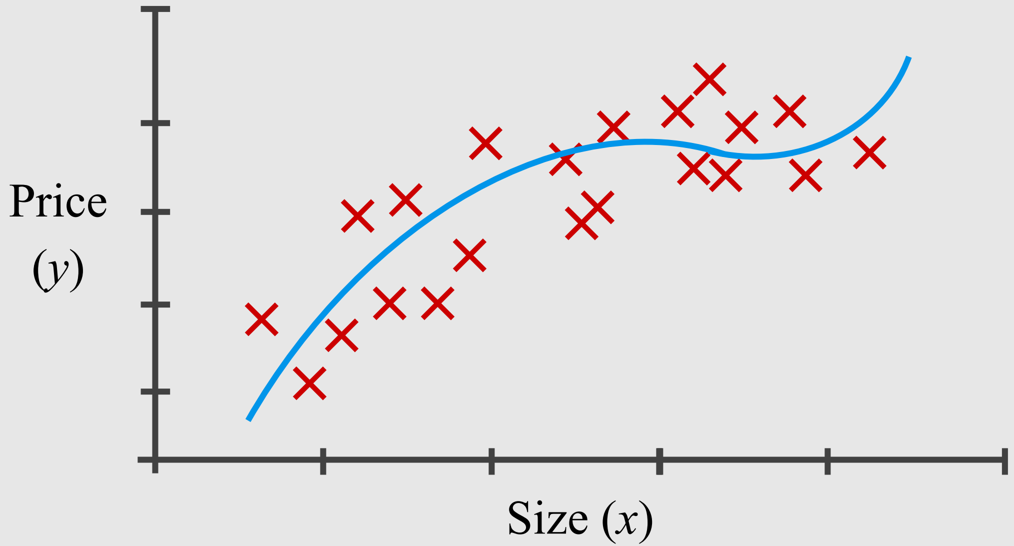
$$\theta_0 + \theta_1 x + \theta_2 x^2$$

