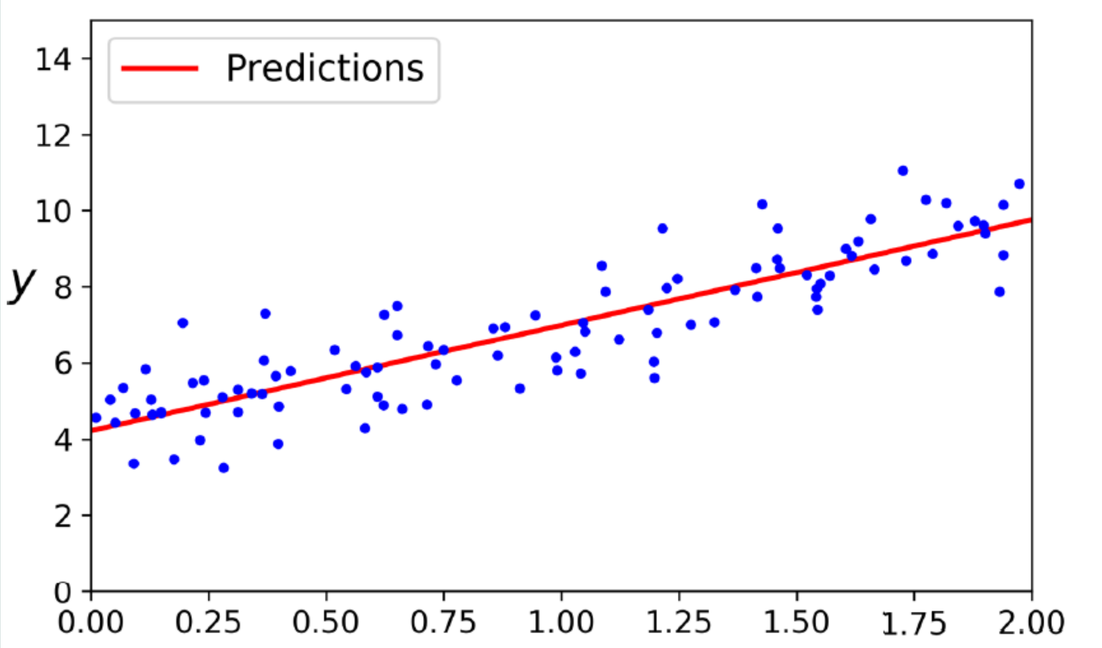
Mustafa Shiple

# Linear Regression

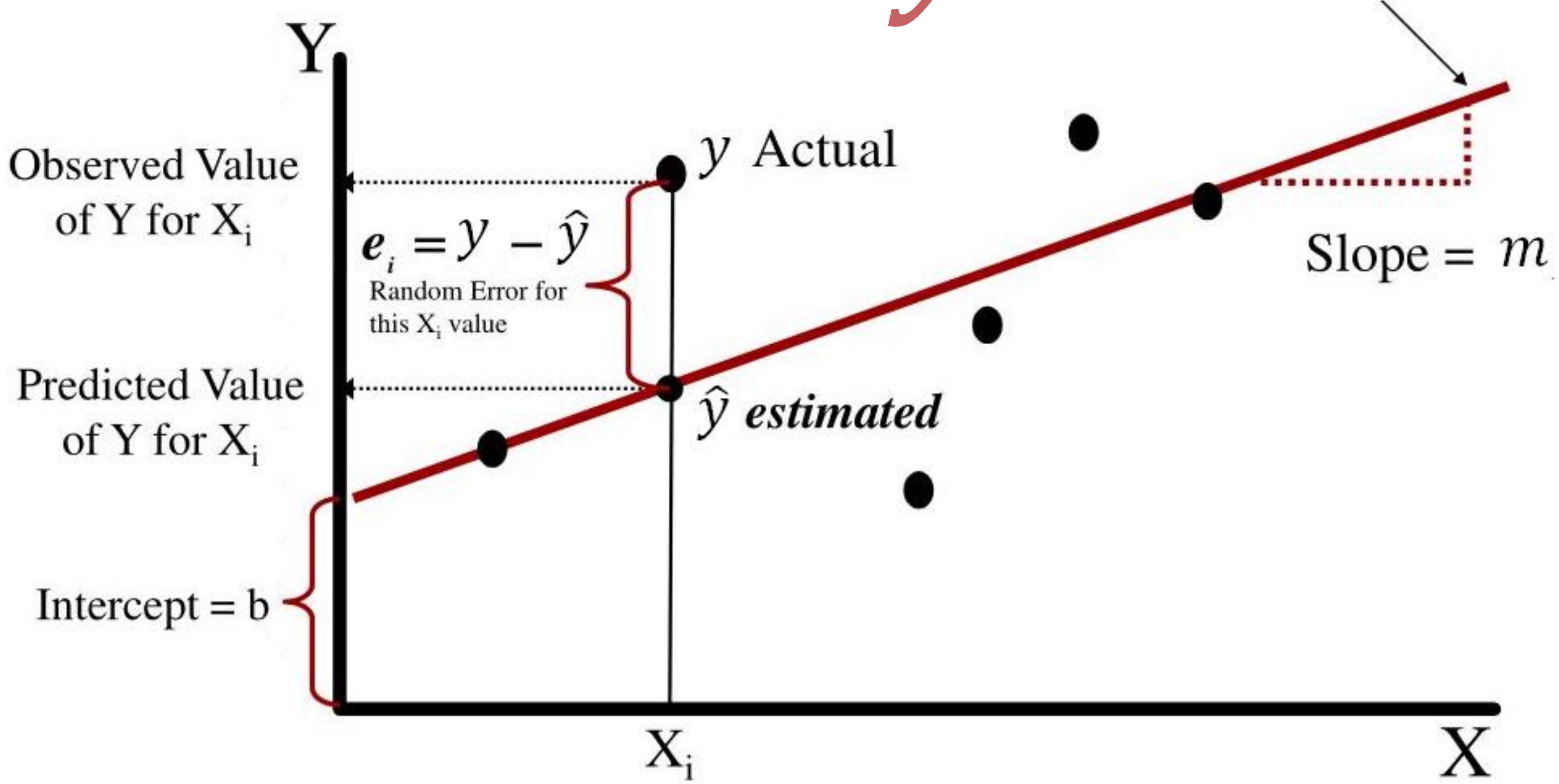
# Introduction

- "fitting a straight line"
- a statistical analysis method to determine the quantitative relationships between two or more variables through regression analysis in mathematical statistics.



# Univariate linear regression (Simple LR)

$$\hat{y} = mx + b$$

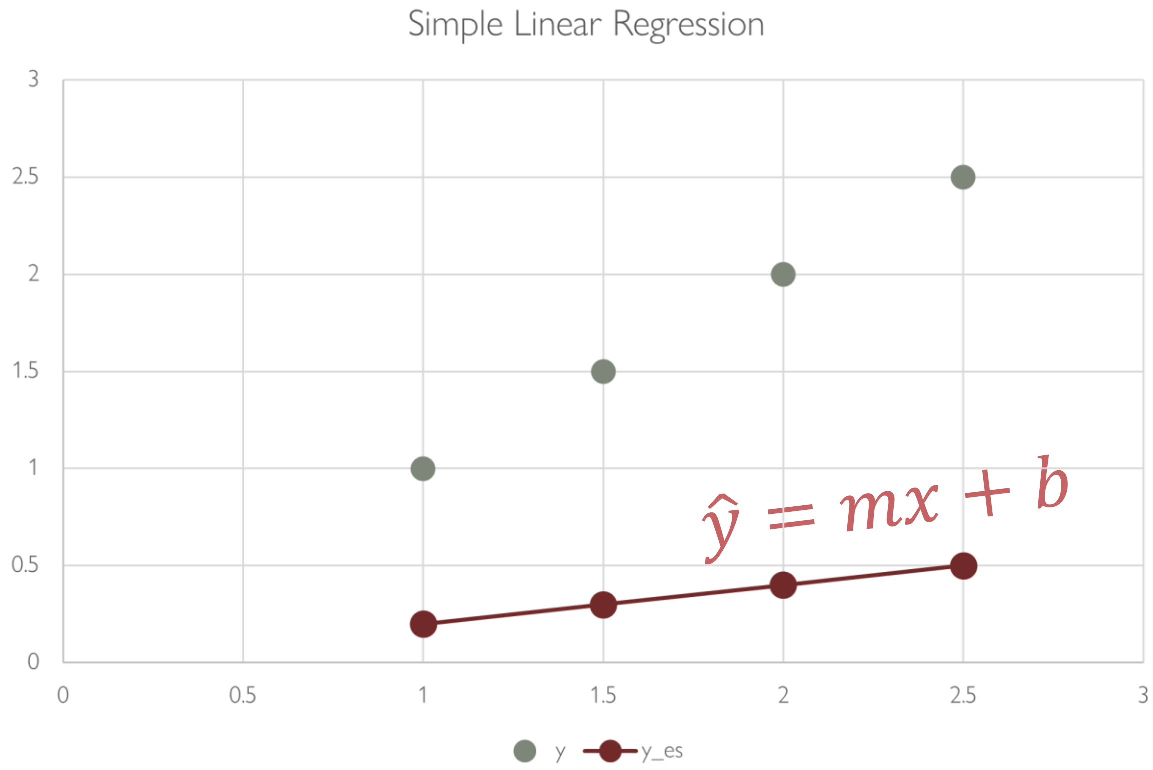


# loss function (**Mean Squared Error (MSE)**)

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$


$$J(w) = \frac{1}{2*4}\sum_{i=1}^{4}\{\,(1-0.2)^2, (1.5-0.3)^2, (2-0.4)^2, (2.5-0.5)^2\}$$

| base | slop | | |
|---|---|---|---|
| 0 | 0.2 | | |
| x | y | y_es | square error |
| 1 | 1 | 0.2 | 0.64 |
| 1.5 | 1.5 | 0.3 | 1.44 |
| 2 | 2 | 0.4 | 2.56 |
| 2.5 | 2.5 | 0.5 | 4 |
| mean square error | | | 1.08 |



