Machine Learning od podstaw

Michał Lipka SAP
20 Listopada, 2019

PUBLIC

Hack Your Career

Fit yourself in IT – Rozegraj swoją rekrutację

15.10.2019
Marek Nawa
Maciej Ogrodnik

Getting started with OAUTH 2.0

29.10.2019
Tomasz Miler

Machine Learning **od podstaw**

20.11.2019
Michał Lipka

Building a city with SCRUM - Workshop

12.12.2019
Michał Drzewiecki

SAP Labs Poland

Top ecommerce, marketing, billing

Development: Go, Java, Cloud Native solutions

> 400 pracowników

Najlepszy Pracodawca w rankingu AON

Jedno z 20 centrów **SAP's Labs Network**

Agenda:

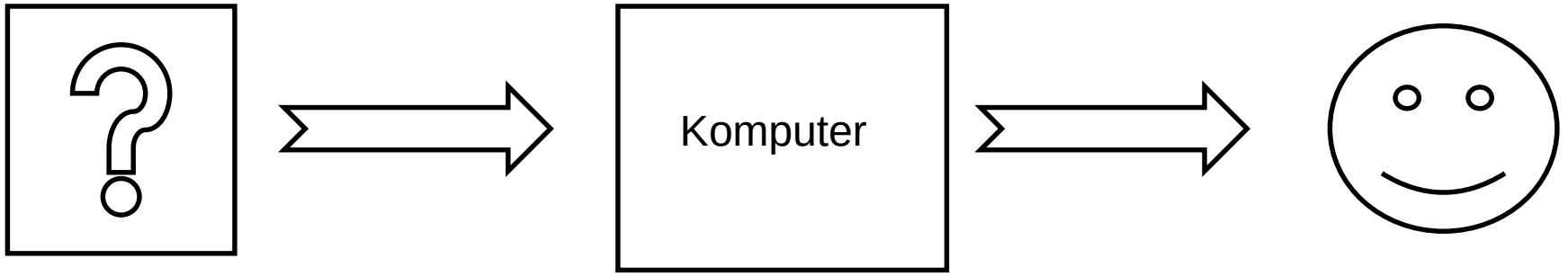
- Co to tak w ogóle jest machine learning?
- Sposoby uczenia
- Reprezentacja modelu: hipoteza, funkcja kosztu, gradient prosty
- Regresja liniowa
- Klasyfikacja
- Sieci neuronowe

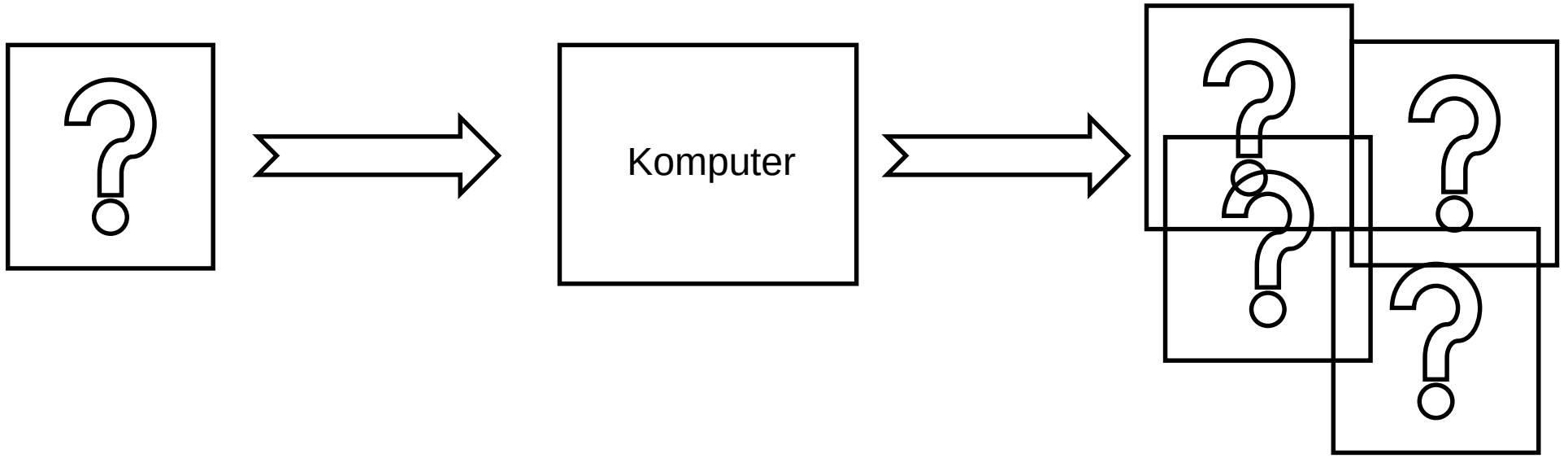
”The field of study that gives computers the ability to learn without being explicitly programmed."

Arthur Samuel

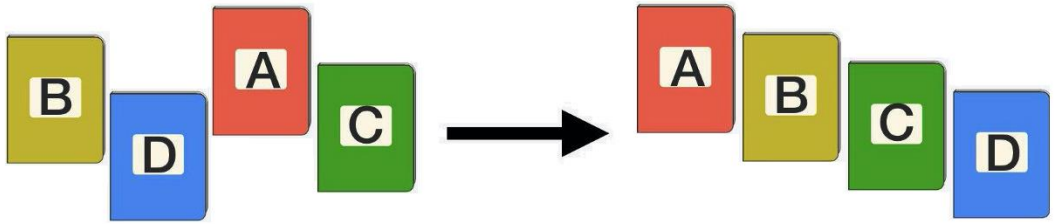
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.”

Tom M. Mitchell

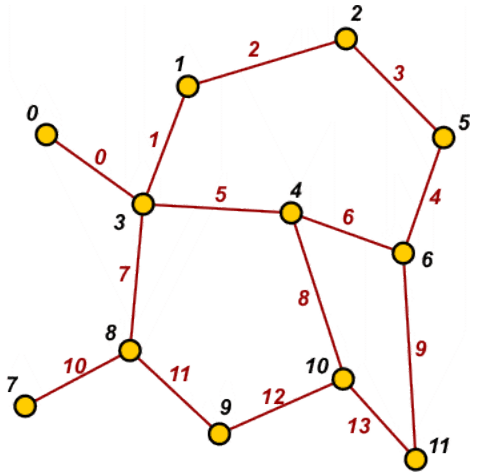




Sortowanie tablic



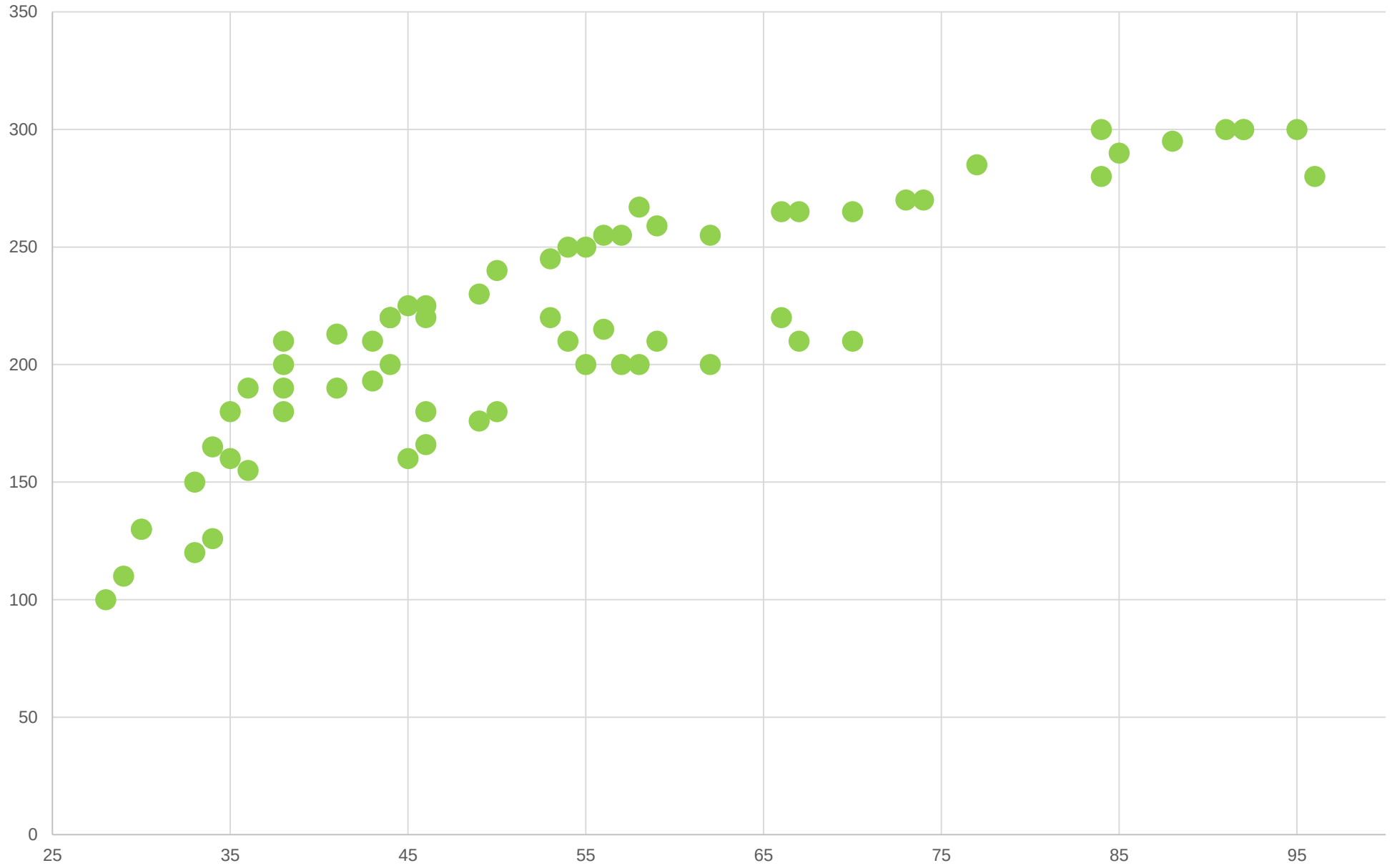
Teoria grafów



Rodzaje uczenia:

- Nadzorowane
- Nienadzorowane
- Inne: drzewa decyzji, uczenie przez wzmacnianie

Hipoteza

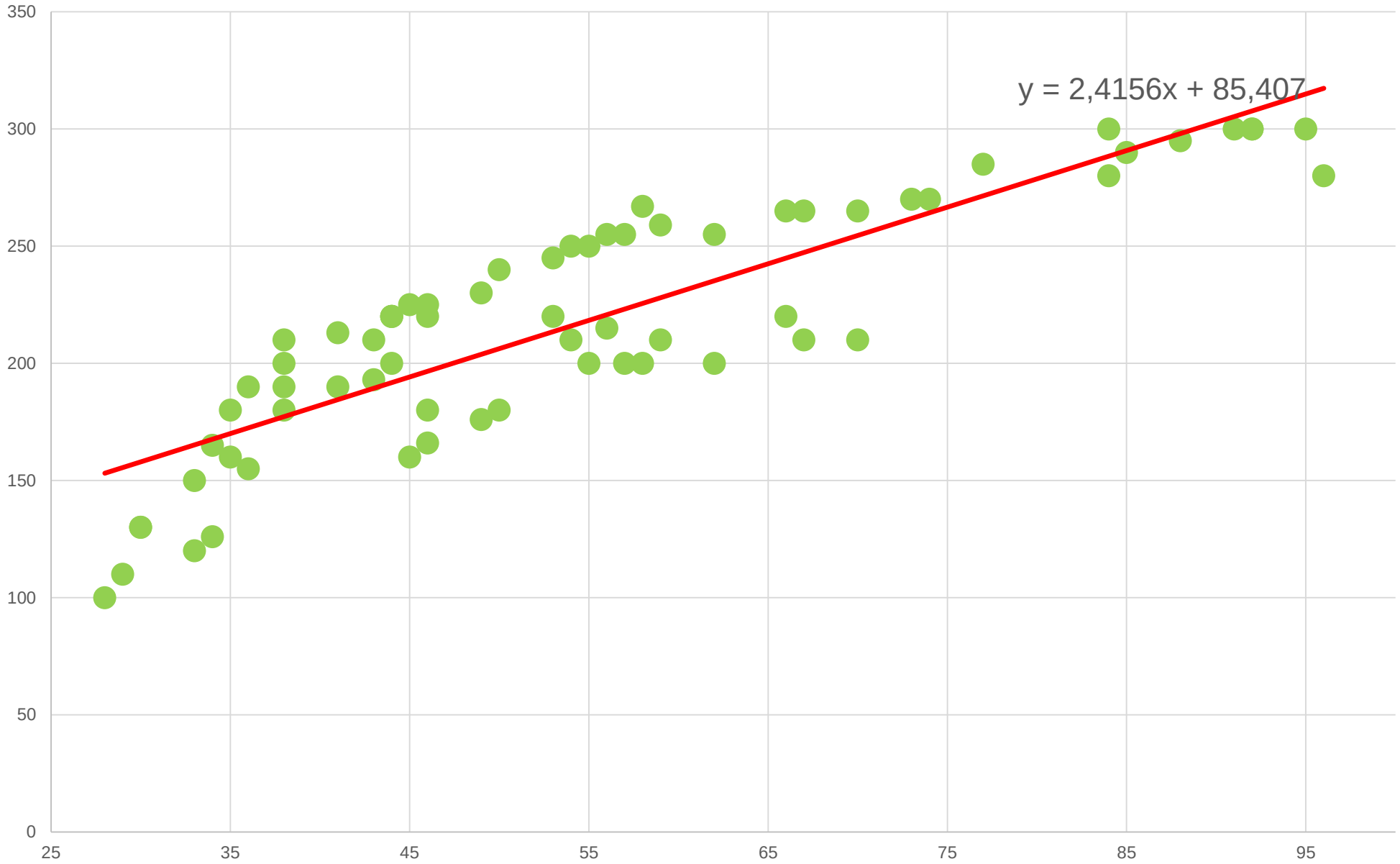


Regresja liniowa

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

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

	X (rozmiar w m2)	Y (cena w tyś)
$x^{(1)}, y^{(1)}$	28	100
$x^{(2)}, y^{(2)}$	29	110
$x^{(3)}, y^{(3)}$	31	130
$x^{(4)}, y^{(4)}$	31	133
$x^{(i)}, y^{(i)}$	…	…

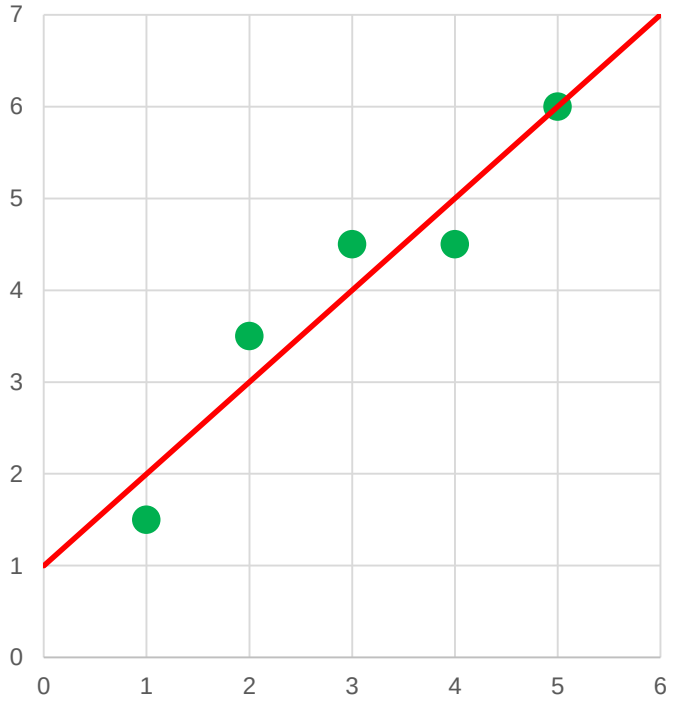


Dwa pytania:

- skąd wiemy, że dana prosta jest „dobra” ?

- jak ją policzyć?