Polynomial Regression



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

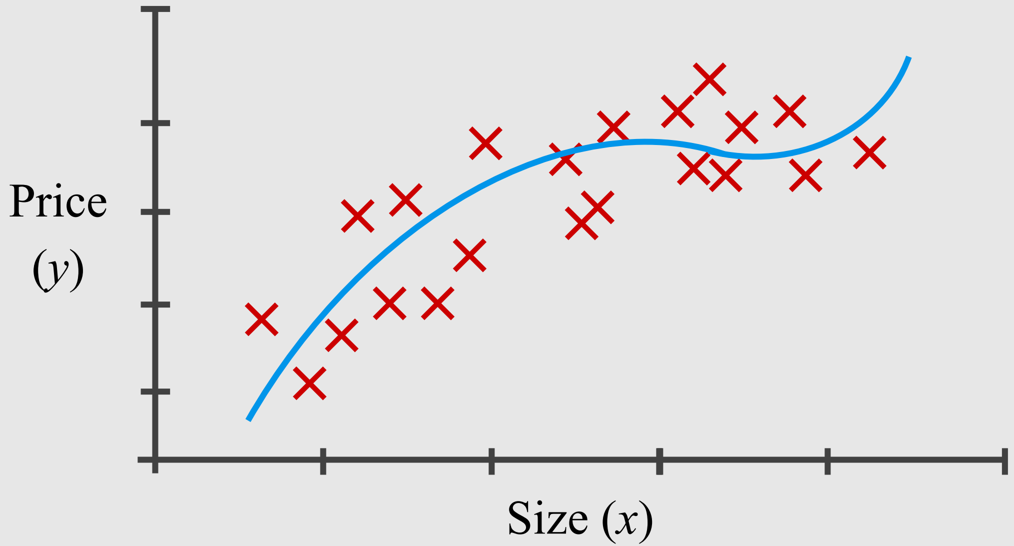
Polynomial Regression



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

Polynomial Regression



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

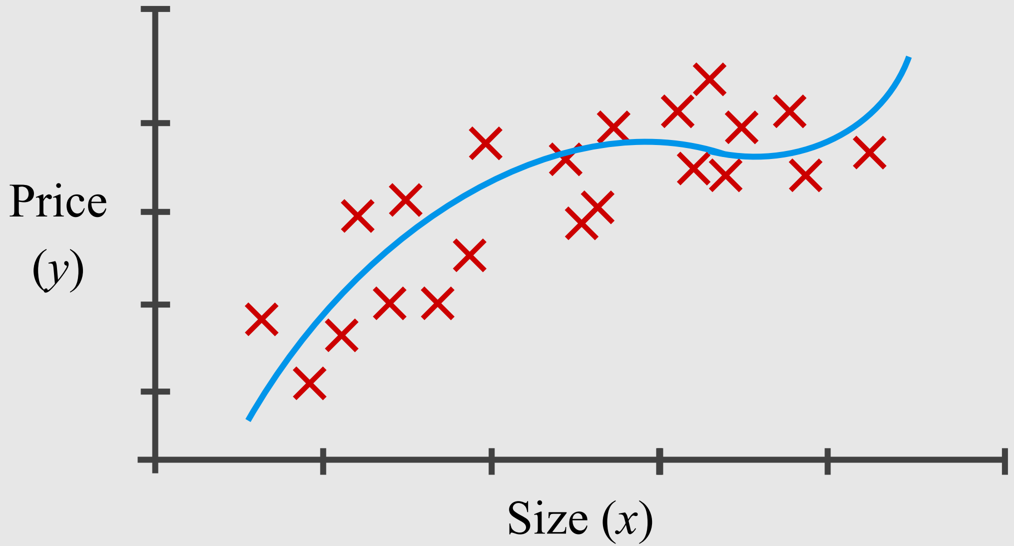
$$x_1 = (\text{size})$$

$$x_2 = (\text{size})^2$$

$$x_3 = (\text{size})^3$$

$$\begin{aligned} h_\theta(x) &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \\ &= \theta_0 + \theta_1(\text{size}) + \theta_2(\text{size})^2 + \theta_3(\text{size})^3 \end{aligned}$$

Polynomial Regression



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

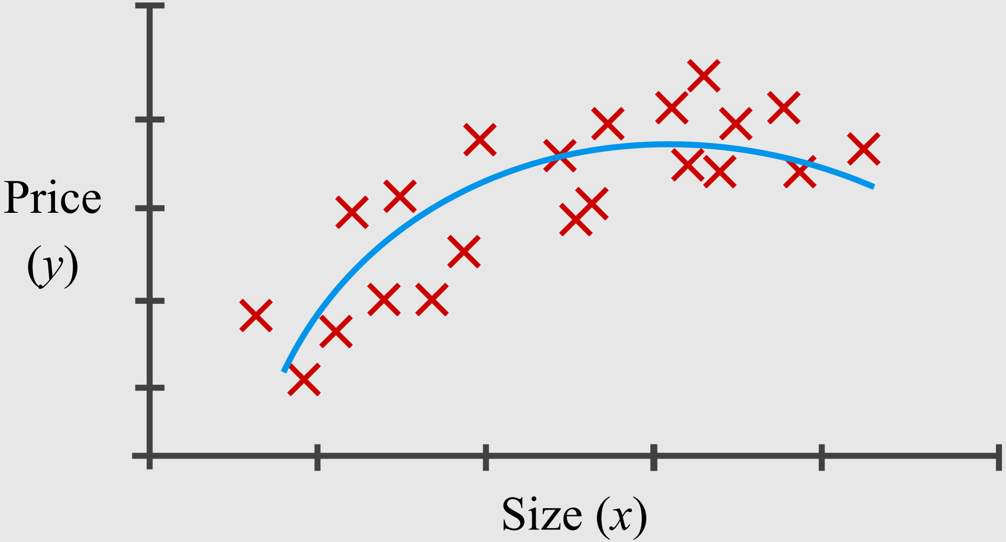
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$= \theta_0 + \theta_1(\text{size}) + \theta_2(\text{size})^2 + \theta_3(\text{size})^3$$

$x_1 = (\text{size})$ $:1\text{–}1{,}000$

$x_2 = (\text{size})^2$ $:1\text{–}1{,}000{,}000$

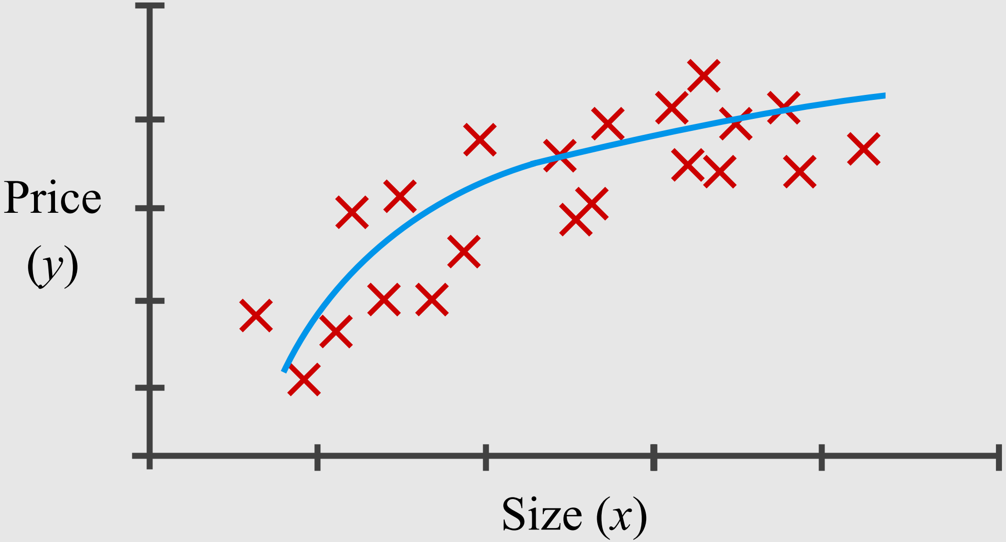
$x_3 = (\text{size})^3$ $:1\text{–}10^9$

