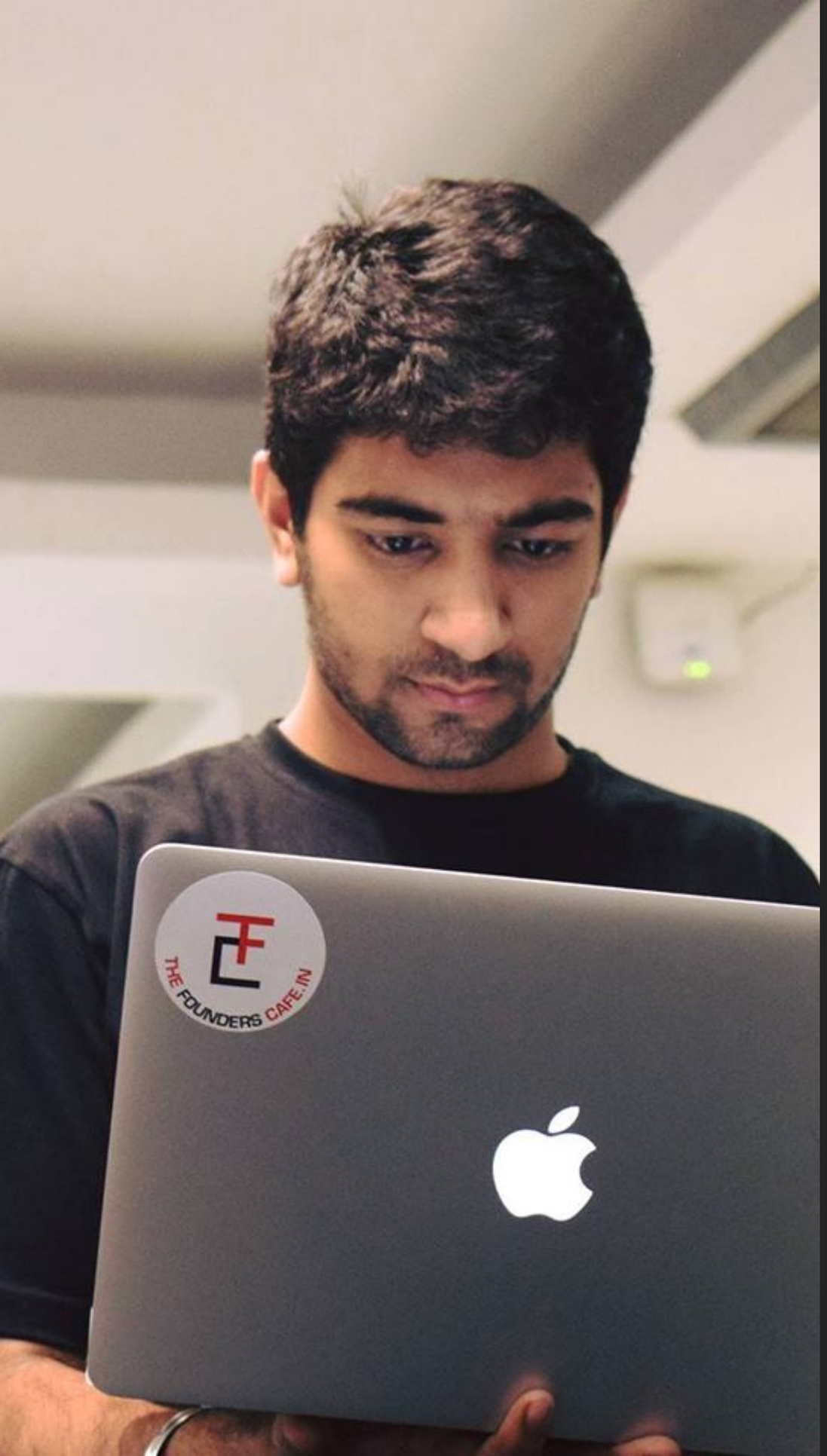
INTERNATIONAL ORGANISATION OF SOFTWARE DEVELOPERS

MACHINE LEARNING

Ronak Sakhuja

https://www.github.com/ronaksakhuja

https://www.linkedin.com/ronaksakhuja



I am a 3rd year student at Delhi Technological University. Having tried my hands on Android Development, Web Development using PHP, and now Machine learning, I have a broad scope of knowledge. I have previously interned at CISF(Central Industrial Security Forces), Le Jensoft, Echoes, Coding Ninjas (as a Machine learning content creator). I will be interning at SAP Labs this summer. I love participating in hackathons and since my 1st year I have participated in 19 hackathons and won 4 of them.

ABOUT ME

COMPUTERS ARE ABLE TO SEE, HEAR AND LEARN. WELCOME TO THE FUTURE.

Dave Waters

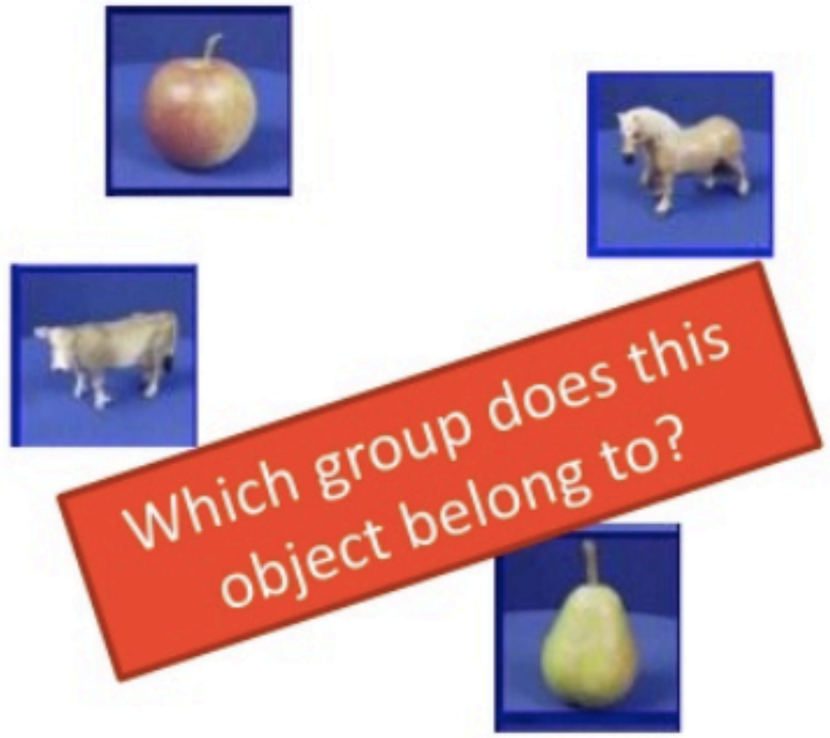
AI

ARTIFICIAL INTELLIGENCE IS THE SCIENCE AND ENGINEERING OF MAKING COMPUTERS BEHAVE IN WAYS THAT, UNTIL RECENTLY, WE THOUGHT REQUIRED HUMAN INTELLIGENCE.

