

Machine Learning

Prepared by Dr. Yasser Abdelhamid

DS 620
Machine Learning

Lecture 3





OUTLINE

- ◉ Linear regression
- ◉ Cost function
- ◉ Gradient descent
- ◉ Normal equation

LINEAR REGRESSION

- ❖ Linear regression is a popular algorithm used in machine learning and statistical analysis to model the relationship between a **dependent** variable and **one or more independent variables**.
- ❖ It aims to find a **linear equation** that **best fits** the given data points and can be used to **predict future values**.
- ❖ In regression problems we predict **real-valued** output, where in classification problems we predict **discrete values**.

DATASET, FEATURES, TARGET VARIABLES, HYPOTHESIS

- ❖ The general Idea of ML depends on the availability of sufficient **instances/cases** that we call **Dataset**.
- ❖ A dataset is composed of two parts:
 - **features/predictors** parameters, and
 - **labels/target** parameter.
- ❖ There is a **hypothesis** of predicting target given some values of features/input.

EXAMPLE 1: IRIS DATASET

❖ Classification

Attribute Information:

1. Sepal length in cm
2. Sepal width in cm
3. Petal length in cm
4. Petal width in cm

5. class:

- Iris Setosa
- Iris Versicolour
- Iris Virginica

Sample instances

5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
7.1,3.0,5.9,2.1,Iris-virginica



EXAMPLE 2: AUTO MPG DATASET

❖ Regression

1. mpg: continuous
2. cylinders: multi-valued discrete
3. displacement: continuous
4. horsepower: continuous
5. weight: continuous
6. acceleration: continuous
7. model year: multi-valued discrete
8. origin: multi-valued discrete
9. car name: string (unique for each instance)



18.0	8	307.0	130.0	3504.	12.0	70	1	"chevrolet chevelle malibu"
15.0	8	350.0	165.0	3693.	11.5	70	1	"buick skylark 320"
18.0	8	318.0	150.0	3436.	11.0	70	1	"plymouth satellite"
16.0	8	304.0	150.0	3433.	12.0	70	1	"amc rebel sst"
17.0	8	302.0	140.0	3449.	10.5	70	1	"ford torino"
15.0	8	429.0	198.0	4341.	10.0	70	1	"ford galaxie 500"

NOTATIONS

❖ Instances.

❖ x input features/predictors

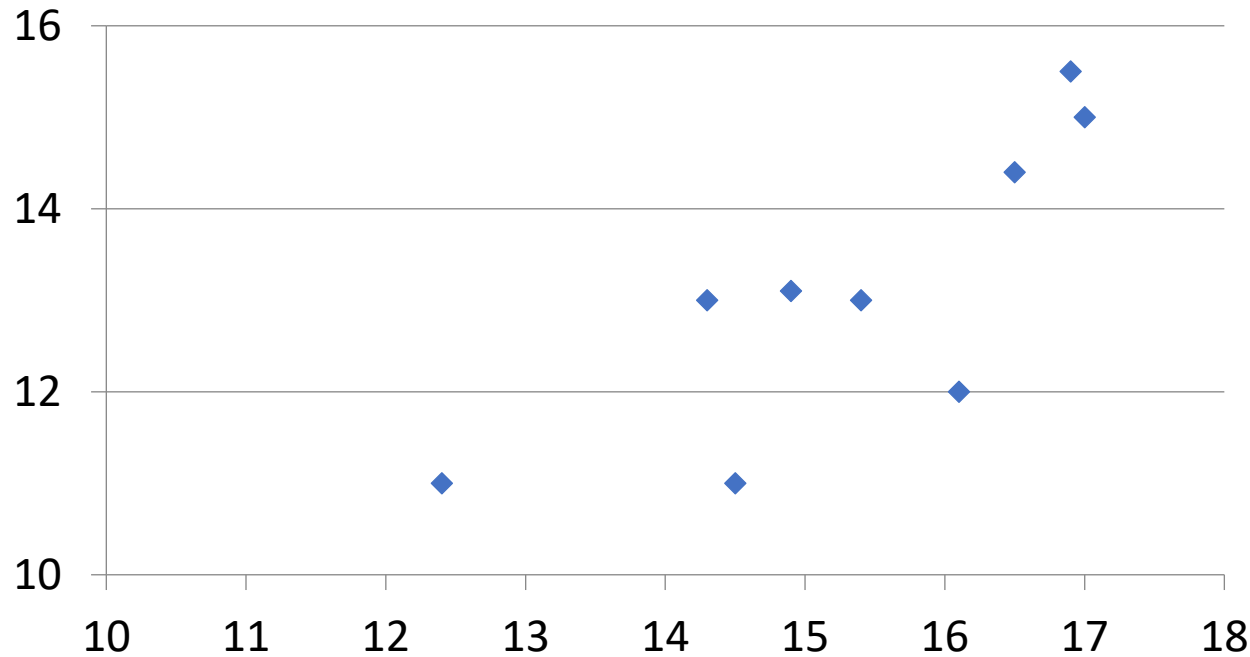
❖ y output/target/label

❖ $(x^{(i)}, y^{(i)})$ row number i in the dataset.

X				Y
5.1	3.5	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.3	3.3	6.0	2.5	Iris-virginica

X	Y
12.4	11.2
14.3	12.5
14.5	12.7
14.9	13.1
16.1	14.1
16.9	14.8
16.5	14.4
15.4	13.4
17	14.9

HOW DOES IT LOOK LIKE?



X	Y
12.4	11
14.3	13
14.5	11
14.9	13.1
16.1	12
16.9	15.5
16.5	14.4
15.4	13
17	15

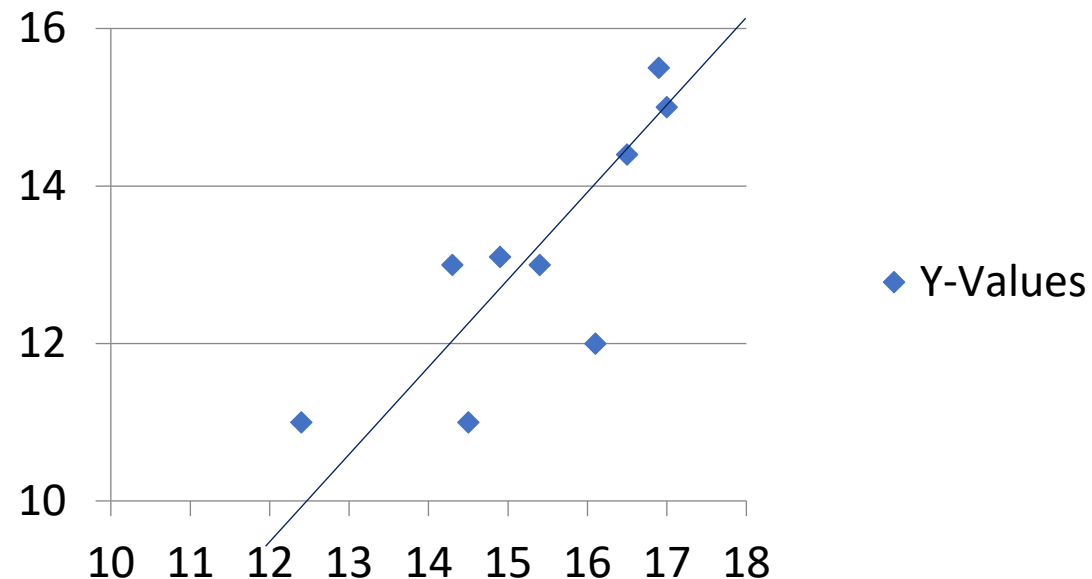
LINEAR REGRESSION

- ❖ It is not always the case that we model the dataset using a linear function.
- ❖ Linear regression with only one input variable is called **Uni-variate (only one independent variable) Linear Regression**
- ❖ it looks like this:

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

LINEAR REGRESSION

❖ To fit the best possible line within our data, we need to know the values of Θ_0 and Θ_1 to make $h(x)$ as close as possible to y .



