GATE WAY TO ML

UNDERSTANDING THE MATH

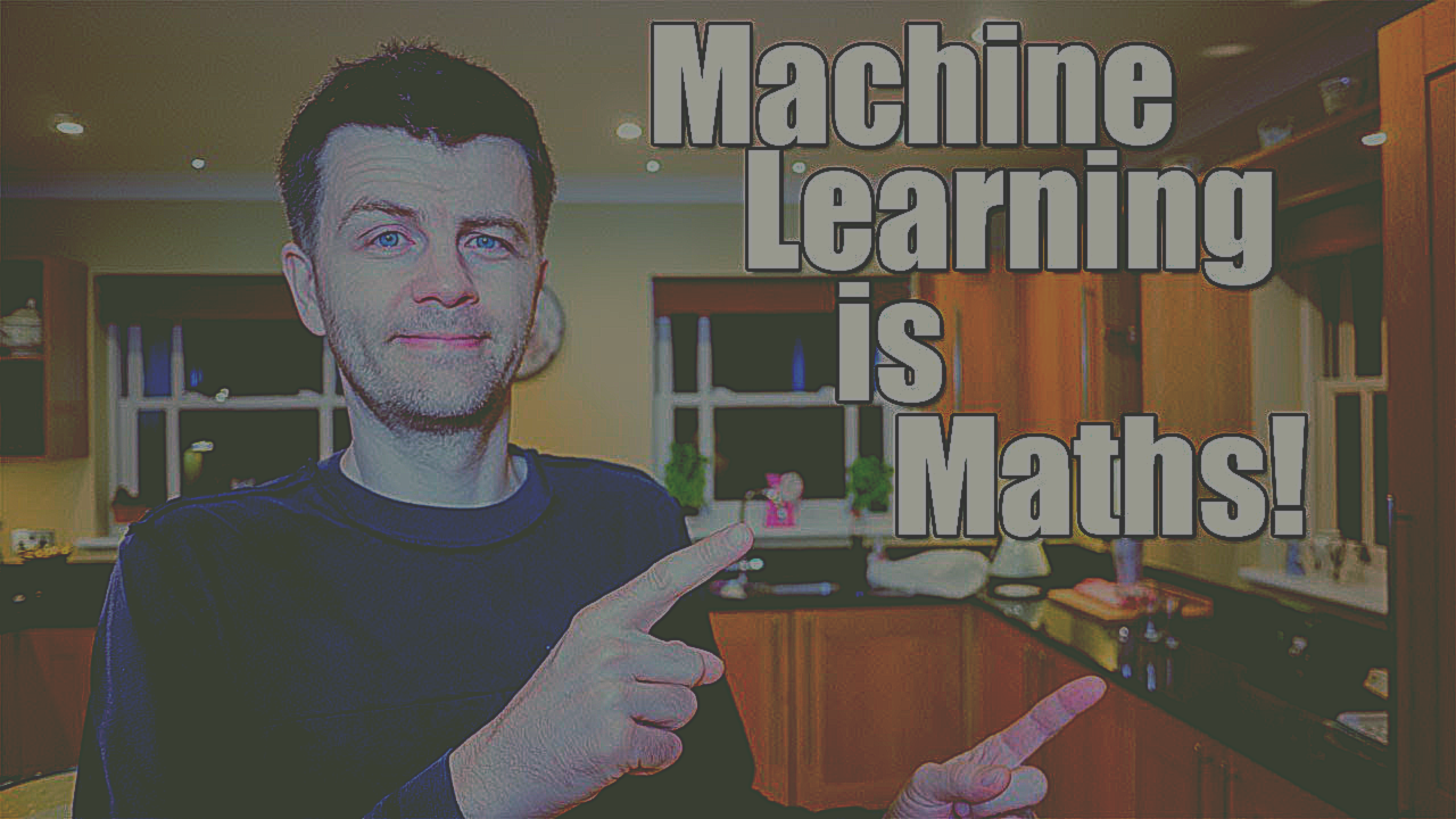
MATHAMATICAL TOOLS

YOU ALREADY KNOW.... RIGHT??

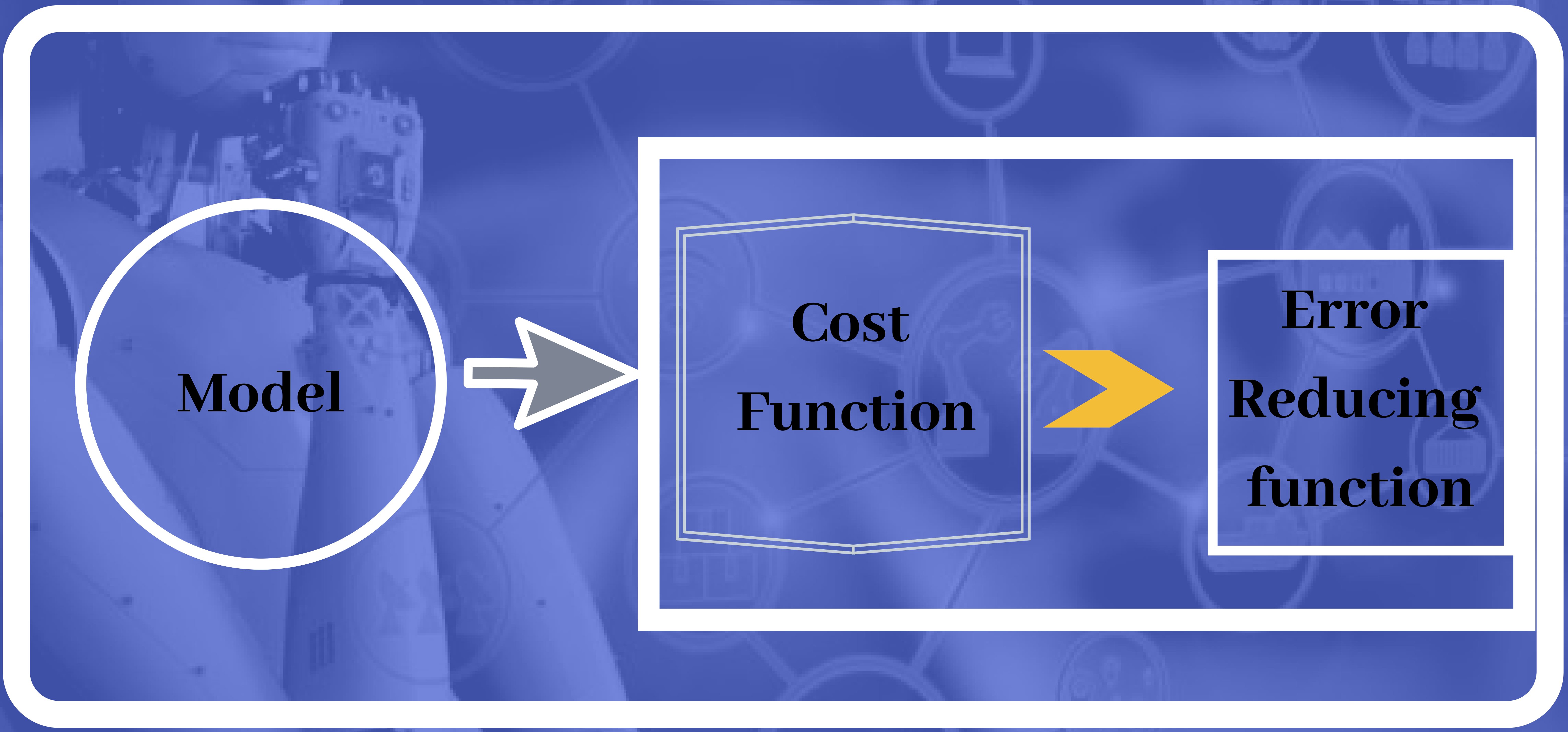
Linear Algebra

Calculus and optimization

probability


Machine
Learning
is
Maths!