ML

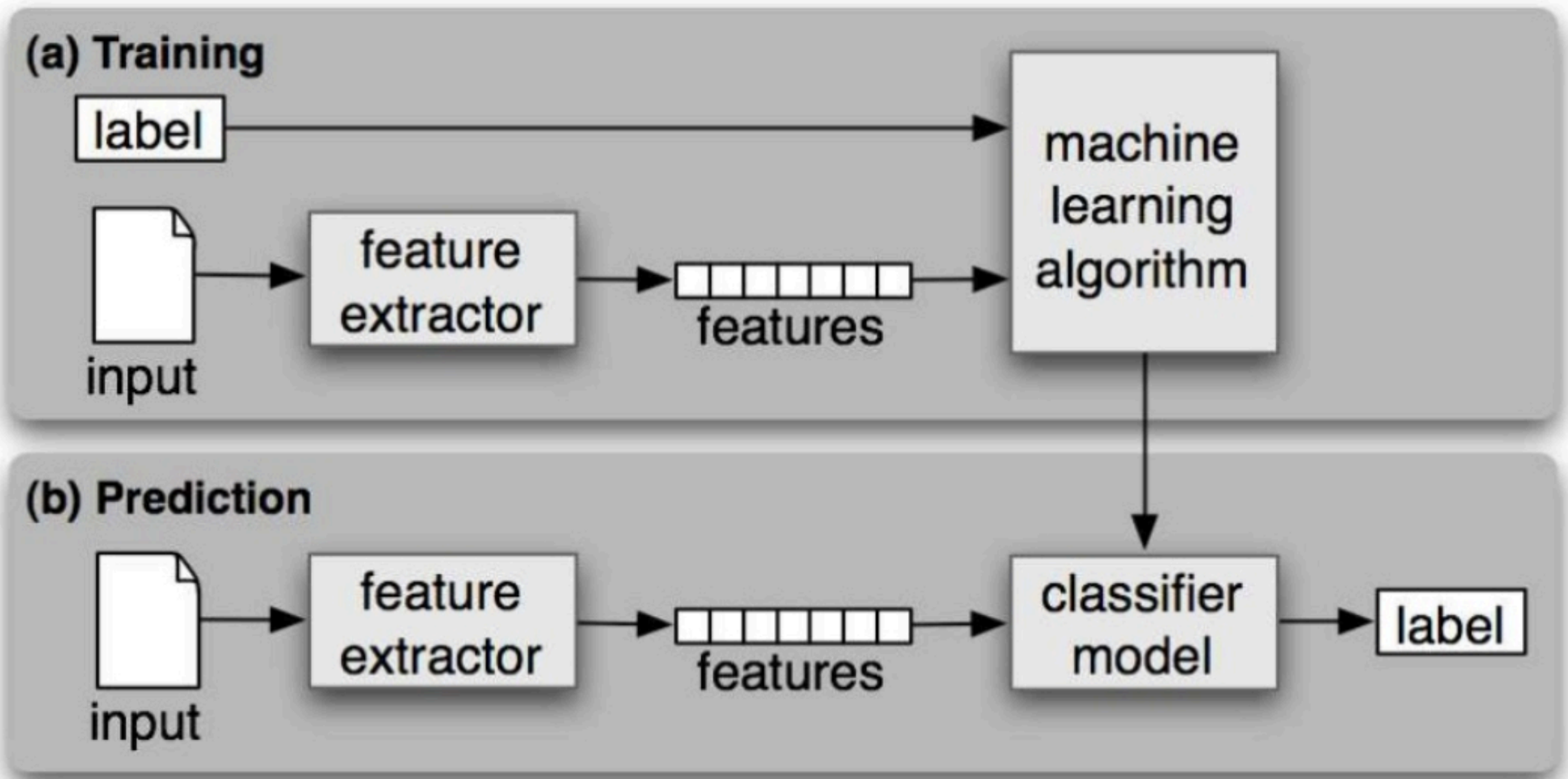
MACHINE LEARNING IS THE STUDY OF COMPUTER ALGORITHMS THAT IMPROVE AUTOMATICALLY THROUGH EXPERIENCE.

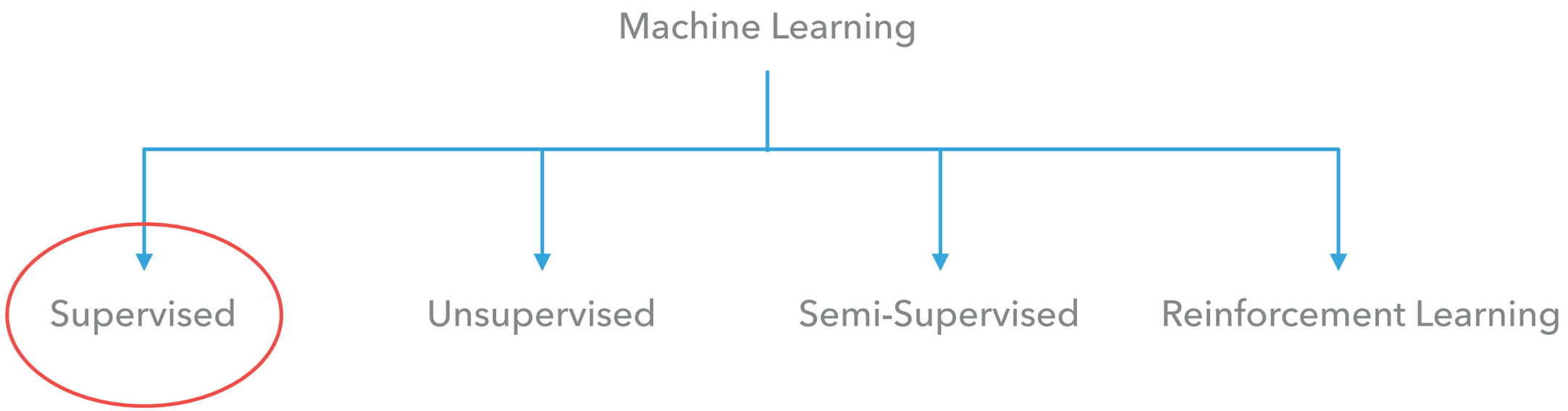
The Learning



WHAT IS MACHINE LEARNING?

- How does our brain process an image?
- How does grouping happen?
- Human brain processed the given images - learning
- After learning, the brain simply looked at the new images and compared with the groups classified the image to the closest group - Classification
- If a machine has to perform the same operation, we use machine learning.
- We are writing programs for learning and then classification, this is nothing but machine learning.


(a) Training
label
input
feature extractor
features
machine learning algorithm
(b) Prediction
input
feature extractor
features
classifier model
label