❖ keep $(h(x) - y) >>>>$ at Minimum

COST FUNCTION

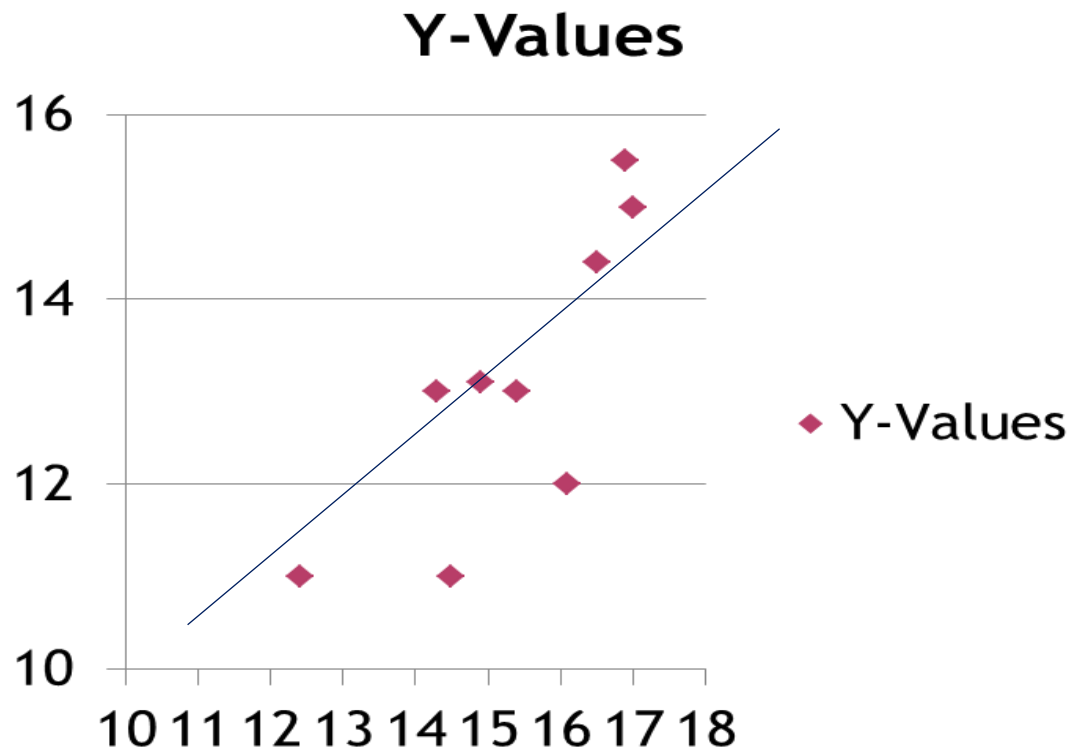
❖ We need to find values of Θ_0 and Θ_1 to keep this function at minimum, this can be accomplished using the cost function as follows.

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

$$h_{\theta}(x) = \Theta_0 + \Theta_1 x$$

x	y	$h_{\theta}(x)$
12.4	11	
14.3	13	
14.5	11	
14.9	13.1	
16.1	12	
16.9	15.5	
16.5	14.4	
15.4	13	
17	15	

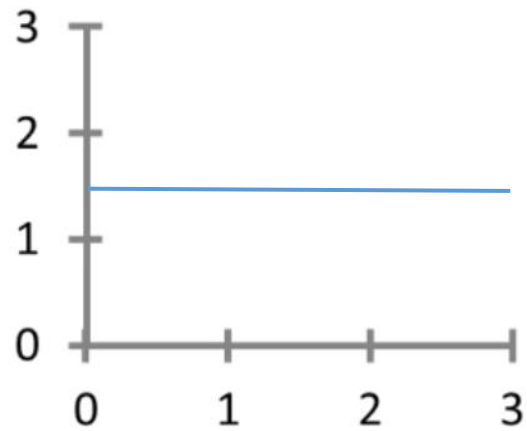
HYPOTHESIS FUNCTION



X	Y	$h_{\theta}(x)$
12.4	11	
14.3	13	
14.5	11	
14.9	13.1	
16.1	12	
16.9	15.5	
16.5	14.4	
15.4	13	
17	15	