Funkcja kosztu $J(\theta)$



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

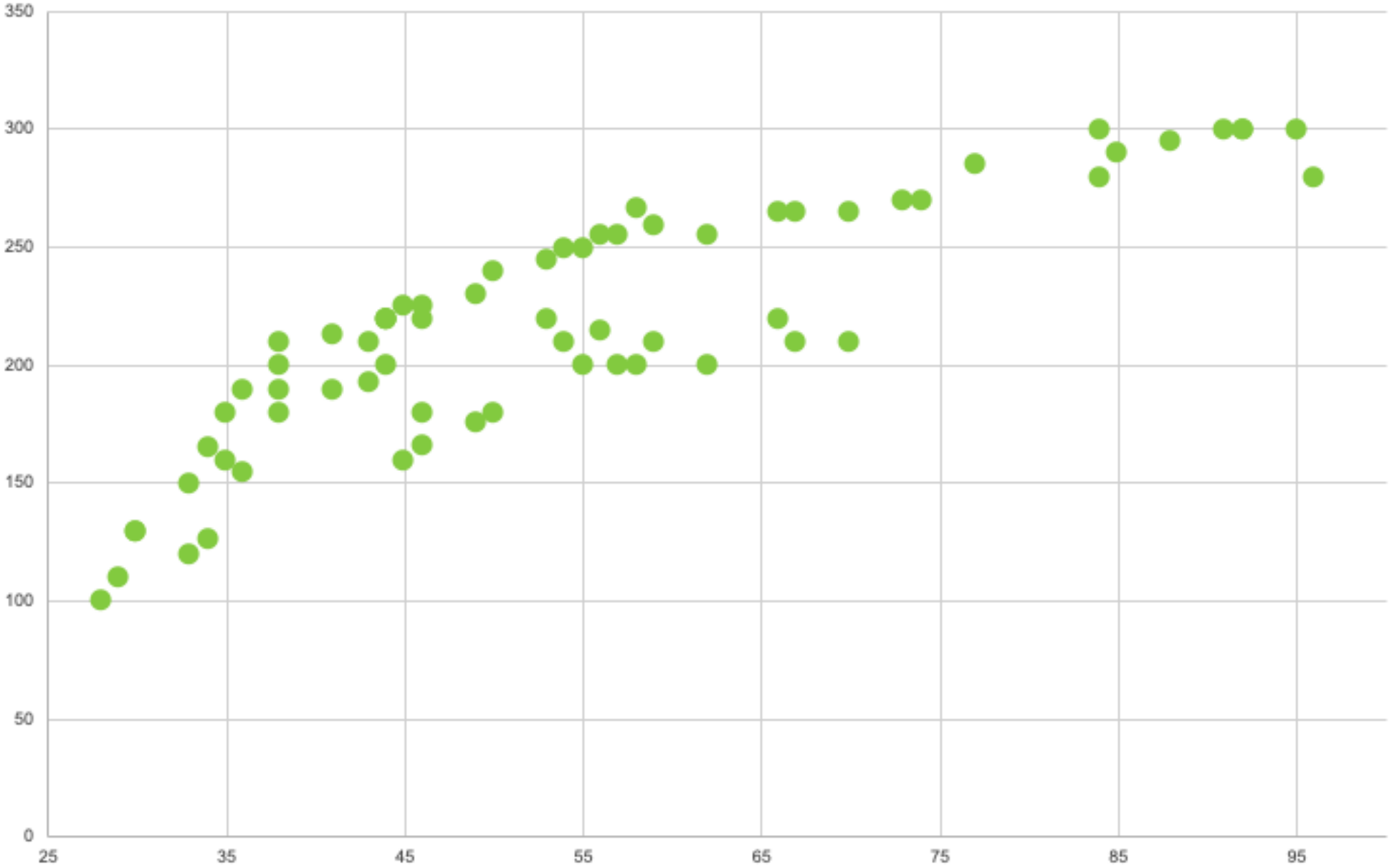
$$J(\theta_0, \theta_1) = \qquad h_\theta\left(x^{(i)}\right) - y^{(i)}$$

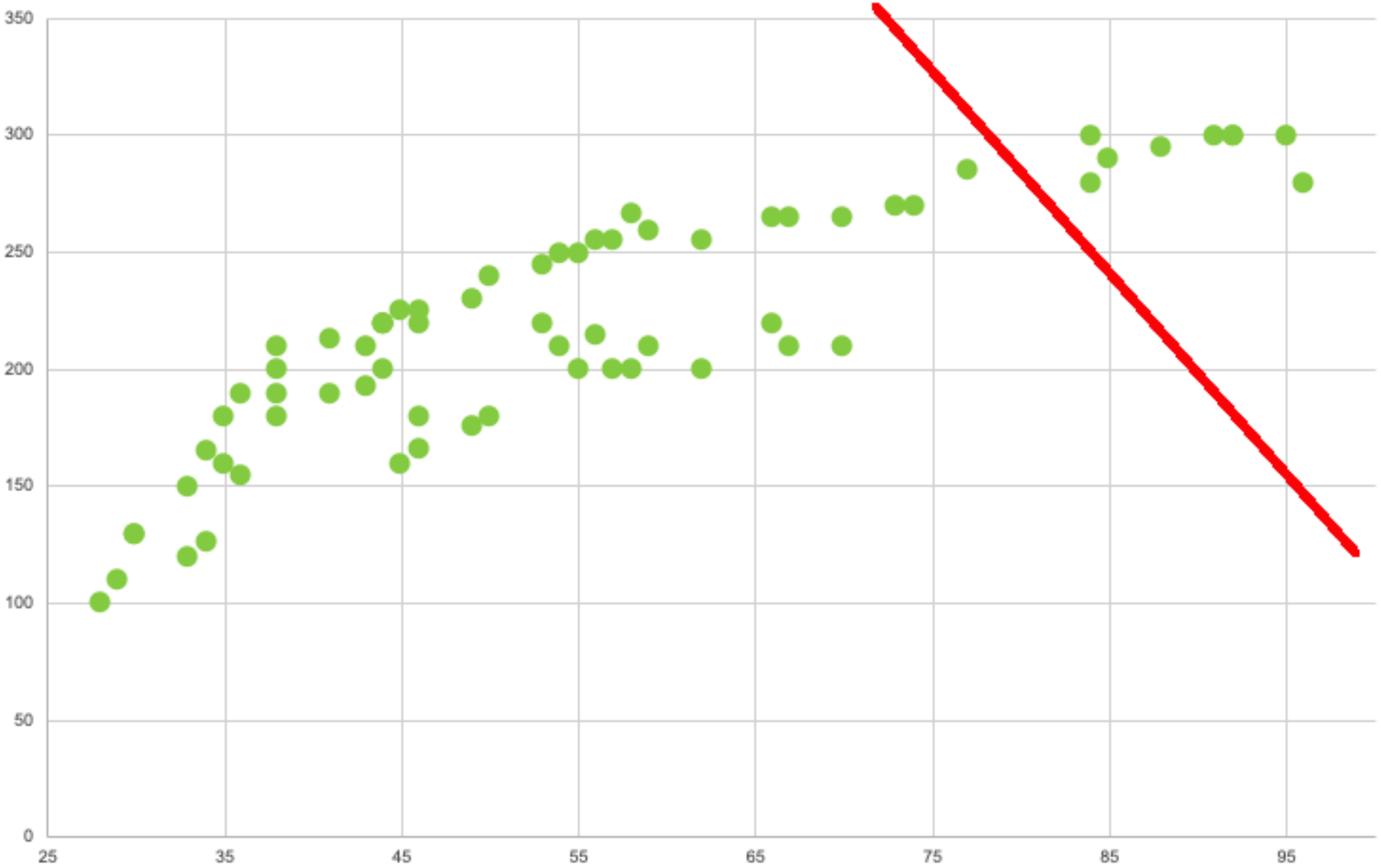
$$J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

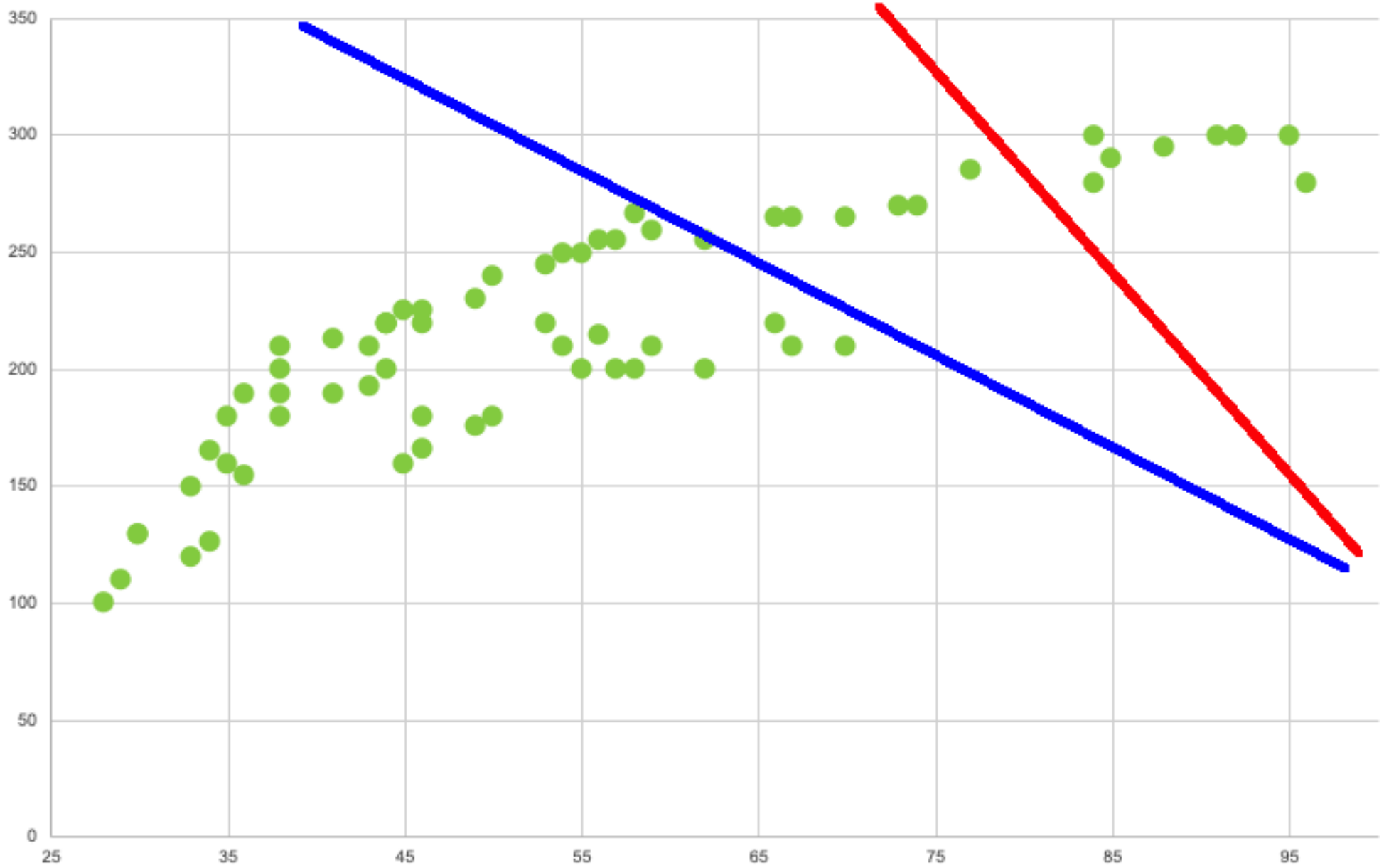
$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

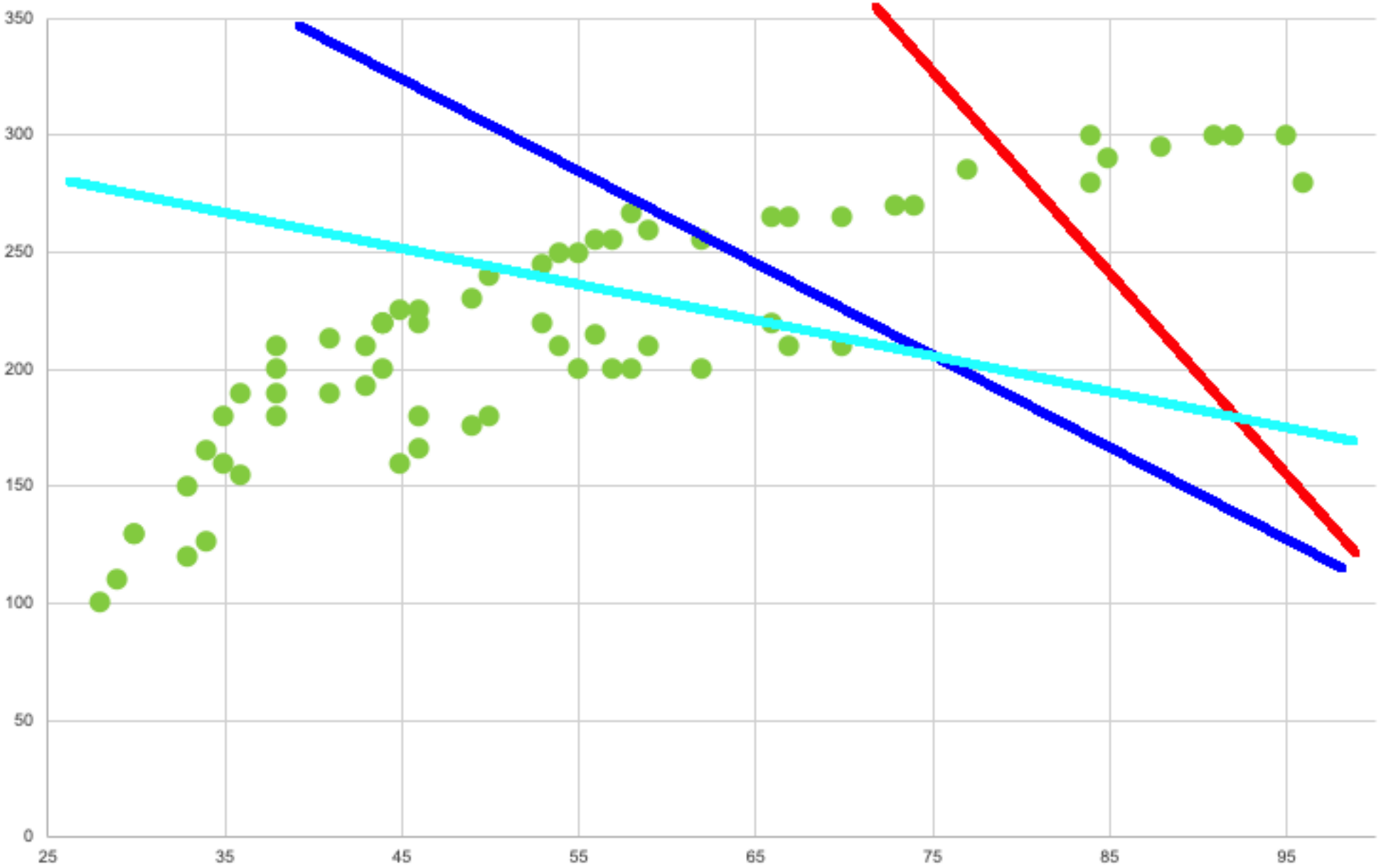
$$\underset{(\theta_0,\theta_1)}{\text{minimaze}}\, J(\theta_0, \theta_1)$$