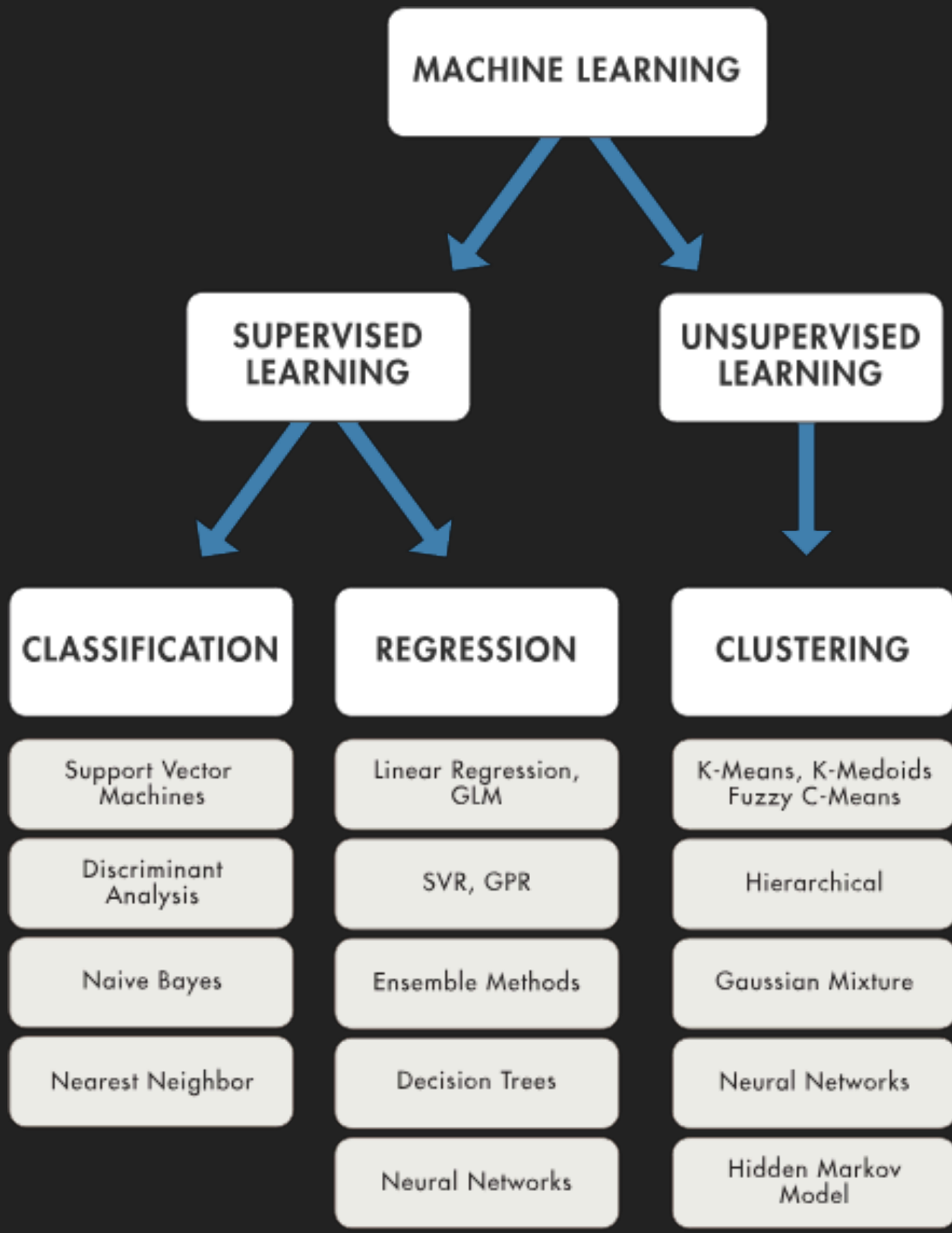


The correct classes of the training data are known

The correct classes of the training data are not known

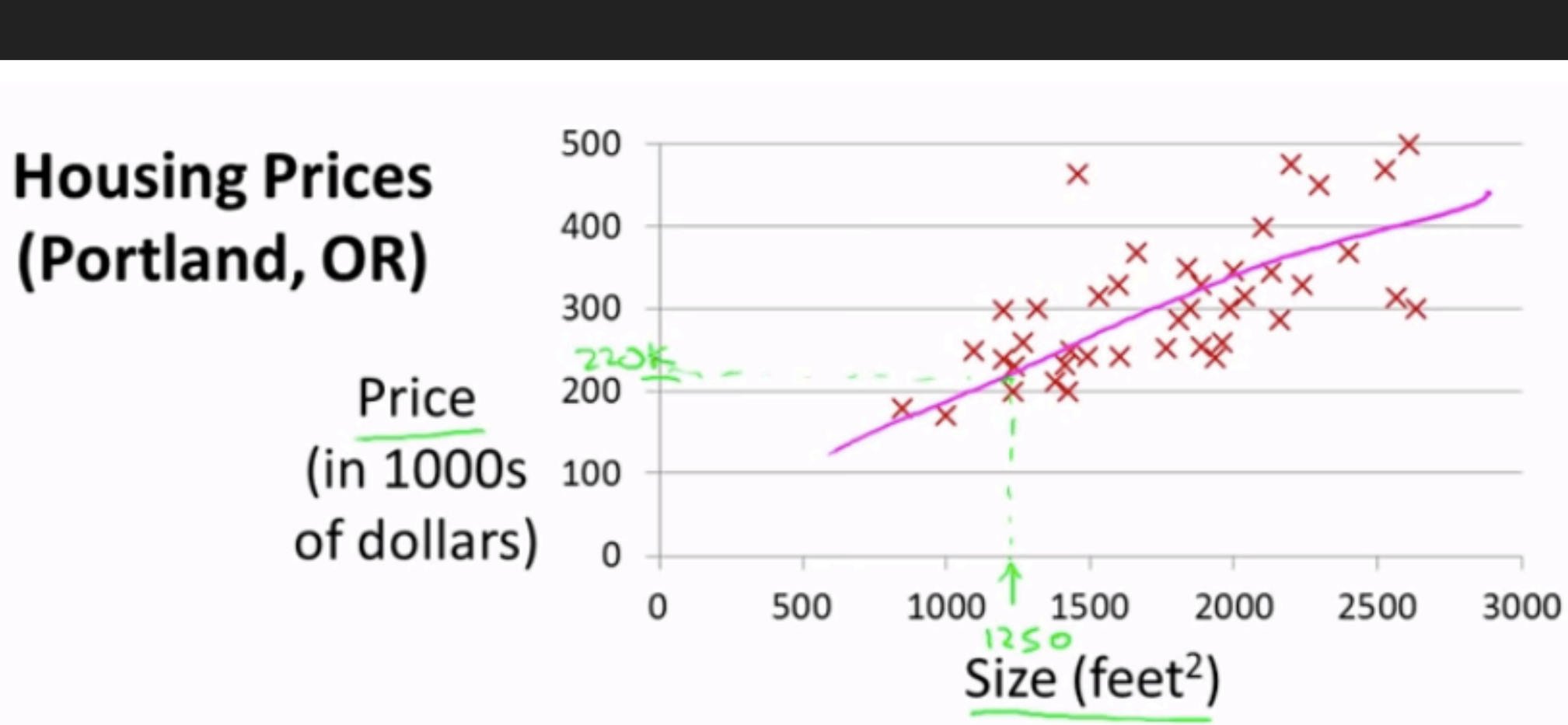
This is a mixture of Supervised and unsupervised.

Allows the machine or software agent to learn based on the feedback from the environment.


MACHINE LEARNING
SUPERVISED LEARNING
UNSUPERVISED LEARNING
CLASSIFICATION
Support Vector Machines
Discriminant Analysis
Naive Bayes
Nearest Neighbor
REGRESSION
Linear Regression, GLM
SVR, GPR
Ensemble Methods
Decision Trees
Neural Networks
CLUSTERING
K-Means, K-Medoids Fuzzy C-Means
Hierarchical
Gaussian Mixture
Neural Networks
Hidden Markov Model

LINEAR REGRESSION

- Evaluating trends
- Finding relationship between two variables.



Training set of housing prices (Portland, OR)

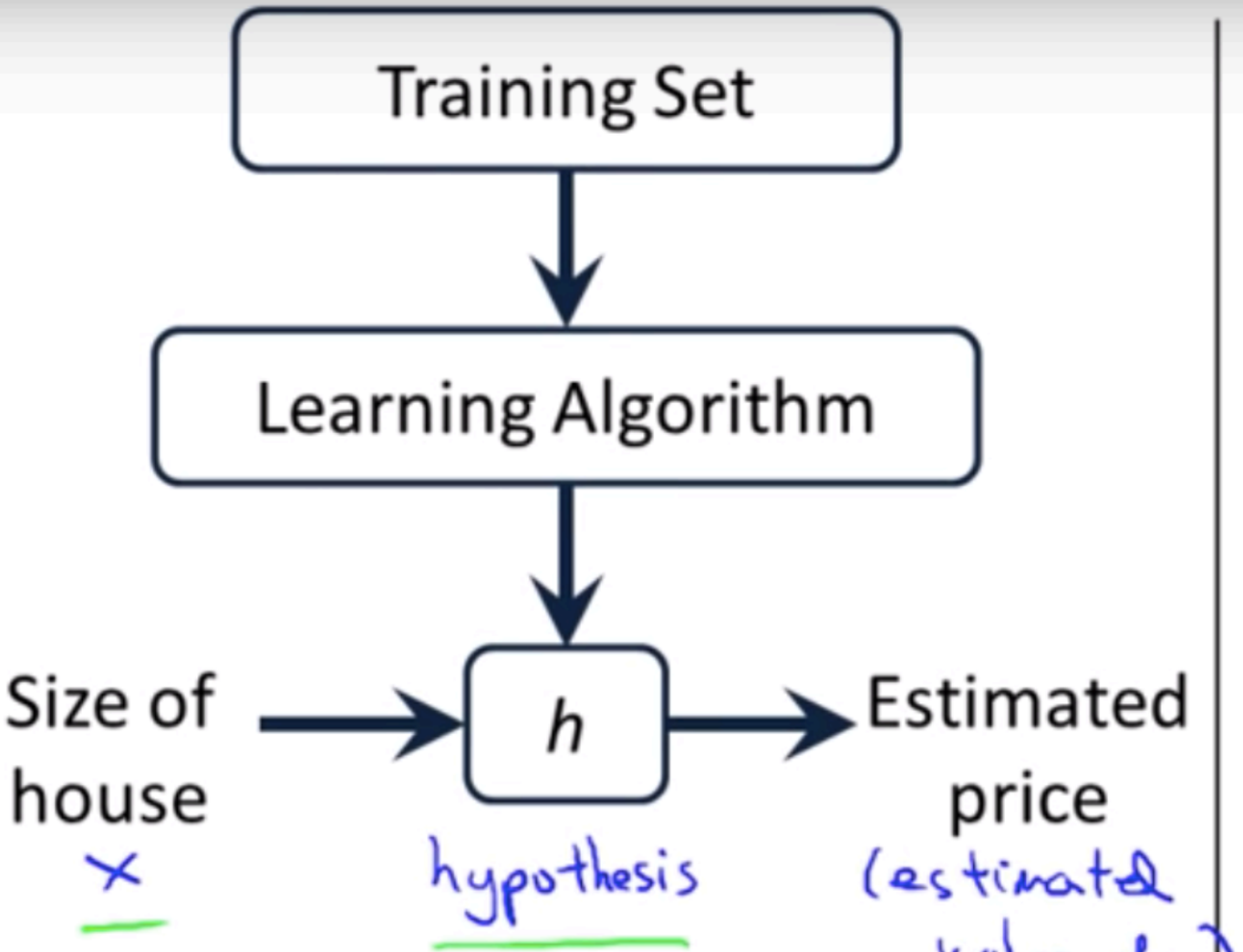
Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Notation:

m = Number of training examples

x's = "input" variable / features

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

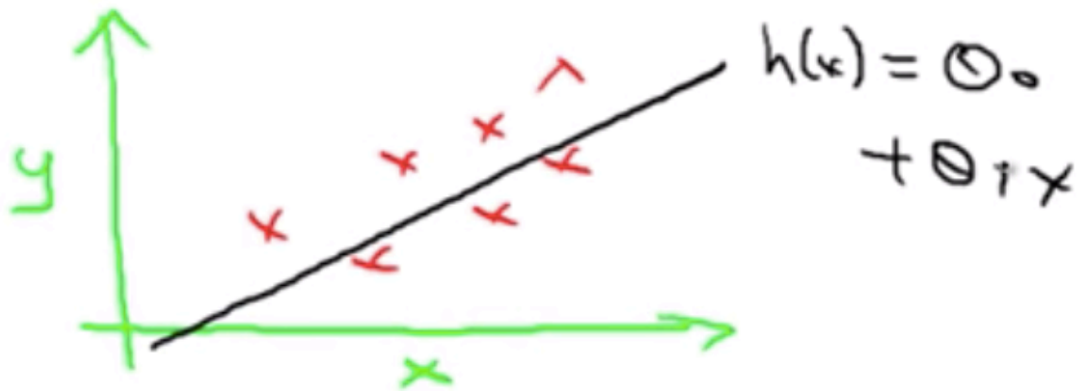


h maps from x's to y's.