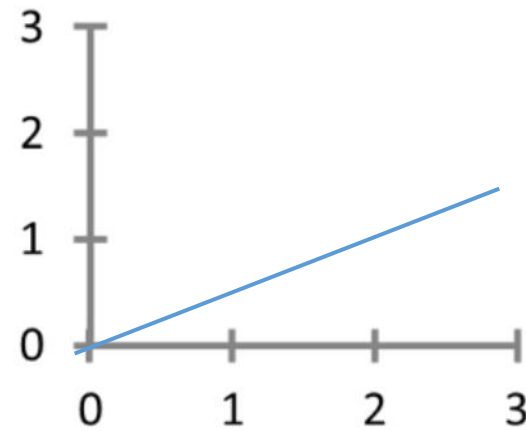
DIFFERENT CHOICES OF θ VALUES

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



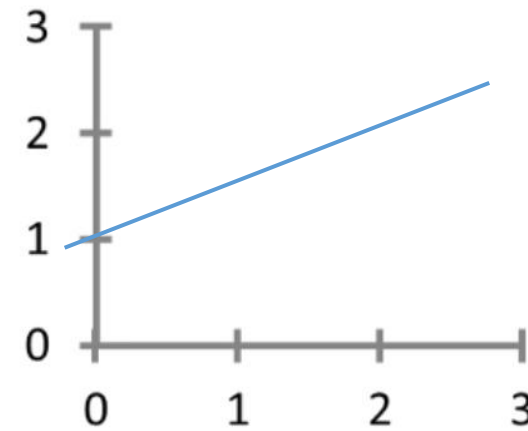
$$\theta_0 = 1.5$$

$$\theta_1 = 0$$



$$\theta_0 = 0$$

$$\theta_1 = 0.5$$



$$\theta_0 = 1$$

$$\theta_1 = 0.5$$

HYPOTHESIS FUNCTION AND COST FUNCTION

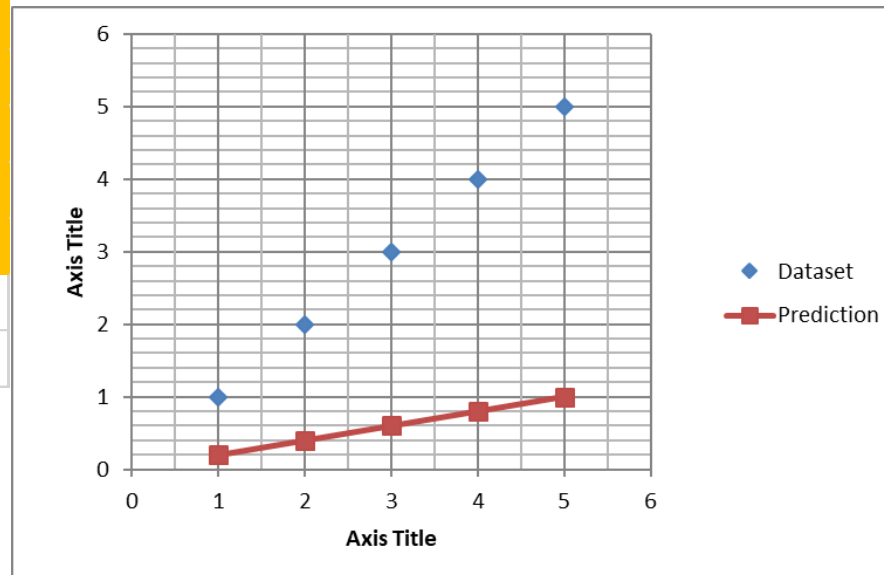
Dataset

i	x	y	h(x)	(h(x)-y)	(h(x)-y) ²
1	1	1	0.2	-0.8	0.64
2	2	2	0.4	-1.6	2.56
3	3	3	0.6	-2.4	5.76
4	4	4	0.8	-3.2	10.24
5	5	5	1	-4	16
MSE					3.52

θ ₁	J(θ ₁)
0.2	3.52

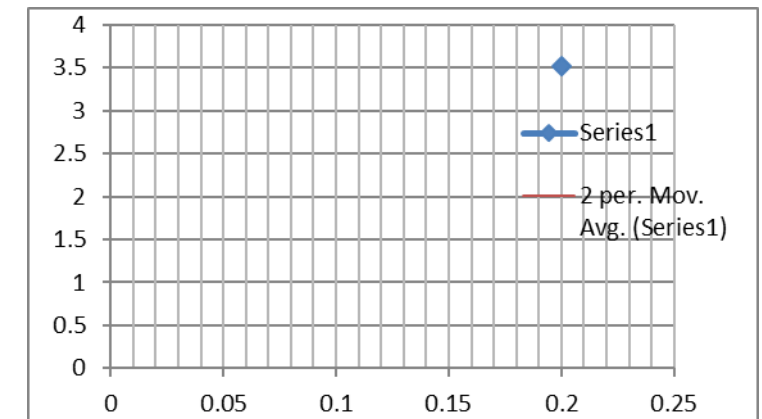
Hypothesis function

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



Cost function

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



HYPOTHESIS FUNCTION AND COST FUNCTION

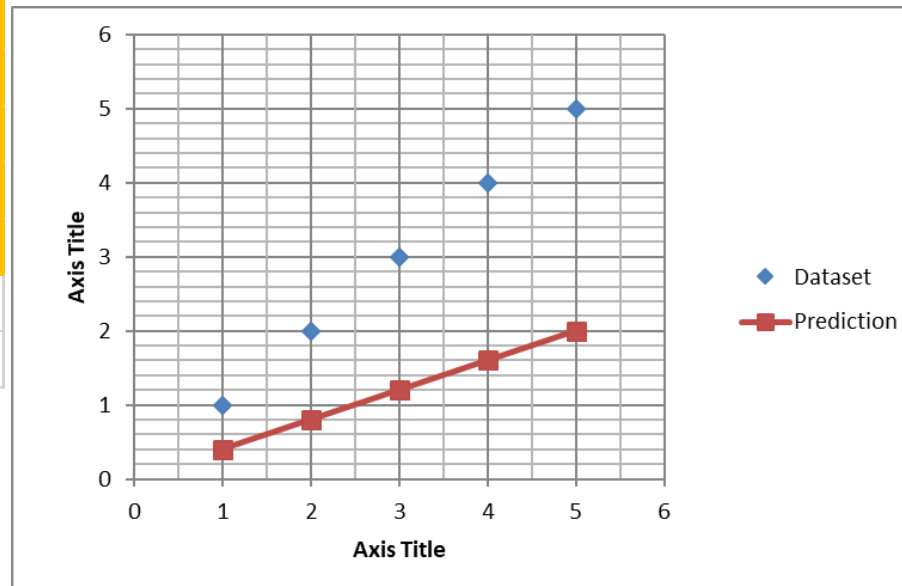
Dataset

i	x	y	$h(x)$	$(h(x)-y)$	$(h(x)-y)^2$
1	1	1	0.4	-0.6	0.36
2	2	2	0.8	-1.2	1.44
3	3	3	1.2	-1.8	3.24
4	4	4	1.6	-2.4	5.76
5	5	5	2	-3	9
MSE					1.98