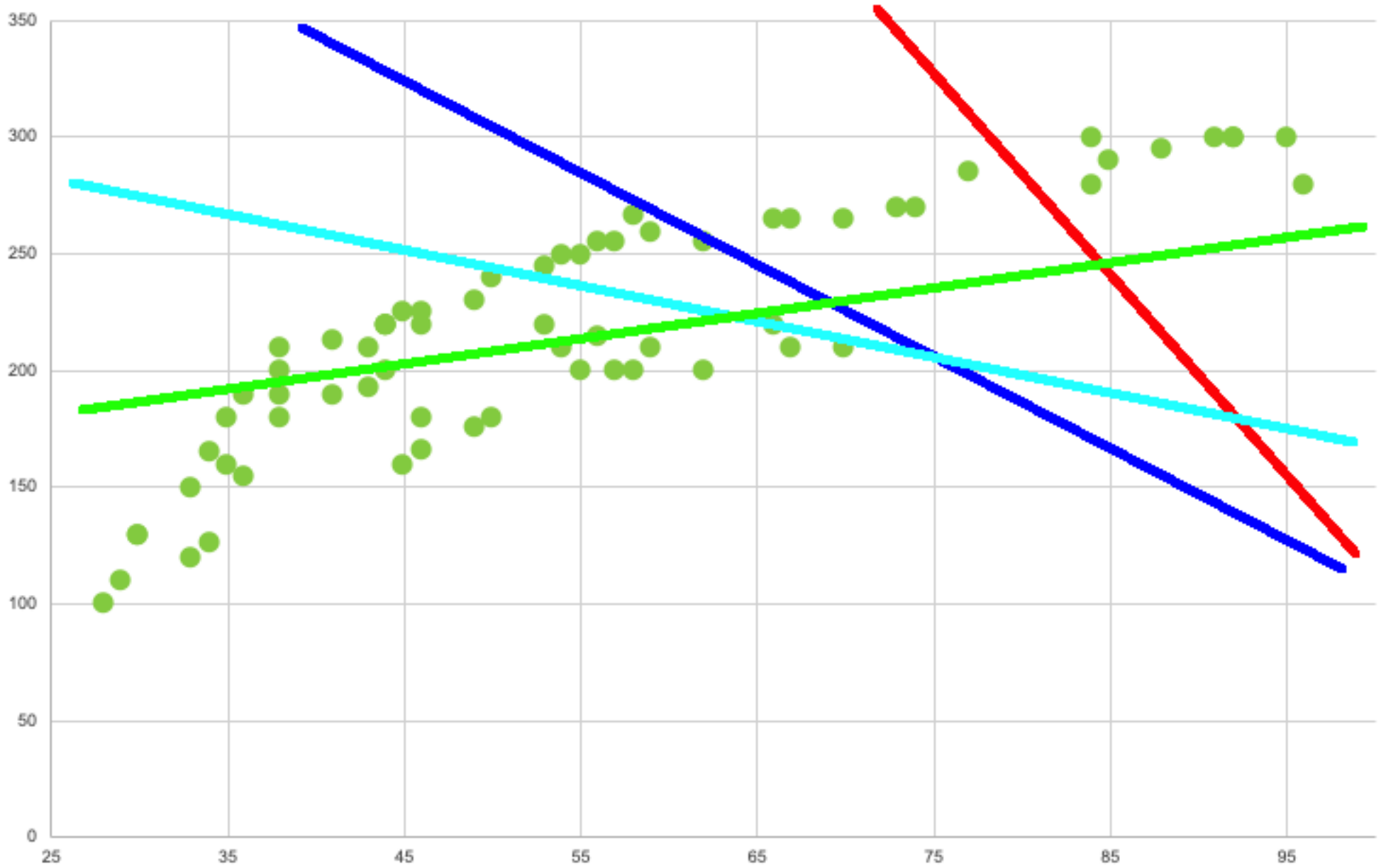
Jak wygląda proces uczenia?