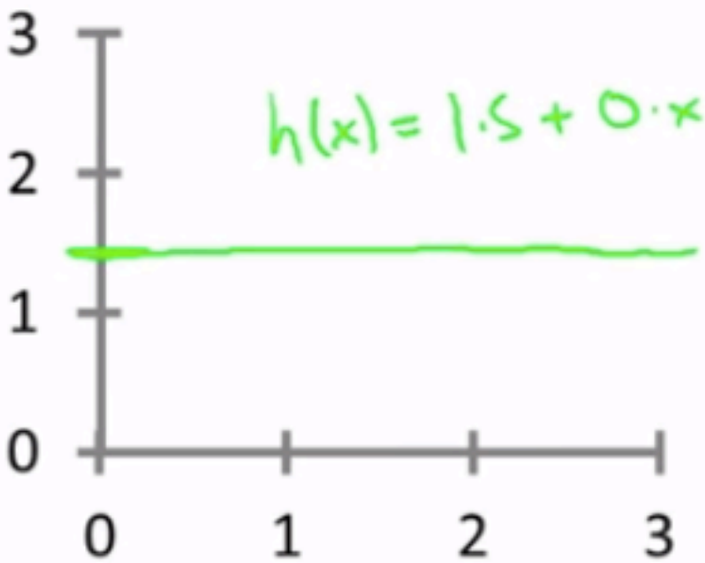
How do we represent *h* ?