θ_1	$J(\theta_1)$
0.2	3.52
0.4	1.98

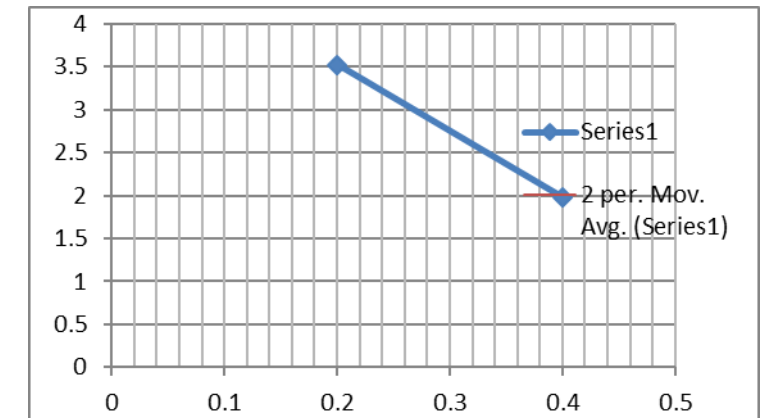
Hypothesis function

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



Cost function

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



HYPOTHESIS FUNCTION AND COST FUNCTION

Dataset

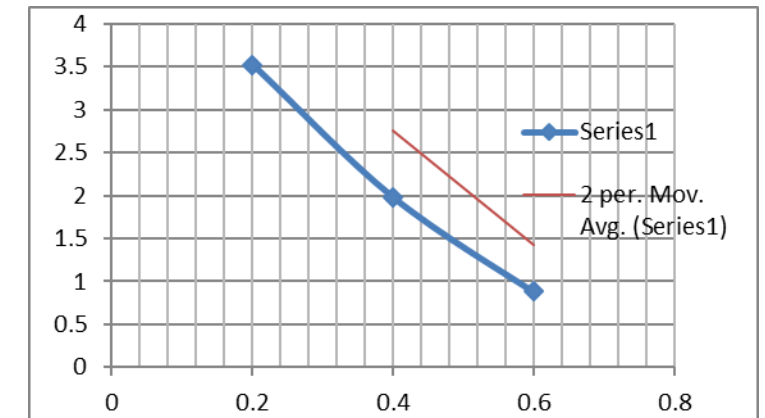
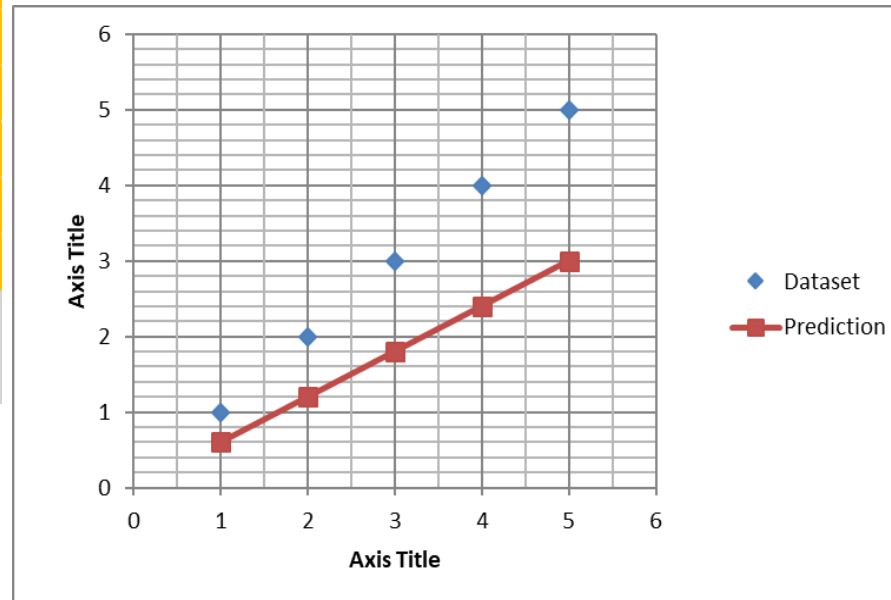
Hypothesis
function

Cost function

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

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

i	x	y	h(x)	(h(x)-y)	(h(x)-y) ²
1	1	1	0.6	-0.4	0.16
2	2	2	1.2	-0.8	0.64
3	3	3	1.8	-1.2	1.44
4	4	4	2.4	-1.6	2.56
5	5	5	3	-2	4
MSE					0.88



θ_1	$J(\theta_1)$
0.2	3.52
0.4	1.98
0.6	0.88

HYPOTHESIS FUNCTION AND COST FUNCTION

Dataset

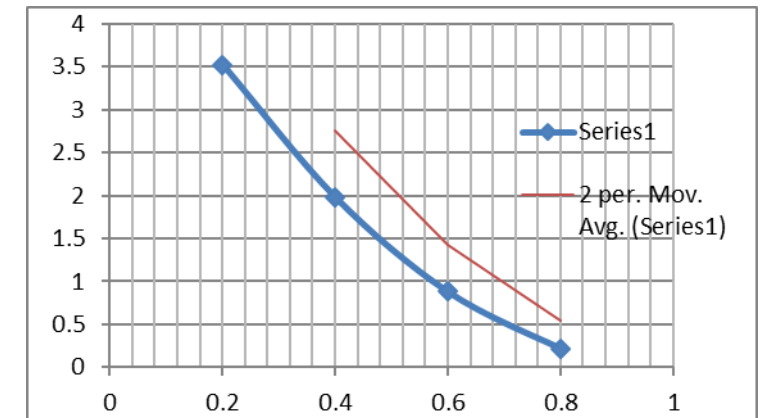
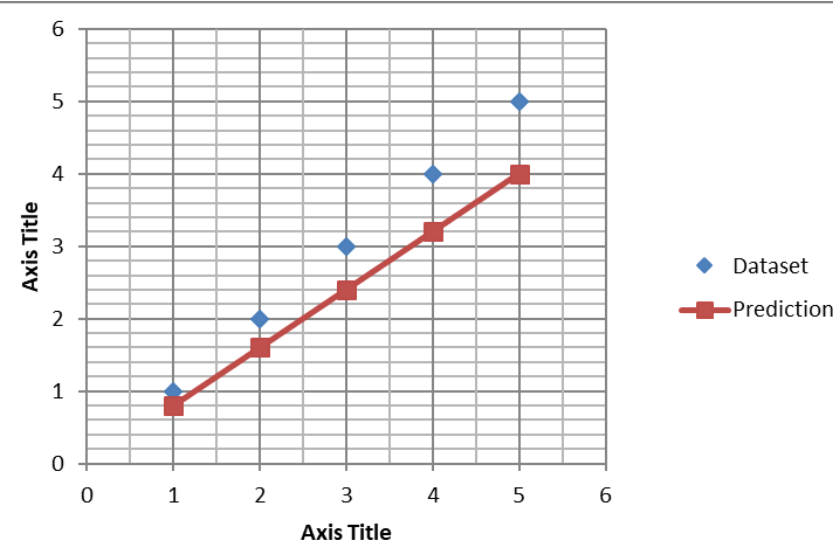
Hypothesis
function

Cost function

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

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

i	x	y	h(x)	(h(x)-y)	(h(x)-y) ²
1	1	1	0.8	-0.2	0.04
2	2	2	1.6	-0.4	0.16
3	3	3	2.4	-0.6	0.36
4	4	4	3.2	-0.8	0.64
5	5	5	4	-1	1



MSE **0.22**

θ_1	$J(\theta_1)$
0.2	3.52
0.4	1.98
0.6	0.88
0.8	0.22

HYPOTHESIS FUNCTION AND COST FUNCTION

Dataset

i	x	y	$h(x)$	$(h(x)-y)$	$(h(x)-y)^2$
1	1	1	1	0	0
2	2	2	2	0	0
3	3	3	3	0	0
4	4	4	4	0	0
5	5	5	5	0	0

