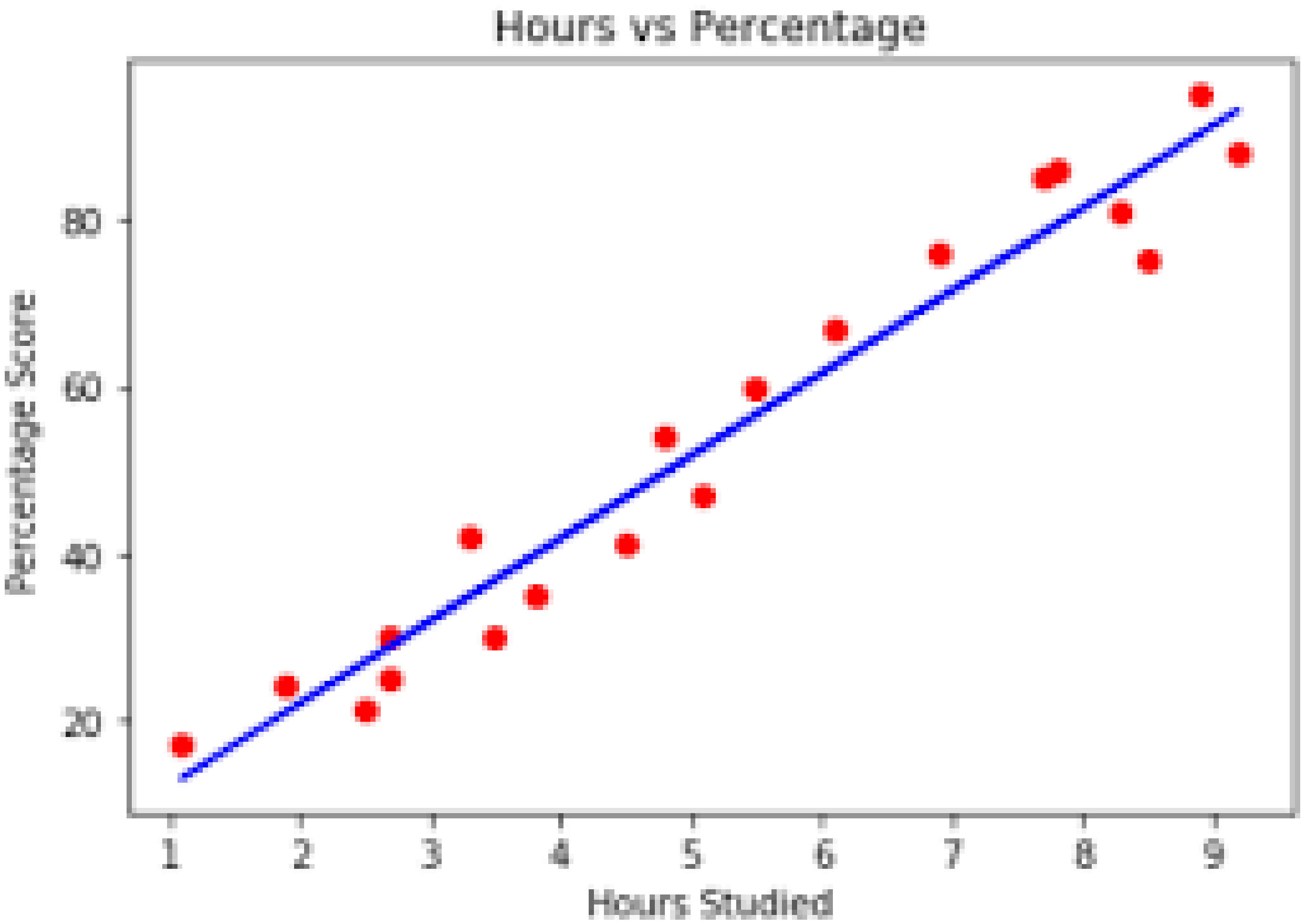
Lecture 2

Linear Regression

By: Nazerke Sultanova

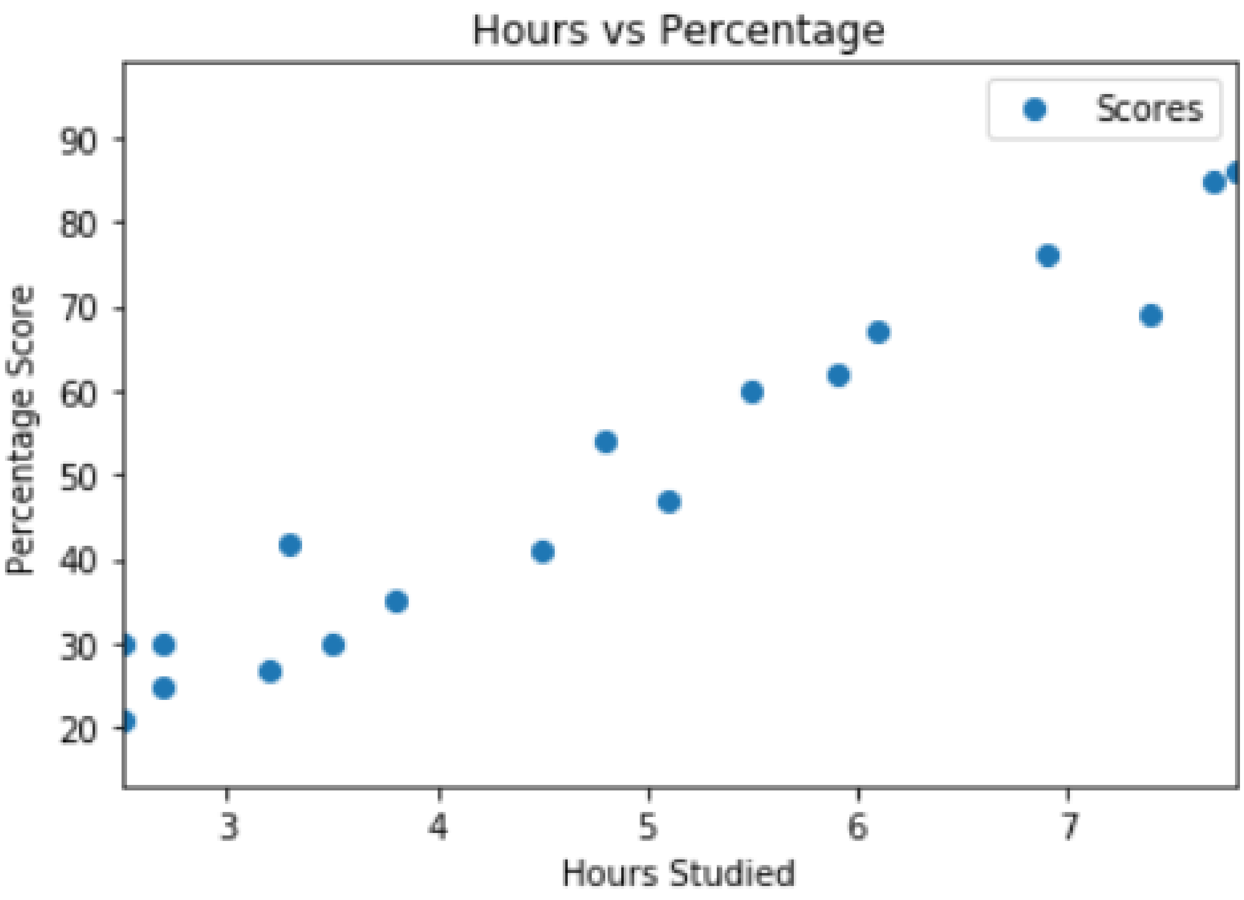
Linear Regression



Linear Regression

- There are two types of supervised machine learning algorithms: Regression and classification.
- Regression predicts continuous value outputs
- For instance, predicting the price of a house in dollars is a regression problem