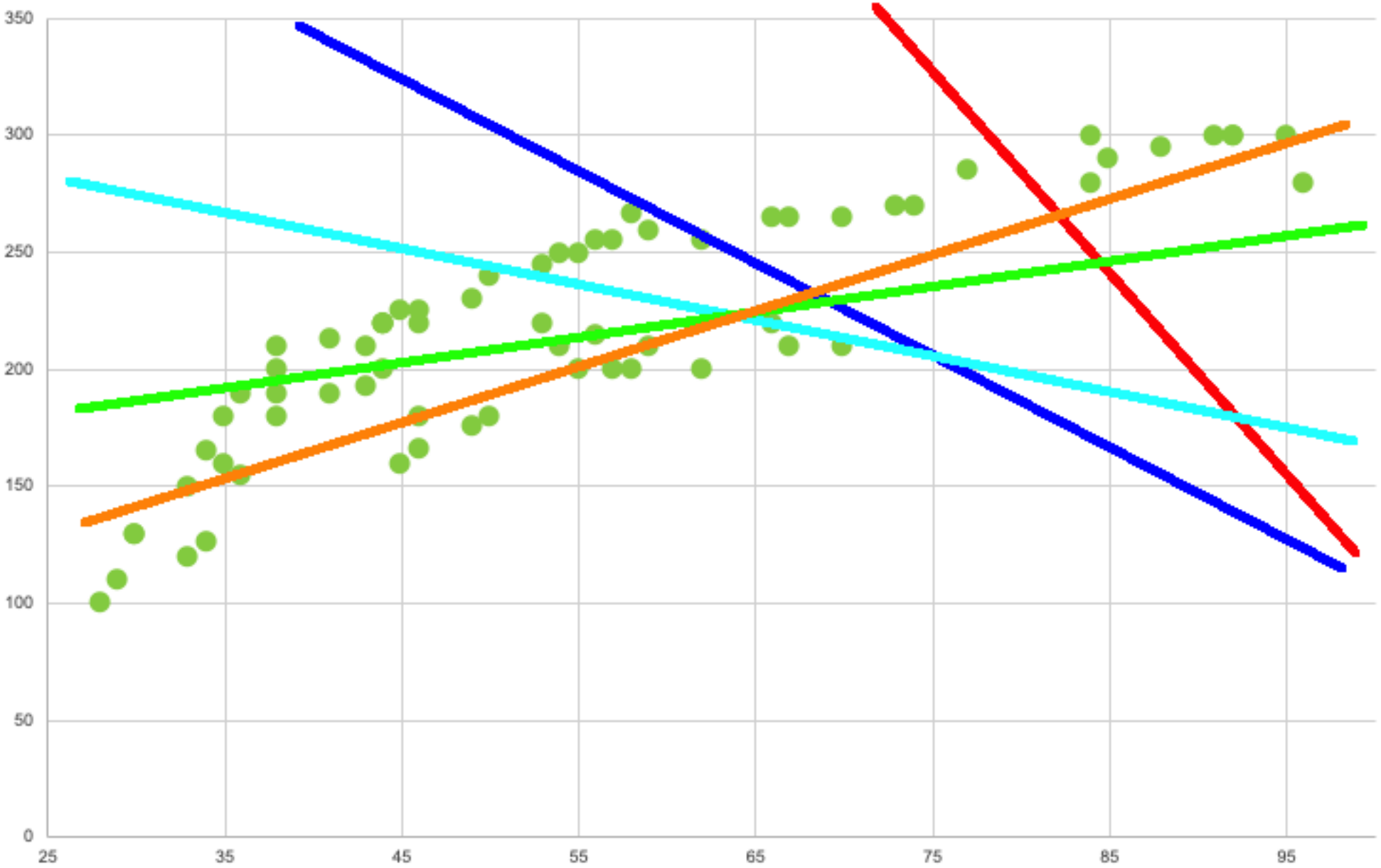


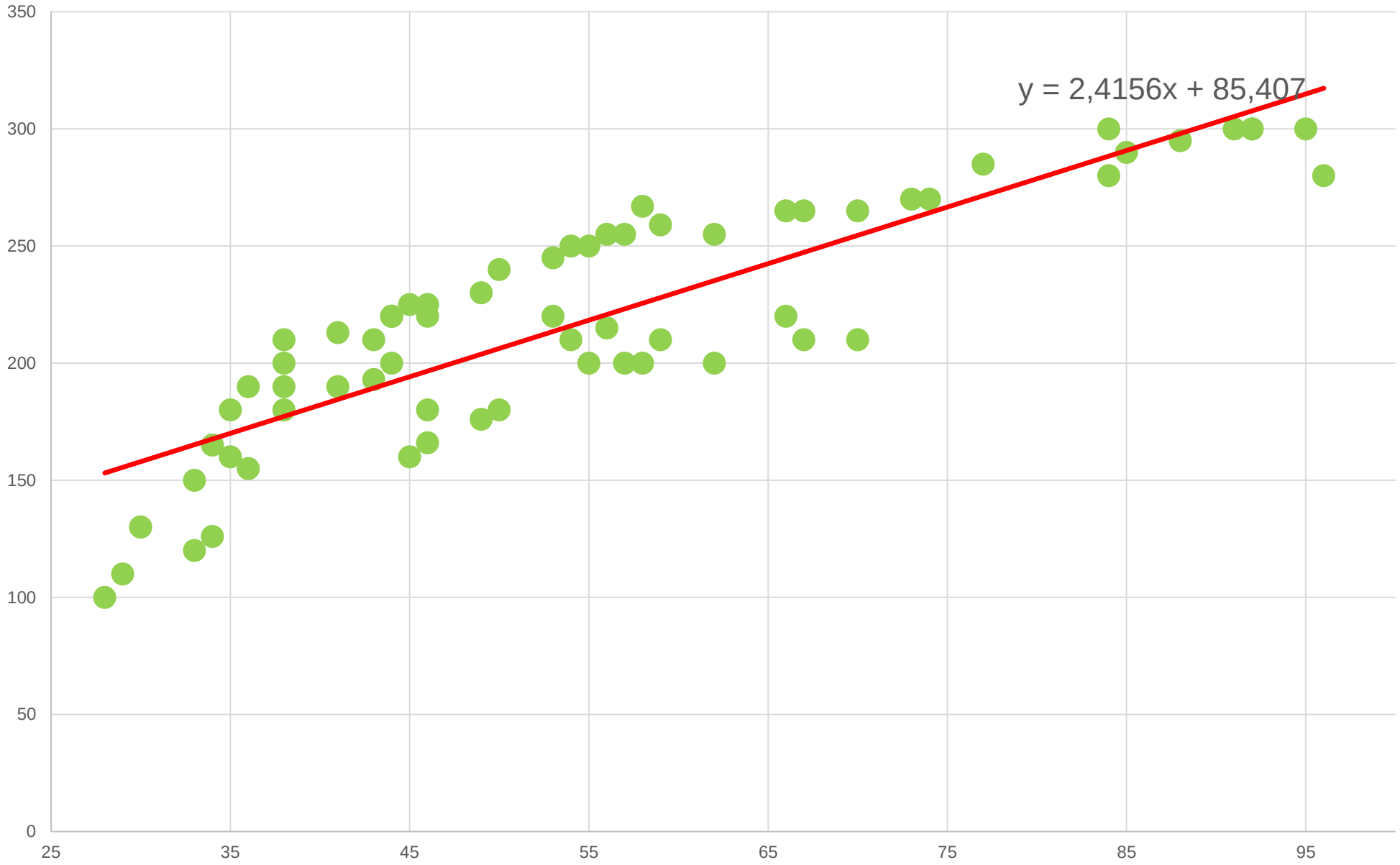




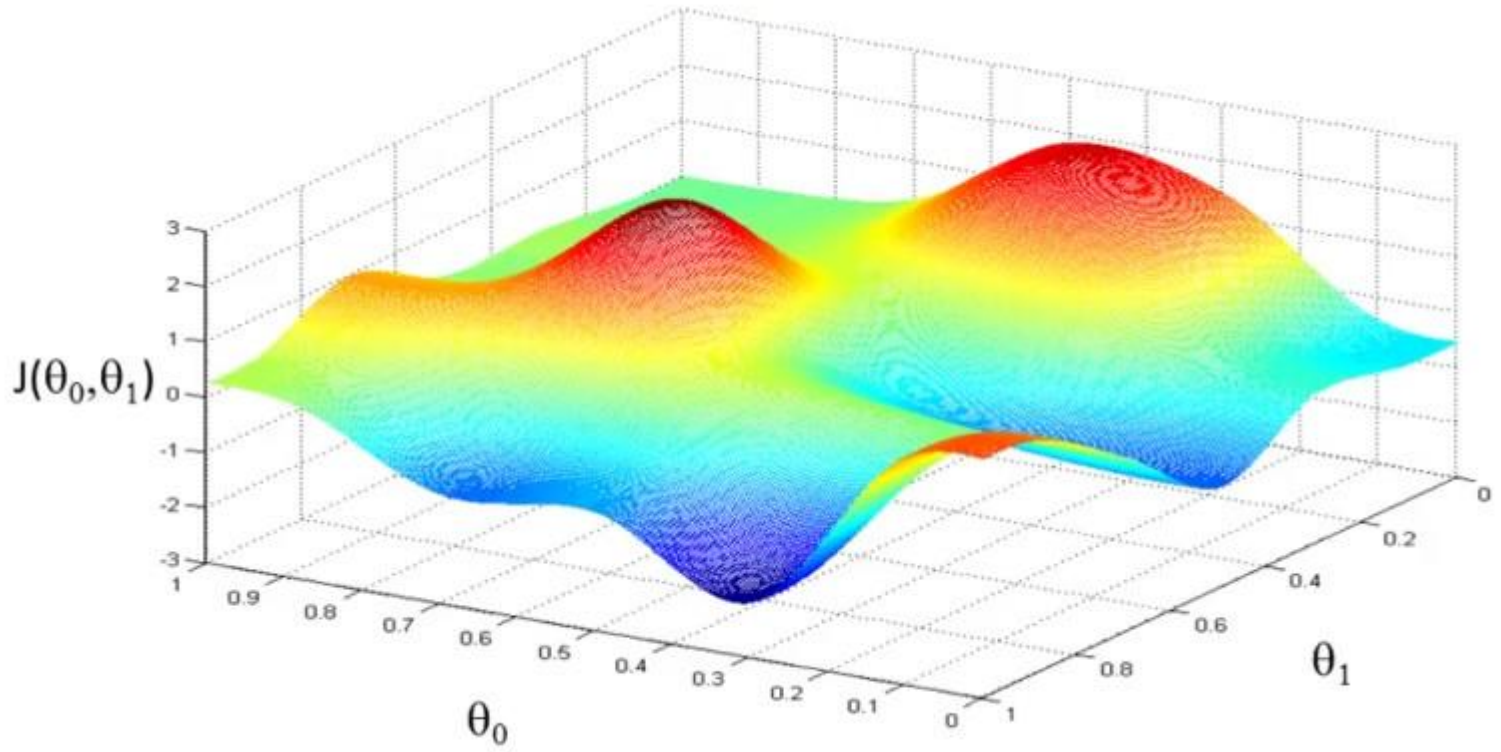




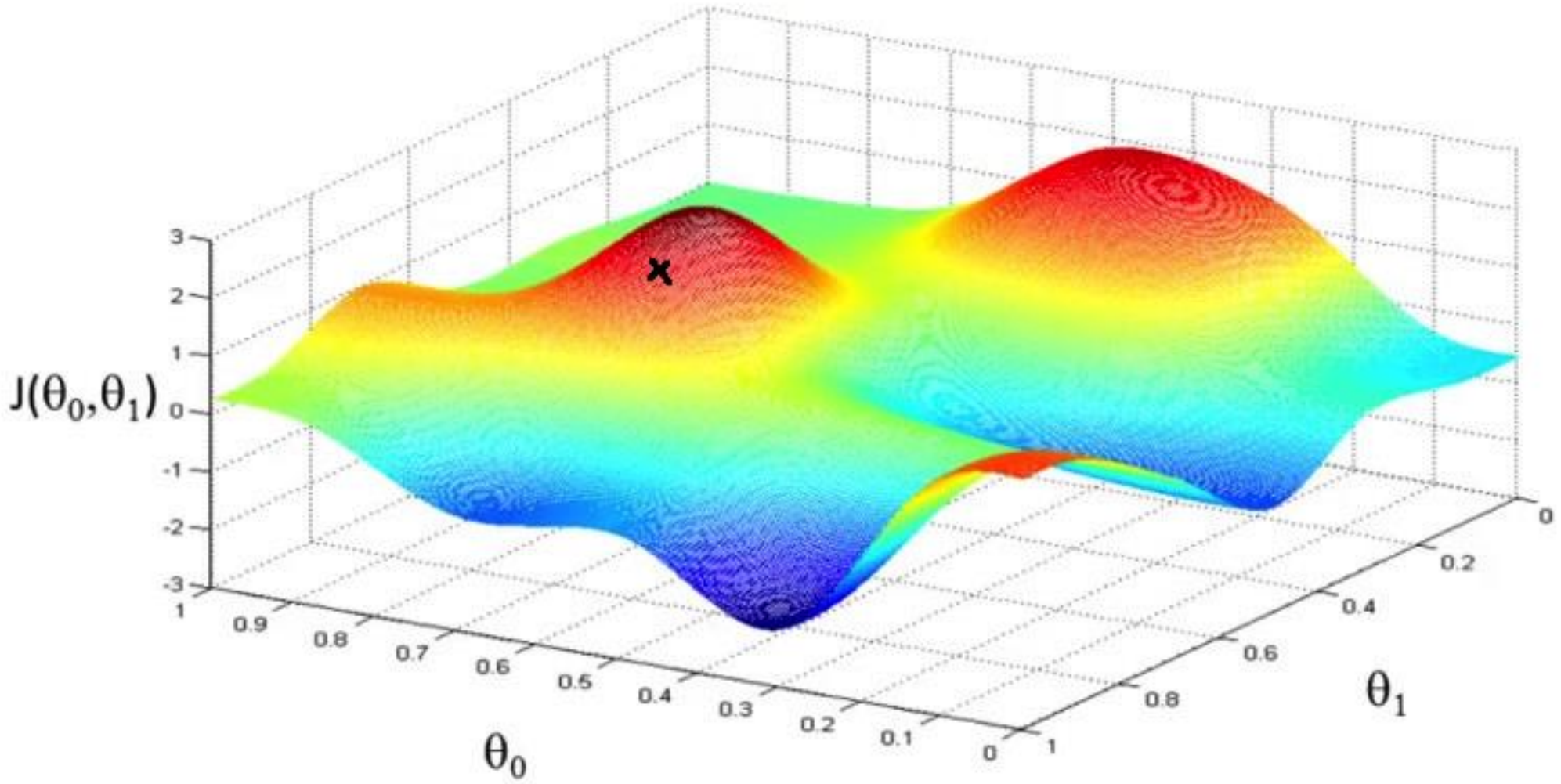




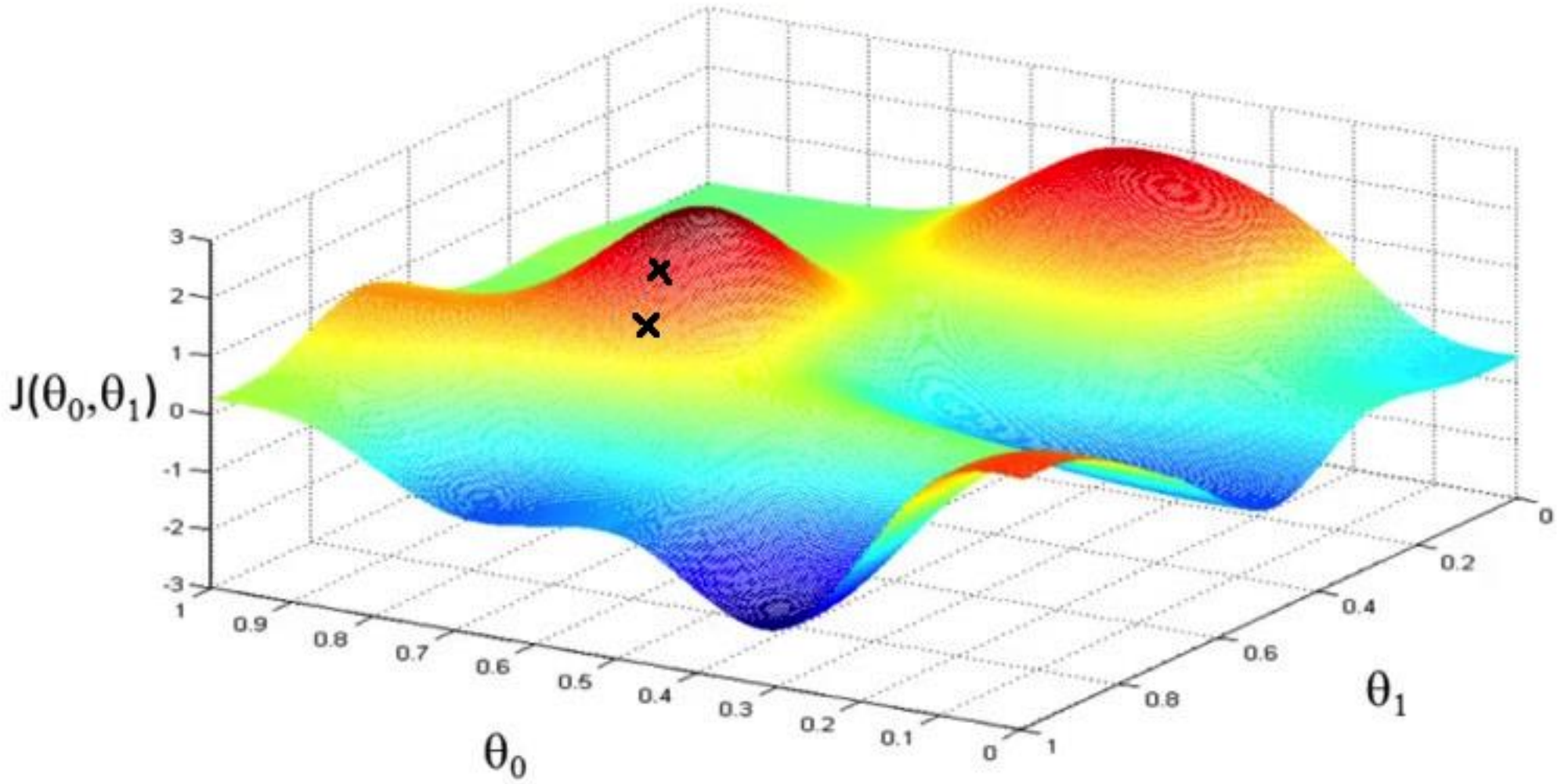
Gradient prosty



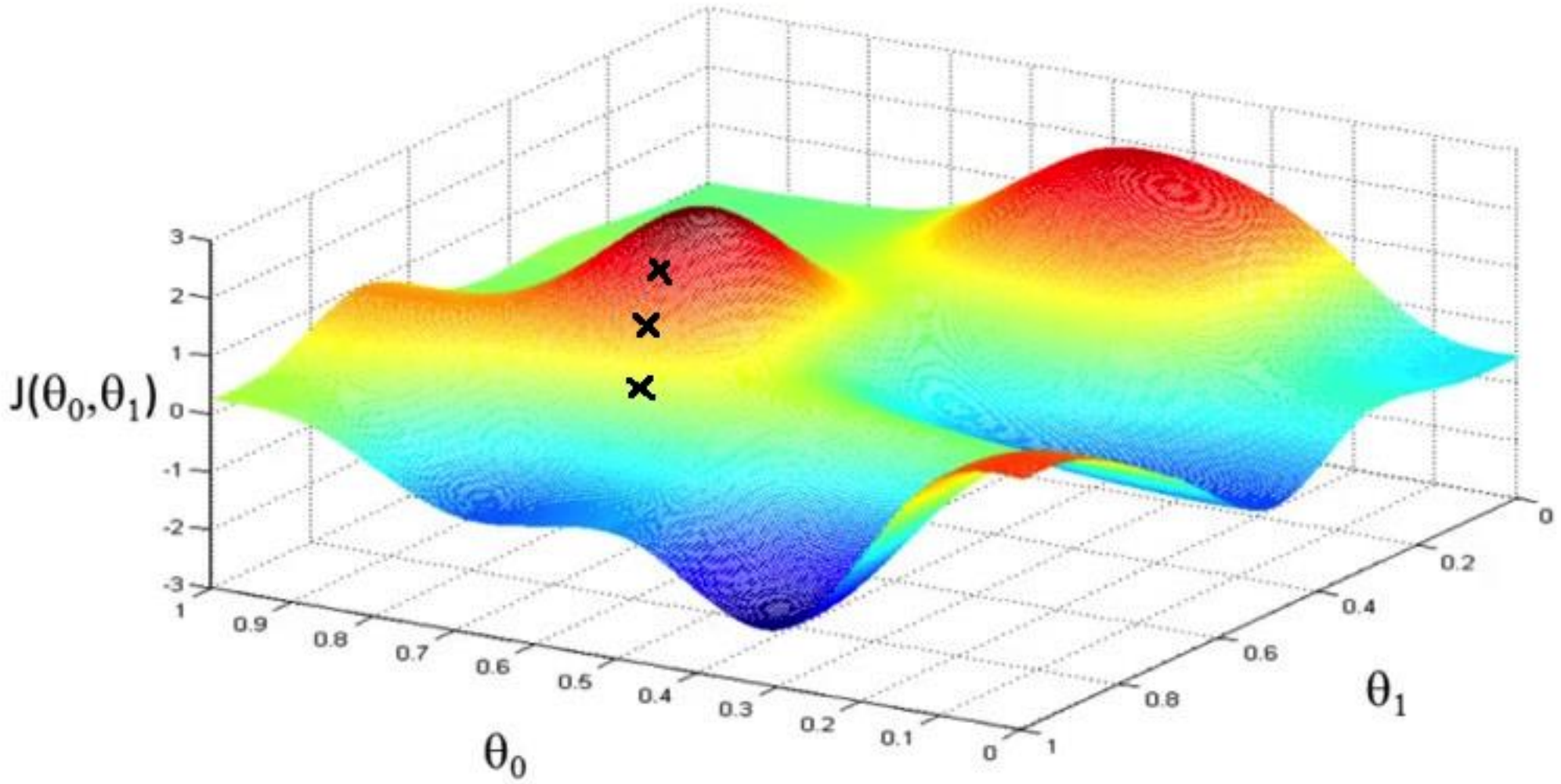
Coursera, Machine Learning, Andrew Ng

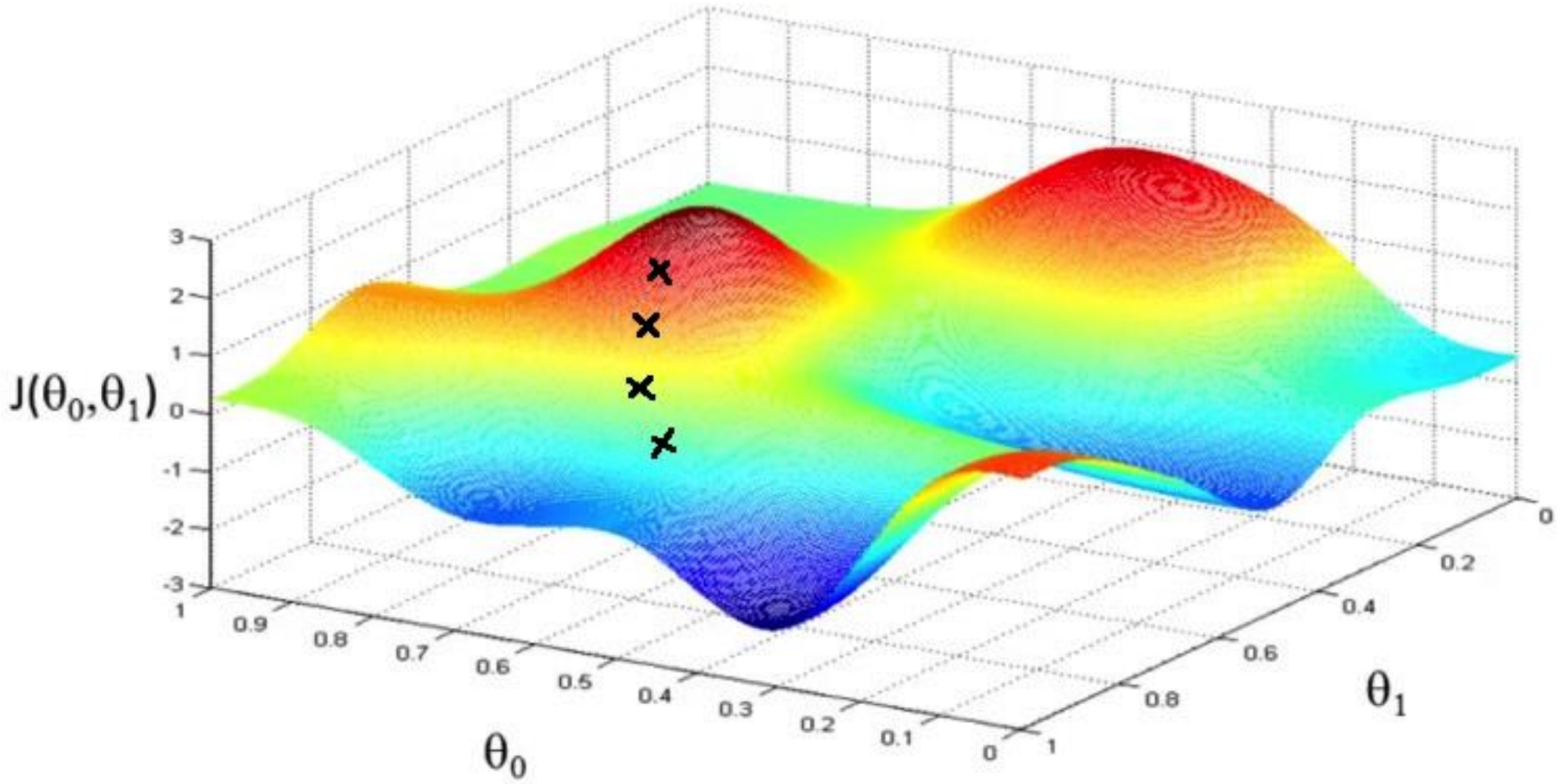


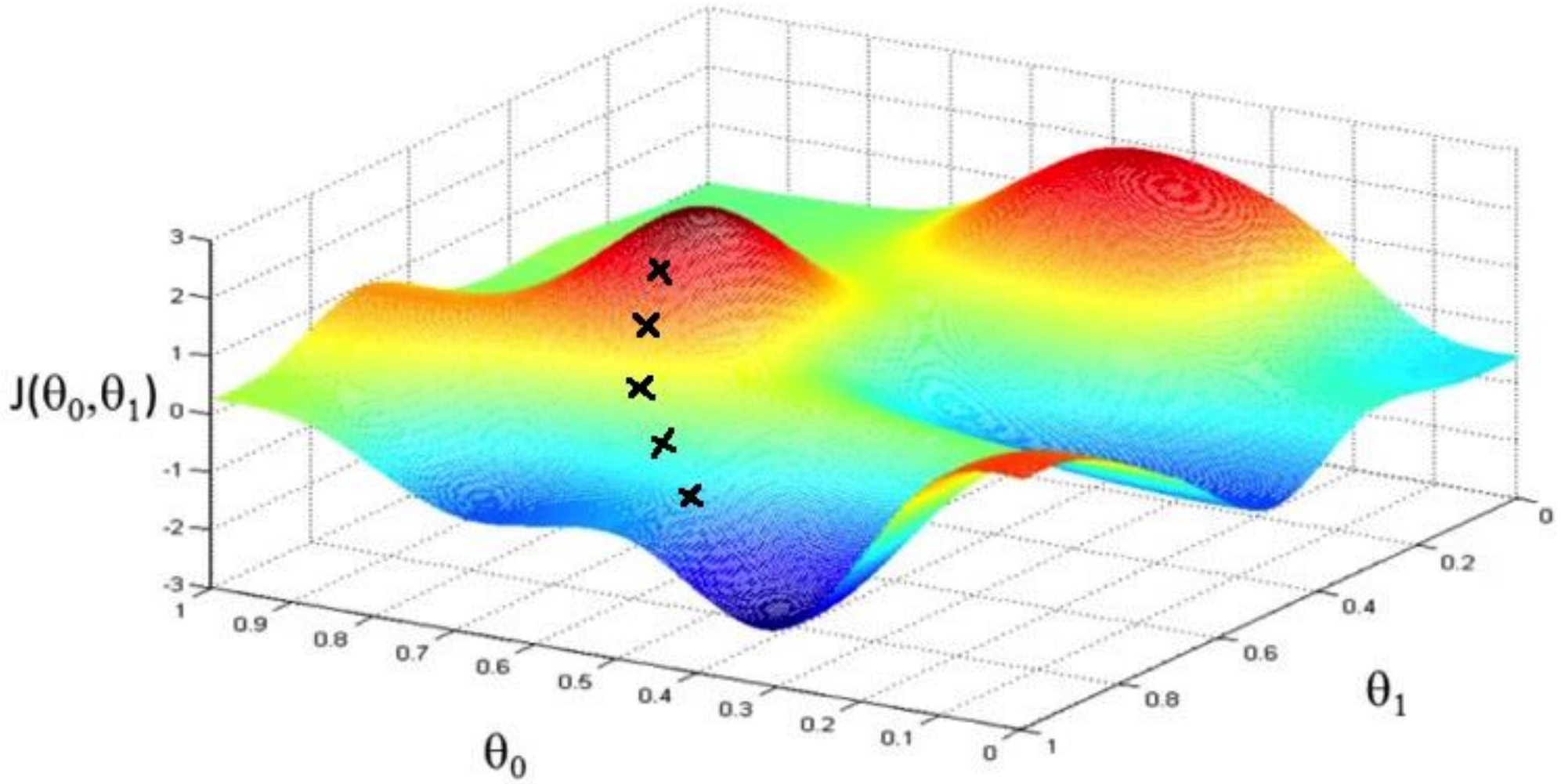
Coursera, Machine Learning, Andrew Ng