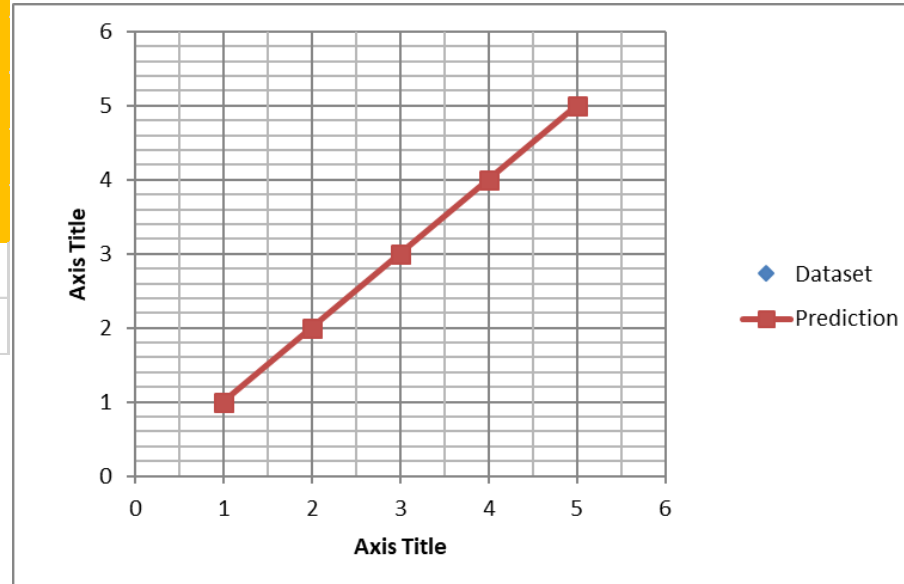
MSE

0

θ_1	$J(\theta_1)$
0.2	3.52
0.4	1.98
0.6	0.88
0.8	0.22
1	0

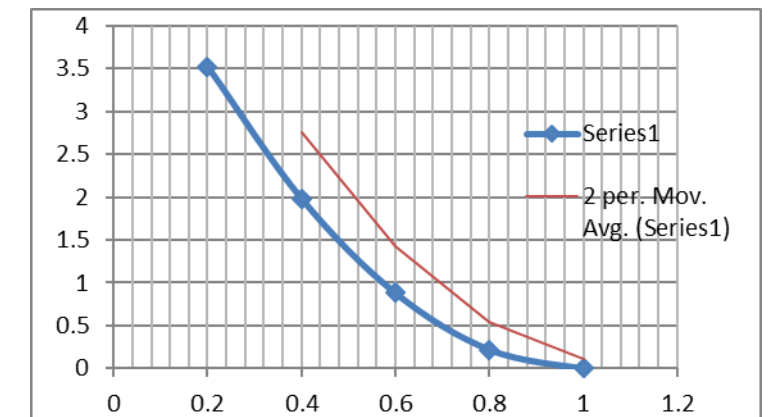
Hypothesis function

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



Cost function

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



HYPOTHESIS FUNCTION AND COST FUNCTION

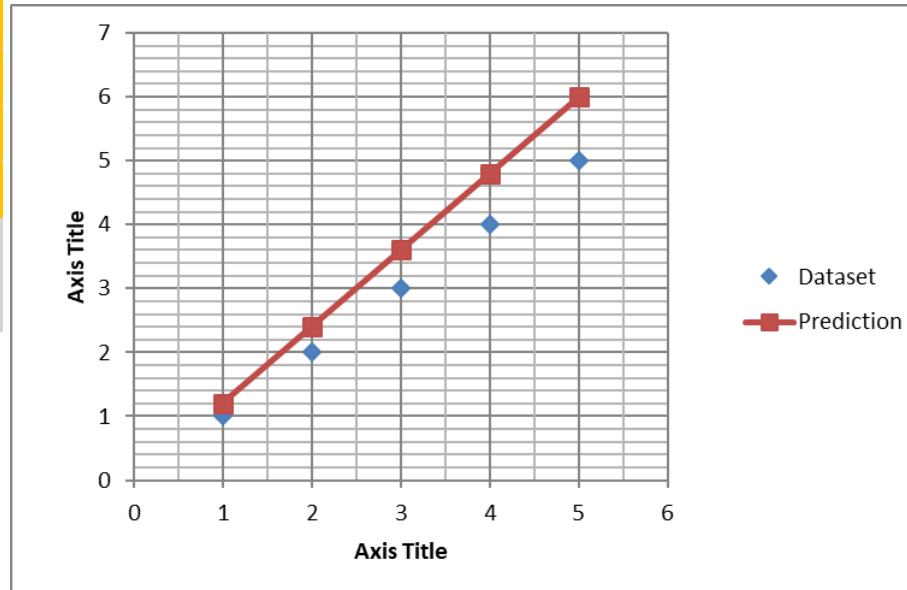
Dataset

i	x	y	h(x)	(h(x)-y)	(h(x)-y) ²
1	1	1	1.2	0.2	0.04
2	2	2	2.4	0.4	0.16
3	3	3	3.6	0.6	0.36
4	4	4	4.8	0.8	0.64
5	5	5	6	1	1
MSE					0.22

θ ₁	J(θ ₁)
0.2	3.52
0.4	1.98
0.6	0.88
0.8	0.22
1	0
1.2	0.22

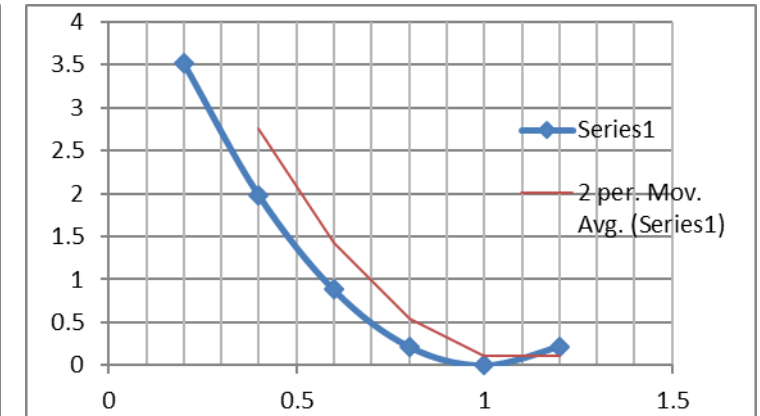
Hypothesis function

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



Cost function

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



HYPOTHESIS FUNCTION AND COST FUNCTION

Dataset

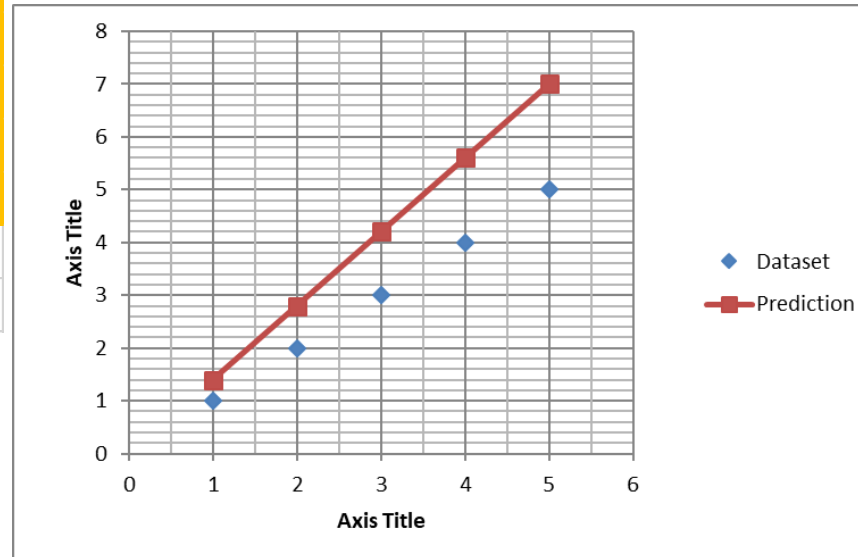
i	x	y	h(x)	(h(x)-y)	(h(x)-y) ²
1	1	1	1.4	0.4	0.16
2	2	2	2.8	0.8	0.64
3	3	3	4.2	1.2	1.44
4	4	4	5.6	1.6	2.56
5	5	5	7	2	4