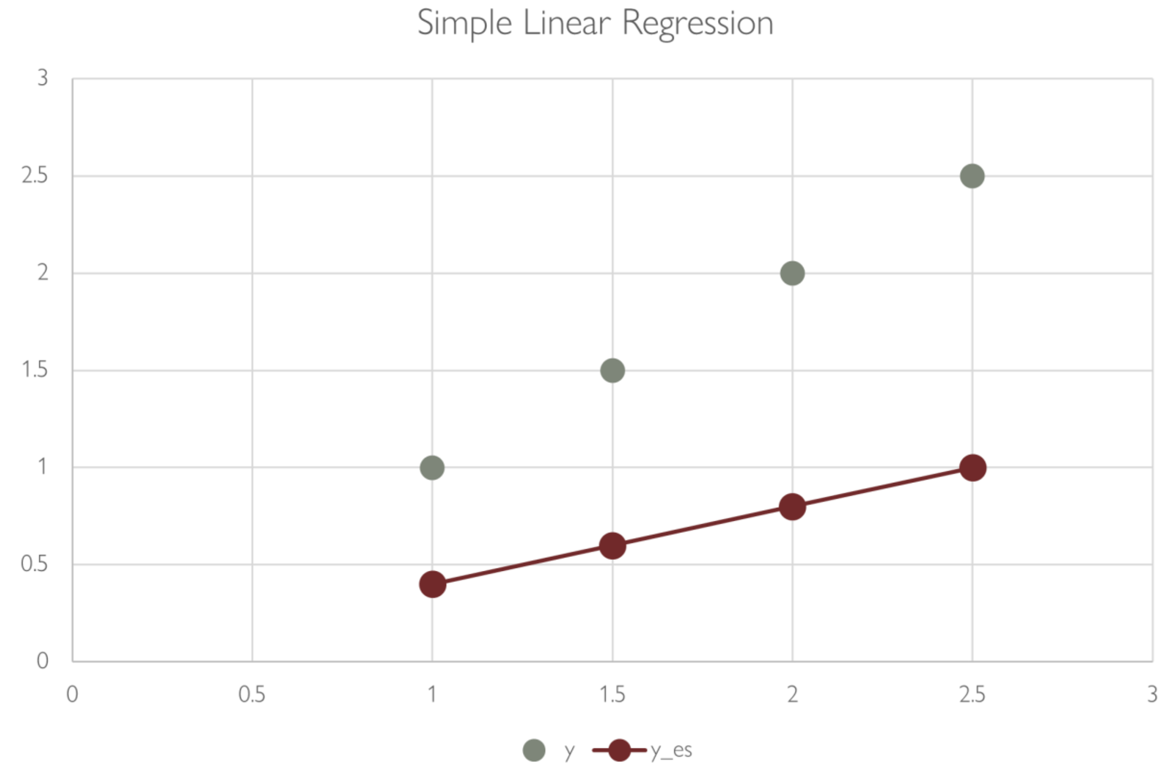
# squared-error loss function

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$

0.61

$$J(w) = \frac{1}{2 * 4}\sum_{i=1}^{4}\{ (1 - 0.4)^2, (1.5 - 0.6)^2, (2 - 0.8)^2, (2.5 - 1)^2\}$$

| base | slop | | | |
|---|---|---|---|---|
| 0 | 0.4 | | | |
| x | y | y_es | square error | square error |
| 1 | 1 | 0.4 | 0.36 | 0.64 |
| 1.5 | 1.5 | 0.6 | 0.81 | 1.44 |
| 2 | 2 | 0.8 | 1.44 | 2.56 |
| 2.5 | 2.5 | 1 | 2.25 | 4 |
| mean square error | | | 0.61 | 1.08 |

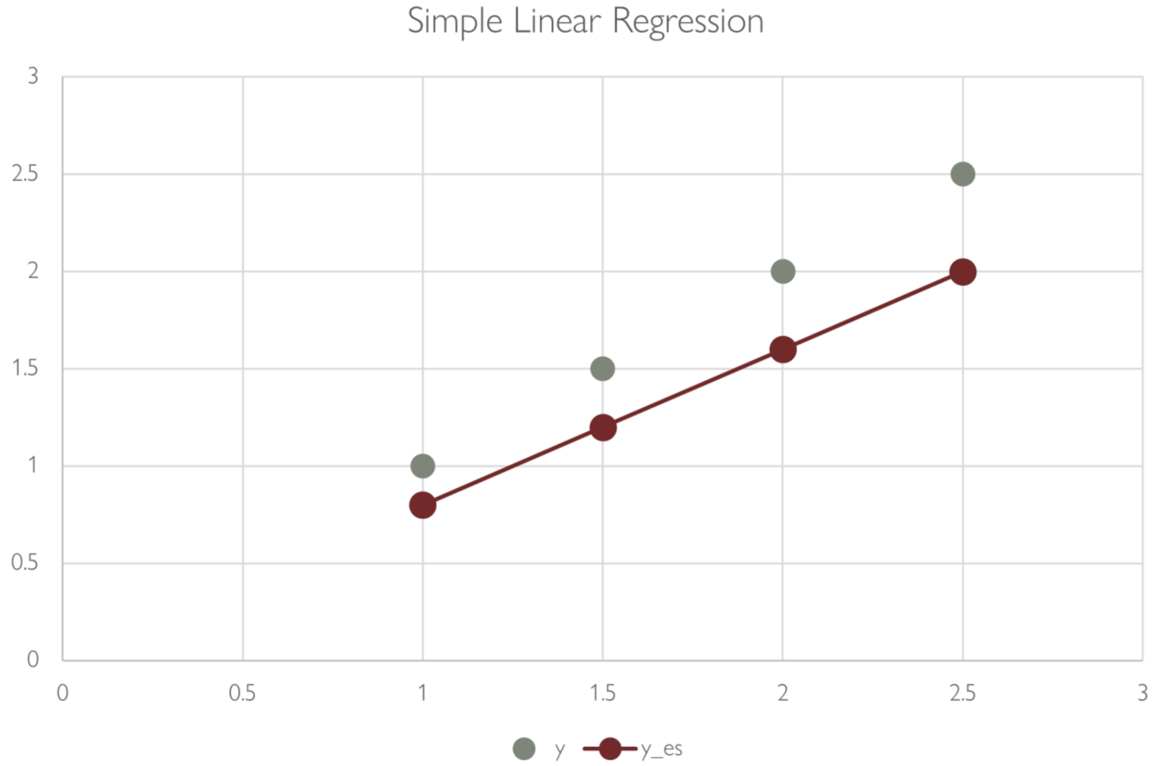


# loss function

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$


$$J(w) = \frac{1}{2 * 4}\sum_{i=1}^{4}\{\,(1-0.8)^2, (1.5-1.2)^2\\ ,(2-1.6)^2, (2.5-2)^2\}$$

| base | slop | | | | |
|---|---|---|---|---|---|
| 0 | 0.8 | | | | |
| x | y | y_es | square error | square error | square error |
| 1 | 1 | 0.8 | 0.04 | 0.36 | 0.64 |
| 1.5 | 1.5 | 1.2 | 0.09 | 0.81 | 1.44 |
| 2 | 2 | 1.6 | 0.16 | 1.44 | 2.56 |
| 2.5 | 2.5 | 2 | 0.25 | 2.25 | 4 |
| mean square error | | | 0.07 | 0.61 | 1.08 |