Choice of Features



$$h_\theta(x) = \theta_0 + \theta_1(\text{ size }) + \theta_2(\text{ size })^2$$

Choice of Features

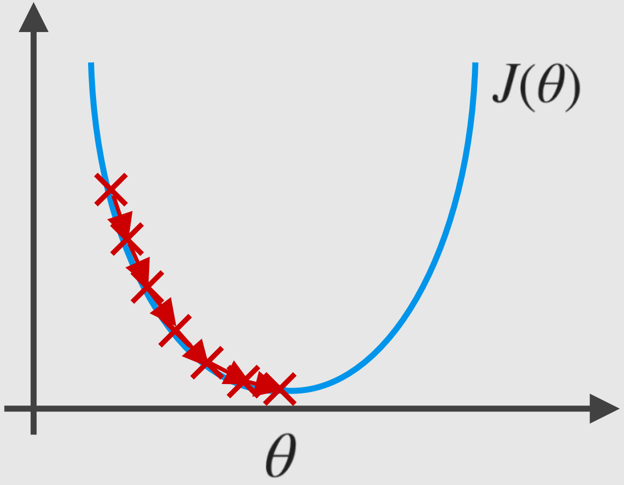


$$h_\theta(x) = \theta_0 + \theta_1(\text{ size }) + \theta_2(\text{ size })^2$$

$$h_\theta(x) = \theta_0 + \theta_1(\text{ size }) + \theta_2\sqrt{(\text{ size })}$$

Normal Equation

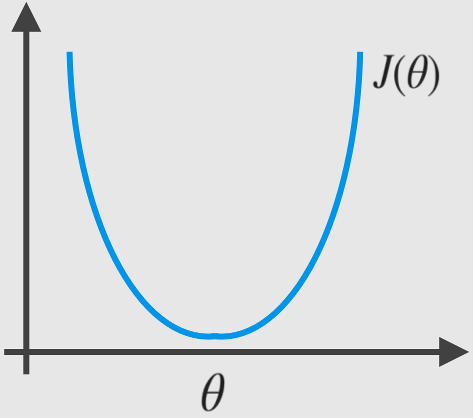
Gradient Descent



Normal equation: Method to solve θ **analytically**.

Intuition: If 1D ($\theta \in \mathbb{R}$)

$$J(\theta) = a\theta^2 + b\theta + c$$

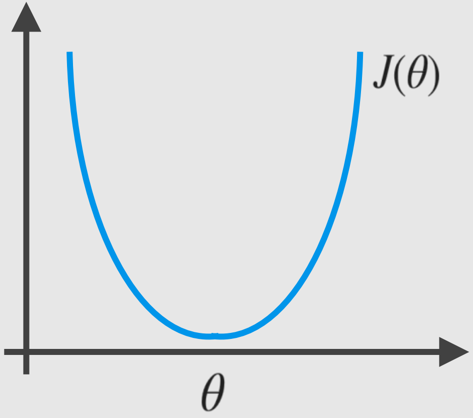


Intuition: If 1D ($\theta \in \mathbb{R}$)

$$J(\theta) = a\theta^2 + b\theta + c$$

$$\frac{d}{d\theta} J(\theta) = \cdots = 0$$

Solve for θ

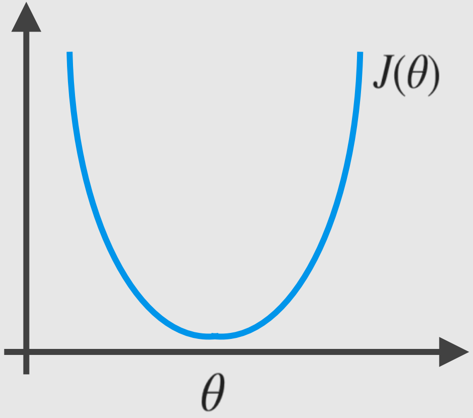


Intuition: If 1D ($\theta \in \mathbb{R}$)

$$J(\theta) = a\theta^2 + b\theta + c$$

$$\frac{d}{d\theta} J(\theta) = \cdots = 0$$

Solve for θ



$\theta \in \mathbb{R}^{n+1}$

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

$$\frac{\partial}{\partial \theta_j} J(\theta) = \cdots = 0$$

Solve for $\theta_0, \theta_1, \ldots, \theta_n$

Examples: $m = 4$.