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

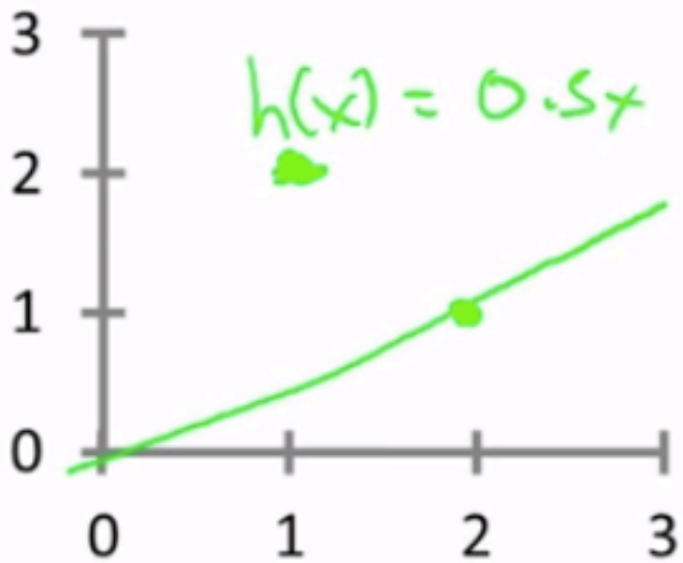
Shorthand: $h(x)$



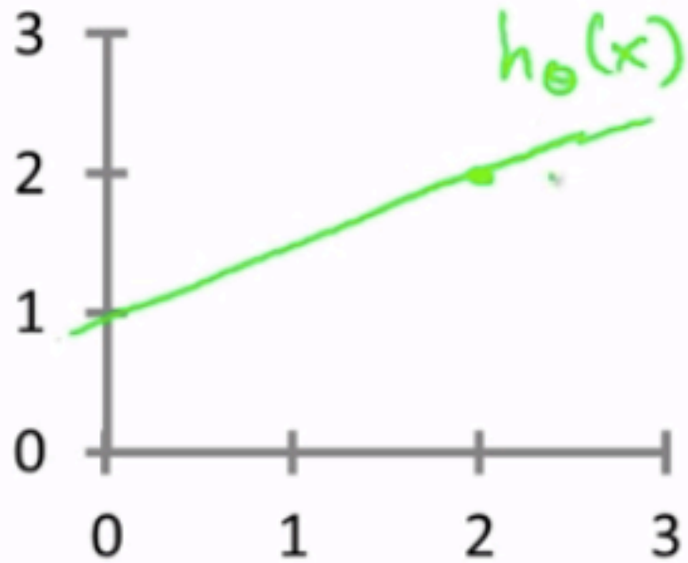
$$h_\theta(x) = \theta_0 + \theta_1 x$$



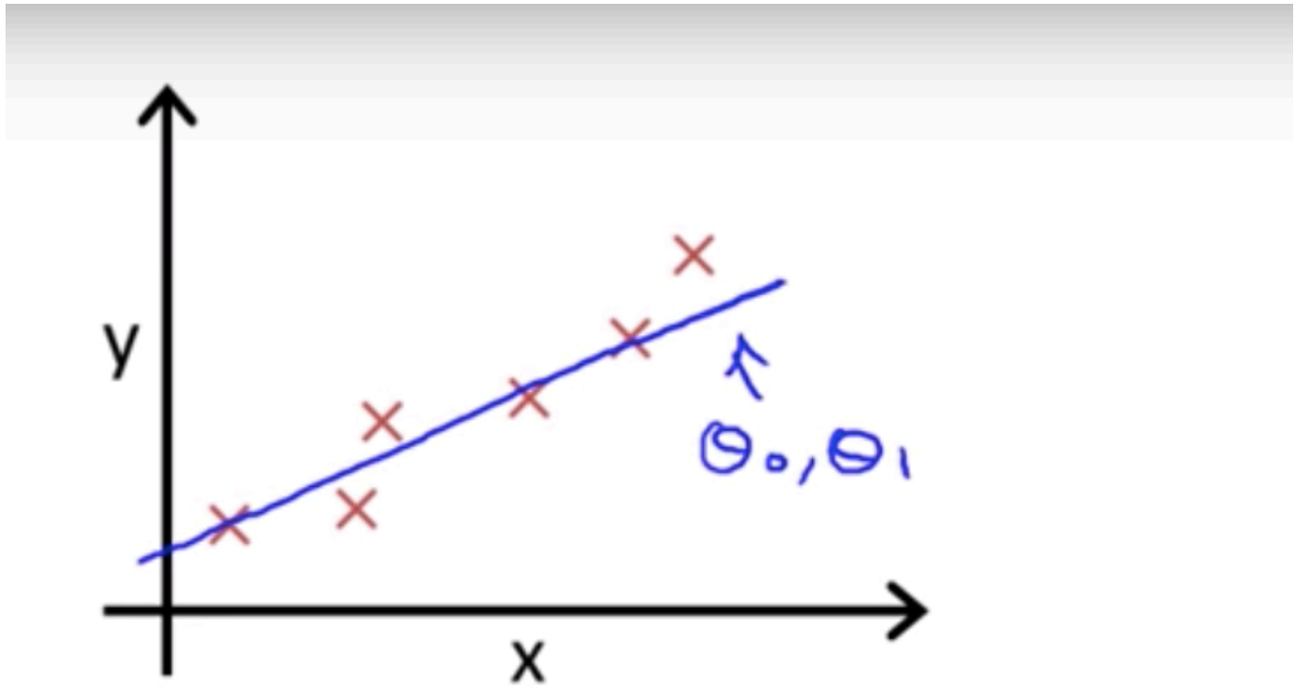
→ $\theta_0 = 1.5$
→ $\theta_1 = 0$