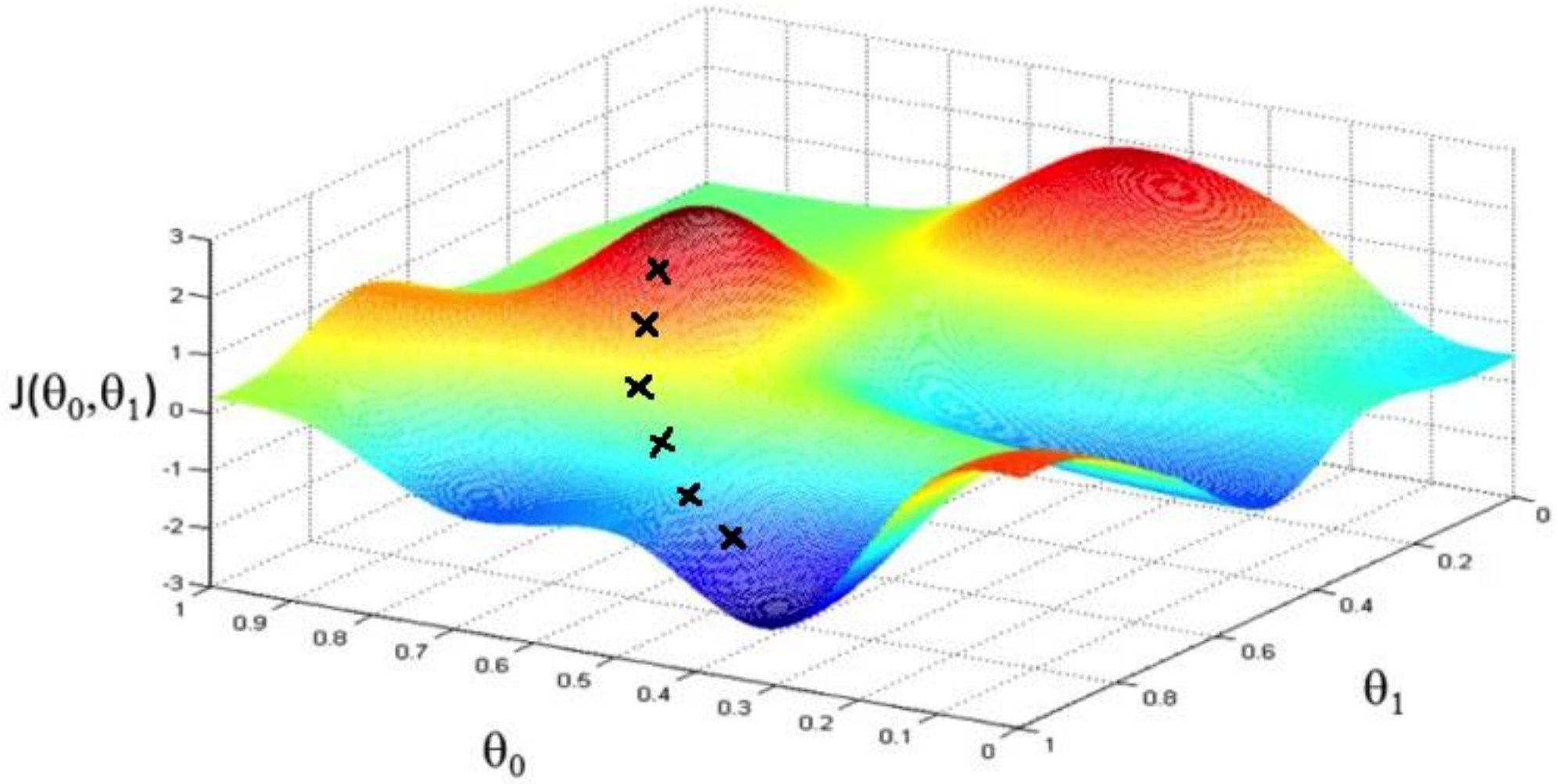
Coursera, Machine Learning, Andrew Ng



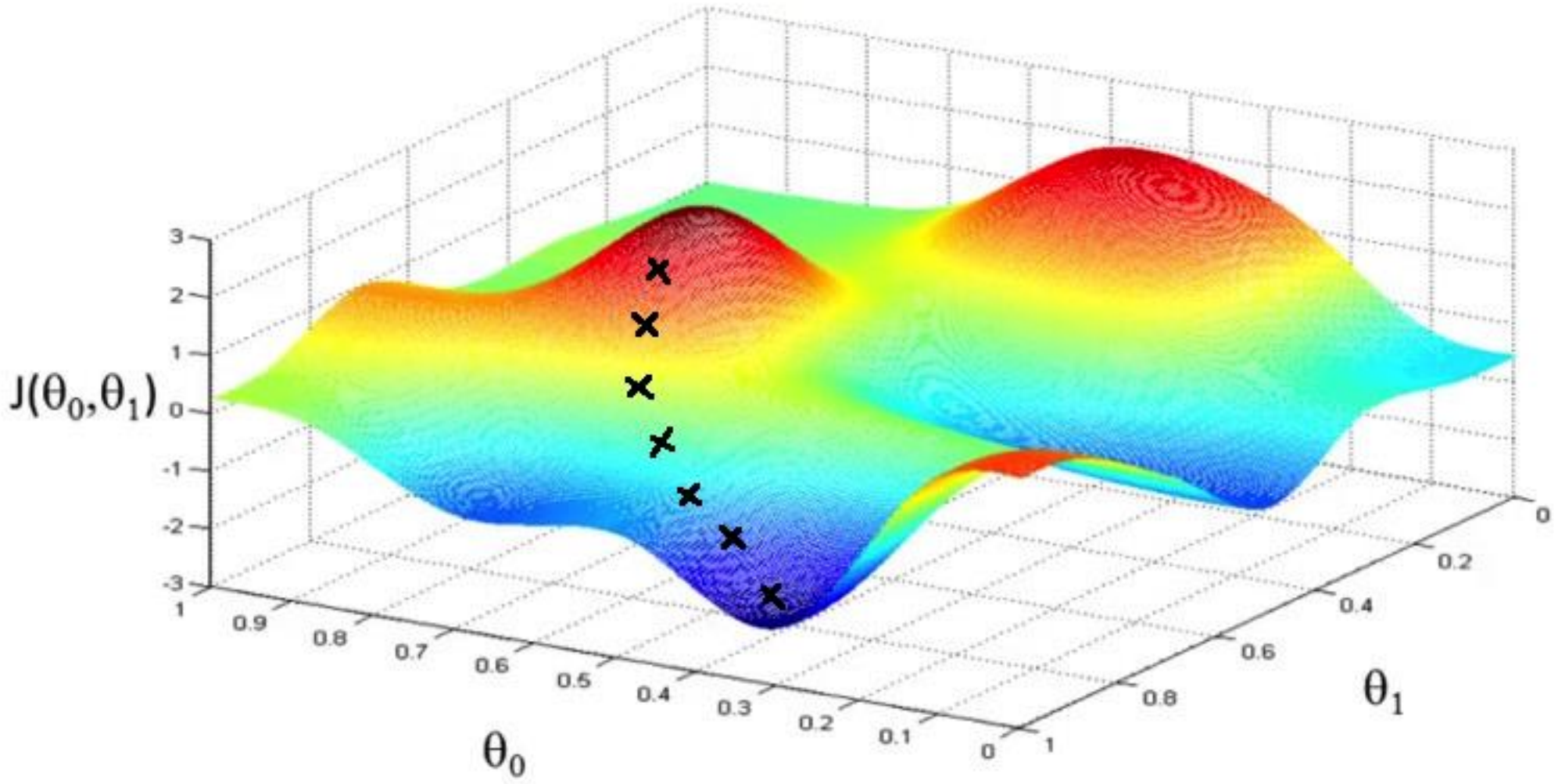




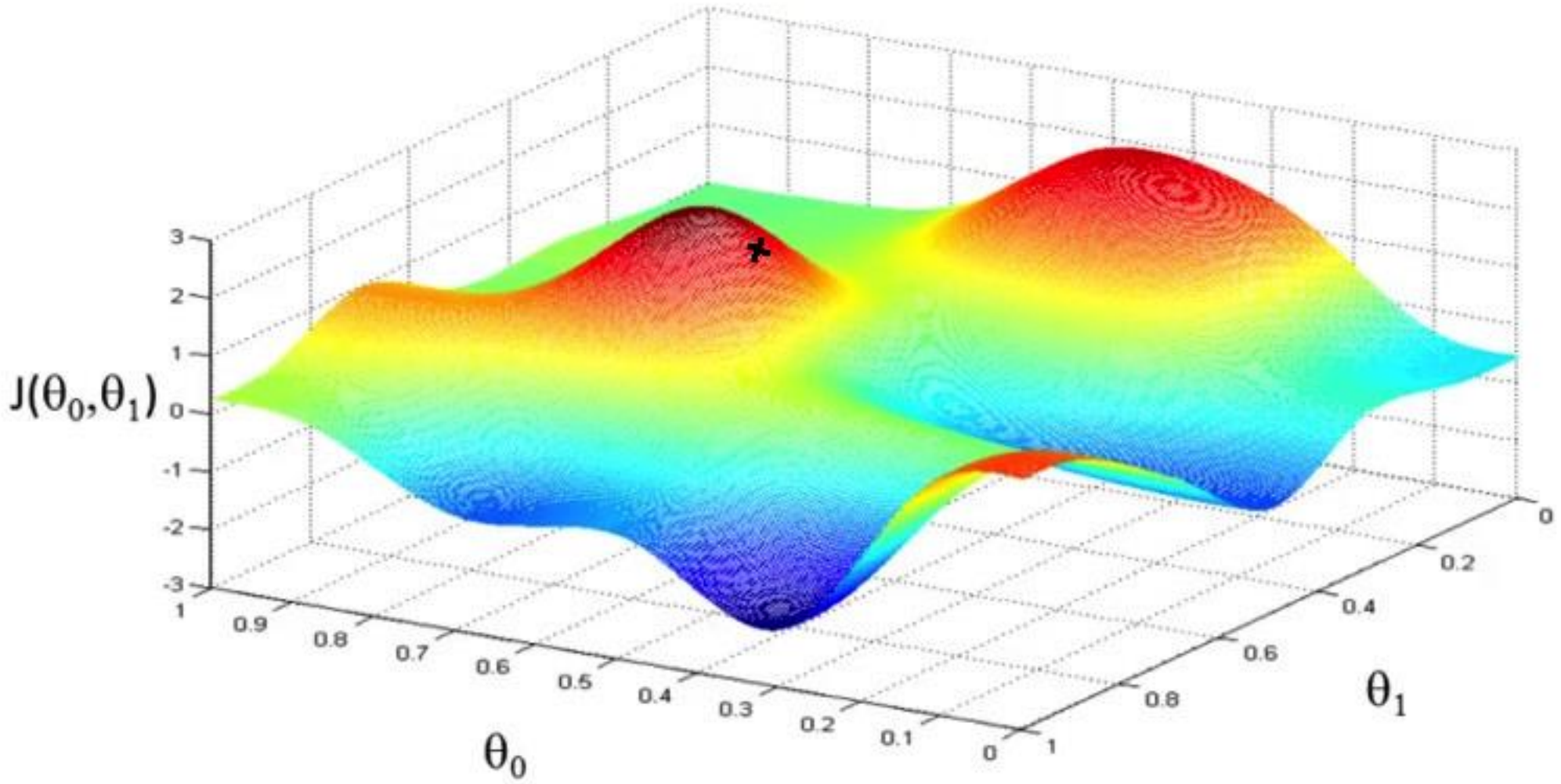
Coursera, Machine Learning, Andrew Ng



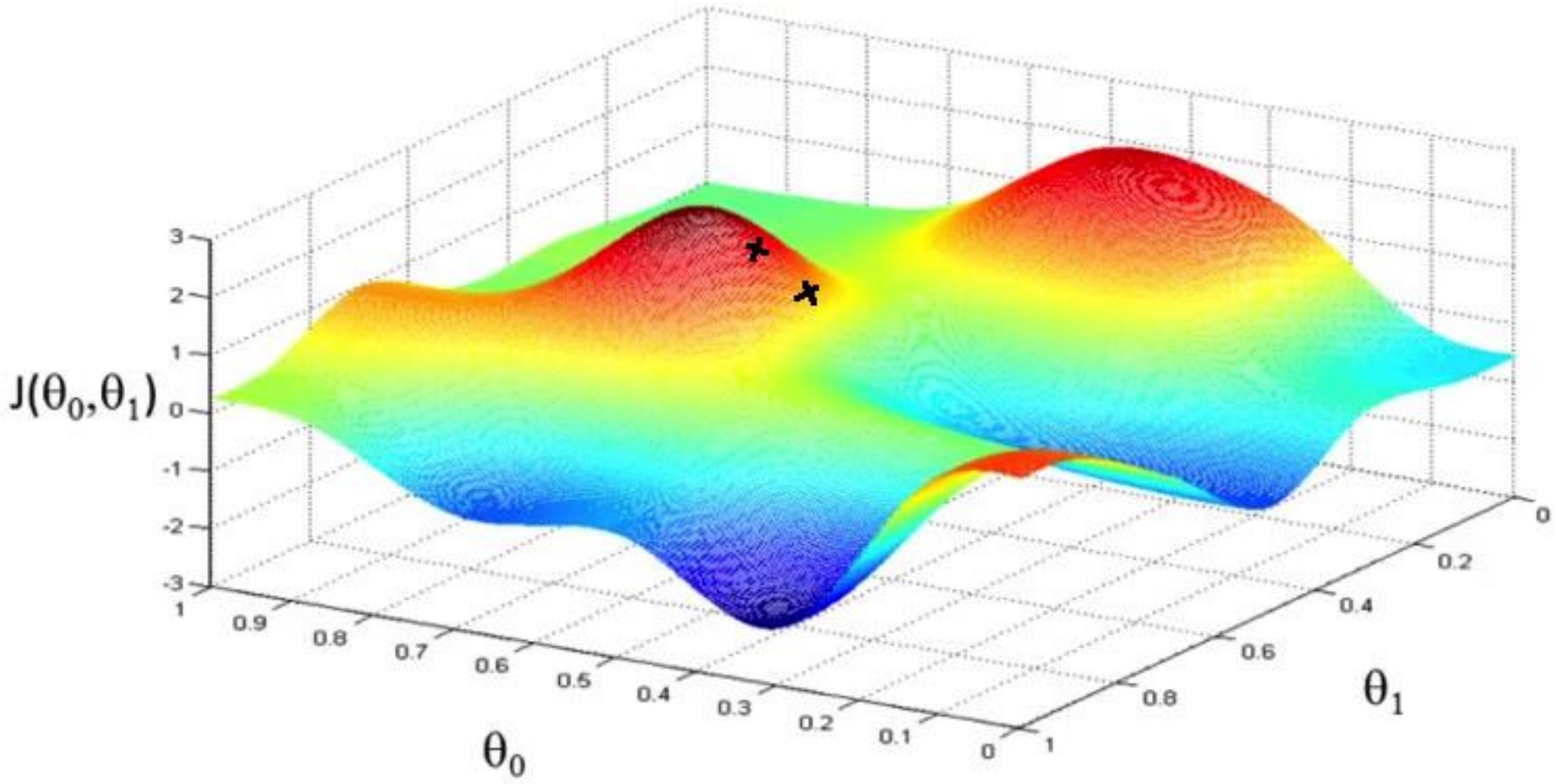
Coursera, Machine Learning, Andrew Ng



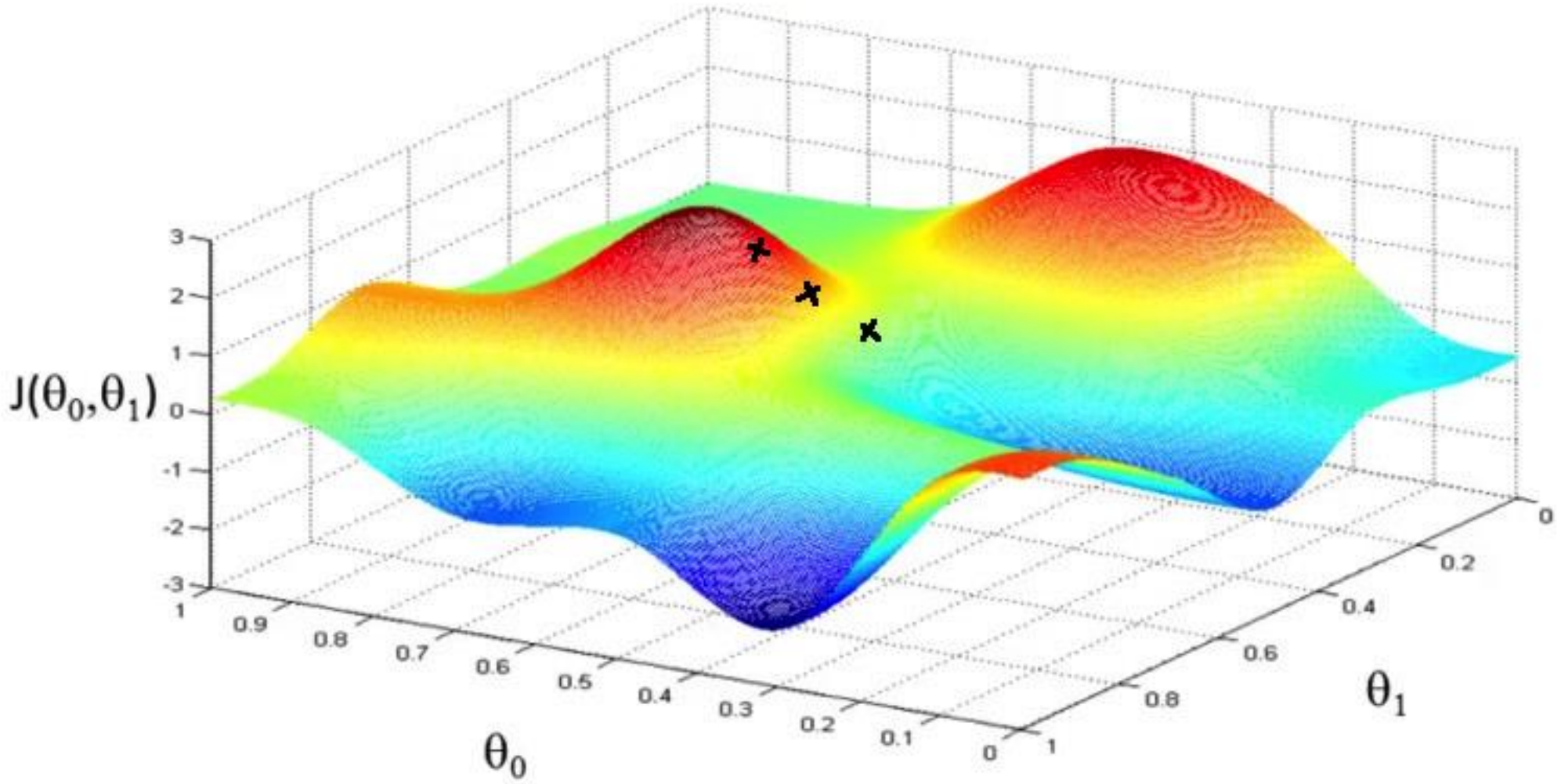
Coursera, Machine Learning, Andrew Ng



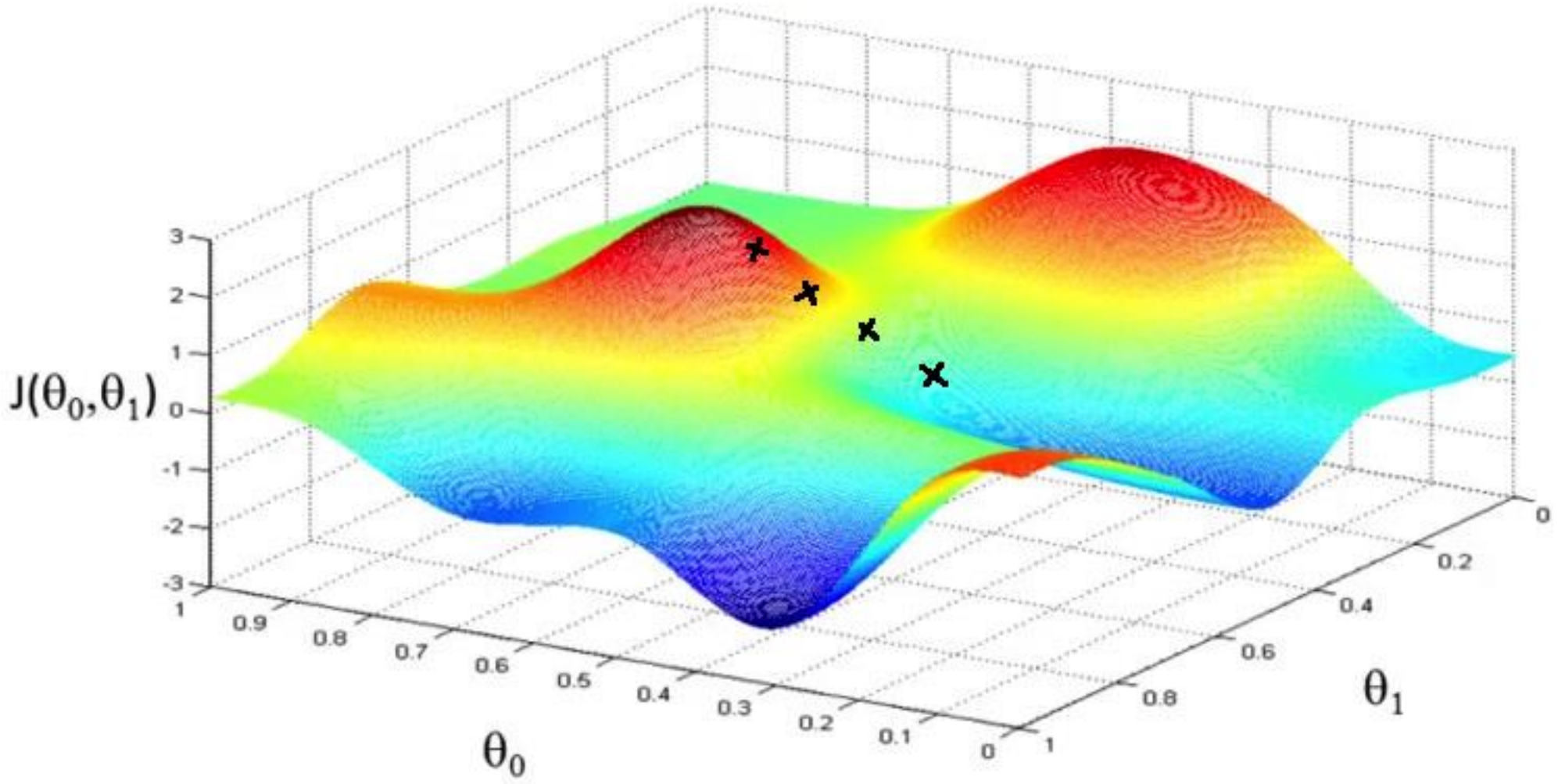
Coursera, Machine Learning, Andrew Ng