Mathematical Understanding of Algorithms



WORDS OF INSPIRATION

WHAT WE WANT IS A MACHINE WHICH LEARNS FROM EXPERIENCE

- ALAN TURING

HOW MATH HELP TO DO SO

lets see

DATA

Let input be X and output be Y

There will be either one or multiple input features that can be represented as input vector

Similarly output an be a discrete or continuous value as we have seen earlier

Linear Regression with One Variable

Model/The Hypothesis Function

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

seen this before ?!...

Note that this is like the equation of a straight line. We give to $h_\theta(x)$ values for θ_0 and θ_1 to get our estimated output $\hat{y}$. In other words, we are trying to create a function called h_θ that is trying to map our input data (the x's) to our output data (the y's).

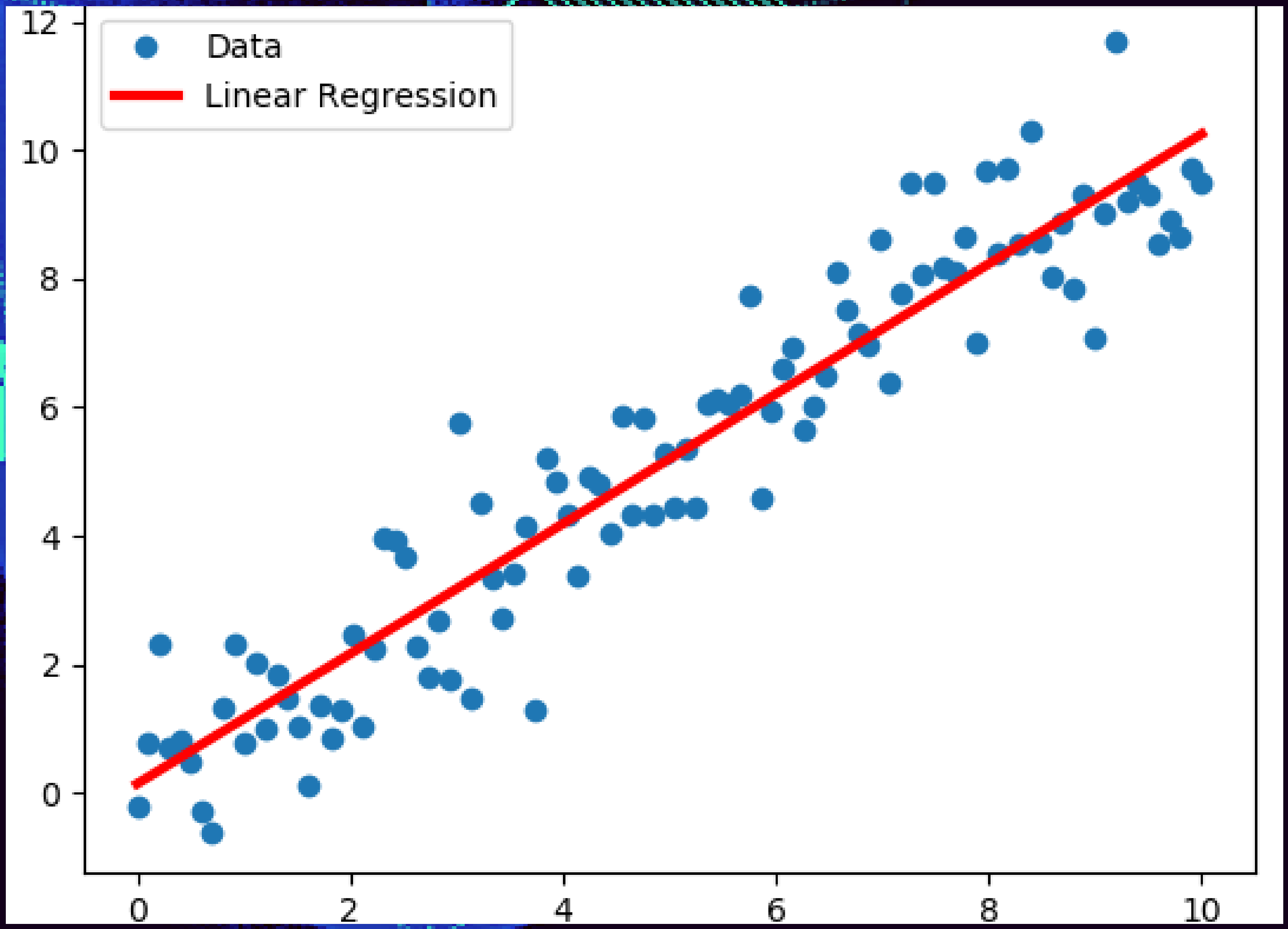
Linear Regression with One Variable

- YES! IT IS EQUATION OF ALINE
- TWO UNKNOWNS AND WE NEED ONLY TWO PAIR OF (X,Y)
- BUT HERE WE HAVE N NUMBER OF (X,Y)

Linear Regression with One Variable

FIND THE BEST FIT LINE WITH LOWEST ERROR

N INTRODUCTION TO
MACHINE LEARNI



BUT HOW?