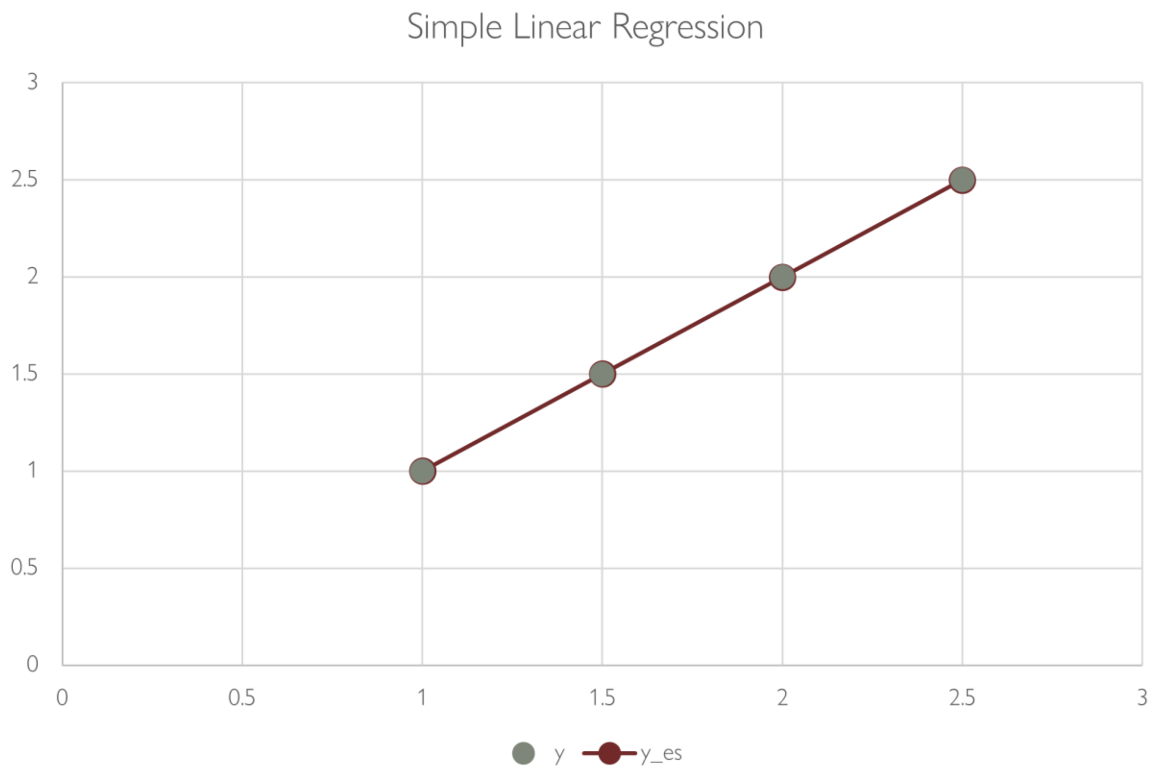


# loss function

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$


$$J(w) = \frac{1}{2*4}\sum_{i=1}^{4}\{(1-1)^2, (1.5-1.5)^2, (2-2)^2, (2.5-2.5)^2\}$$

| base | slop | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | | | | | |
| x | y | y_es | square error | square error | square error | square error |
| 1 | 1 | 1 | 0 | 0.04 | 0.36 | 0.64 |
| 1.5 | 1.5 | 1.5 | 0 | 0.09 | 0.81 | 1.44 |
| 2 | 2 | 2 | 0 | 0.16 | 1.44 | 2.56 |
| 2.5 | 2.5 | 2.5 | 0 | 0.25 | 2.25 | 4 |
| mean square error | | | 0 | 0.07 | 0.61 | 1.08 |

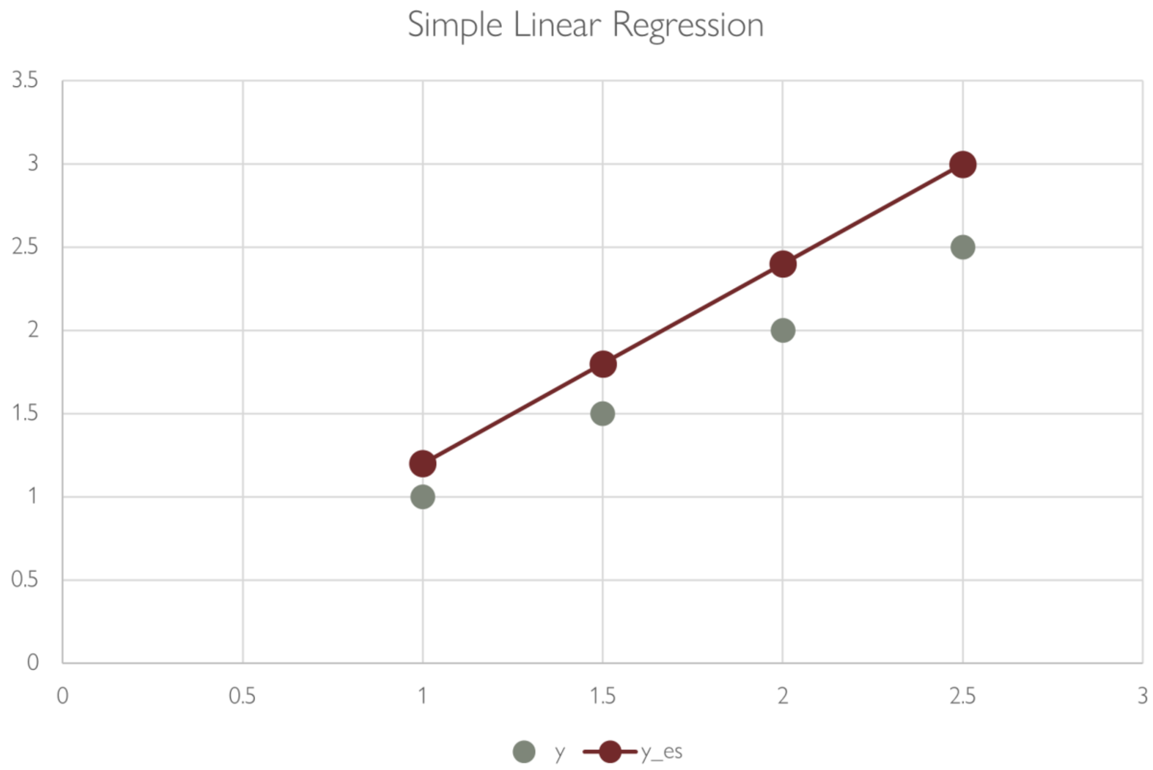


# squared-error loss function

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$


$$J(w) = \frac{1}{2*4}\sum_{i=1}^{4}\{\,(1-1.2)^2, (1.5-1.8)^2, (2-2.4)^2, (2.5-3)^2\}$$

| base | slop | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1.2 | | | | | | |
| x | y | y_es | square error | square error | square error | square error | square error |
| 1 | 1 | 1.2 | 0.04 | 0 | 0.04 | 0.36 | 0.64 |
| 1.5 | 1.5 | 1.8 | 0.09 | 0 | 0.09 | 0.81 | 1.44 |
| 2 | 2 | 2.4 | 0.16 | 0 | 0.16 | 1.44 | 2.56 |
| 2.5 | 2.5 | 3 | 0.25 | 0 | 0.25 | 2.25 | 4 |
| mean square error | | | 0.07 | 0 | 0.07 | 0.61 | 1.08 |



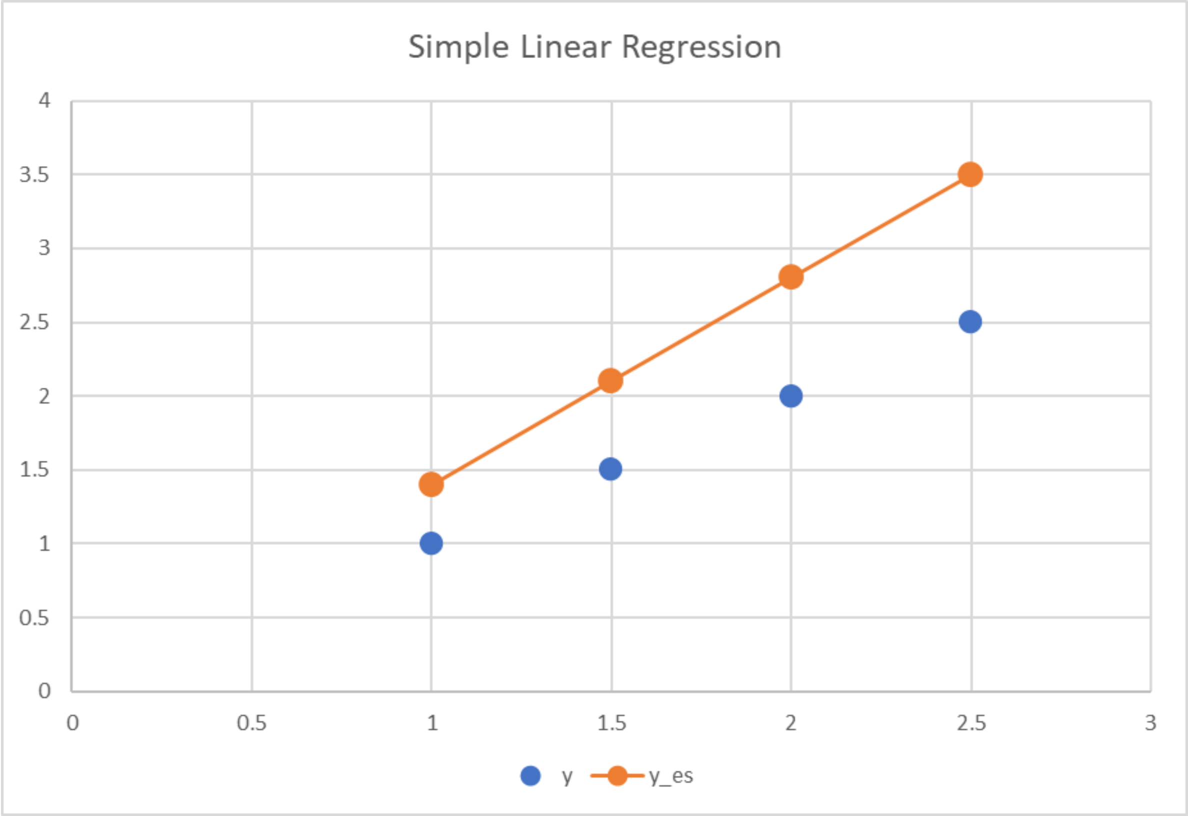
# loss function

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$

# 0.27

$$J(w) = \frac{1}{2*4}\sum_{i=1}^{4}\{(1-1.4)^2, (1.5-2.1)^2, (2-2.8)^2, (2.5-3.5)^2\}$$

| base | slop | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1.4 | | | | | | | |
| x | y | y_es | square error | square error | square error | square error | square error | square error |
| 1 | 1 | 1.4 | 0.16 | 0.04 | 0 | 0.04 | 0.36 | 0.64 |
| 1.5 | 1.5 | 2.1 | 0.36 | 0.09 | 0 | 0.09 | 0.81 | 1.44 |
| 2 | 2 | 2.8 | 0.64 | 0.16 | 0 | 0.16 | 1.44 | 2.56 |
| 2.5 | 2.5 | 3.5 | 1 | 0.25 | 0 | 0.25 | 2.25 | 4 |
| mean square error | | | 0.27 | 0.07 | 0 | 0.07 | 0.61 | 1.08 |