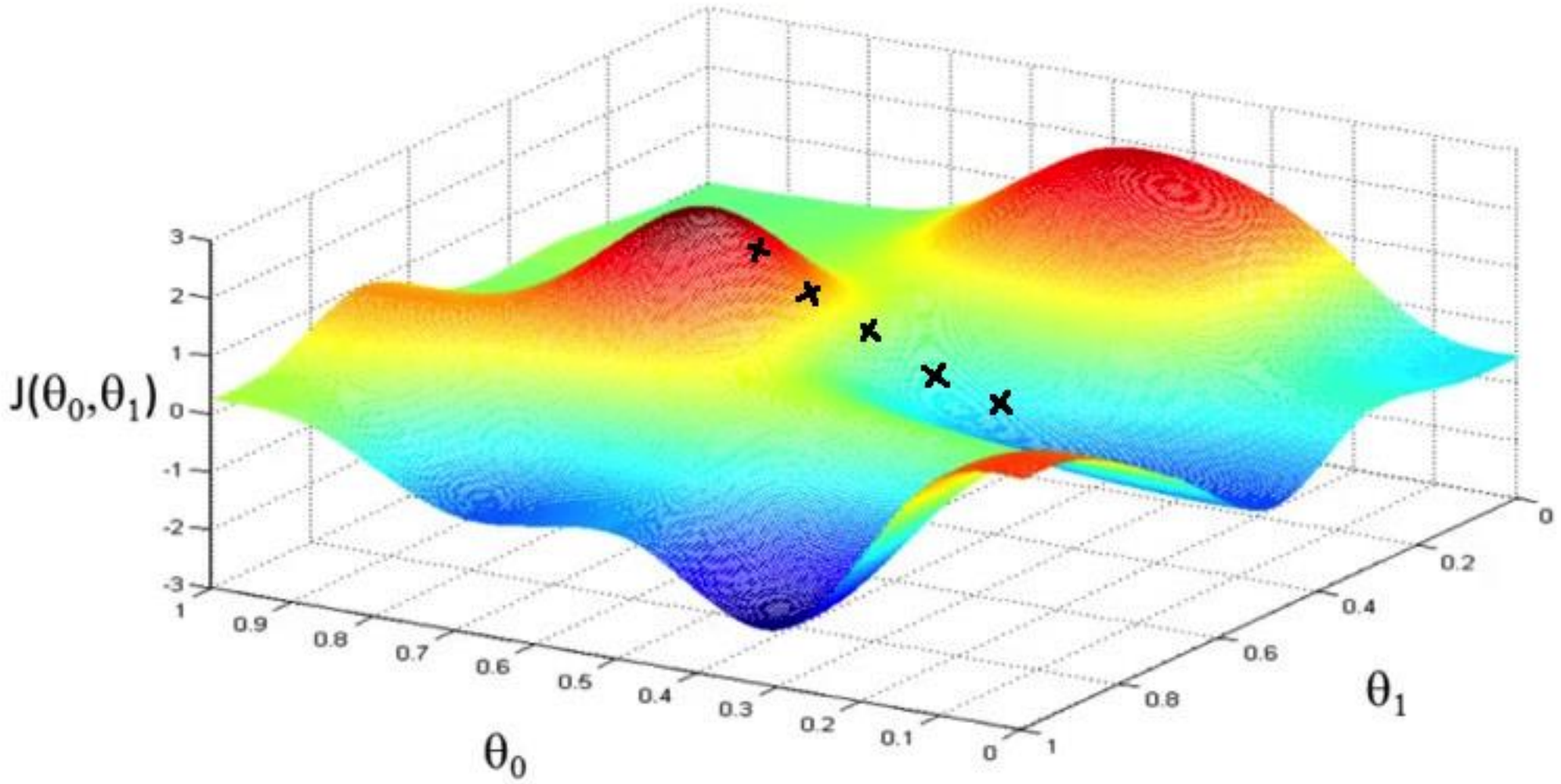
Coursera, Machine Learning, Andrew Ng



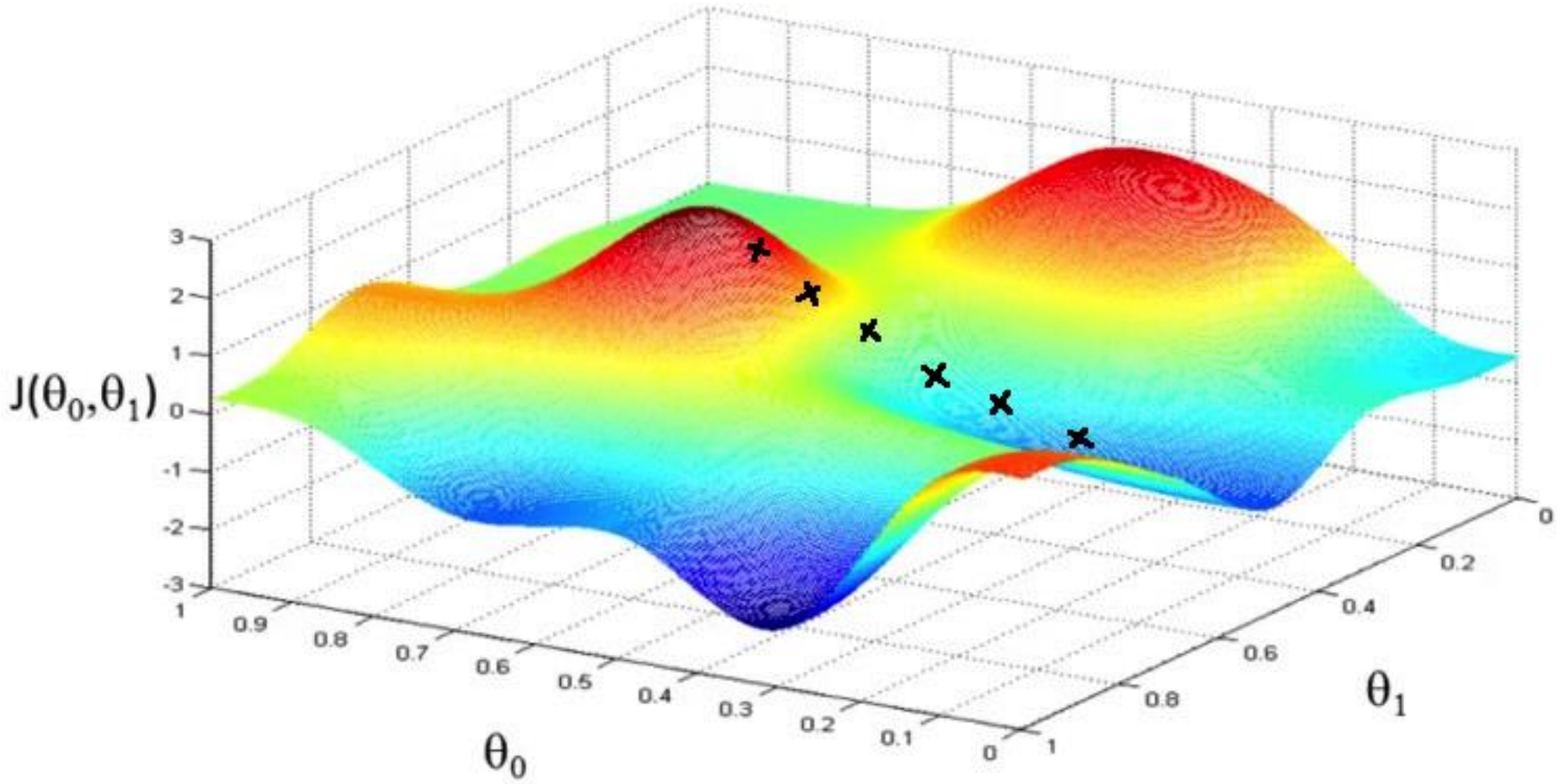
Coursera, Machine Learning, Andrew Ng



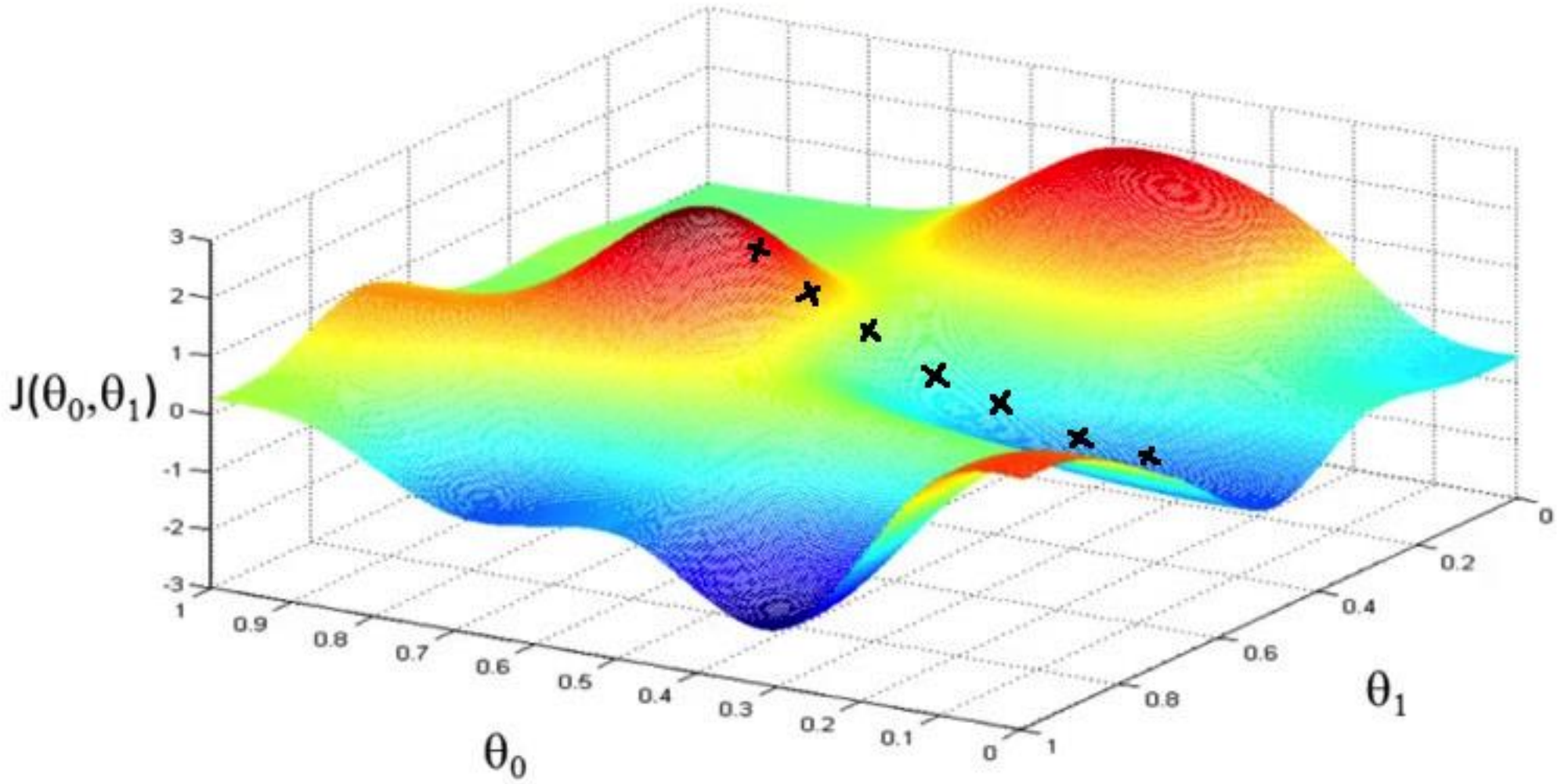
Coursera, Machine Learning, Andrew Ng



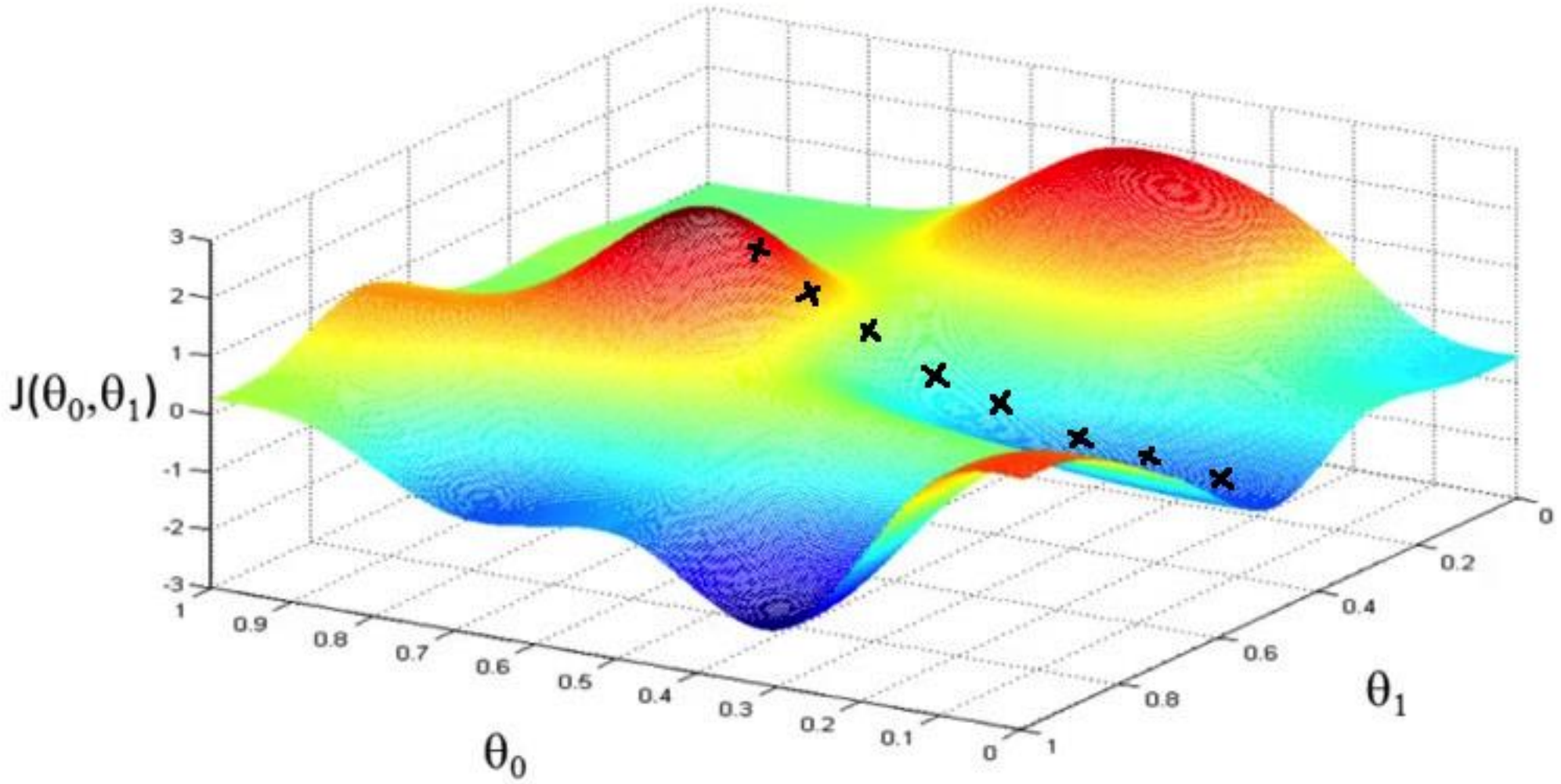
Coursera, Machine Learning, Andrew Ng



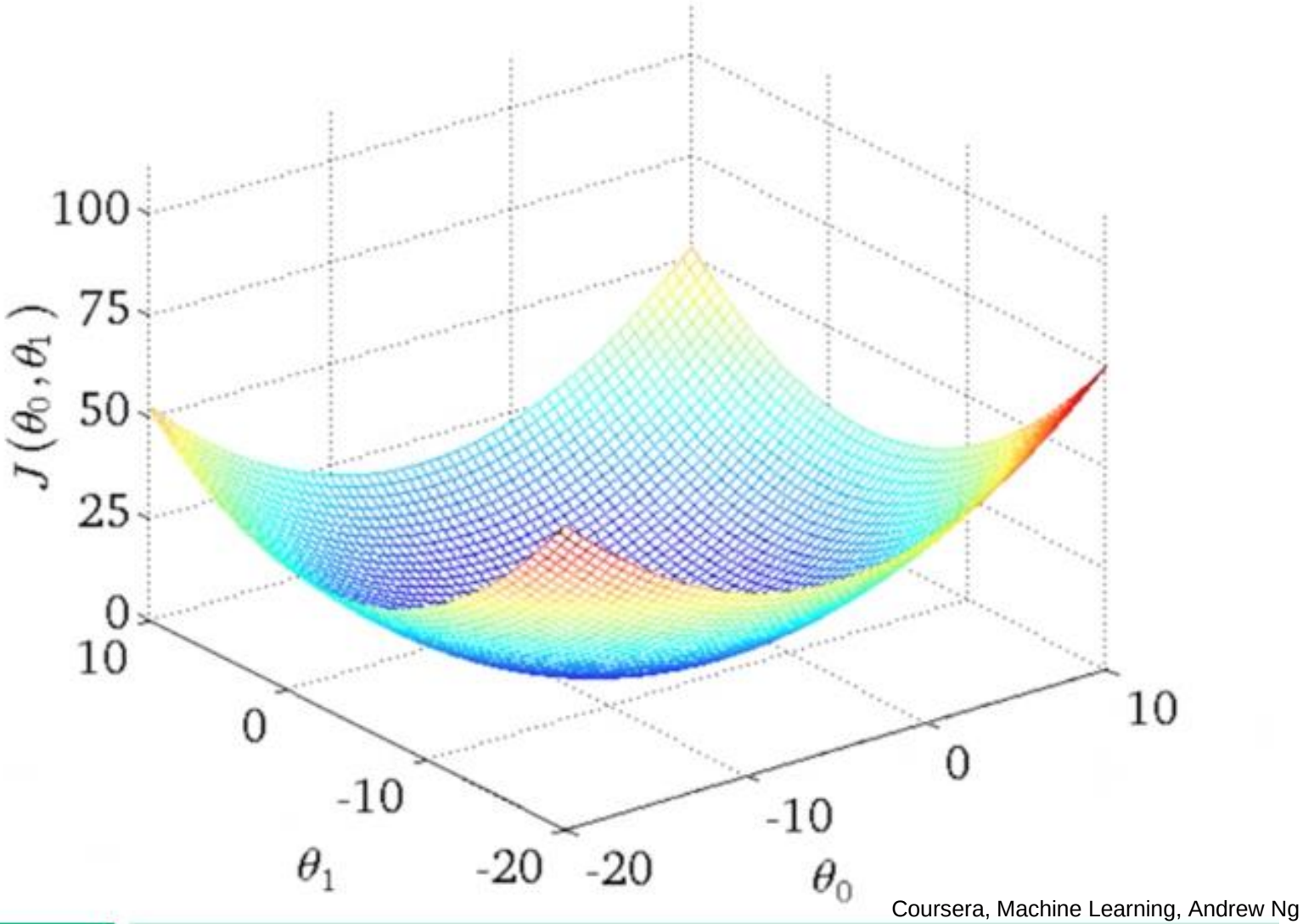
Coursera, Machine Learning, Andrew Ng



Coursera, Machine Learning, Andrew Ng