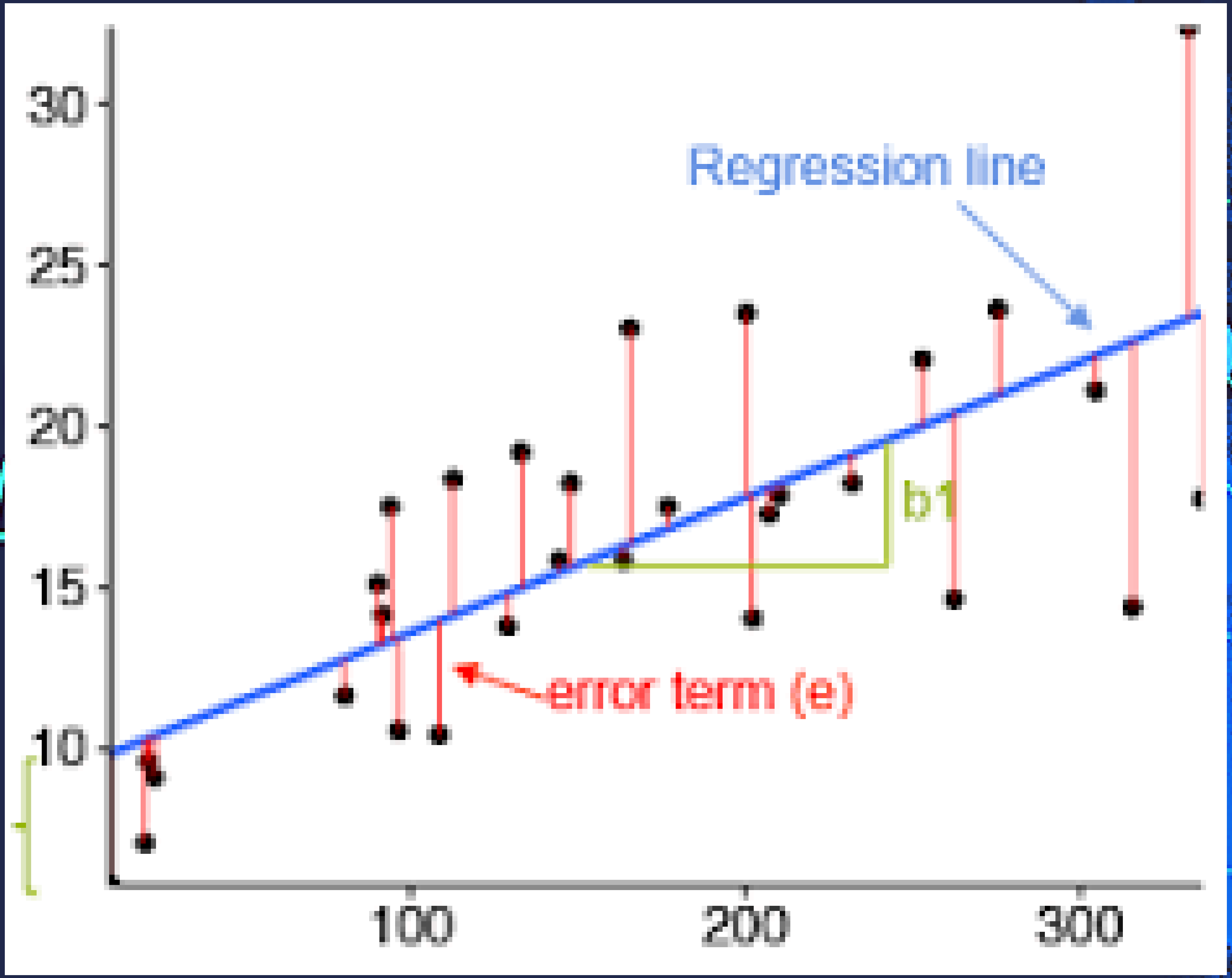
Linear Regression with One Variable

THE COST FUNCTION

We can measure the accuracy of our hypothesis function by using a cost function. This takes an average (actually a fancier version of an average) of all the results of the hypothesis with inputs from x's compared to the actual output y's.

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

Linear Regression with One Variable



THE COST FUNCTION

Linear Regression with One Variable

THE COST FUNCTION

THE BEST POSSIBLE LINE WILL BE SUCH SO THAT THE AVERAGE SQUARED VERTICAL DISTANCES OF THE SCATTERED POINTS FROM THE LINE WILL BE THE LEAST.

Linear Regression with One Variable

THE COST FUNCTION

IN THE BEST CASE, THE LINE SHOULD PASS THROUGH ALL THE POINTS OF OUR TRAINING DATA SET. IN SUCH A CASE WHAT WILL BE THETHE VALUE OF $J(\Theta0,\Theta1)$?

Linear Regression with One Variable

THE GRADIENT DESCENT

SO WE HAVE OUR HYPOTHESIS FUNCTION AND WE HAVE A WAY OF MEASURING HOW WELL IT FITS INTO THE DATA. NOW WE NEED TO ESTIMATE THE PARAMETERS IN HYPOTHESIS FUNCTION. THAT'S WHERE GRADIENT DESCENT COMES IN.

Linear Regression with One Variable

THE GRADIENT DESCENT

The gradient descent algorithm is:

repeat until convergence:

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

where

j=0,1 represents the feature index number.

Intuitively, this could be thought of as:

repeat until convergence:

$\theta_j := \theta_j - \alpha[\text{Slope of tangent aka derivative in j dimension}]$[Slope of tangent aka derivative in j dimension]

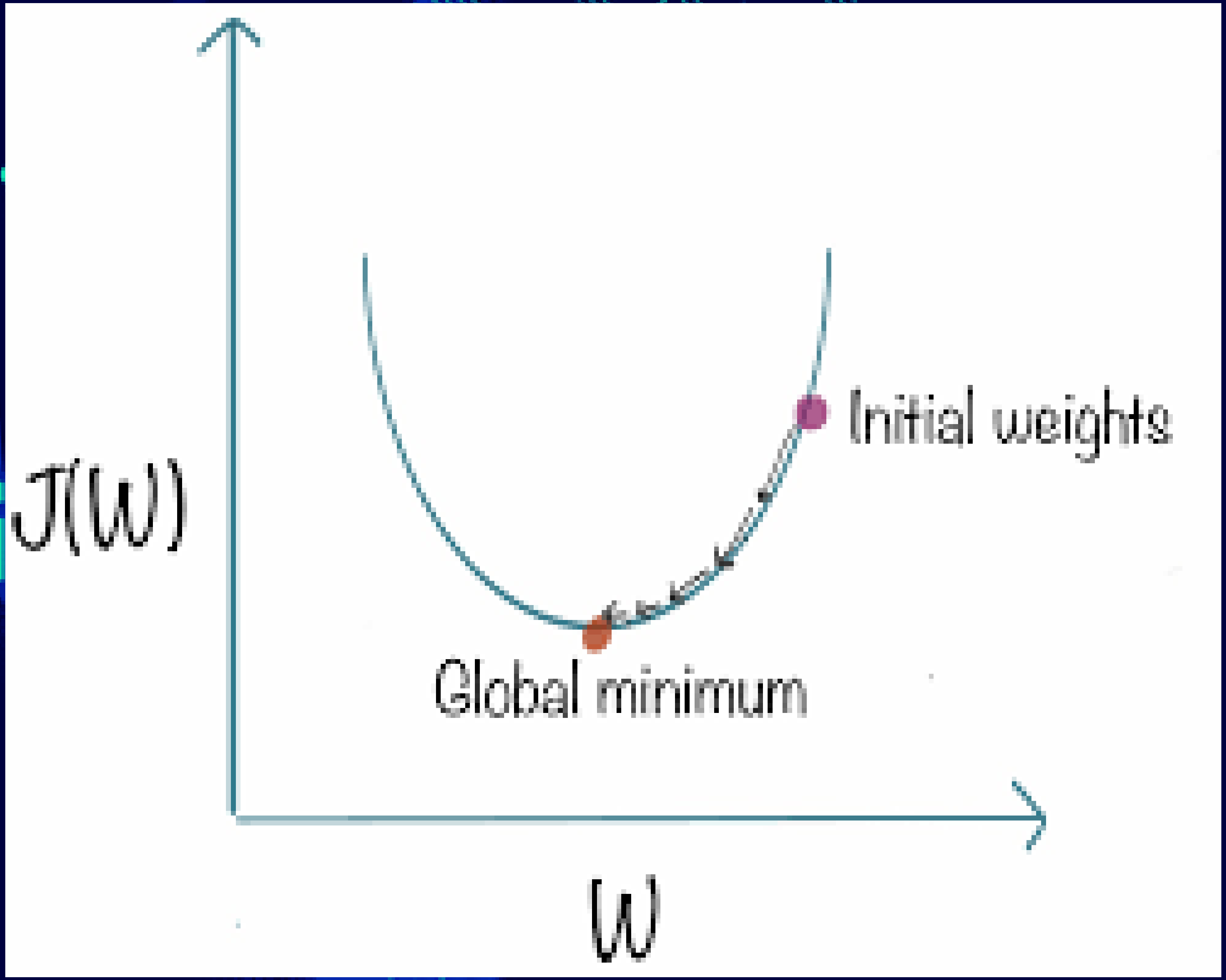
Linear Regression with One Variable

THE GRADIENT DESCENT

The way we do this is by taking the derivative (the tangential line to a function) of our cost function. The slope of the tangent is the derivative at that point and it will give us a direction to move towards. We make steps down the cost function in the direction with the steepest descent, and the size of each step is determined by the parameter α, which is called the learning rate.