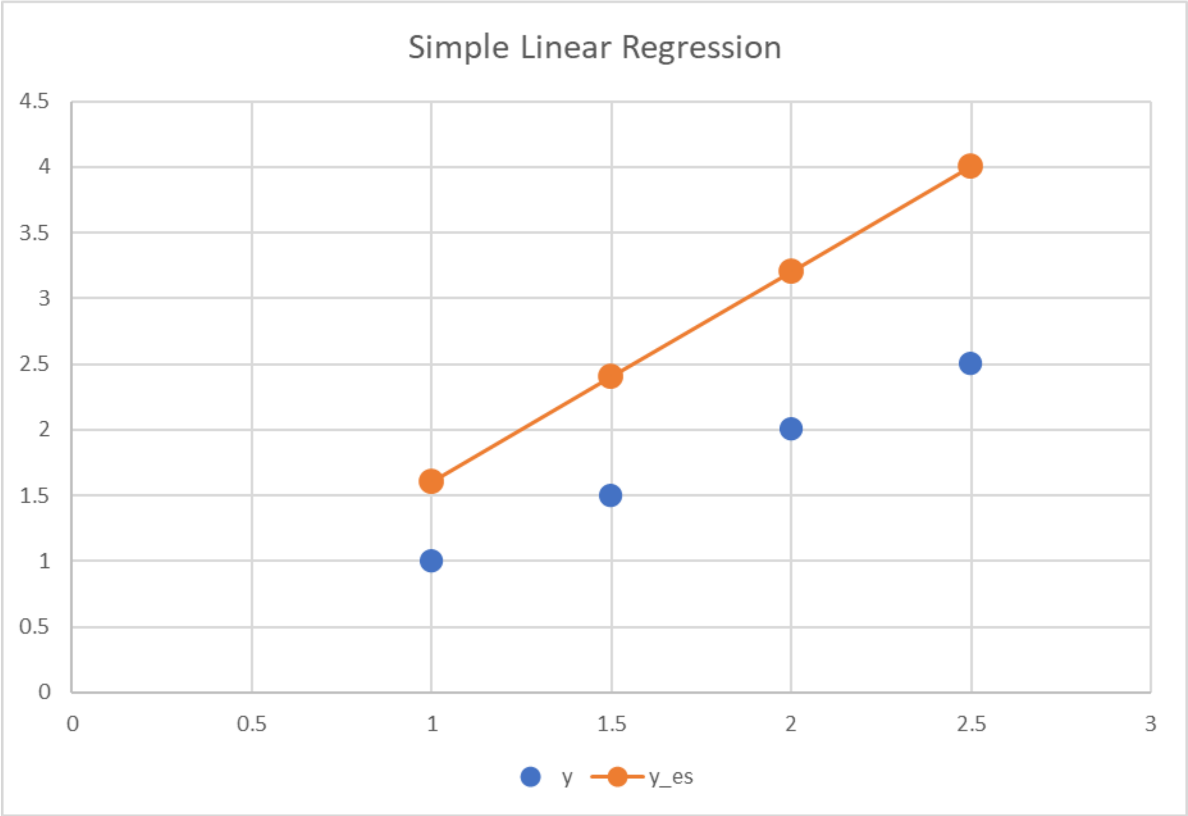


# loss function

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$

| base | slop | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.6 | | | | | | | | |
| x | y | y_es | square error | square error | square error | square error | square error | square error | square error |
| 1 | 1 | 1.6 | 0.36 | 0.16 | 0.04 | 0 | 0.04 | 0.36 | 0.64 |
| 1.5 | 1.5 | 2.4 | 0.81 | 0.36 | 0.09 | 0 | 0.09 | 0.81 | 1.44 |
| 2 | 2 | 3.2 | 1.44 | 0.64 | 0.16 | 0 | 0.16 | 1.44 | 2.56 |
| 2.5 | 2.5 | 4 | 2.25 | 1 | 0.25 | 0 | 0.25 | 2.25 | 4 |
| mean square error | | | 0.61 | 0.27 | 0.07 | 0 | 0.07 | 0.61 | 1.08 |

0.61



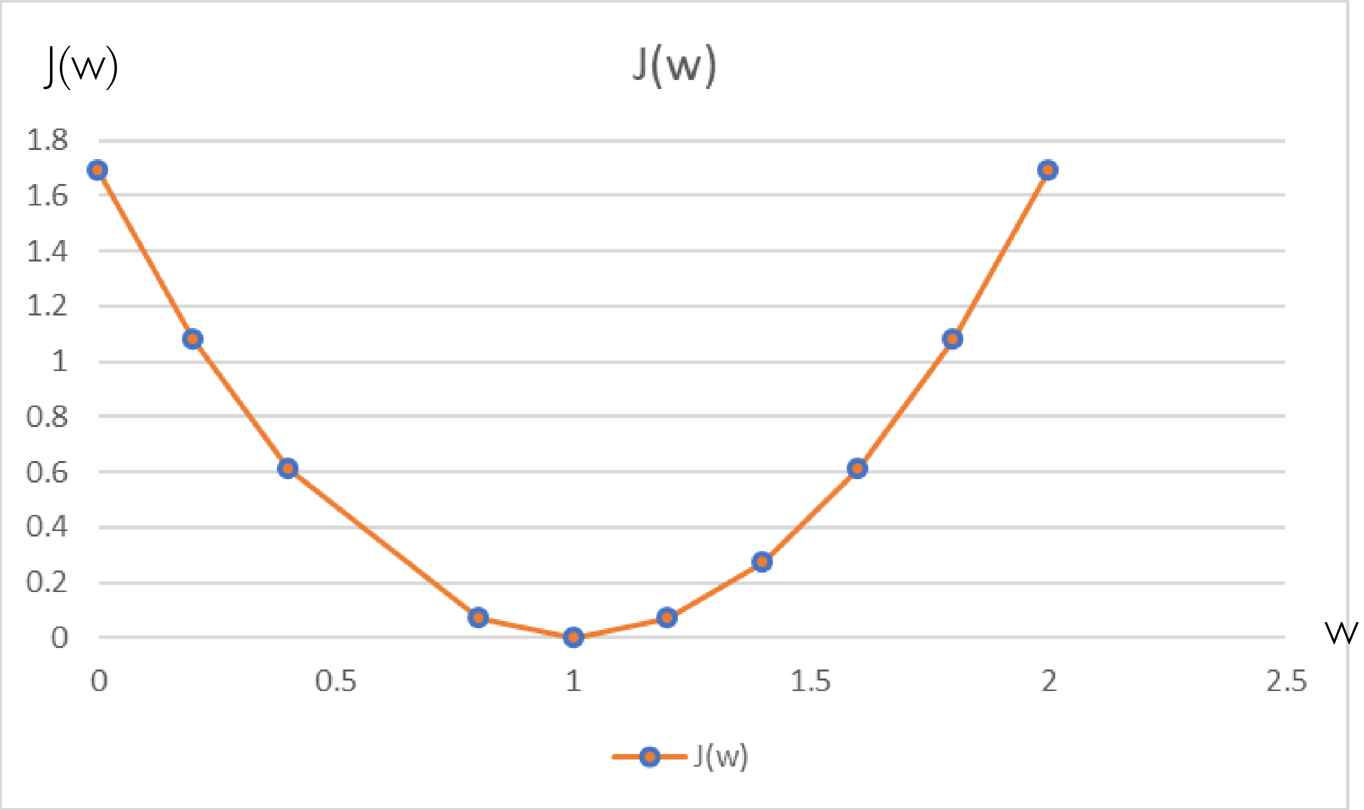

$$J(w) = \frac{1}{2 * 4}\sum_{i=1}^{4}\{ (1 - 1.6)^2, (1.5 - 2.4)^2, (2 - 3.2)^2, (2.5 - 4)^2\}$$