MSE 0.88

θ ₁	J(θ ₁)
0.2	3.52
0.4	1.98
0.6	0.88
0.8	0.22
1	0
1.2	0.22
1.4	0.88

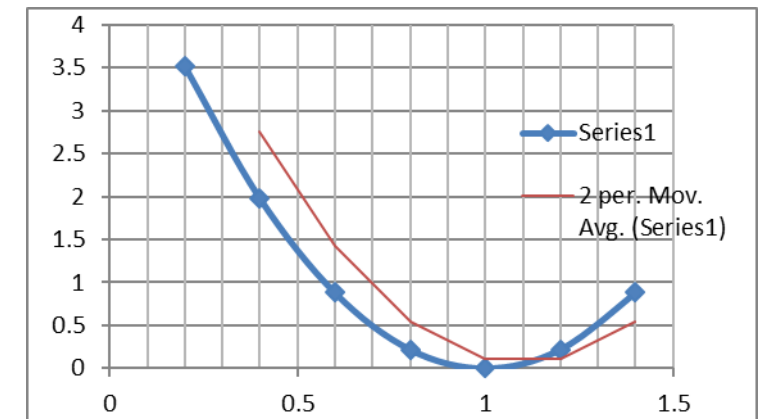
Hypothesis function

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



Cost function

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



HYPOTHESIS FUNCTION AND COST FUNCTION

Dataset

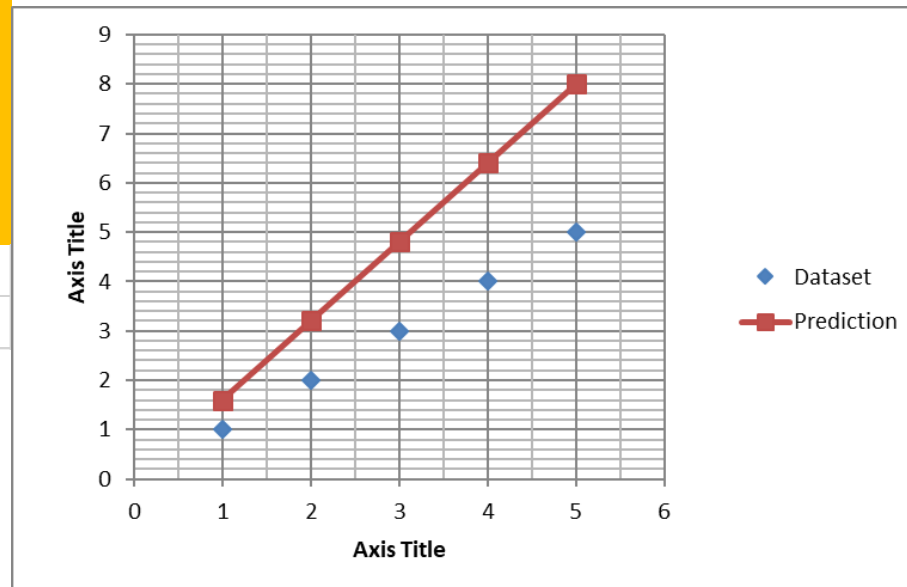
i	x	y	$h(x)$	$(h(x)-y)$	$(h(x)-y)^2$
1	1	1	1.6	0.6	0.36
2	2	2	3.2	1.2	1.44
3	3	3	4.8	1.8	3.24
4	4	4	6.4	2.4	5.76
5	5	5	8	3	9

MSE 1.98

θ_1	$J(\theta_1)$
0.2	3.52
0.4	1.98
0.6	0.88
0.8	0.22
1	0
1.2	0.22
1.4	0.88
1.6	1.98

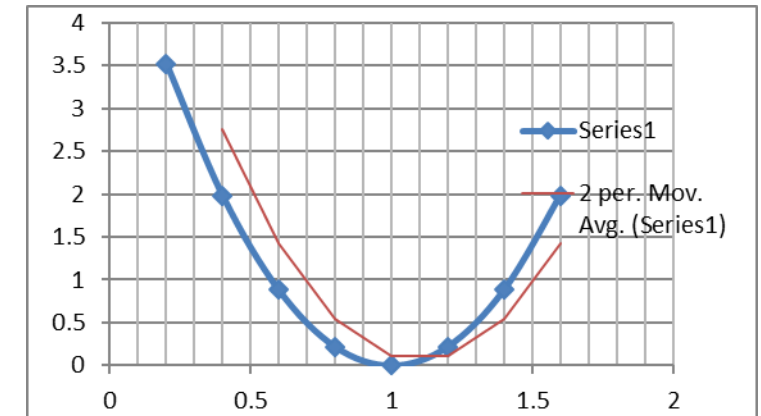
Hypothesis function

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



Cost function

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



HYPOTHESIS FUNCTION AND COST FUNCTION

Dataset

Hypothesis
function

Cost function

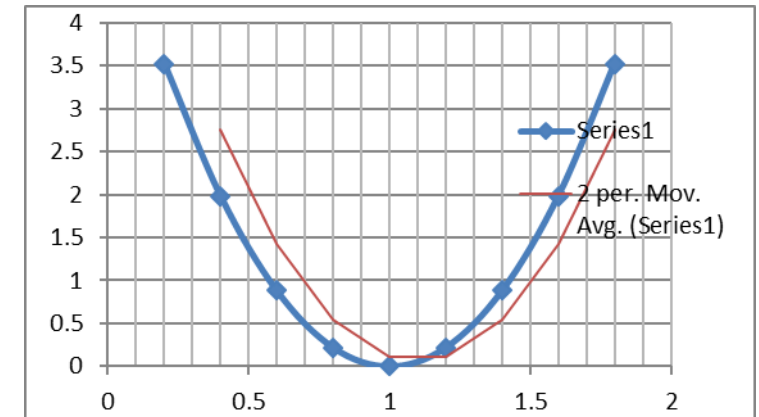
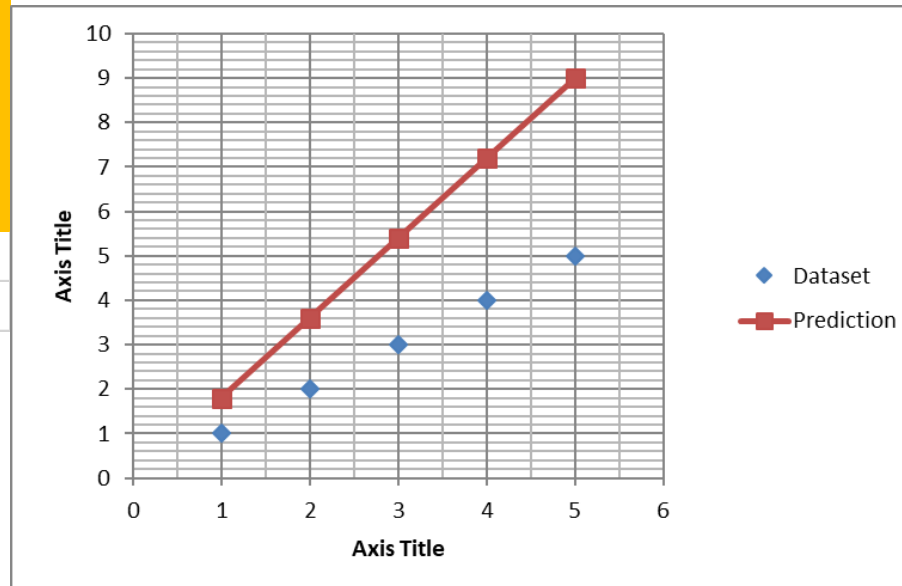
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