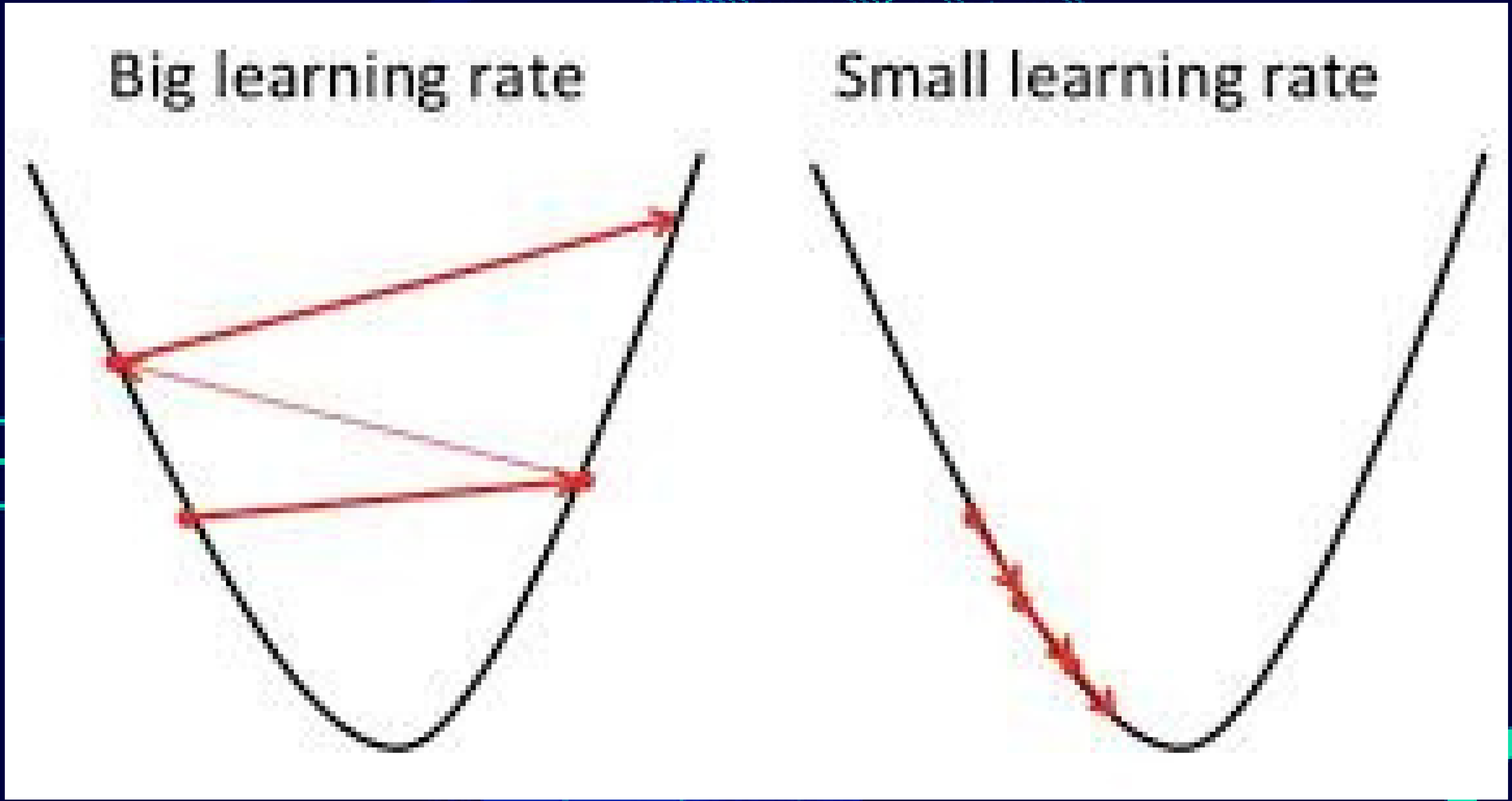
Linear Regression with One Variable

THE GRADIENT DESCENT



Linear Regression with One Variable

THE GRADIENT
DESCENT
LEARNING RATE



Linear Regression with One Variable

GRADIENT DESCENT FOR LINEAR REGRESSION

When specifically applied to the case of linear regression, a new form of the gradient descent equation can be derived.