# loss function

$$J(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$

Cost function

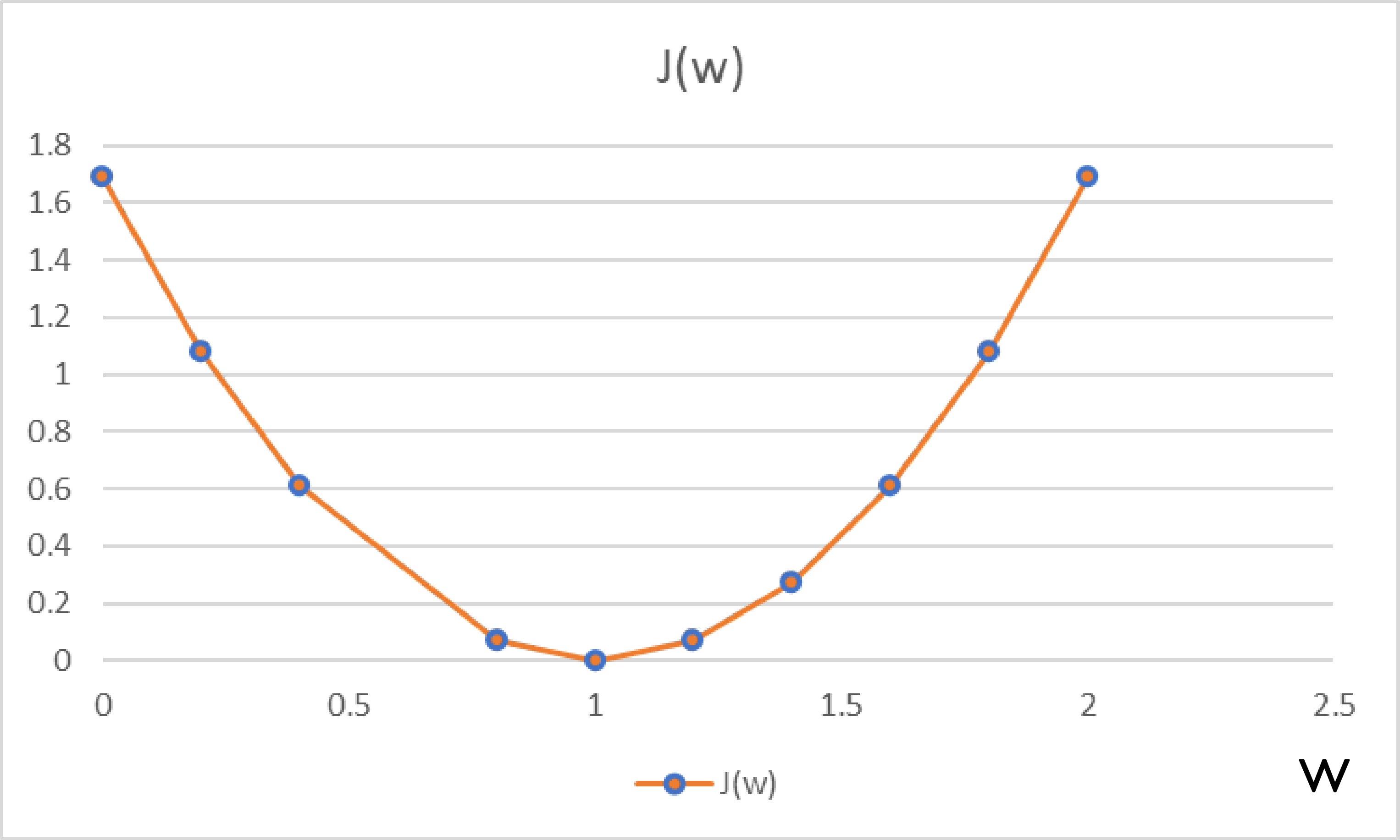
| base | slop | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.6 | | | | | | | | |
| x | y | y_es | square error | square error | square error | square error | square error | square error | square error |
| 1 | 1 | 1.6 | 0.36 | 0.16 | 0.04 | 0 | 0.04 | 0.36 | 0.64 |
| 1.5 | 1.5 | 2.4 | 0.81 | 0.36 | 0.09 | 0 | 0.09 | 0.81 | 1.44 |
| 2 | 2 | 3.2 | 1.44 | 0.64 | 0.16 | 0 | 0.16 | 1.44 | 2.56 |
| 2.5 | 2.5 | 4 | 2.25 | 1 | 0.25 | 0 | 0.25 | 2.25 | 4 |
| mean square error | | | 0.61 | 0.27 | 0.07 | 0 | 0.07 | 0.61 | 1.08 |



# Gradient Descent

1. Keep changing w,b to reduce J(w, b)
2. Until we settle at or near a minimum

| base | slop | | |
|---|---|---|---|
| 0 | 1.6 | | |
| x | y | y_es | square error |
| 1 | 1 | 1.6 | 0.36 |
| 1.5 | 1.5 | 2.4 | 0.81 |
| 2 | 2 | 3.2 | 1.44 |
| 2.5 | 2.5 | 4 | 2.25 |
| mean square error | | | 0.61 |



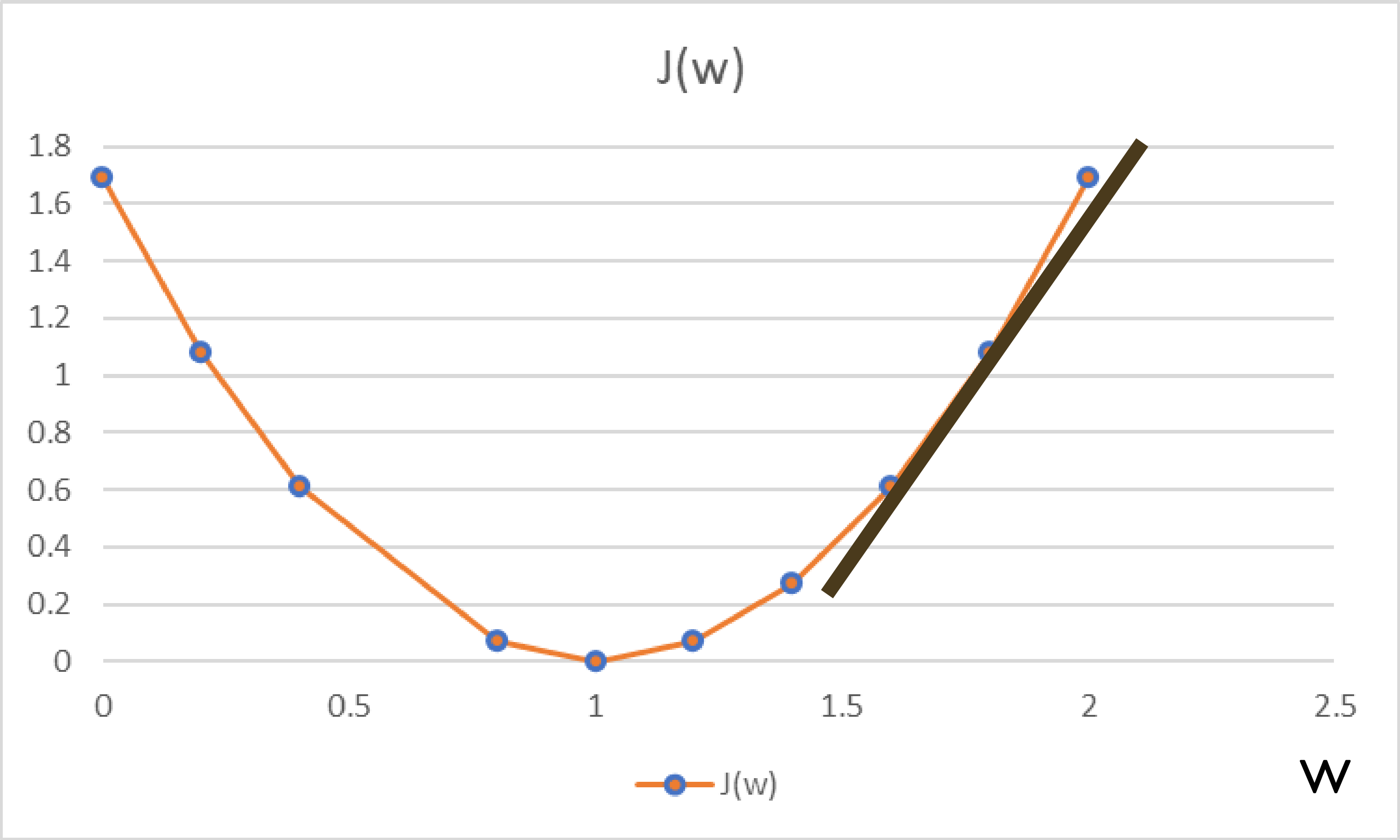

$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

Learning Rate (step)

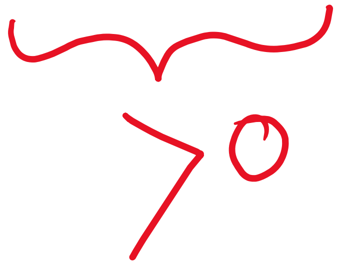
$$b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

1. Keep changing w,b to reduce J(w, b)
2. Until we settle at or near a minimum

| base | slop | | |
|---|---|---|---|
| 0 | 1.6 | | |
| x | y | y_es | square error |
| 1 | 1 | 1.6 | 0.36 |
| 1.5 | 1.5 | 2.4 | 0.81 |
| 2 | 2 | 3.2 | 1.44 |
| 2.5 | 2.5 | 4 | 2.25 |
| mean square error | | | 0.61 |