→ $\theta_0 = 0$
→ $\theta_1 = 0.5$



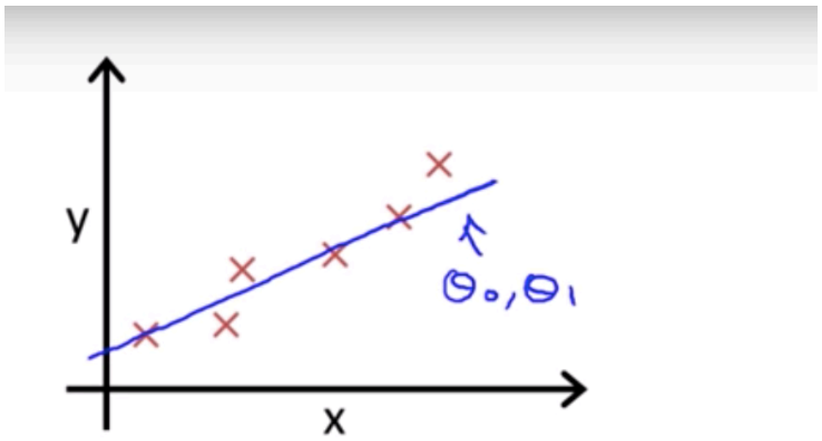
→ $\theta_0 = 1$
→ $\theta_1 = 0.5$



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

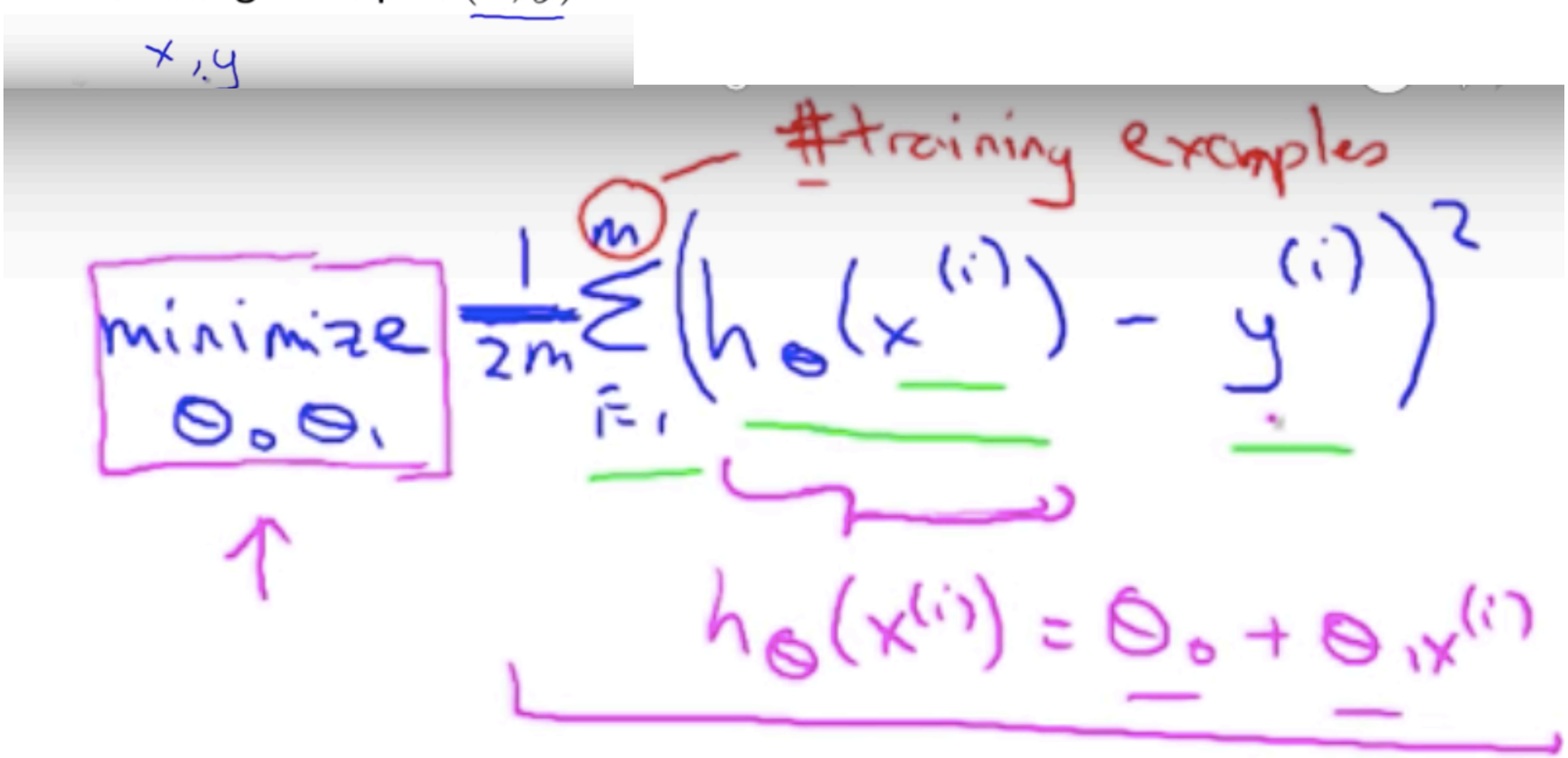
x, y

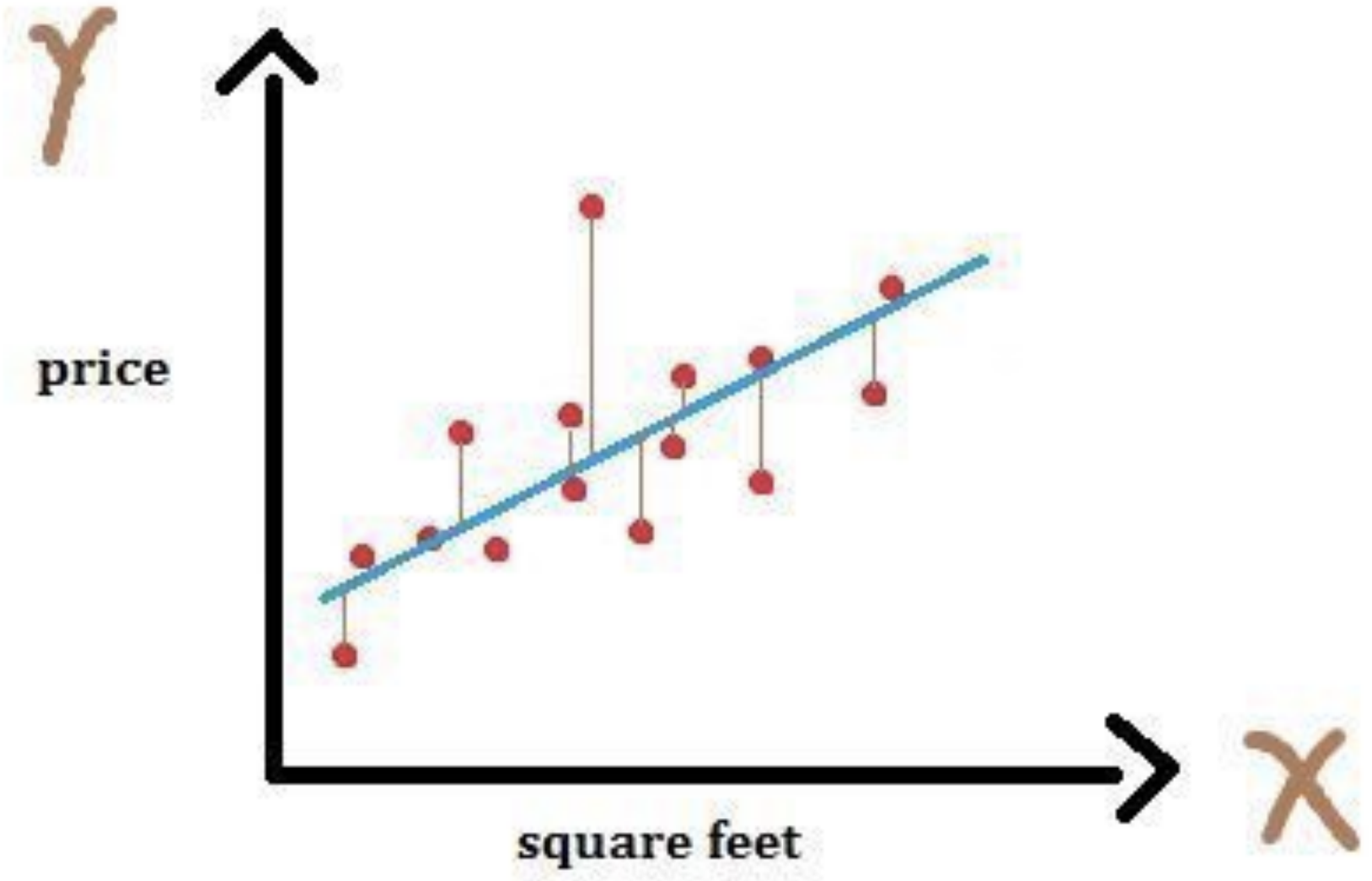
Squared Error Function