repeat until convergence {

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

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

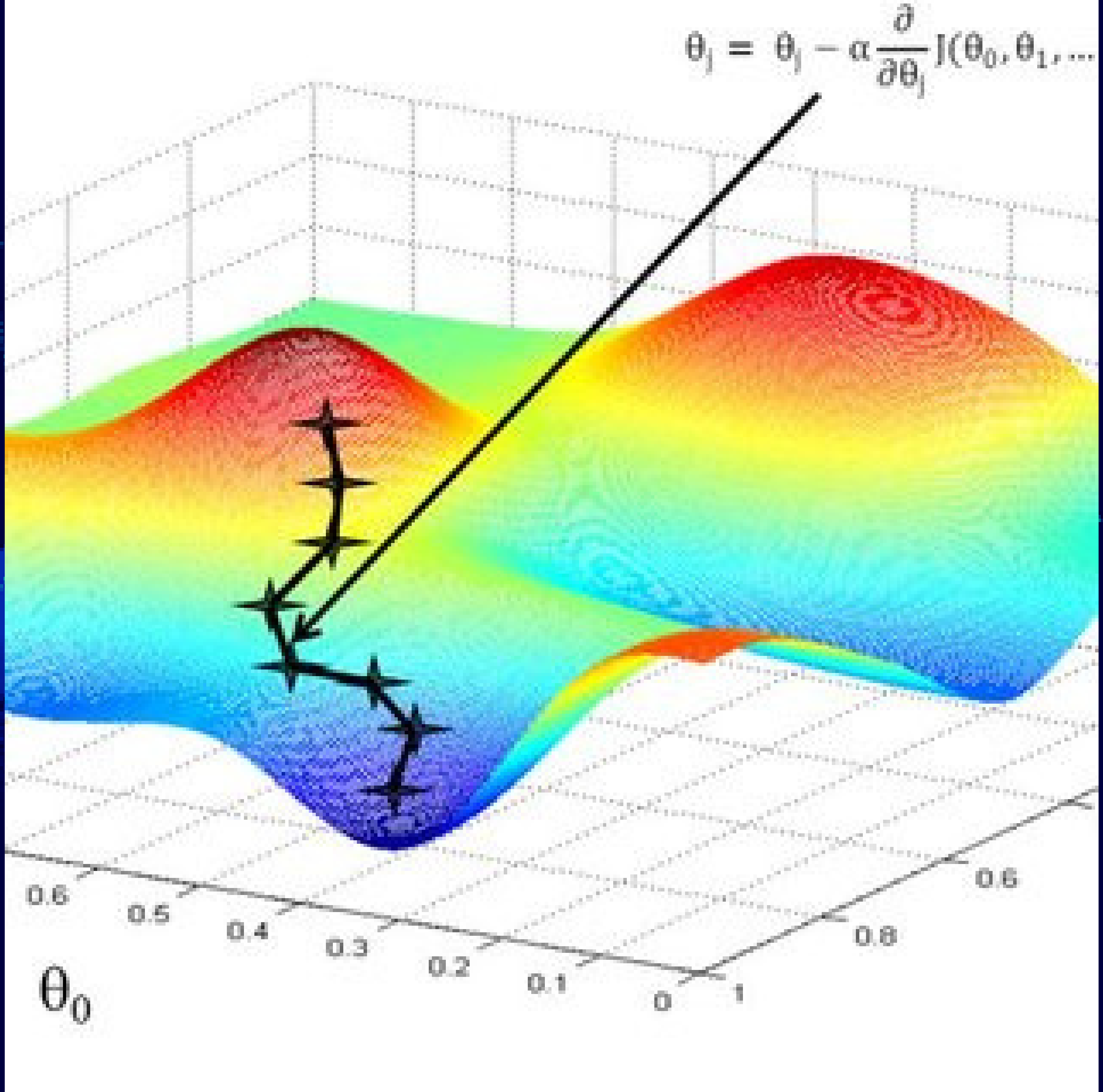
}

Linear Regression with One Variable

GRADIENT DESCENT

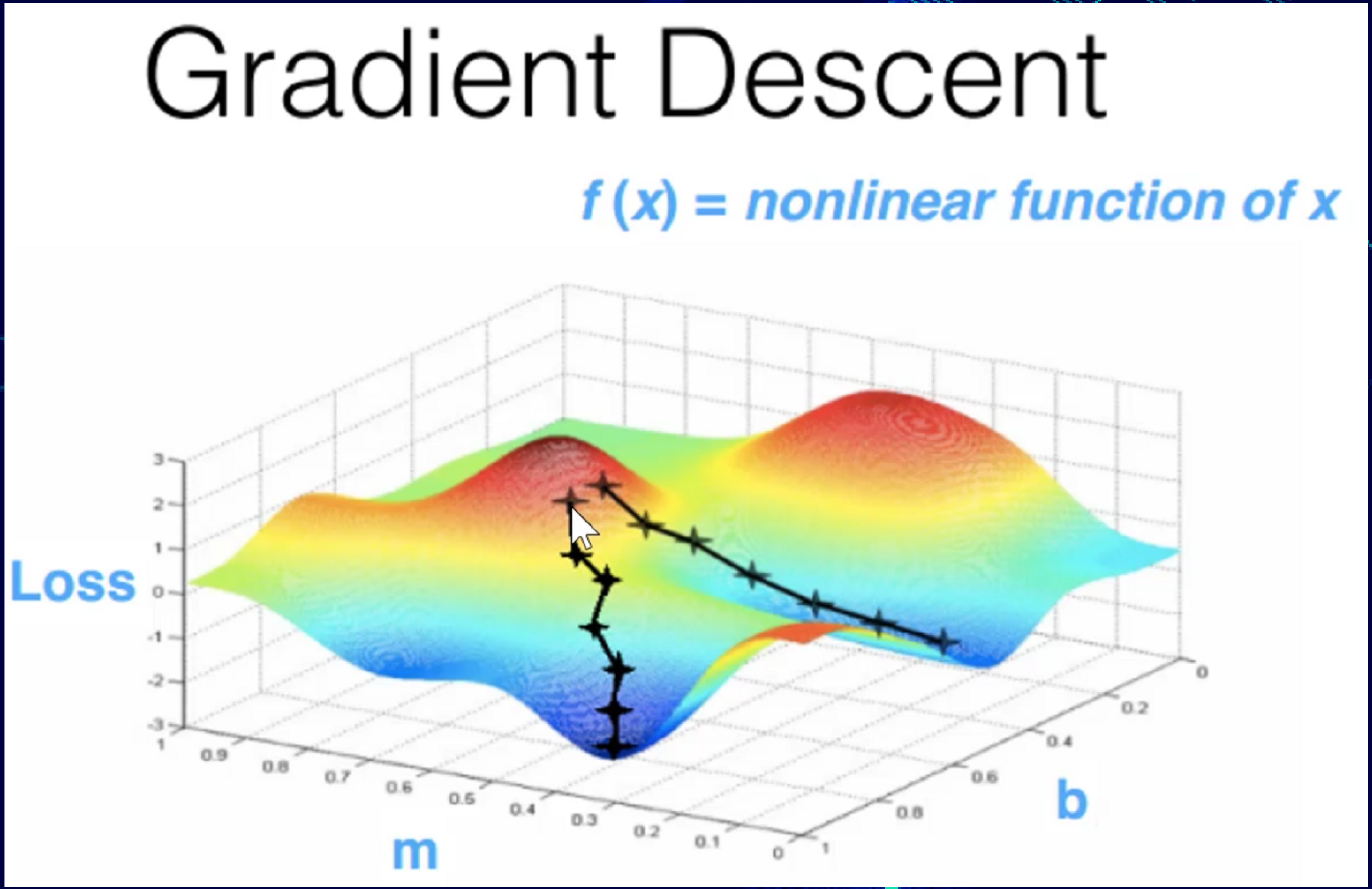
FOR

LINEAR REGRESSION



Linear Regression with One Variable

GRADIENT DESCENT FOR LINEAR REGRESSION



RECAP

Gradient descent algorithm

$$
\begin{array}{l}
\text{repeat until convergence } \{ \\
\quad \theta_j := \theta_j - \alpha \dfrac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
\qquad \text{(for } j = 1 \text{ and } j = 0) \\
\}
\end{array}
$$