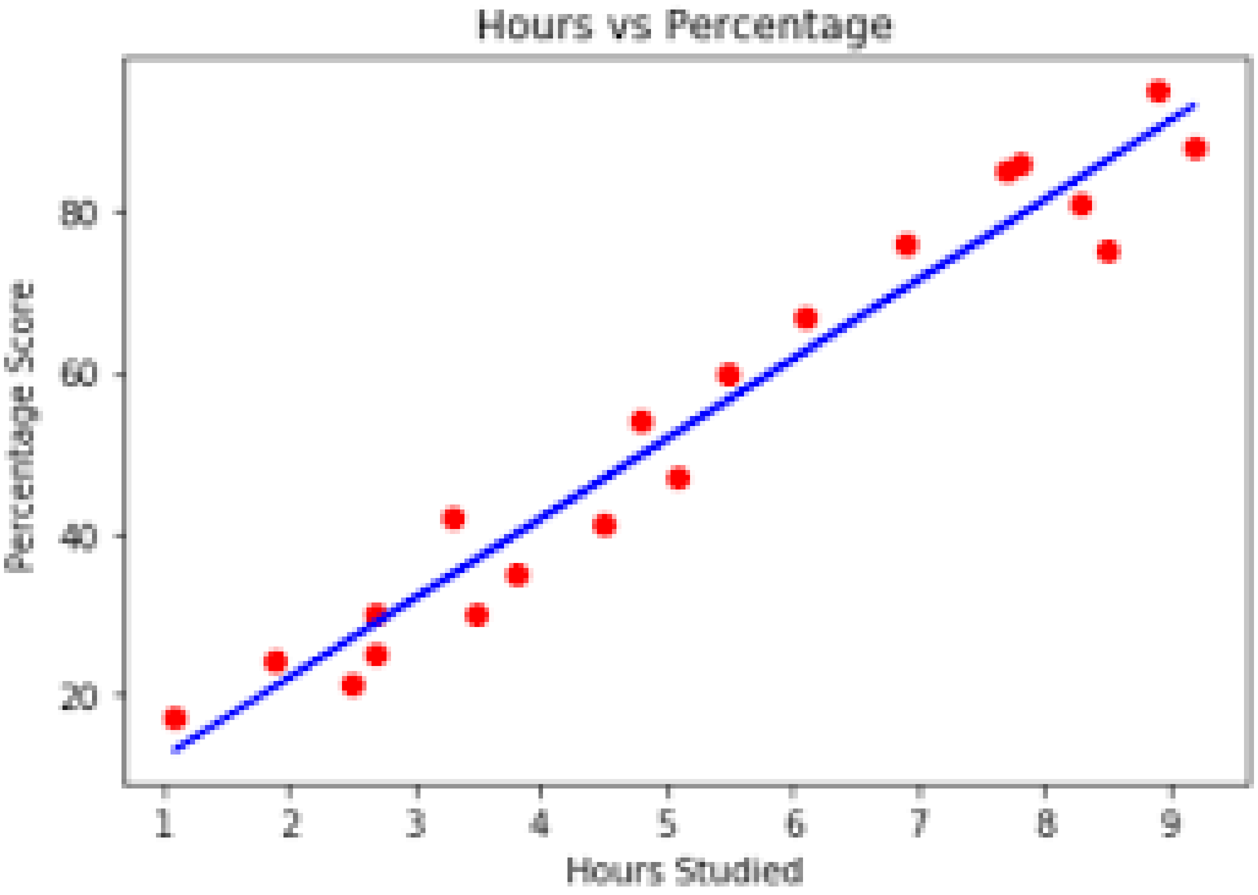
Example



Hours	Scores
2.5	21
5.1	47
3.2	27
8.5	75
3.5	30
1.5	20
9.2	88
5.5	60
8.3	81
2.7	25
7.7	85
5.9	62
4.5	41
3.3	42
1.1	17
8.9	95
2.5	30
1.9	24
6.1	67
7.4	69
2.7	30
4.8	54
3.8	35