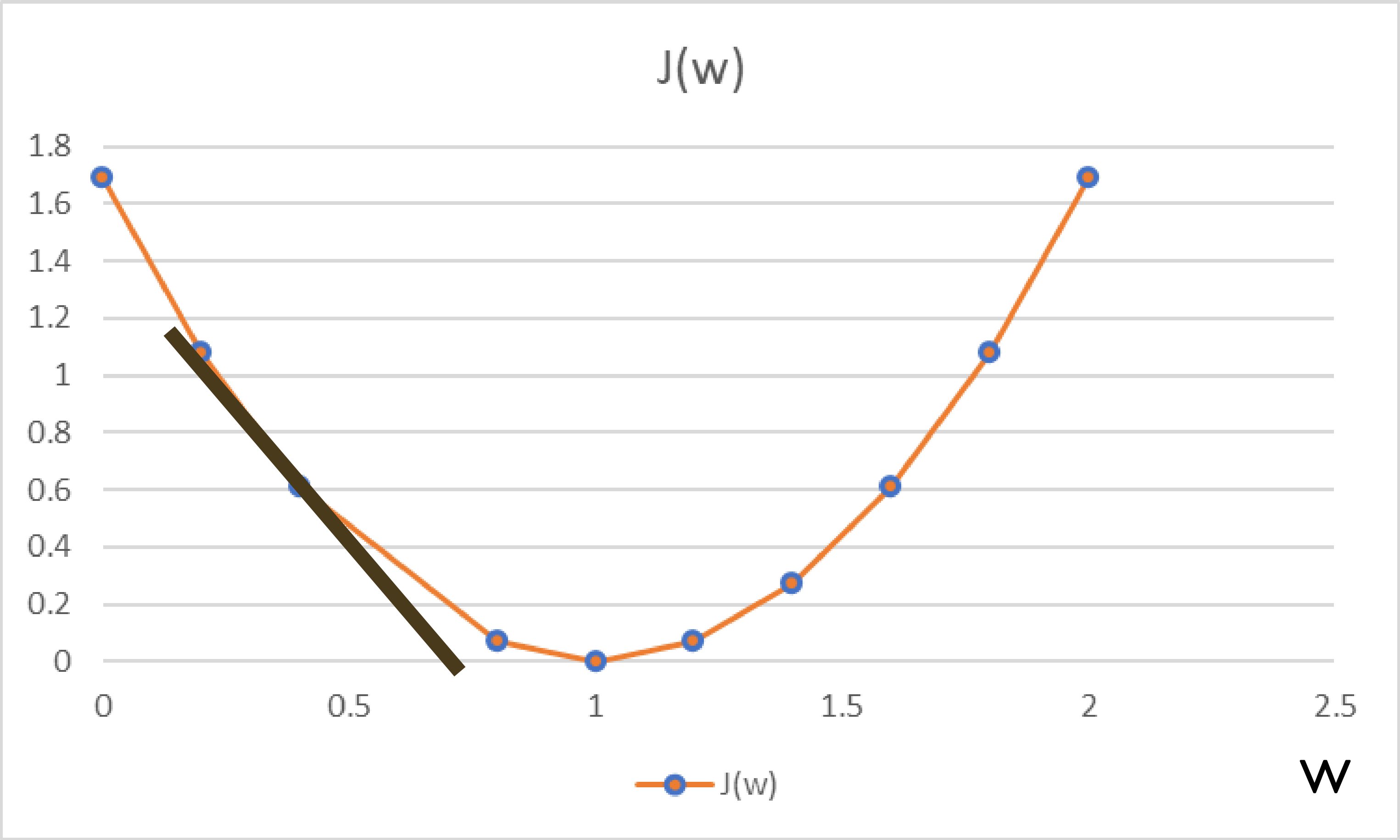



$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$



1. Keep changing w,b to reduce J(w, b)
2. Until we settle at or near a minimum

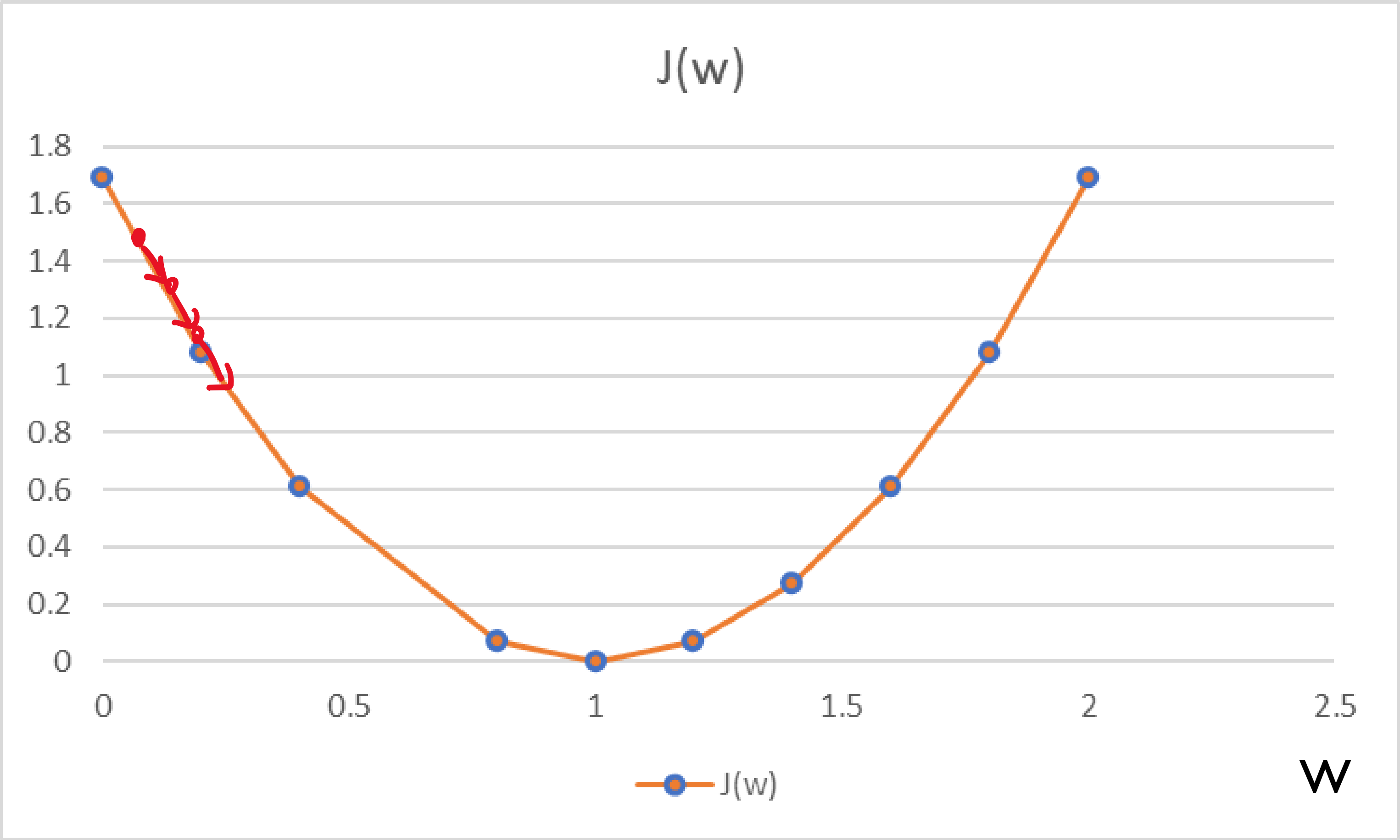
| base | slop | | |
|---|---|---|---|
| 0 | 1.6 | | |
| x | y | y_es | square error |
| 1 | 1 | 1.6 | 0.36 |
| 1.5 | 1.5 | 2.4 | 0.81 |
| 2 | 2 | 3.2 | 1.44 |
| 2.5 | 2.5 | 4 | 2.25 |
| mean square error | | | 0.61 |