Linear Regression Model

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

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

RECAP

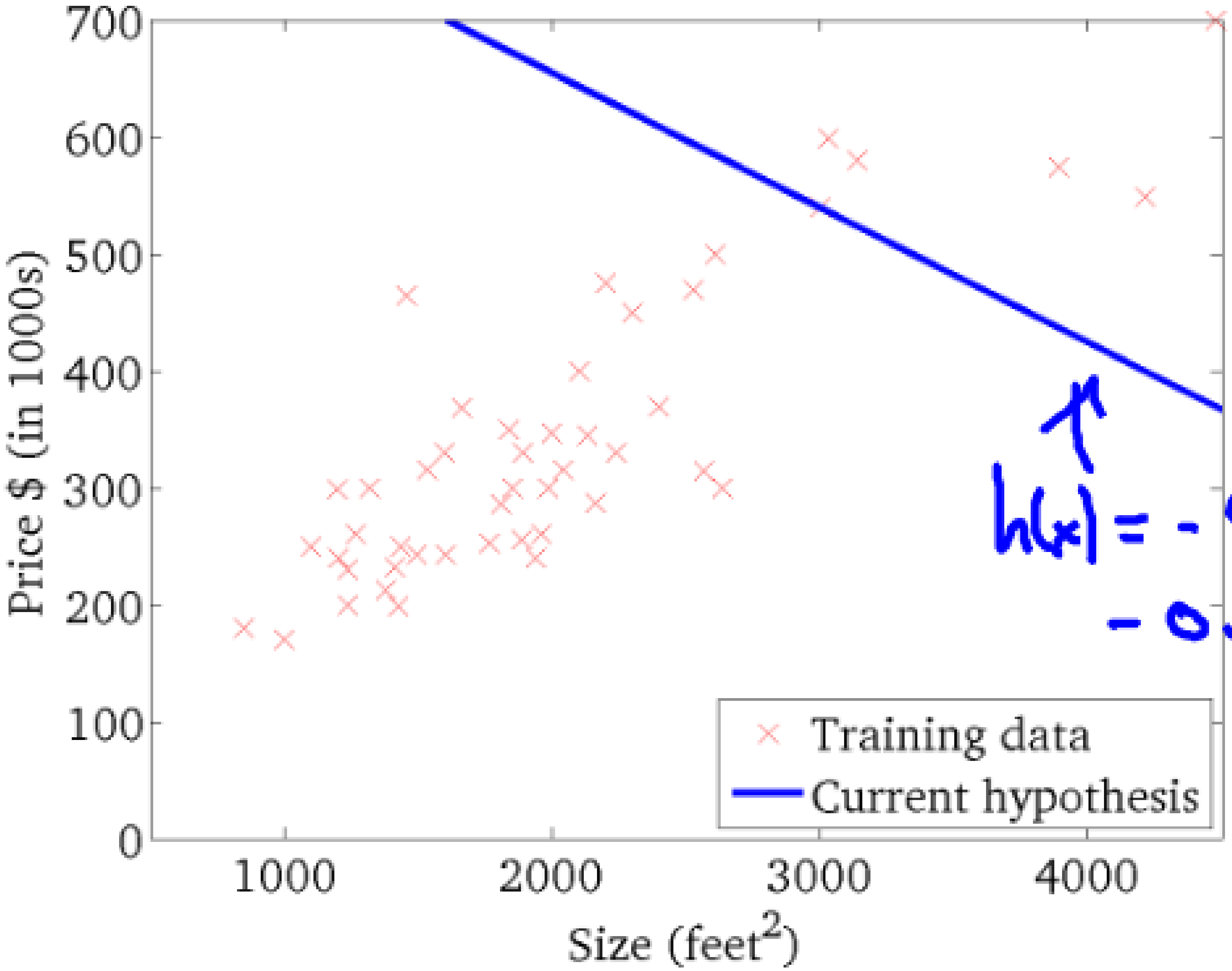
OUR AIM IS TO GET A BEST FIT LINE FOR OUR MODEL

THE LOCAL MINIMUM OF THE COST FUNCTION WILL GIVE US THE BEST FIT LINE

THE GRADIENT DESCENT WILL HELPS TO REACH THE LOCAL MINIMUM

$$h_\theta(x)$$

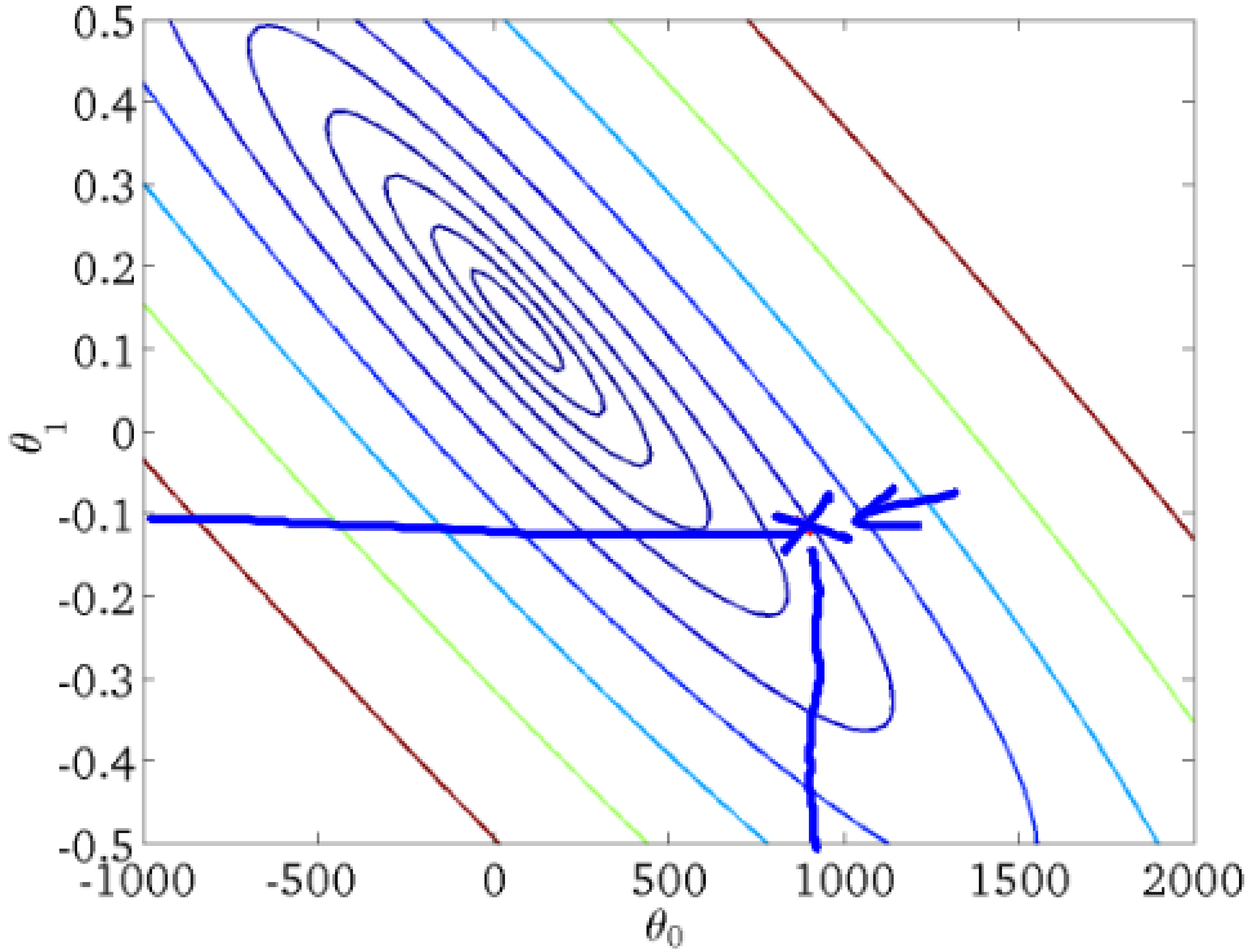
(for fixed θ_0, θ_1, this is a function of x)