Objective

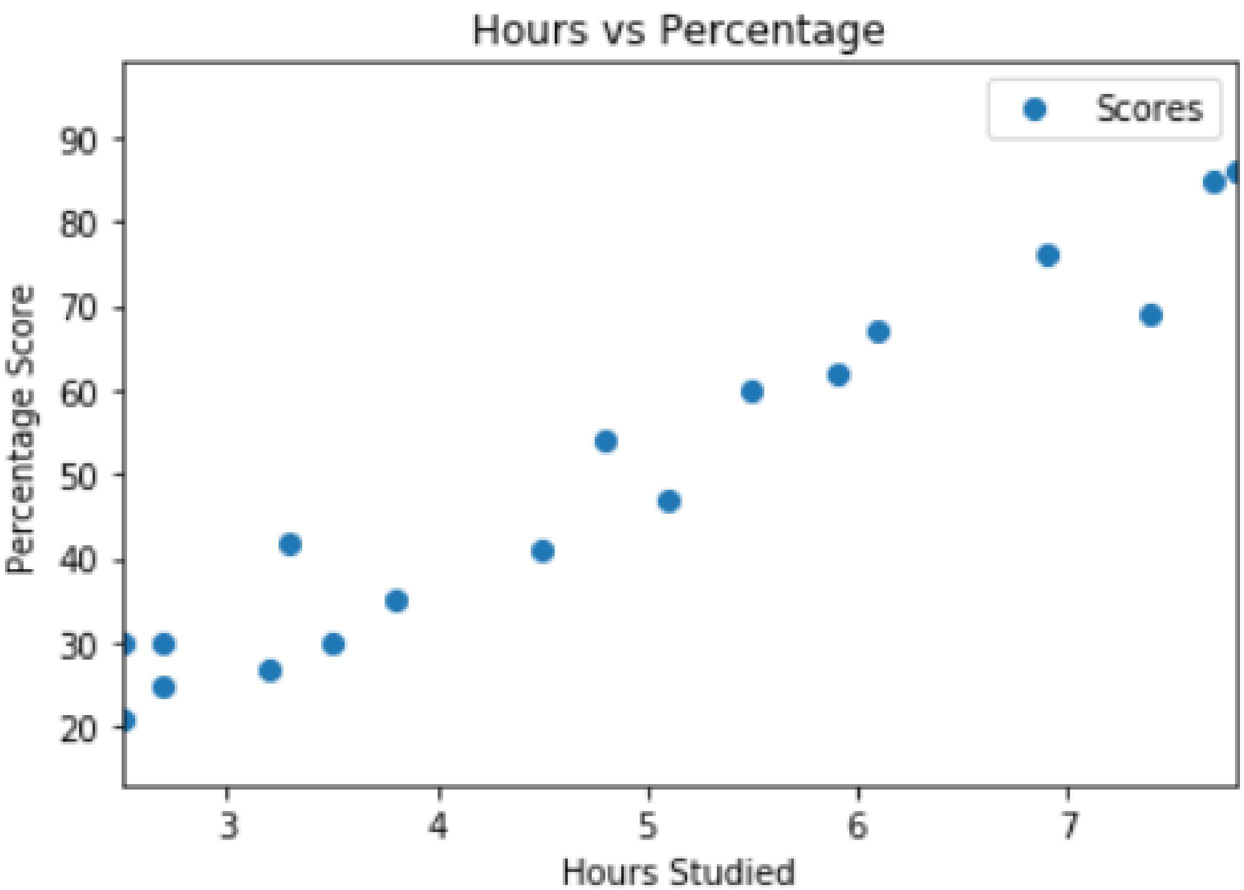
- Our objective is to predict hypothesis function



Hypothesis

- Linear regression => linear hypothesis (straight line)
- $Y = mx+b$
- Slope is m
- Intercept is b

Find a hypothesis?