Coursera, Machine Learning, Andrew Ng



Coursera, Machine Learning, Andrew Ng

Gradient prosty

- Zaczynami z dowolnymi θ_0, θ_1
- Zmieniamy θ_0, θ_1 tak żeby $J(\theta)$ się zmniejszało

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

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

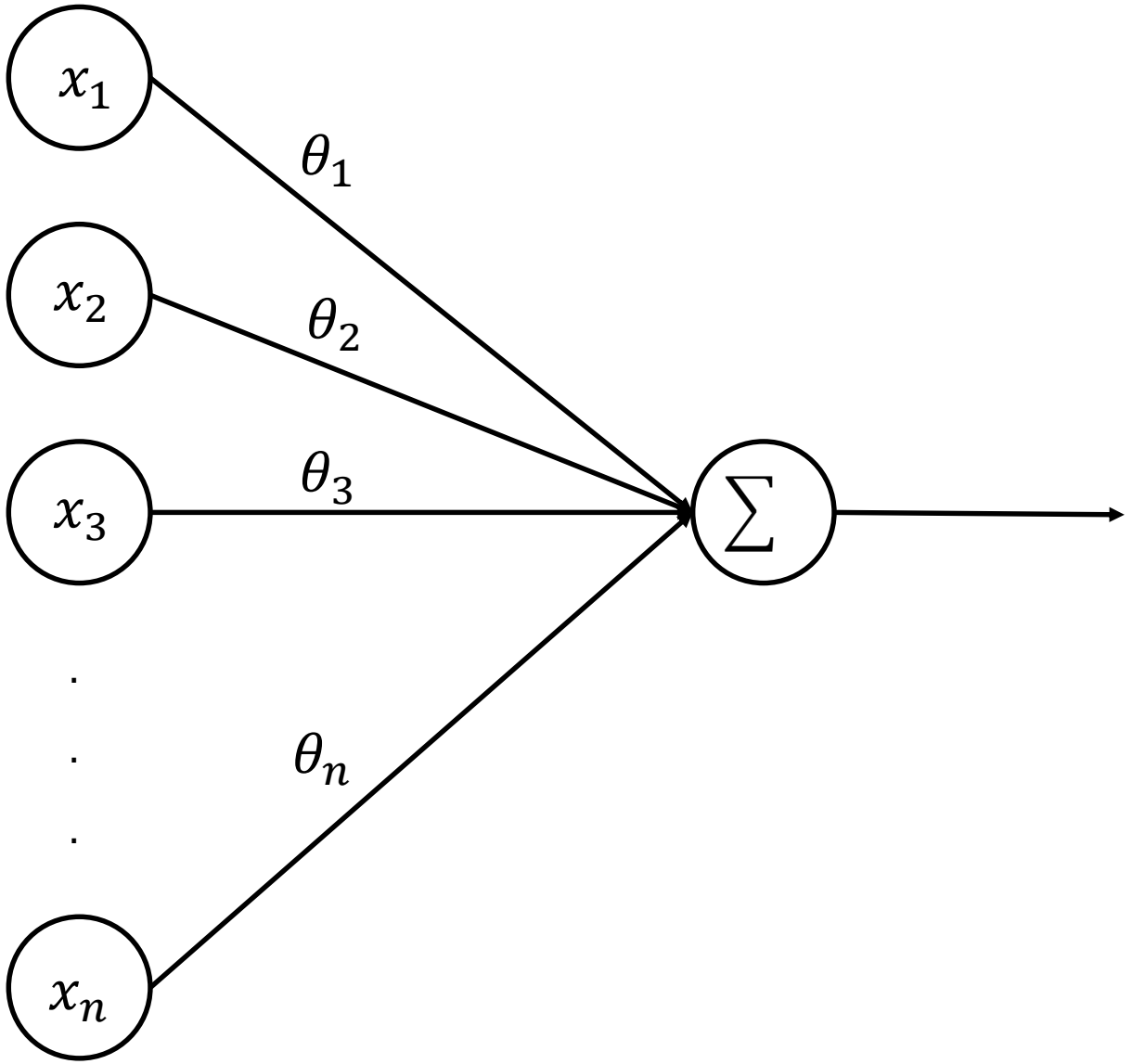
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x^{(i)}) - y^{(i)}) * x^{(i)})$$