$h(x) = -900 - 0.1x$

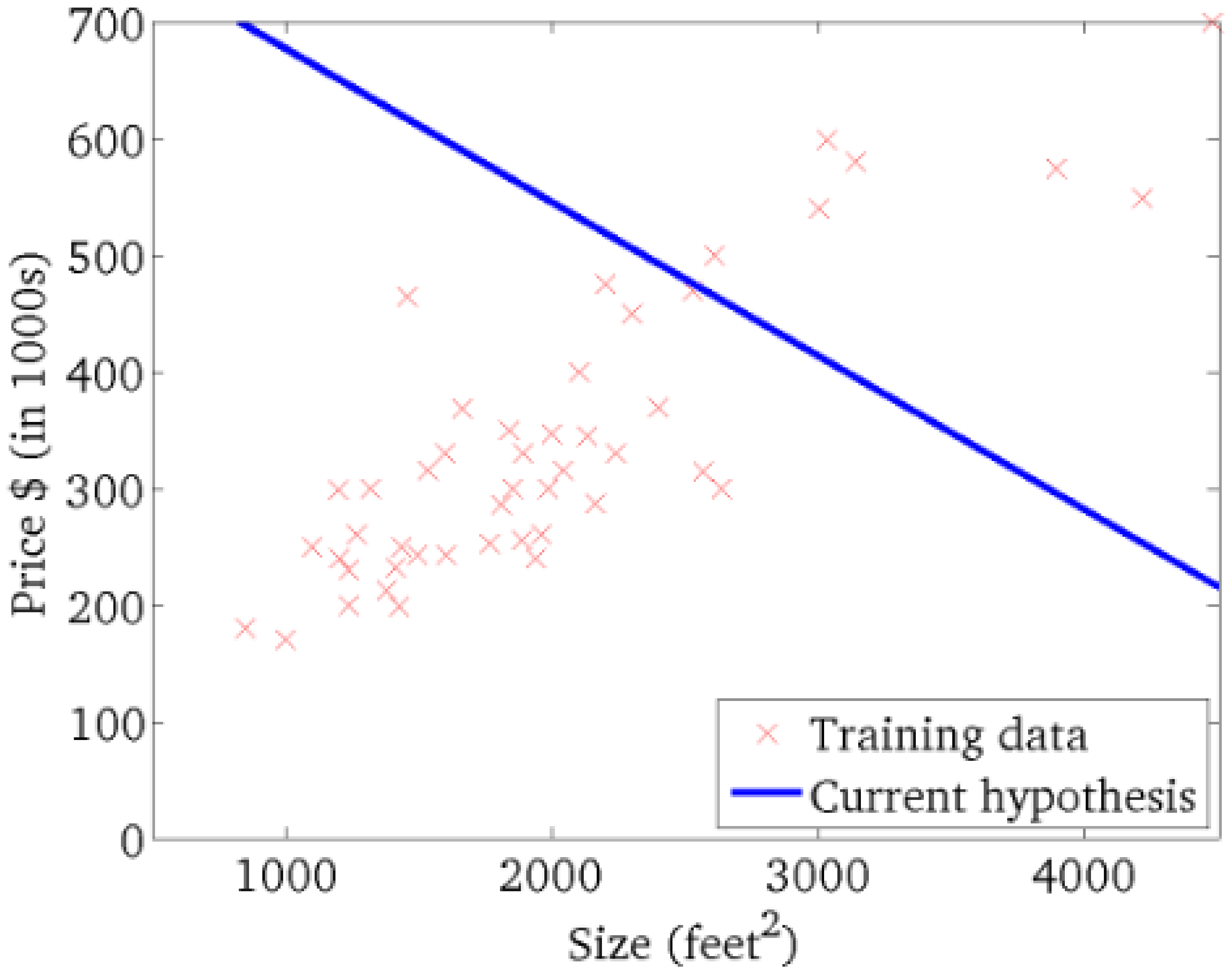
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



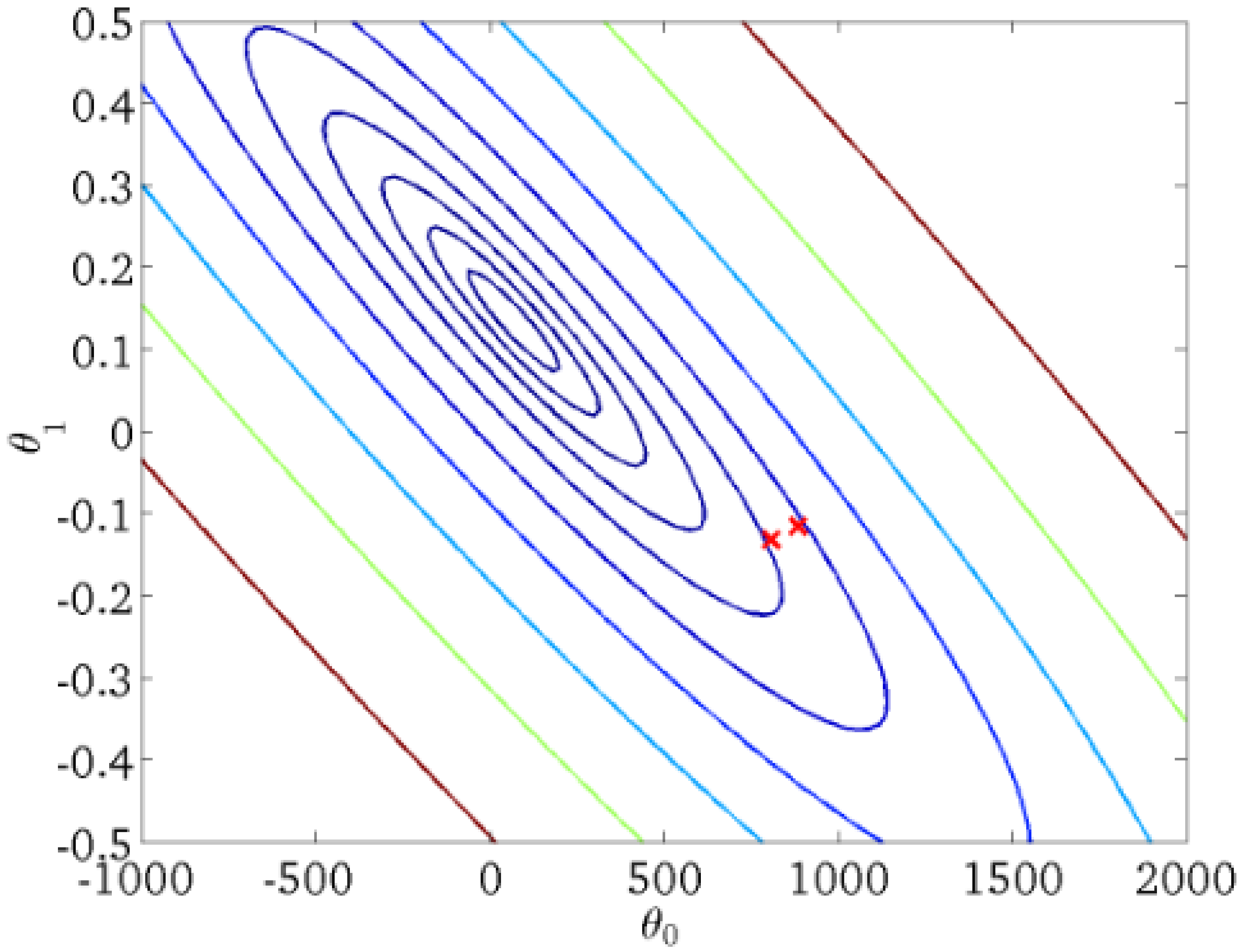
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