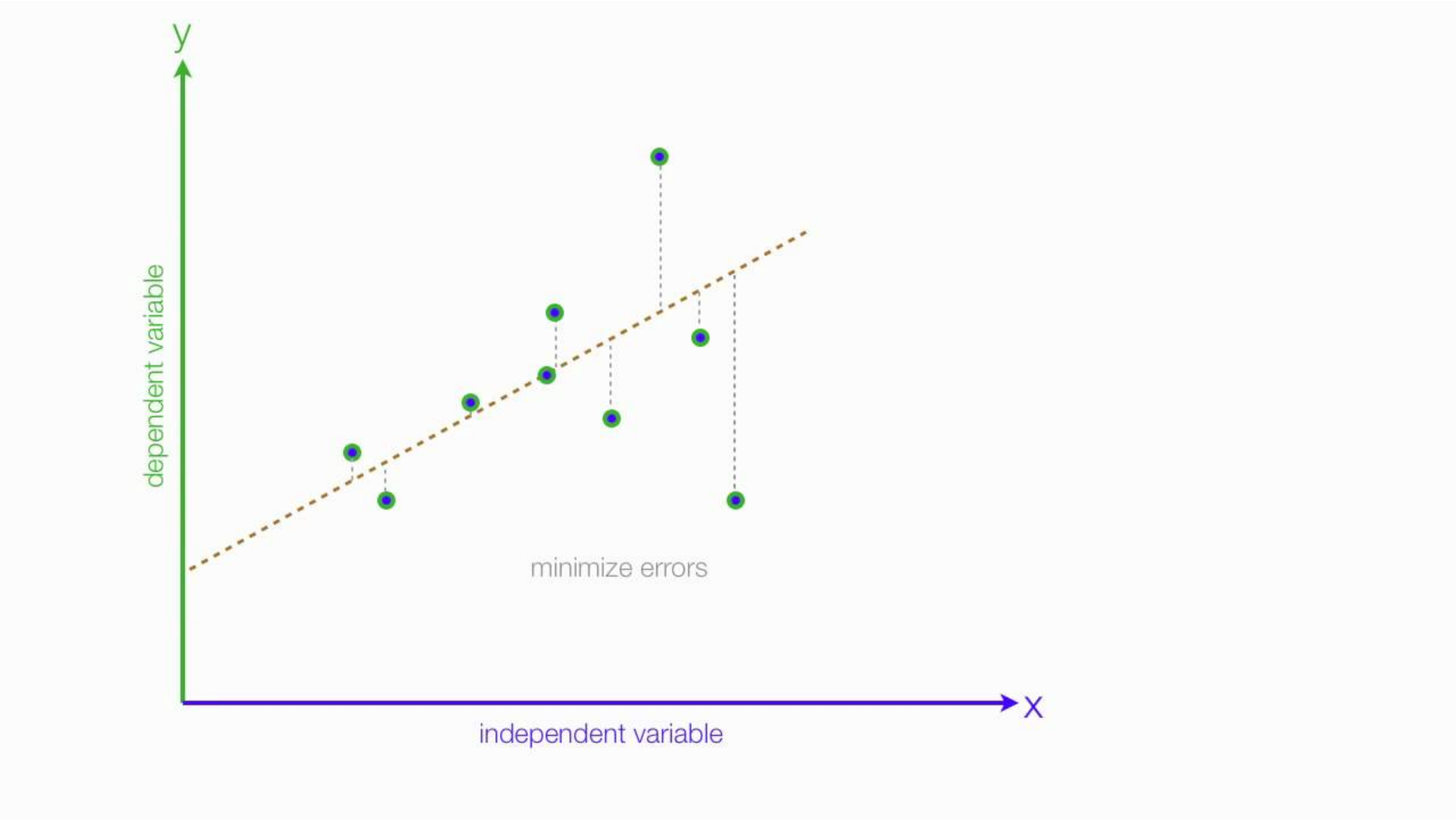


Percentage = __*hours+____

Hours	Scores
2.5	21
5.1	47
3.2	27
8.5	75
3.5	30
1.5	20
9.2	88
5.5	60
8.3	81
2.7	25
7.7	85
5.9	62
4.5	41
3.3	42
1.1	17
8.9	95
2.5	30
1.9	24
6.1	67
7.4	69
2.7	30
4.8	54
3.8	35

How to find best hypothesis?

- Hypothesis that has smallest error



- Sum of squared errors

Cost function

Cost function