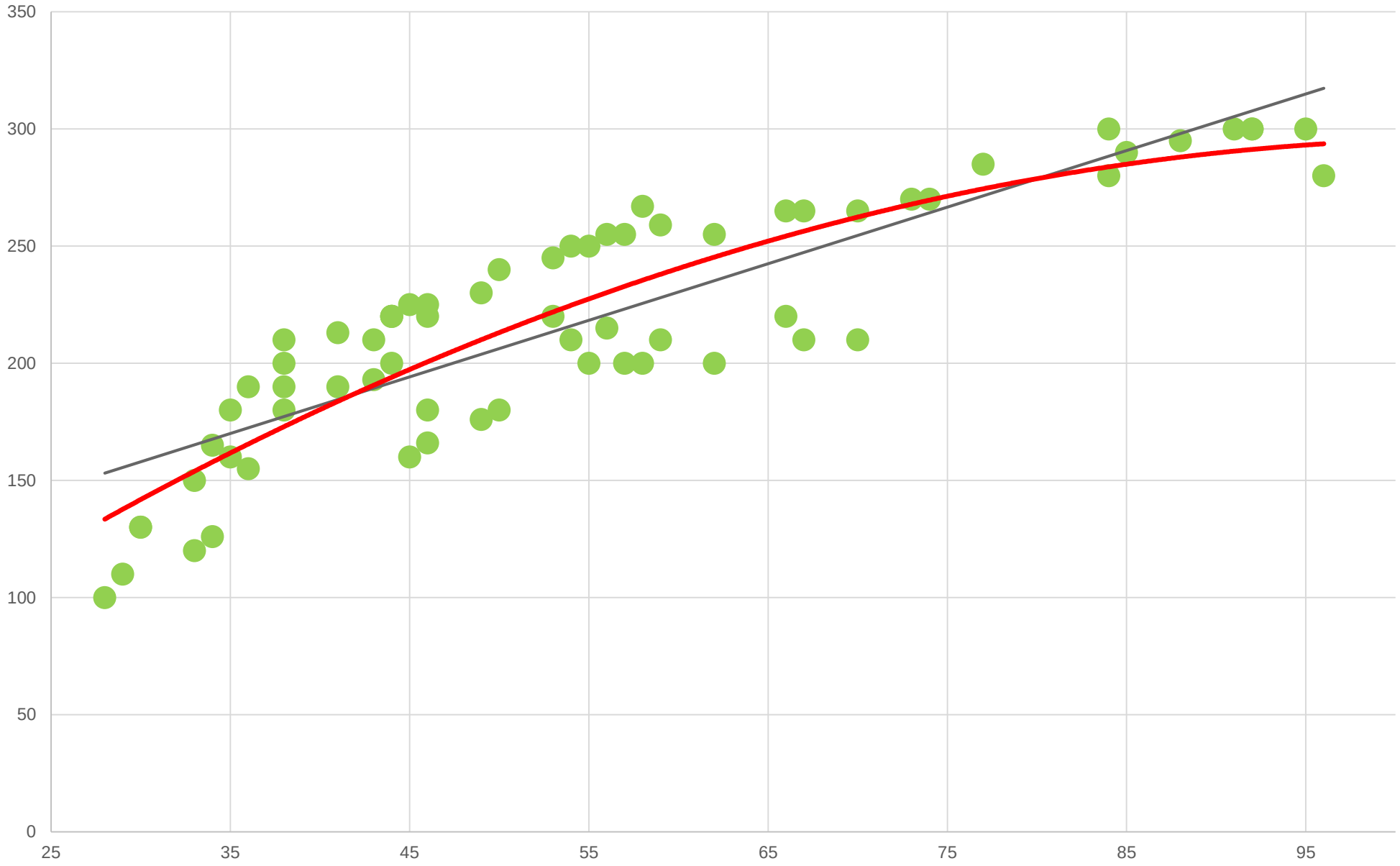
Wiele atrybutów

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

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \qquad \text{gdzie:} \qquad x_0 = 1$$

$$h_\theta(x) = \theta^T x \qquad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$





Nieliniowa hipoteza

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_2 x_1 x_2 + \theta_2 x_2 + \theta_2 x_2^2$$

Klasyfikacja

Regresja logistyczna

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

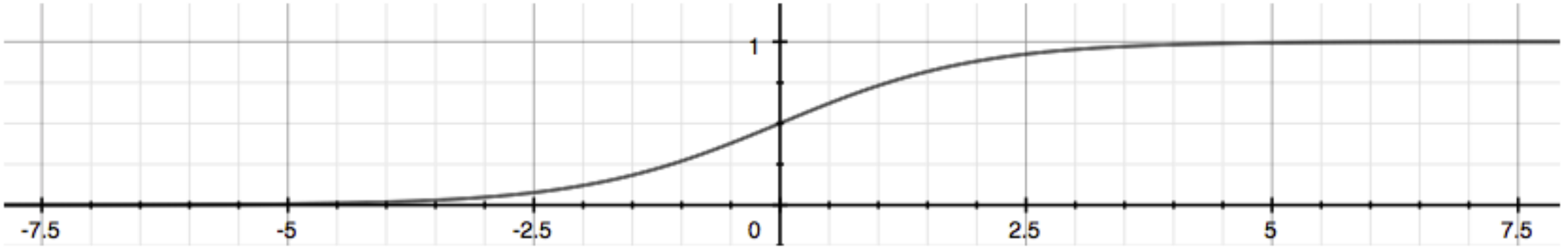
$$z = \theta^T x$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

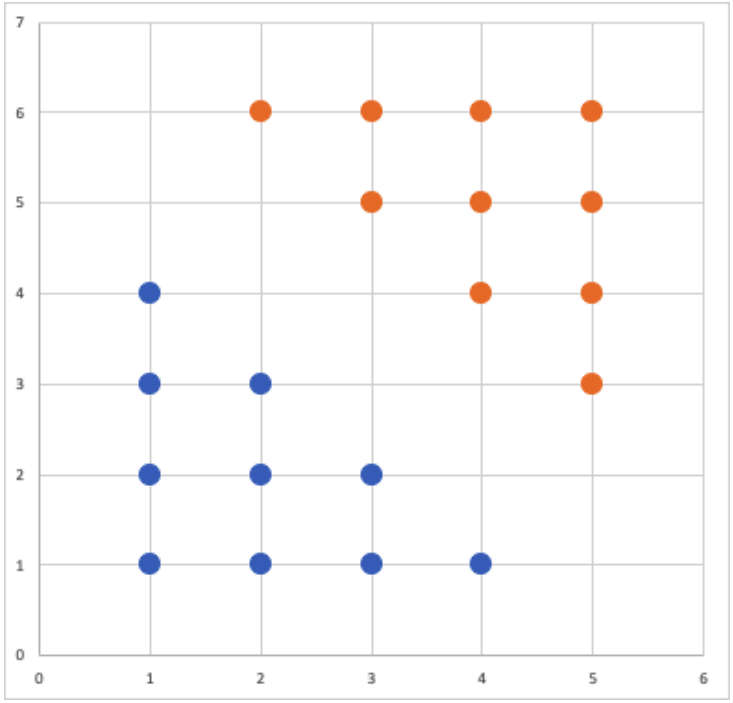
Funkcja logistyczna

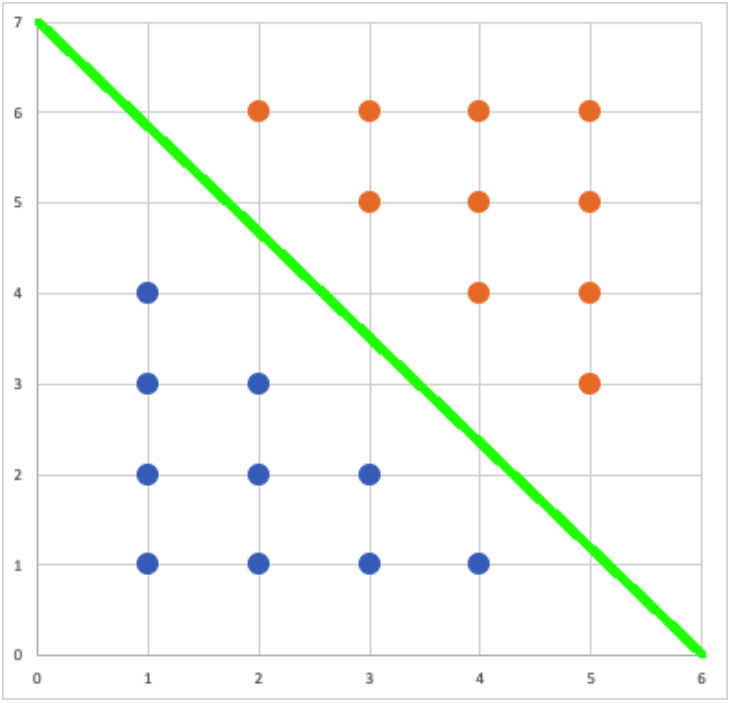
$$g(z) = \frac{1}{1 + e^{-z}}$$



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

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2) \qquad \theta_0 = 7, \theta_1 = -1, \theta_2 = 1$$





Funkcja kosztu dla regresji logistycznej:

$$J(\theta) = \frac{1}{m}\sum_{i=0}^{m} \mathrm{Cost}(h_\theta\left(x^{(i)}\right), y^{(i)})$$

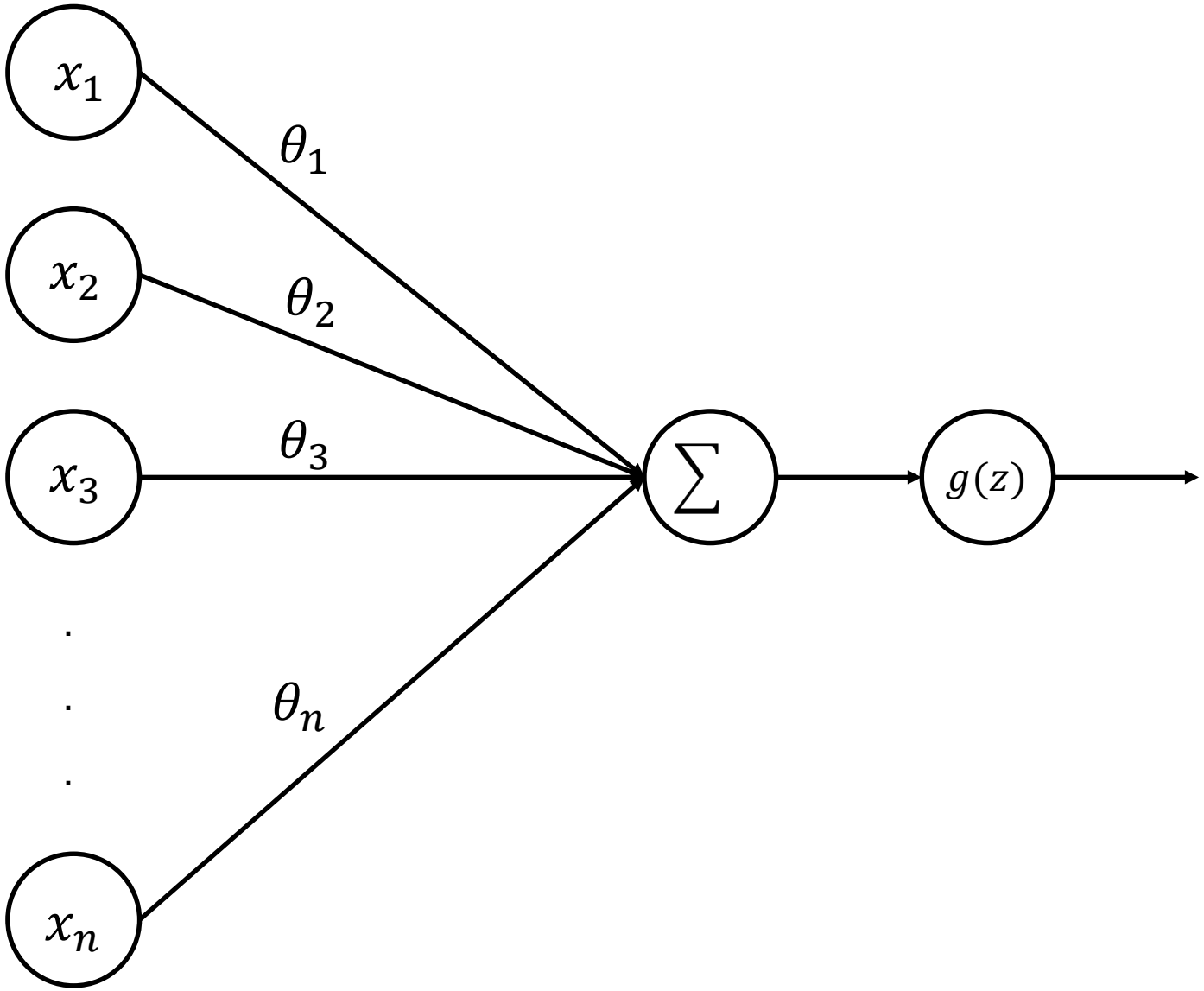
$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & y = 1 \\ -\log(1 - h_\theta(x)) & y = 0 \end{cases}$$

$$J(\theta) = -\frac{1}{m}\sum_{i=0}^{m} \left[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

Gradient prosty

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

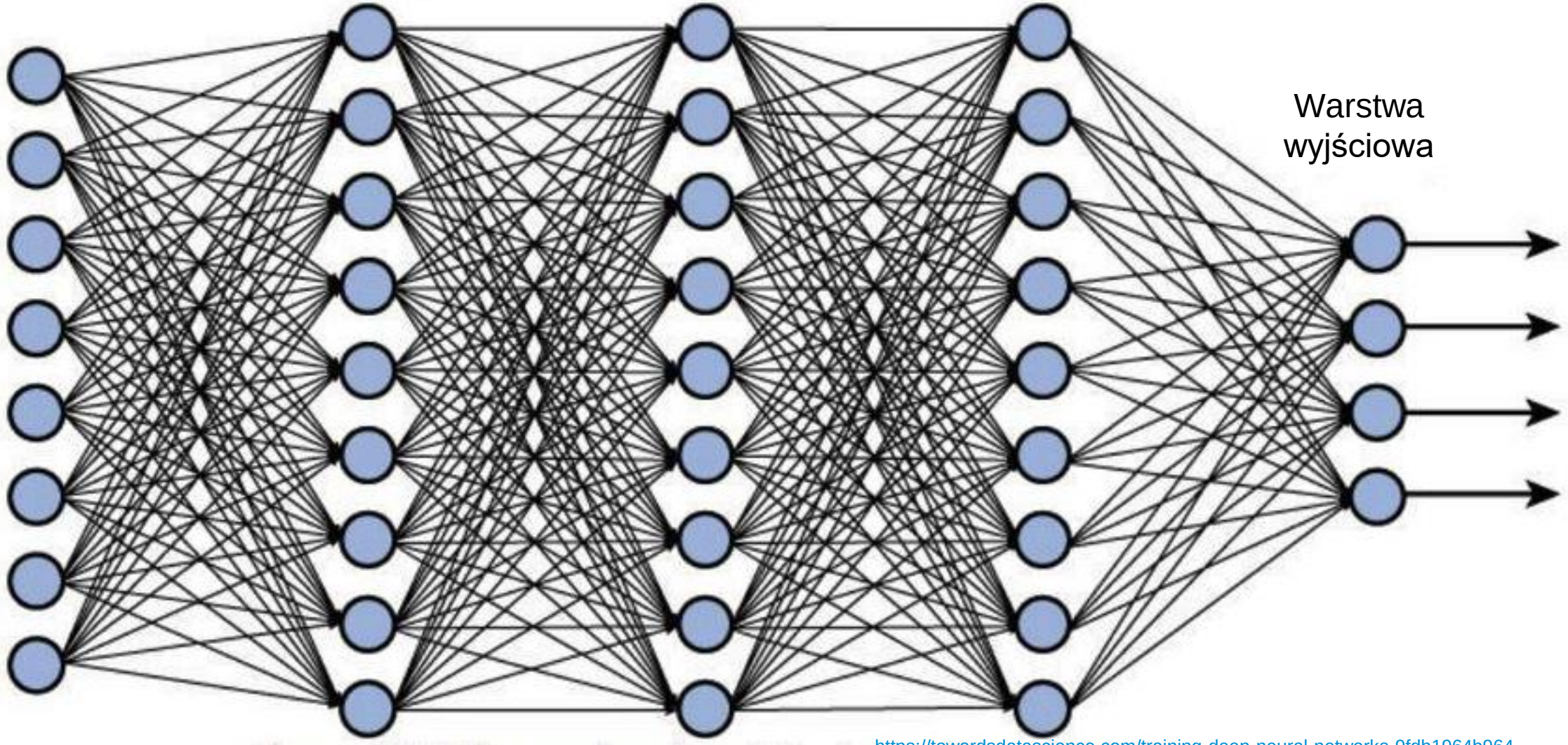
$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x^{(i)}) - y^{(i)}) * x^{(i)})$$



Sieci neuronowe

Warstwy ukryte

Warstwa wejściowa



https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964

Funkcja kosztu sieci neuronowej:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)} log\left(\left(h_\theta\left(x^{(i)}\right)\right)_k\right) + (1 - y_k^{(i)}) log\left(1 - \left(h_\theta\left(x^{(i)}\right)\right)_k\right)\right]$$

https://www.coursera.org/learn/machine-learning/

Dziękuję.

Michał Lipka

michal.lipka@sap.com