$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$< 0$

# 1. Learning rate

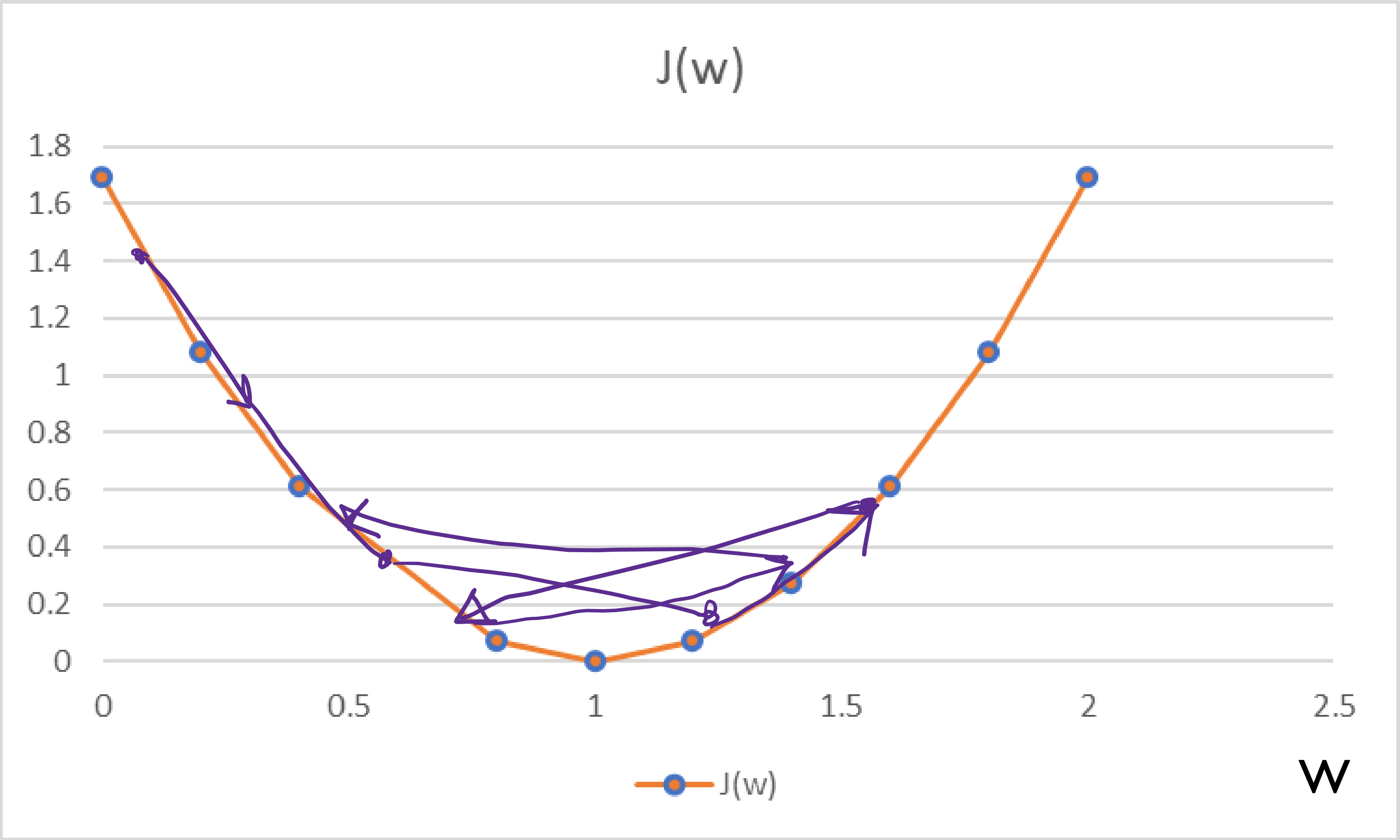



$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

α <<

too many steps

# 1. Learning rate

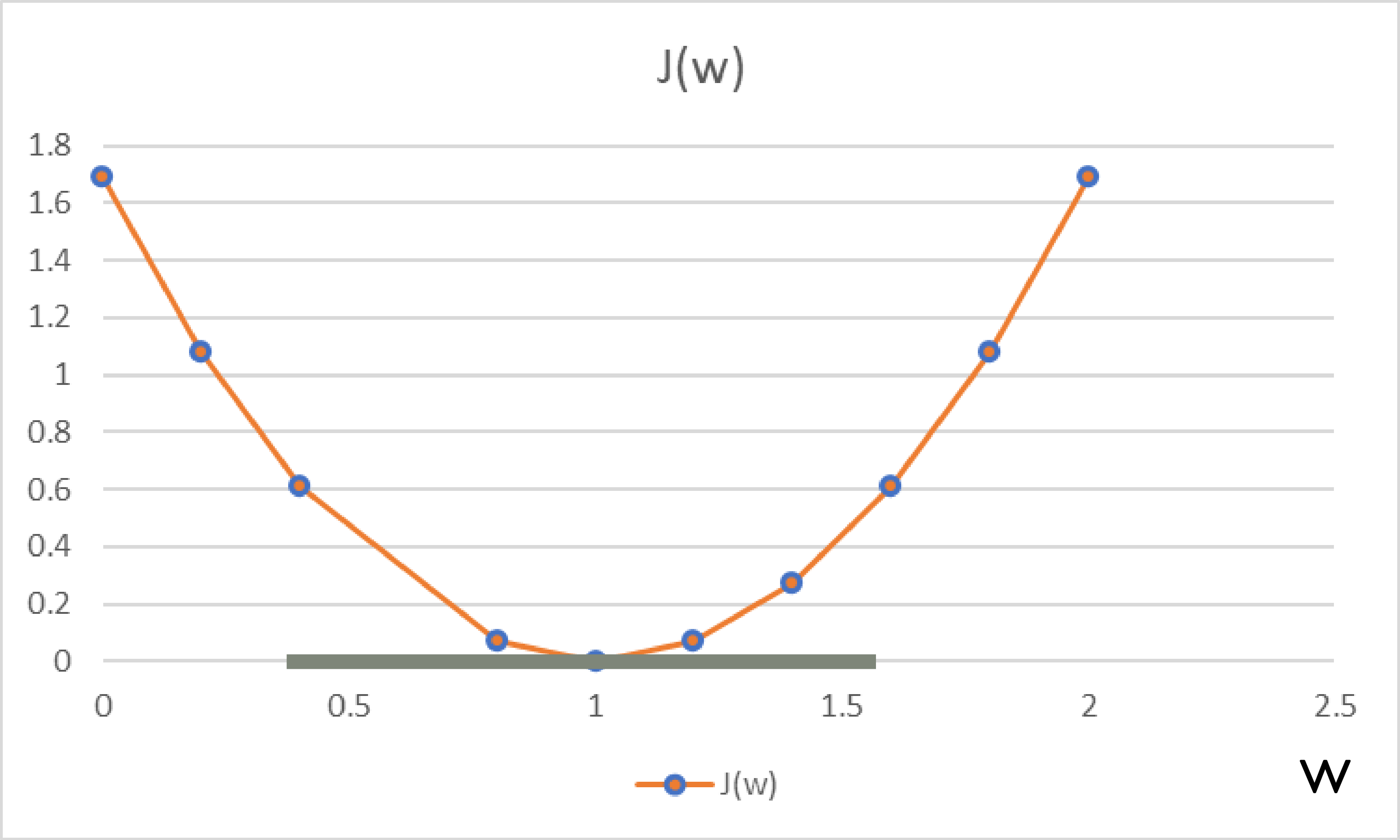



$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

a larger value

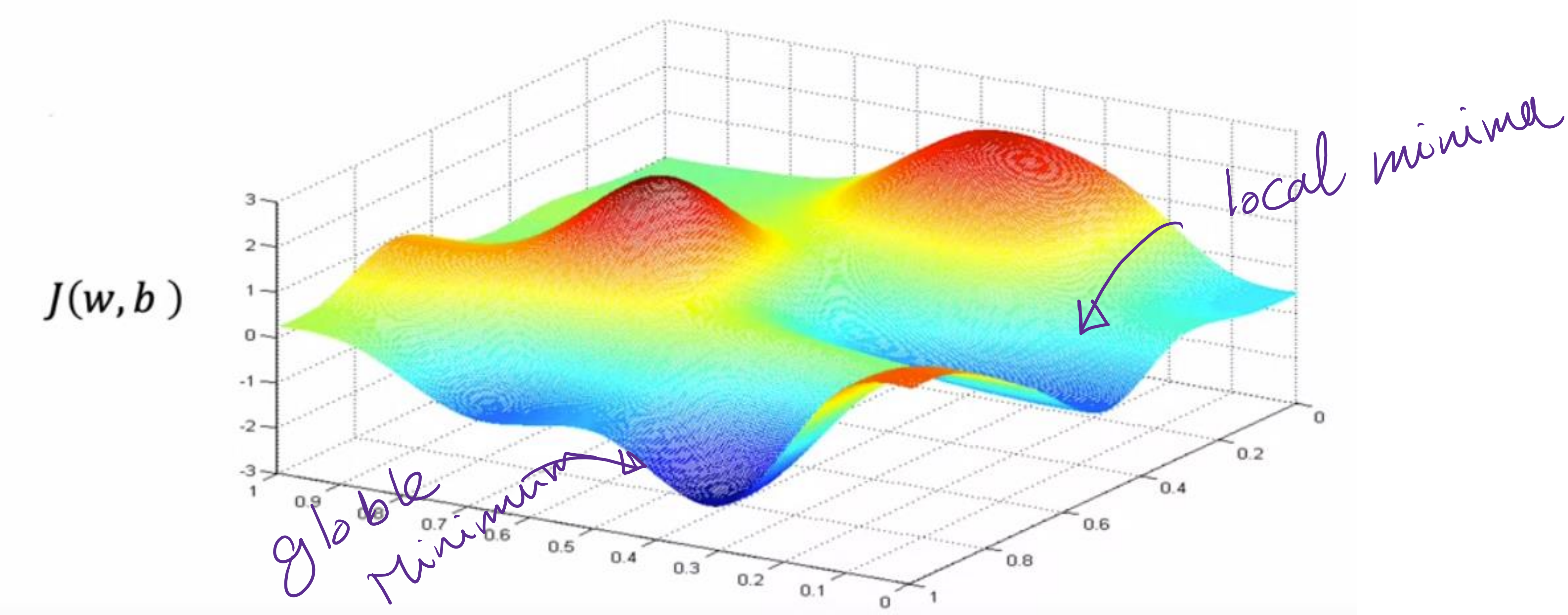
1. Keep changing w,b to reduce J(w, b)
2. Until we settle at or near a minimum

| base | slop | | |
|---|---|---|---|
| 0 | 1.6 | | |
| x | y | y_es | square error |
| 1 | 1 | 1.6 | 0.36 |
| 1.5 | 1.5 | 2.4 | 0.81 |
| 2 | 2 | 3.2 | 1.44 |
| 2.5 | 2.5 | 4 | 2.25 |
| mean square error | | | 0.61 |