For the parameter vector θ (of type $\mathbb{R}^{n+1}$ or in $\mathbb{R}^{(n+1)\times 1}$, the cost function is:

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

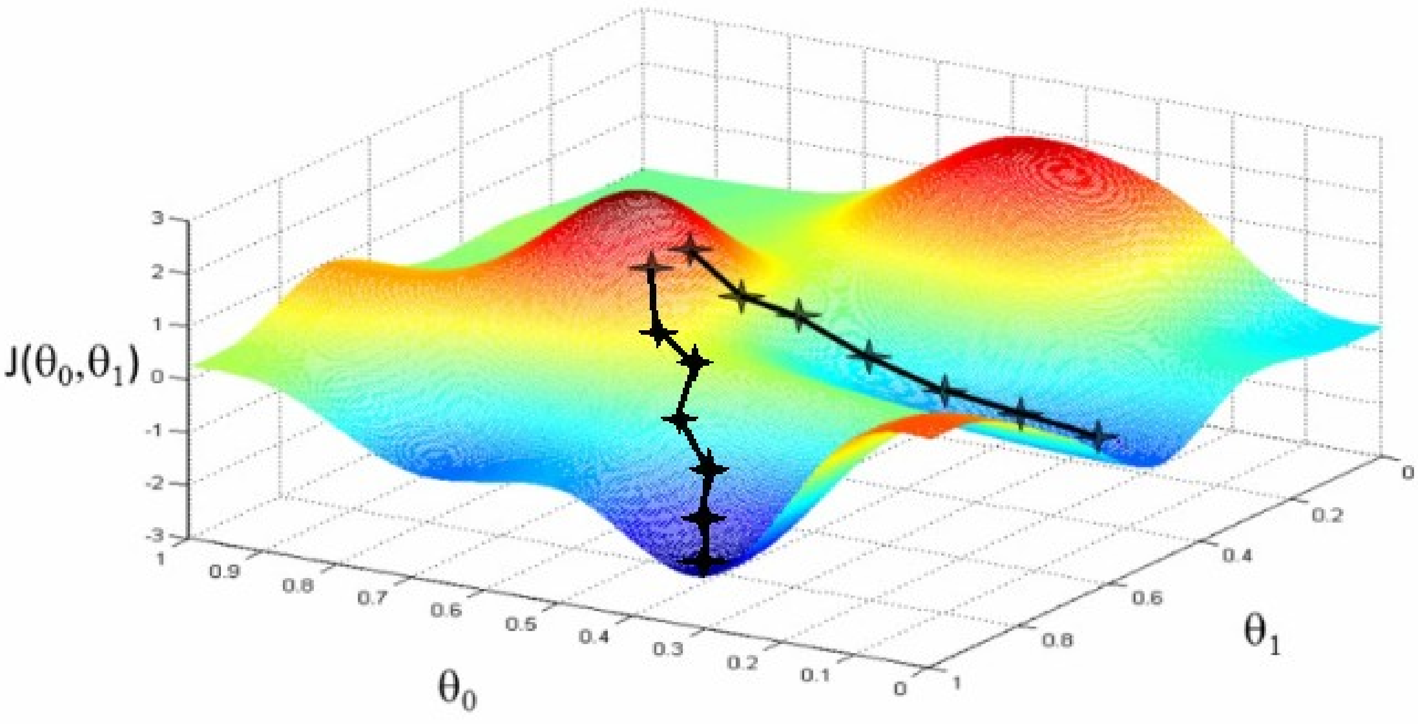
The vectorized version is:

$$J(\theta) = \frac{1}{2m} (X\theta - \vec{y})^T (X\theta - \vec{y})$$

Where $\vec{y}$ denotes the vector of all y values.