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

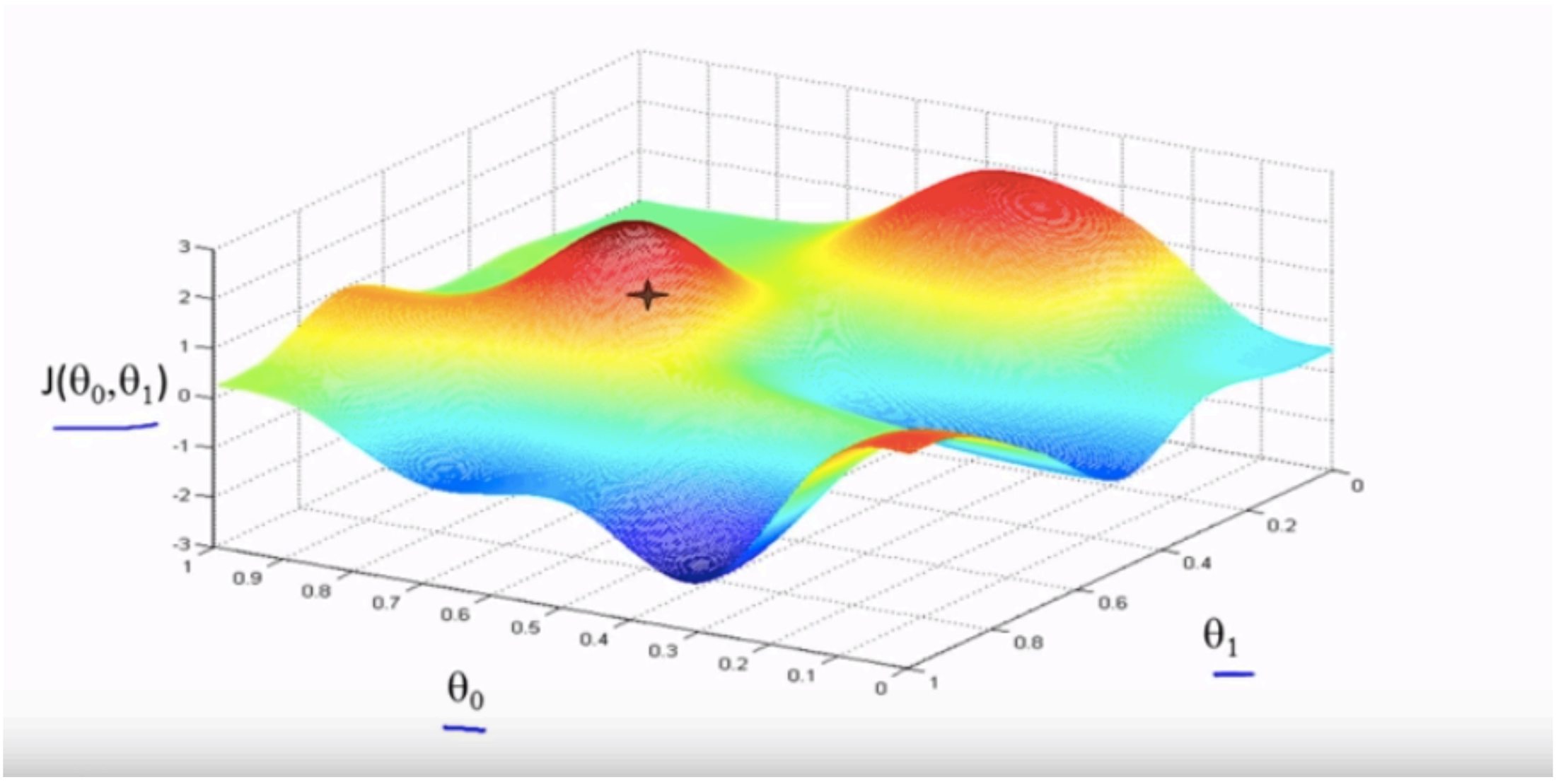
x, y

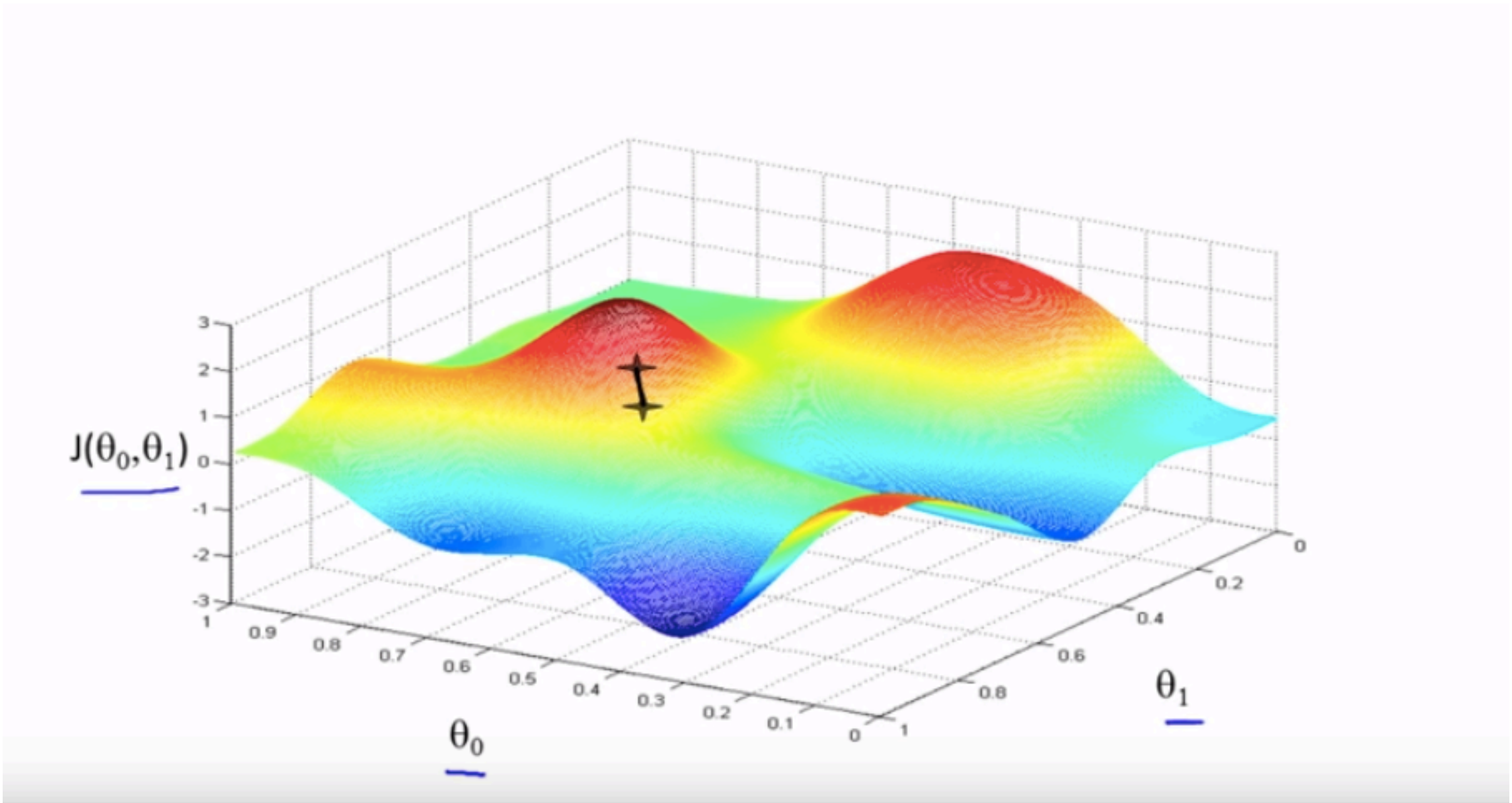


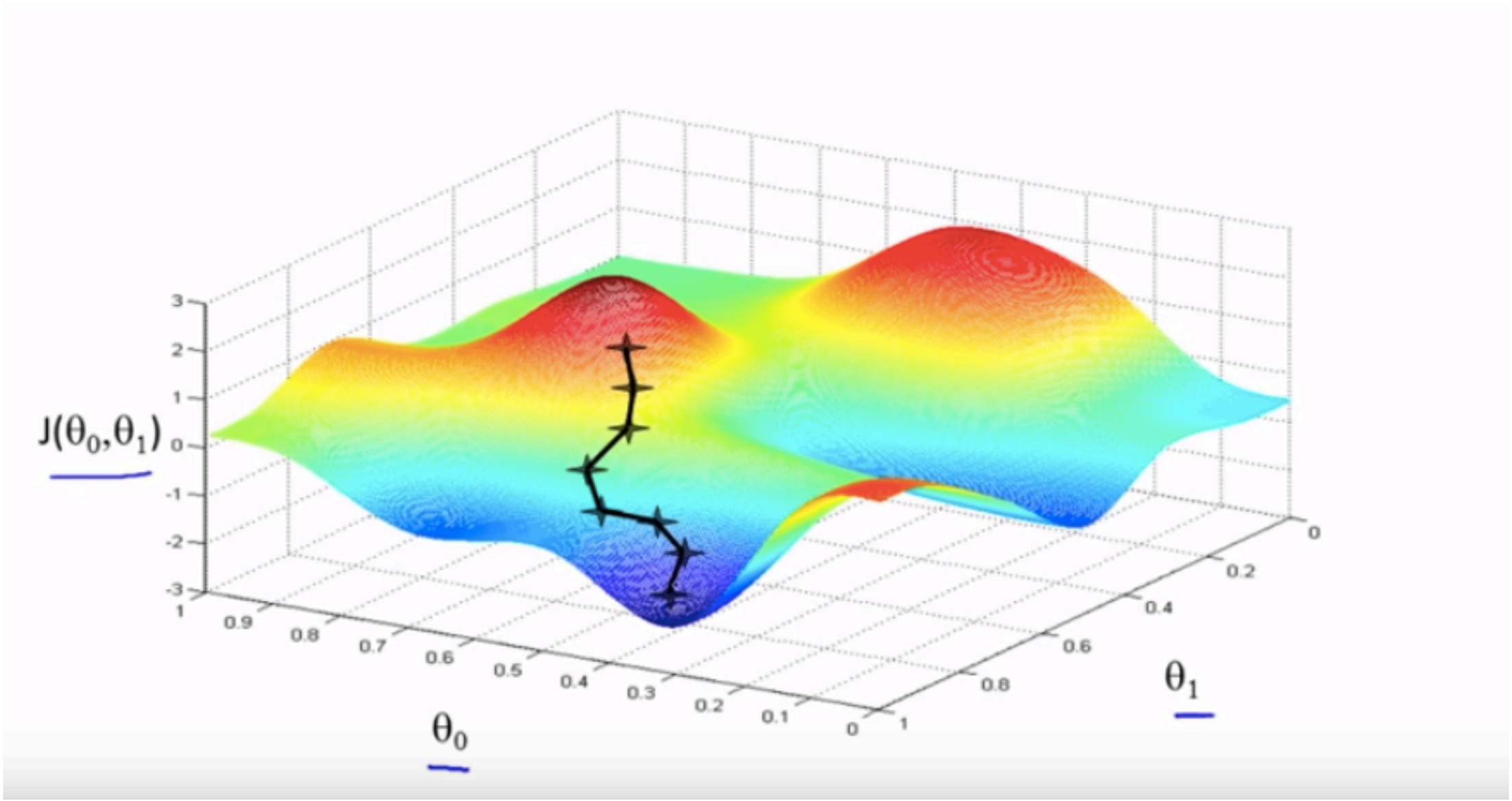


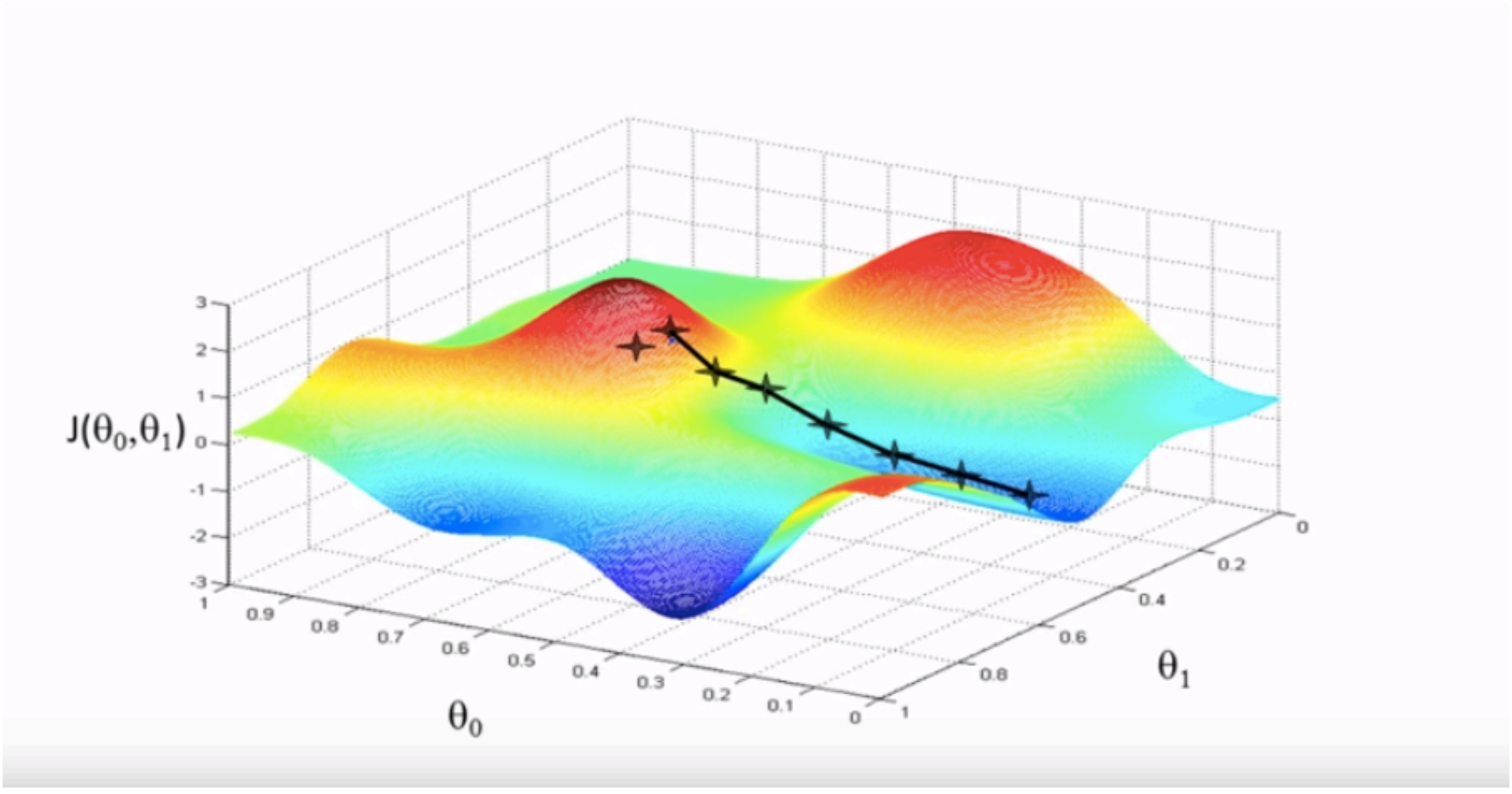
Cost Function

HOW TO EFFICIENTLY MINIMIZE COST FUNCTION?