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

i	x	y	h(x)	(h(x)-y)	(h(x)-y) ²
1	1	1	1.8	0.8	0.64
2	2	2	3.6	1.6	2.56
3	3	3	5.4	2.4	5.76
4	4	4	7.2	3.2	10.24
5	5	5	9	4	16

MSE 3.52

θ_1	$J(\theta_1)$
0.2	3.52
0.4	1.98
0.6	0.88
0.8	0.22
1	0
1.2	0.22
1.4	0.88
1.6	1.98
1.8	3.52



GRADIENT DESCENT ALGORITHM

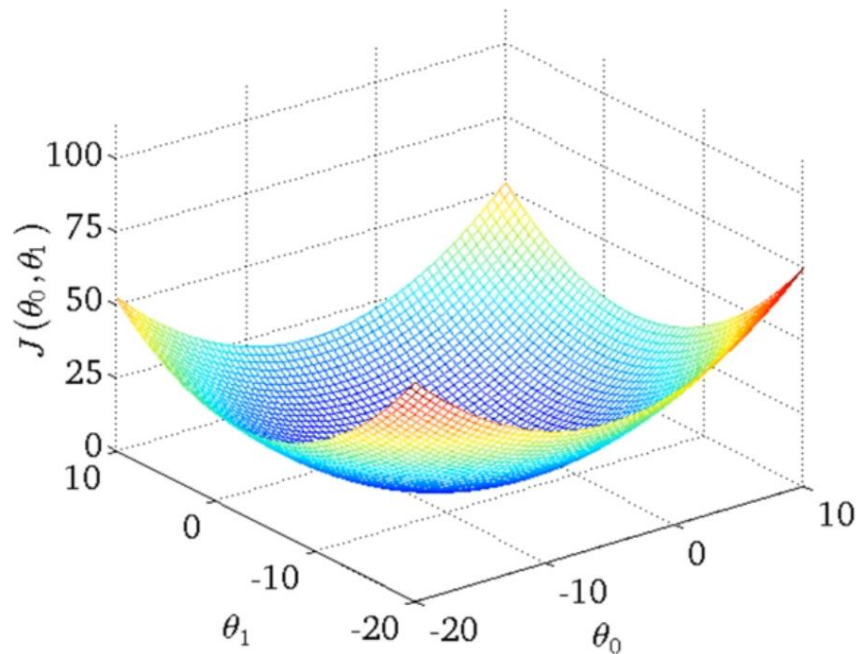
Have some function $J(\theta_0, \theta_1)$

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

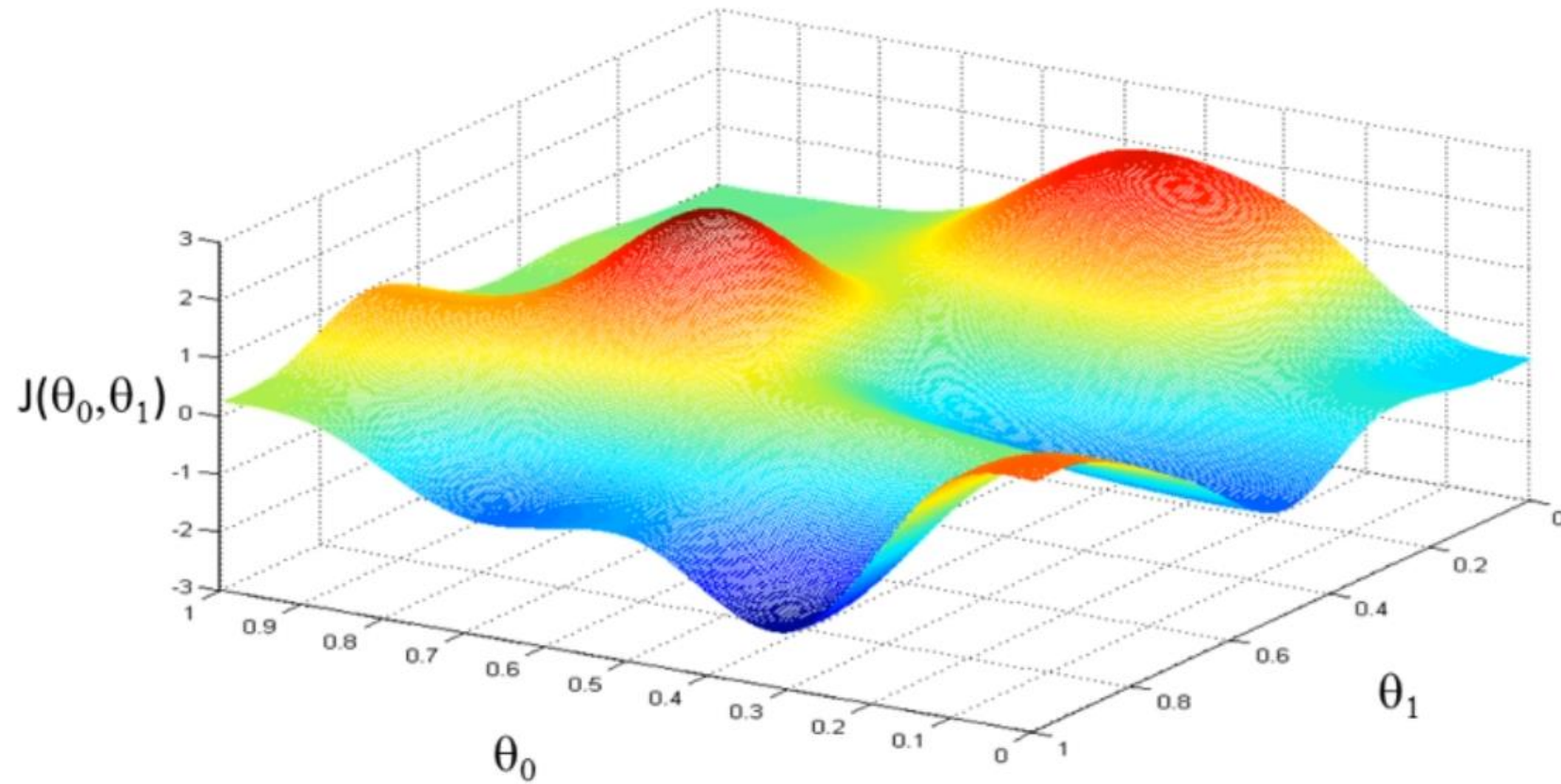
Outline:

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

GRADIENT DESCENT WITH TWO PARAMETERS



GRADIENT DESCENT ALGORITHM



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 $j = 1$)
}

Correct: Simultaneous update

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

Incorrect:

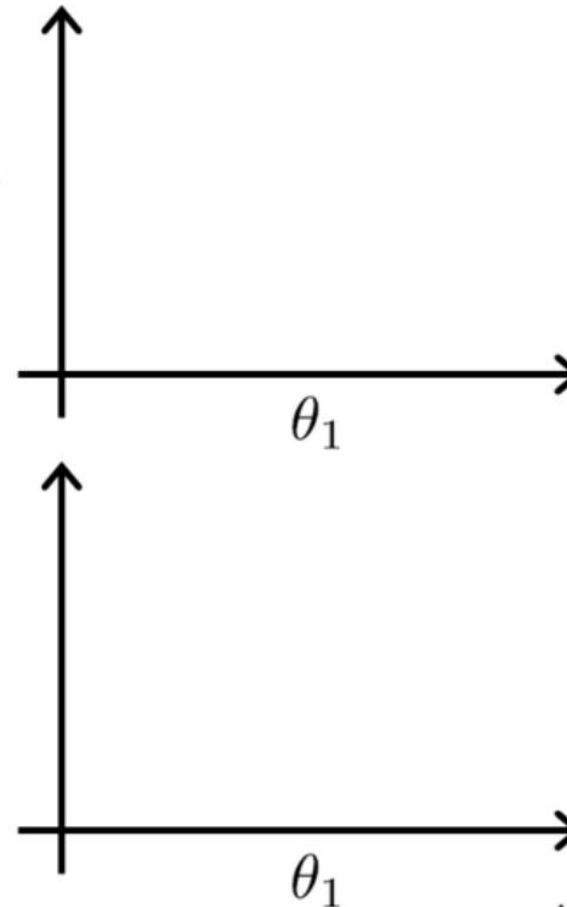
```
temp0 :=  $\theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$   
 $\theta_0 :=$  temp0  
temp1 :=  $\theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$   
 $\theta_1 :=$  temp1
```

GRADIENT DESCENT ALGORITHM

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

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

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

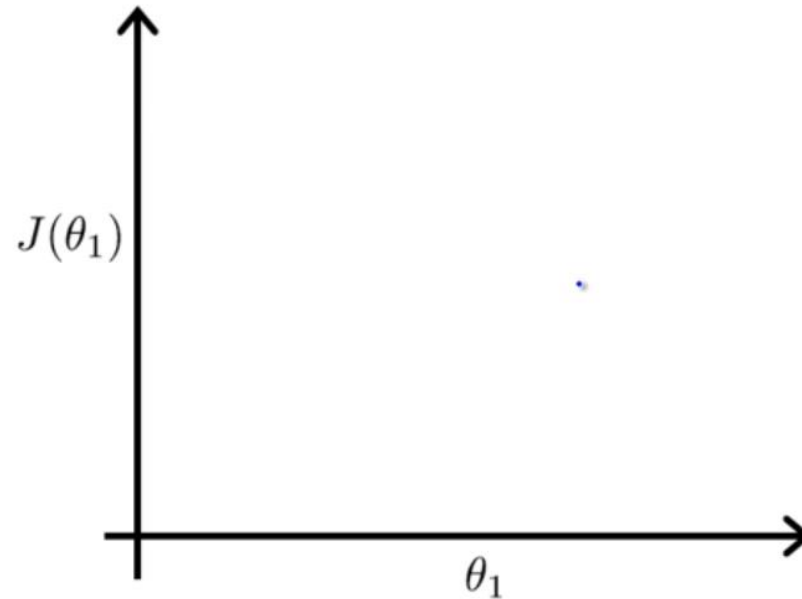


GRADIENT DESCENT ALGORITHM

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

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 $j = 1$)
}

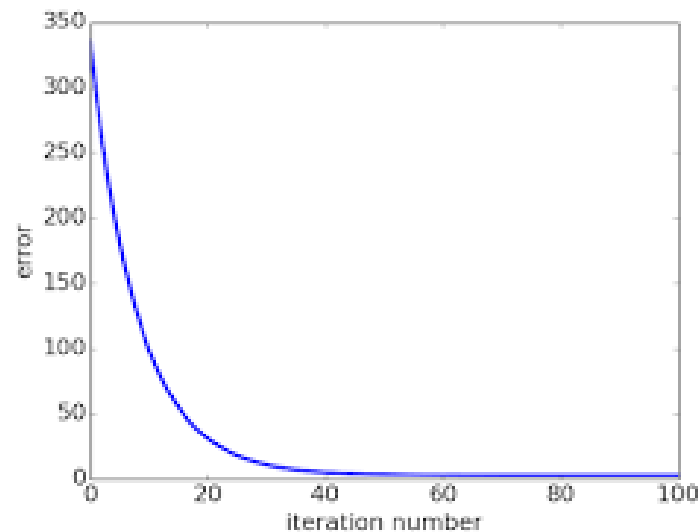
α is the learning rate.

GRADIENT DESCENT ALGORITHM

$$\begin{aligned} &\text{Repeat } \{ \\ &\quad \theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial \theta_0} J(\theta)} \\ &\quad \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x^{(i)} \\ &\quad \text{(simultaneously update } \theta_0, \theta_1) \\ &\} \end{aligned}$$
$$h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

CONVERGENCE

- ❖ Gradient descent is an **iterative algorithm**.
- ❖ It has to keep repeating the process until the cost is **almost constant**, or **has minor changes**.



GENERAL FORM

$$\diamond h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)}$$

$\diamond$ by adding the term x_0 that always has the value 1

$$\diamond h_{\theta}(x^{(i)}) = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)}$$

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
 $j = 0, \dots, n$)

}

i	x_0	x_1	y
1	1	12.4	11
2	1	14.3	13
3	1	14.5	11
4	1	14.9	13.1
5	1	16.1	12
6	1	16.9	15.5
7	1	16.5	14.4
8	1	15.4	13
9	1	17	15

NORMAL EQUATION

❖ The normal equation is a mathematical formula used in linear regression to **find the optimal values for the coefficients (weights)** of the linear regression model.

NORMAL EQUATION

❖ Our hypothesis function:

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$$

❖ Our goal is to minimize the cost function:

$$J(\theta_{0...n}) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

NORMAL EQUATION

❖ We can represent Θ as a vector:

$$\begin{pmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{pmatrix} \in \mathbb{R}^{n+1}$$

❖ The hypothesis function will become: $h_{\theta}(x) = \theta^T x$

❖ The parameters vector: $\theta = (X^T X)^{-1} X^T y$

FEATURE VECTOR

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ \vdots \\ x^{(m)} \end{bmatrix}$$

$$X = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & \cdot & \cdot & \cdot & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \cdot & \cdot & \cdot & x_n^{(2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_0^{(m)} & x_1^{(m)} & \cdot & \cdot & \cdot & x_n^{(m)} \end{bmatrix}$$

is the feature vector and

TARGET VECTOR

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \text{Target vector}$$

PARAMETERS VECTOR

The parameters vector: $\theta = (X^T X)^{-1} X^T y$ $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \cdot \\ \cdot \\ \cdot \\ \theta_n \end{bmatrix}$ is the parameter vector

NORMAL EQUATION VS GRADIENT DESCENT

Normal Equation

- **Directly** solve for the optimum values of the parameters Θ .
- No need for setting a learning rate.
- Not suitable for large number of features.

Gradient descent

- In Gradient descent we start with some value of Θ then we make a number of **iterations** to converge.
- We have to set a suitable value for the learning rate α
- Can be used with large number of features.



SUMMARY

- ◉ Linear regression
- ◉ Hypothesis function
- ◉ Cost function
- ◉ Convergence
- ◉ Gradient descent
- ◉ Normal equation
- ◉ Gradient descent vs Normal equation



Thank You

midocean.edu.km