Gradient Descent

- One of convex optimizations
- Find the global minimum



Gradient Descent Algorithm

Gradient descent algorithm

repeat until convergence {

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

}

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

Gradient descent algorithm

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

repeat until convergence {

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

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

}

update θ_0 and θ_1 simultaneously

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

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

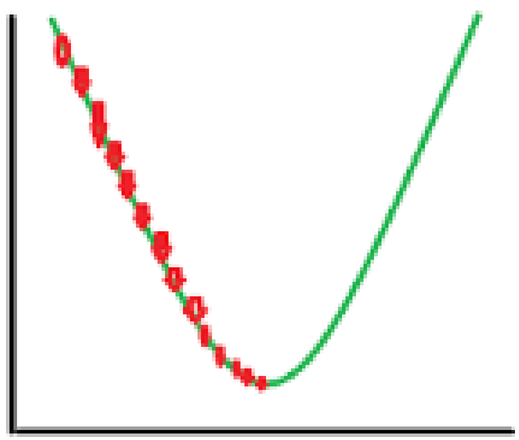
Andrew Ng

Learning rate

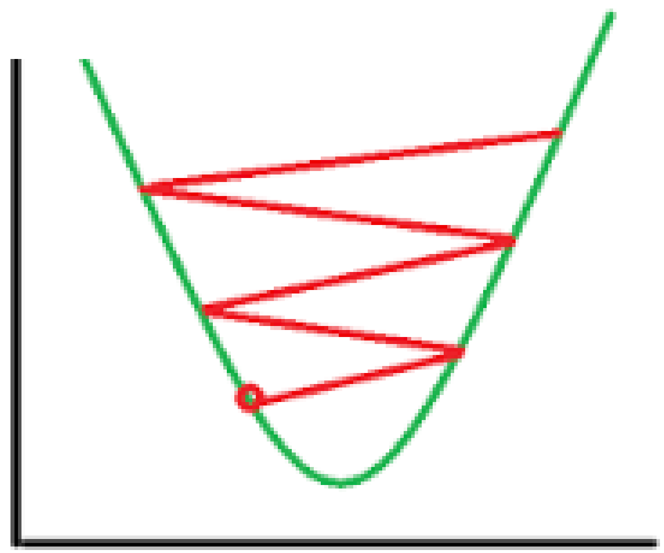
$$\left\{ \theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)} \right\}$$