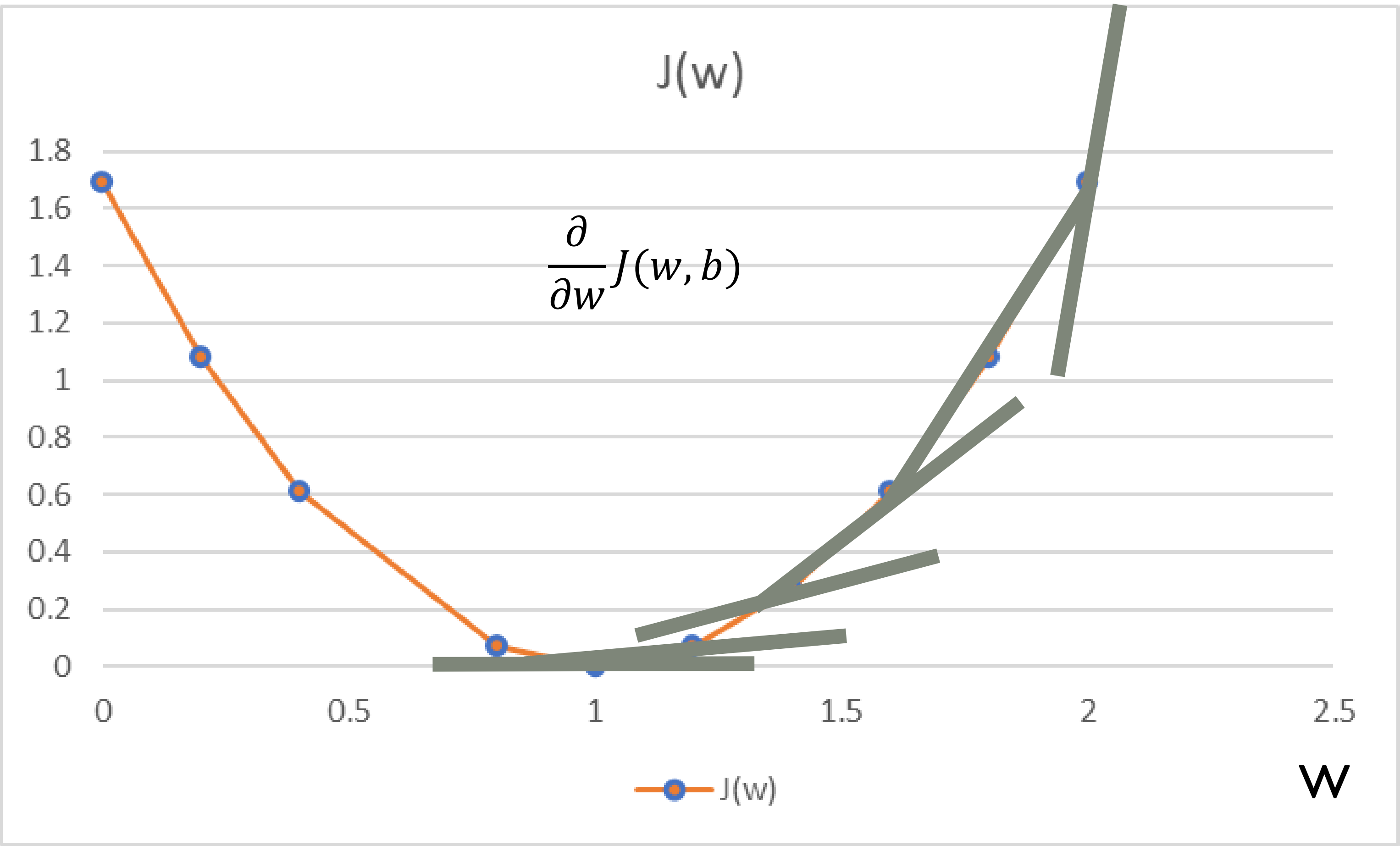

$$w = w - \alpha \times 0$$
$$w = w$$

# Local minima



1. Keep changing w,b to reduce J(w, b)
2. Until we settle at or near a minimum

| base | slop | | |
|---|---|---|---|
| 0 | 1.6 | | |
| x | y | y_es | square error |
| 1 | 1 | 1.6 | 0.36 |
| 1.5 | 1.5 | 2.4 | 0.81 |
| 2 | 2 | 3.2 | 1.44 |
| 2.5 | 2.5 | 4 | 2.25 |
| mean square error | | | 0.61 |




$$\frac{\partial}{\partial w}J(w,b)$$

$$w = w - \alpha \frac{\partial}{\partial w}J(w,b)$$

$\frac{\partial}{\partial w}J(w,b)$ ↓ when go further

# Gradient Descent

$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$\frac{\partial}{\partial w} J(w, b) = \frac{\partial}{\partial w} \left[ \frac{1}{2n} \sum_{i=1}^{n} ((wx + b) - \hat{y}^{(i)})^2 \right]$$

$$= \frac{1}{2n} \times 2\left[\sum_{i=1}^{n} ((wx^{(i)} + b) - \hat{y}^{(i)})) \times x^{(i)}\right]$$

$$= \frac{1}{n} \times \sum_{i=1}^{n} (y - \hat{y}^{(i)}) \times x^{(i)}$$

# Gradient Descent

$$b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$$\frac{\partial}{\partial b} J(w, b) = \frac{\partial}{\partial b}\left[\frac{1}{2n}\sum_{i=1}^{n}((wx + b) - \hat{y}^{(i)})^2\right]$$

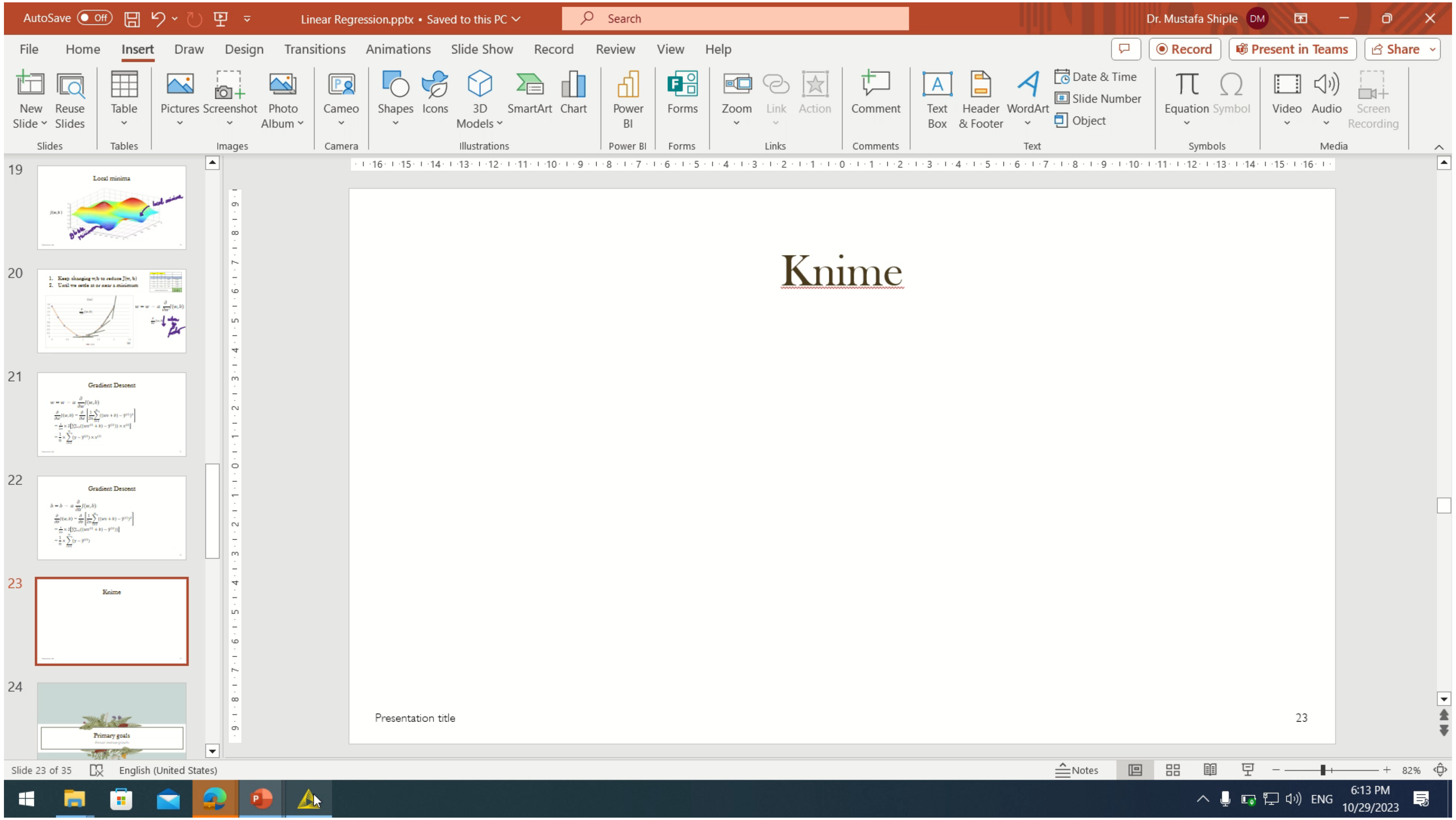
$$= \frac{1}{2n} \times 2\left[\sum_{i=1}^{n}((wx^{(i)} + b) - \hat{y}^{(i)})\right]$$

$$= \frac{1}{n} \times \sum_{i=1}^{n}(y - \hat{y}^{(i)})$$

# Batch gradient decent

- Batch gradient uses all training set data in learning phase.

# Knime


AutoSave Off
Linear Regression.pptx • Saved to this PC
Search
Dr. Mustafa Shiple DM
File Home Insert Draw Design Transitions Animations Slide Show Record Review View Help
Record
Present in Teams
Share
New Slide
Reuse Slides
Slides
Table
Tables
Pictures Screenshot Photo Album
Images
Cameo
Camera
Shapes Icons 3D Models SmartArt Chart
Illustrations
Power BI
Power BI
Forms
Forms
Zoom Link Action
Links
Comment
Comments
Text Box
Header & Footer
WordArt
Date & Time
Slide Number
Object
Text
Equation Symbol
Symbols
Video Audio Screen Recording
Media
19
Local minima
20
21
Gradient Descent
22
Gradient Descent
23
Knime
24
Primary goals
Knime
Presentation title
23
Slide 23 of 35
English (United States)
Notes
82%
ENG
6:13 PM
10/29/2023

# The END