$J(\theta_0,\theta_1)$
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

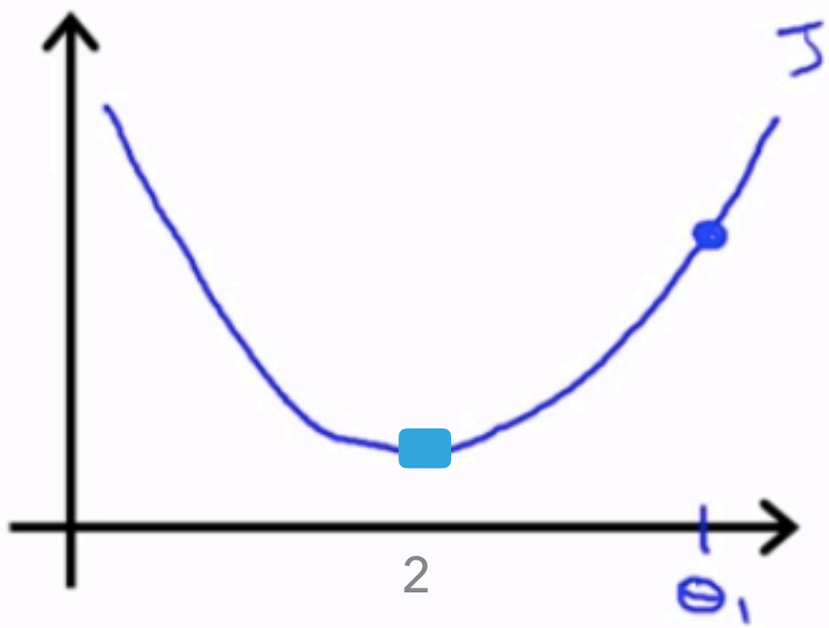
repeat until convergence {

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

}

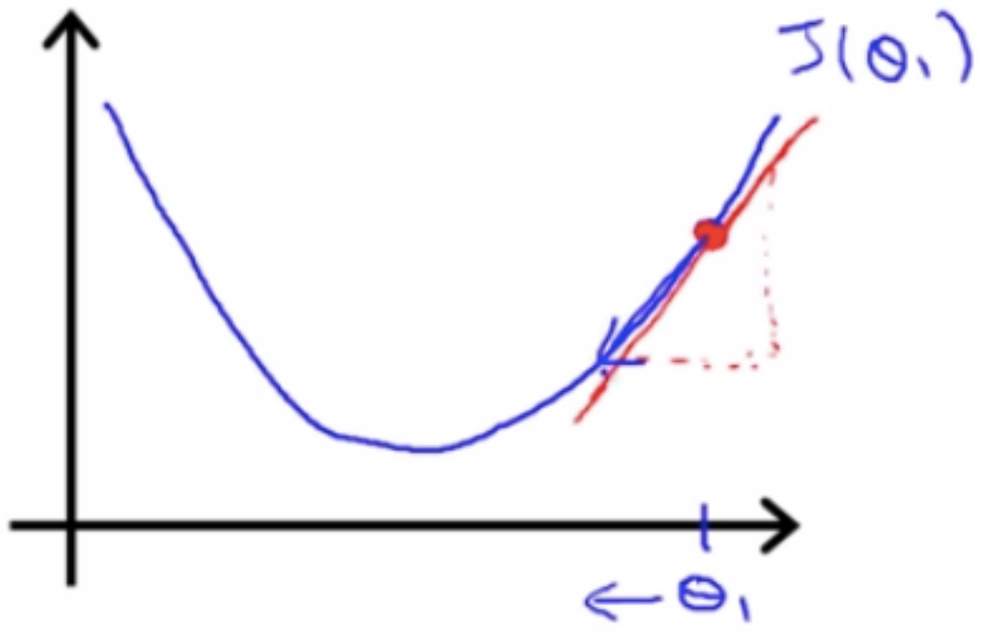
Learning Rate

Slope



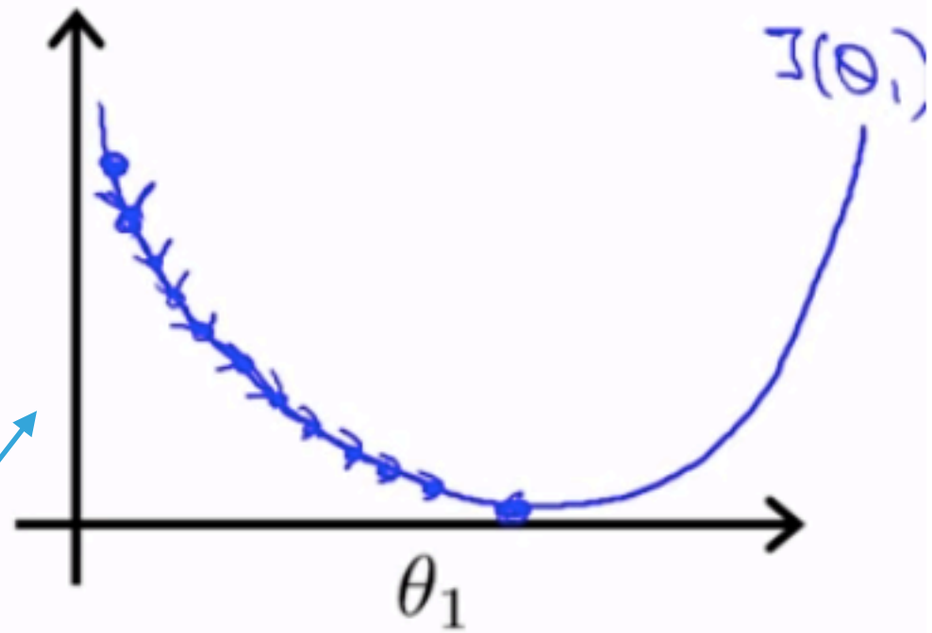
$J(\theta_1) \quad (\theta_1 \in \mathbb{R})$

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

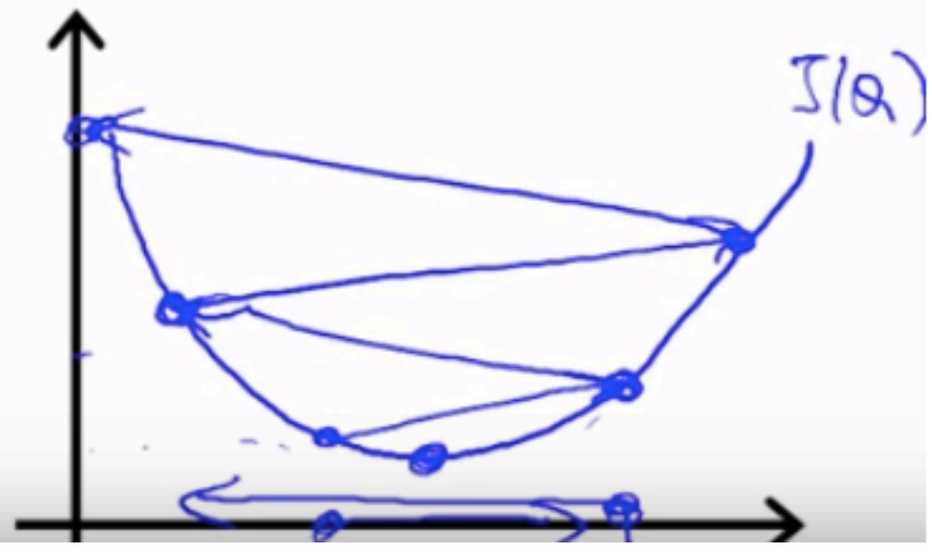


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

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



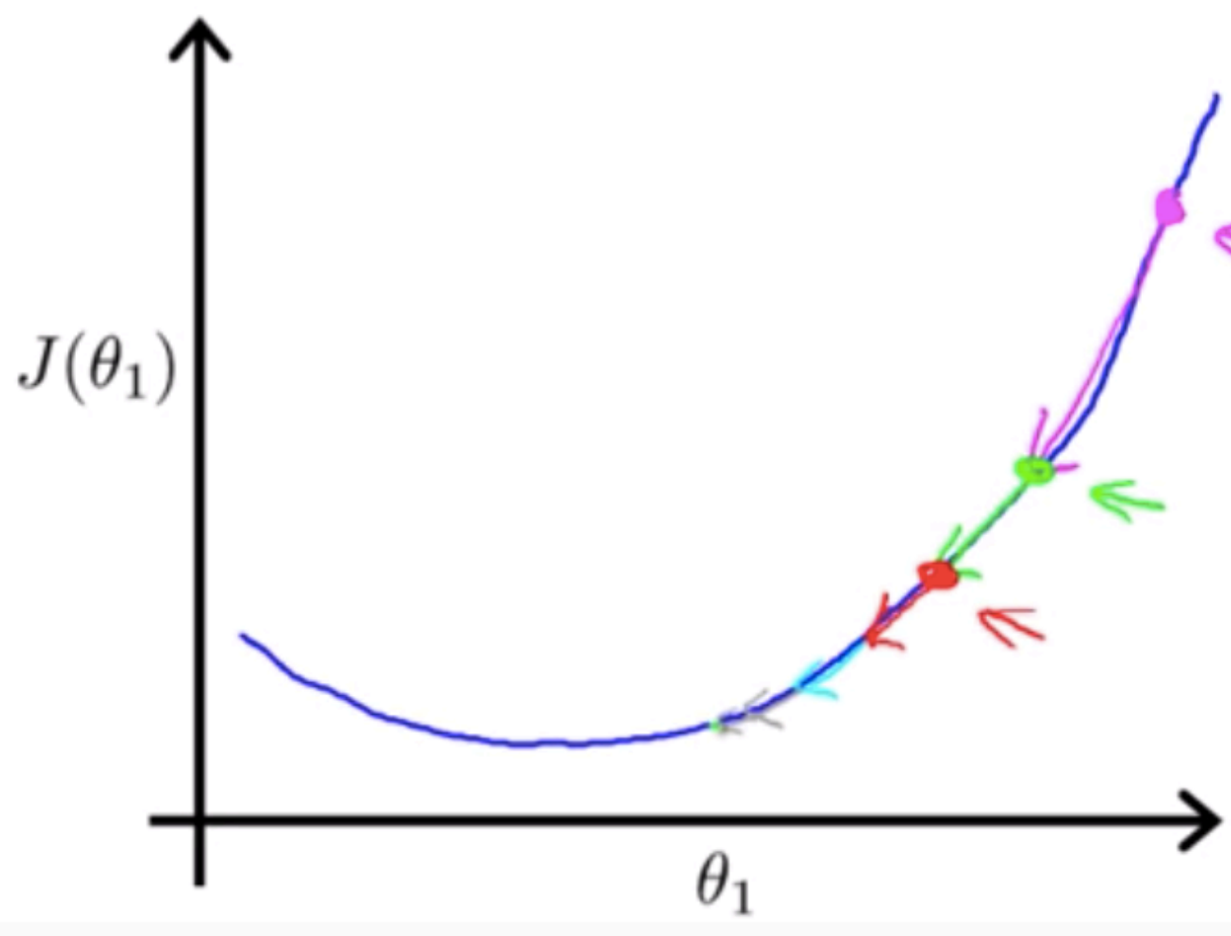
If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



Gradient descent 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.



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

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

θ_0 $j = 0$: $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$

θ_1 $j = 1$: $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}$

Gradient descent algorithm

repeat until convergence {

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

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

}

LET'S CODE

https://colab.research.google.com