$\Downarrow$

Learning Rate

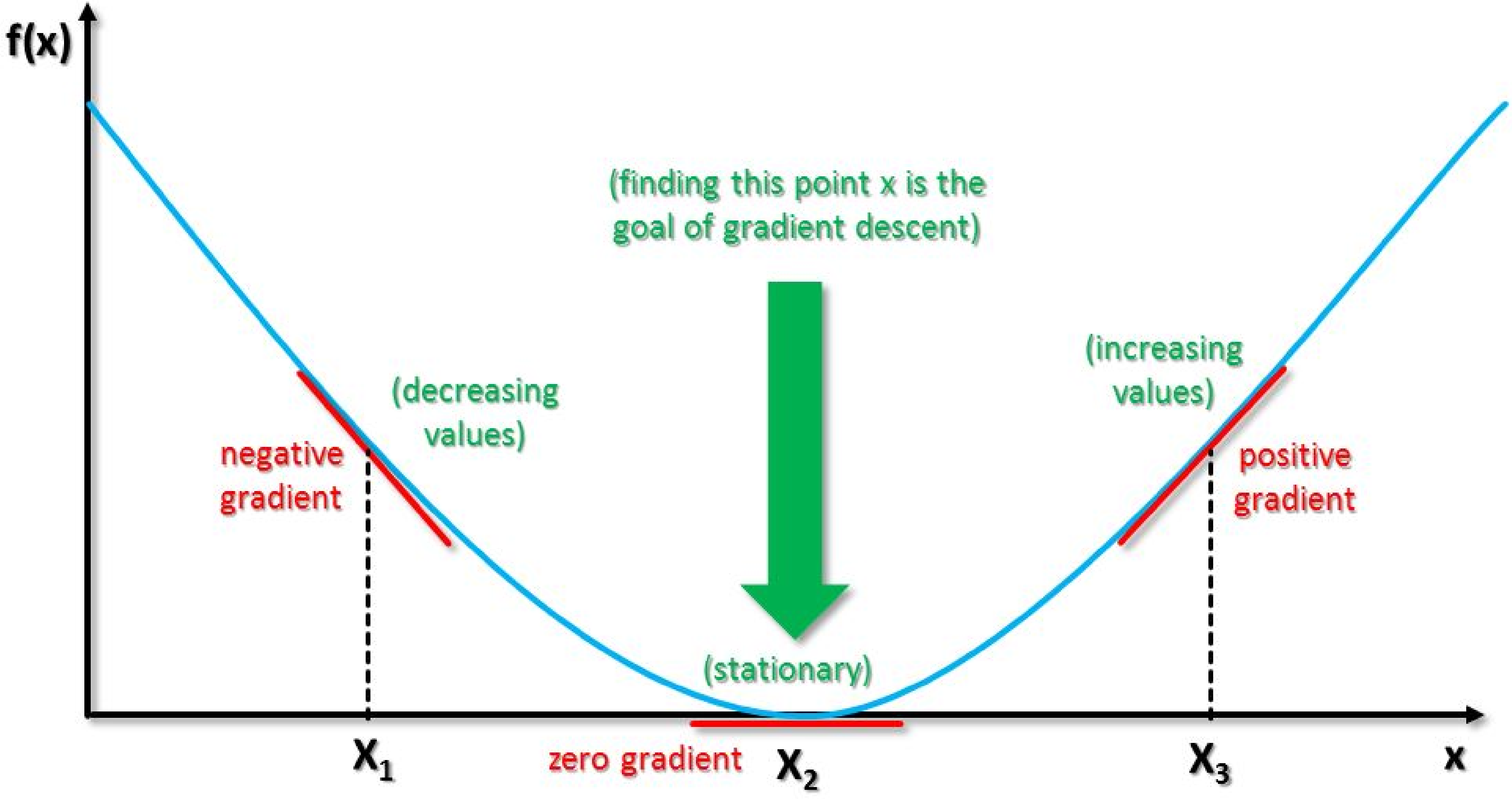


Low α



High α

Find a direction using derivative



Multivariate Linear Regression

Multiple features (variables).

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

$m = 47$

Notation:

- n = number of features $\quad n = 4$
- $x^{(i)}$ = input (features) of i^{th} training example.
- $x_j^{(i)}$ = value of feature j in i^{th} training example.

$$x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix}$$

$$x_3^{(2)} = 2$$

Assignment

- Write a function for ***hypothesis(theta1, theta0)***
- Write a function for ***gradient descent(theta1, theta0)***
- Find best ***theta1, theta0***
- Visualize the dataset and hypothesis line
- Datasets will be posted in google classrooms (three different dataset for three groups)

Hints for Assignment

- Initialize theta’s=0
- Take a few different alphas
- Gradient Descent: repeat until convergence = you can repeat ~2000 times
- Use your intuition

Linear Algebra Review

- Matrix Operations
- Inverse/Transpose of a matrix
- Dot product