Size (feet^2) x_1	**Number of bedrooms** x_2	**Number of floors** x_3	**Age of home (years)** x_4	**Price (\$) in 1000's** y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

Examples: $m = 4$.

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

Examples: $m = 4$.

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

Examples: $m = 4$.

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

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

Examples: $m = 4$.

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

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

Examples: $m = 4$.

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

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

Examples: $m = 4$.

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price (\$) in 1000's y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}_{m \times (n+1)} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}_{m}$$

Examples: $m = 4$.

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

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}_{m \times (n+1)} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}_{m}$$

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

***m* examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and *n* features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

***m* examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and *n* features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}$$

m **examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{---} \; (x^{(1)})^{\mathrm{T}} \; \text{---} \\ \\ \\ \\ \end{bmatrix}$$

***m* examples** $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$ **and *n* features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{---} \ (x^{(1)})^{\mathrm{T}} \ \text{---} \\ \text{---} \ (x^{(2)})^{\mathrm{T}} \ \text{---} \\ \\ \\ \end{bmatrix}$$

m examples $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and n features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{——} & (x^{(1)})^{\mathrm{T}} & \text{——} \\ \text{——} & (x^{(2)})^{\mathrm{T}} & \text{——} \\ \text{——} & \vdots & \text{——} \\ \text{——} & (x^{(m)})^{\mathrm{T}} & \text{——} \end{bmatrix}$$

m examples $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and n features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{——} & (x^{(1)})^{\mathrm{T}} & \text{——} \\ \text{——} & (x^{(2)})^{\mathrm{T}} & \text{——} \\ \text{——} & \vdots & \text{——} \\ \text{——} & (x^{(m)})^{\mathrm{T}} & \text{——} \end{bmatrix}$$

E.g. $x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$

m **examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{---} & (x^{(1)})^{\mathrm{T}} & \text{---} \\ \text{---} & (x^{(2)})^{\mathrm{T}} & \text{---} \\ \text{---} & \vdots & \text{---} \\ \text{---} & (x^{(m)})^{\mathrm{T}} & \text{---} \end{bmatrix}$$

E.g. $x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$ $\quad X = \begin{bmatrix} 1 & x_1^{(1)} \\ \vdots & \vdots \\ 1 & x_m^{(1)} \end{bmatrix}_{m \times 2}$

m **examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and** n **features**

$$X = \begin{bmatrix} \text{——} & (x^{(1)})^{\mathrm{T}} & \text{——} \\ \text{——} & (x^{(2)})^{\mathrm{T}} & \text{——} \\ \text{——} & \vdots & \text{——} \\ \text{——} & (x^{(m)})^{\mathrm{T}} & \text{——} \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

$$\theta = (X^T X)^{-1} X^T y$$

$$\theta = (X^T X)^{-1} X^T y$$

$$\theta = (X^T X)^{-1} X^T y$$

$(X^T X)^{-1}$ is inverse of matrix $X^T X$.

$$\theta = (X^T X)^{-1} X^T y$$

$(X^T X)^{-1}$ is inverse of matrix $X^T X$.

Deriving the Normal Equation using matrix calculus …

👉 https://ayearofai.com/rohan-3-deriving-the-normal-equation-using-matrix-calculus-1a1b16f65dda

$$\theta = (X^T X)^{-1} X^T y$$

$(X^T X)^{-1}$ is inverse of matrix $X^T X$.

Deriving the Normal Equation using matrix calculus …

👉 https://ayearofai.com/rohan-3-deriving-the-normal-equation-using-matrix-calculus-1a1b16f65dda

What if $X^T X$ is noninvertible?

What if $X^T X$ is noninvertible?

The common causes might be having :

- Redundant features, where two features are very closely related (i.e. they are linearly dependent).
- Too many features (e.g. $m \leq n$). In this case, delete some features or use "regularization".

Gradient Descent

- 😞 Need to choose α.
- 😞 Needs many iterations.

Normal Equation

- 😃 No need to choose α.
- 😃 Don’t need to iterate.

m examples and n features

Gradient Descent

- 😞 Need to choose α.
- 😞 Needs many iterations.
- 😃 Works well even when n is large.

Normal Equation

- 😃 No need to choose α.
- 😃 Don't need to iterate.
- 😞 Need to compute $(X^T X)^{-1} \rightarrow O(n^3)$.
- 😞 Slow if n is very large.

m examples and n features

References

Machine Learning Books

- Hands-On Machine Learning with Scikit-Learn and TensorFlow, Chap. 2 & 4
- Pattern Recognition and Machine Learning, Chap. 3
- Machine Learning: a Probabilistic Perspective, Chap. 7

Machine Learning Courses

- https://www.coursera.org/learn/machine-learning, Week 1 & 2