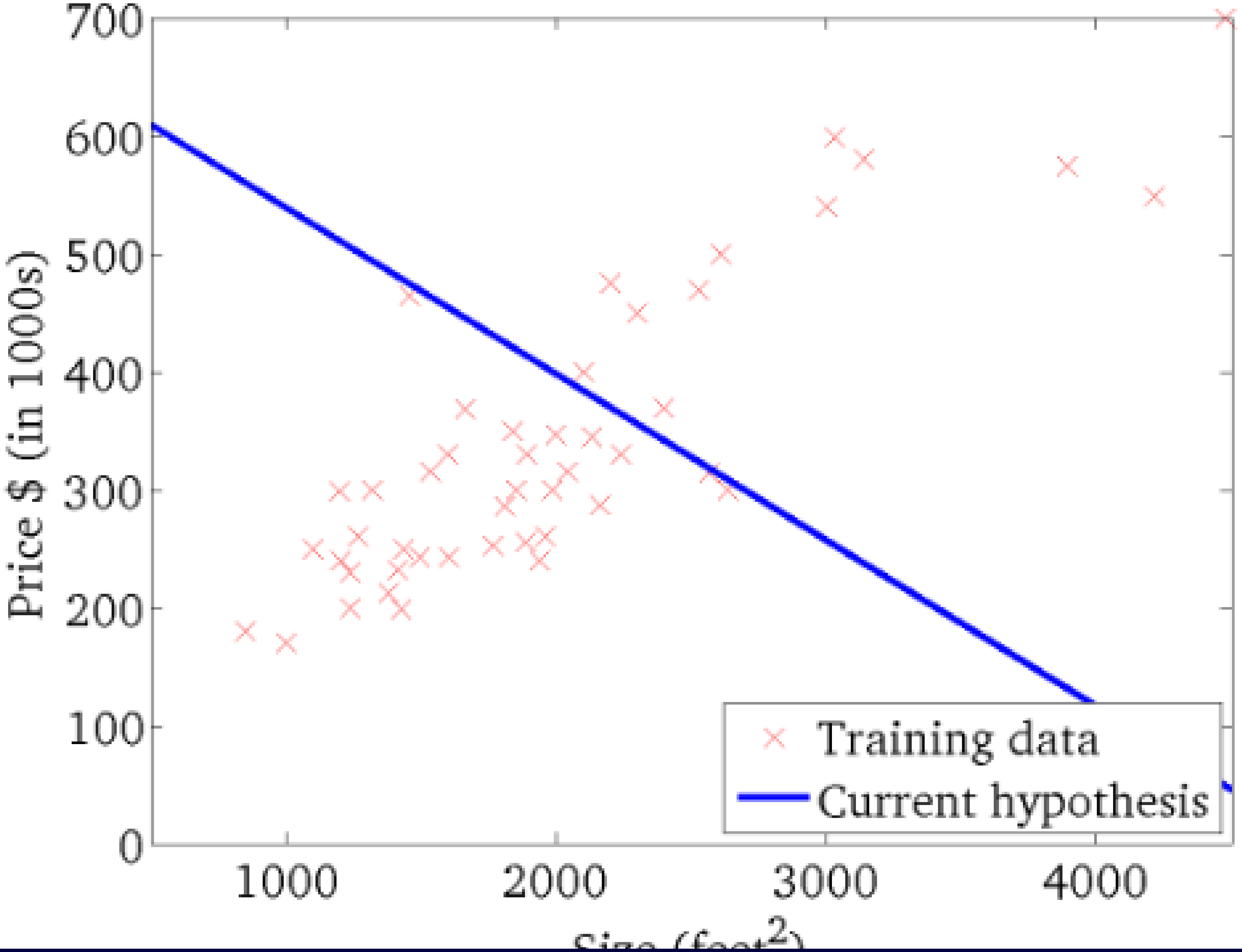
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



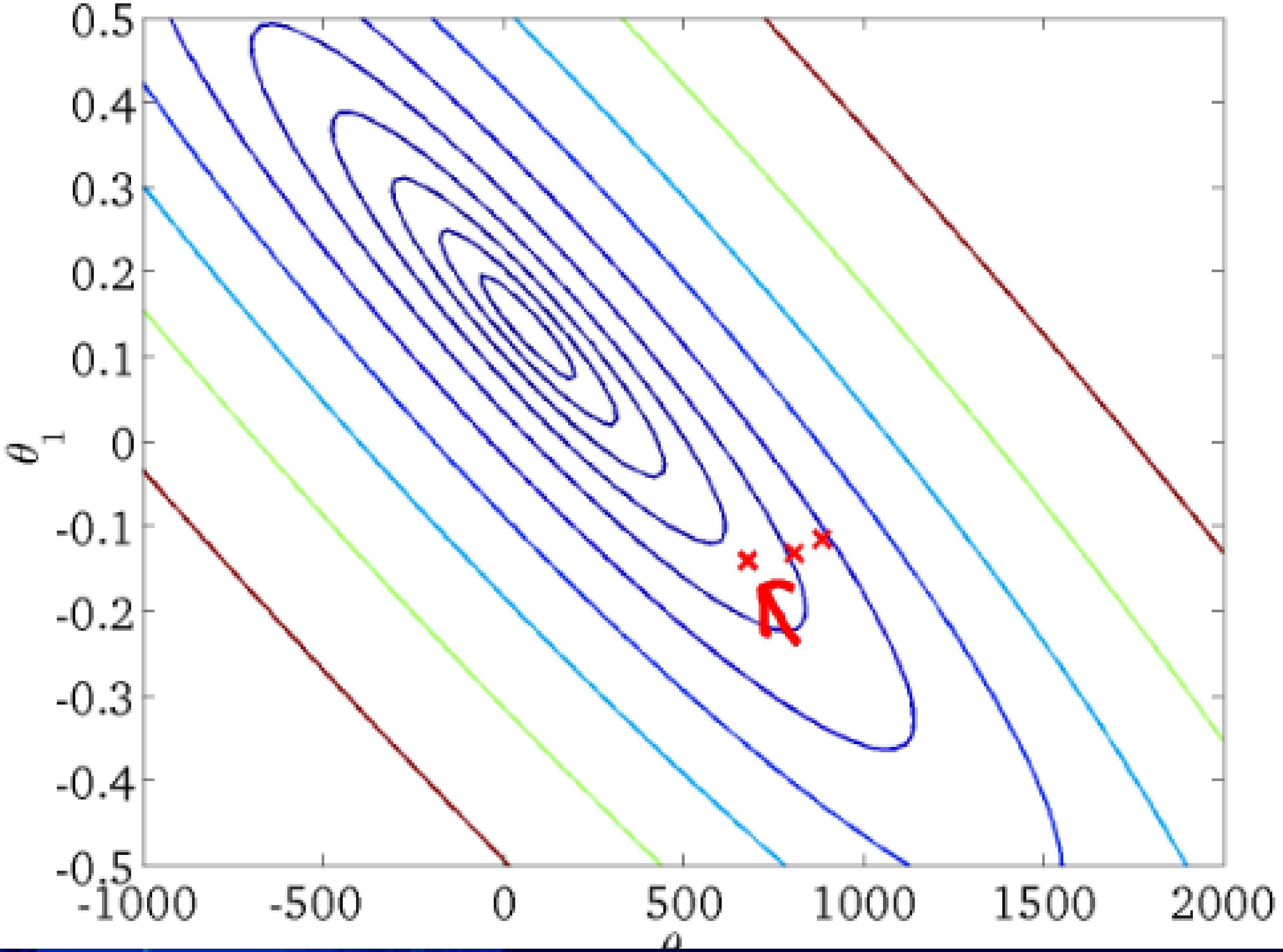
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



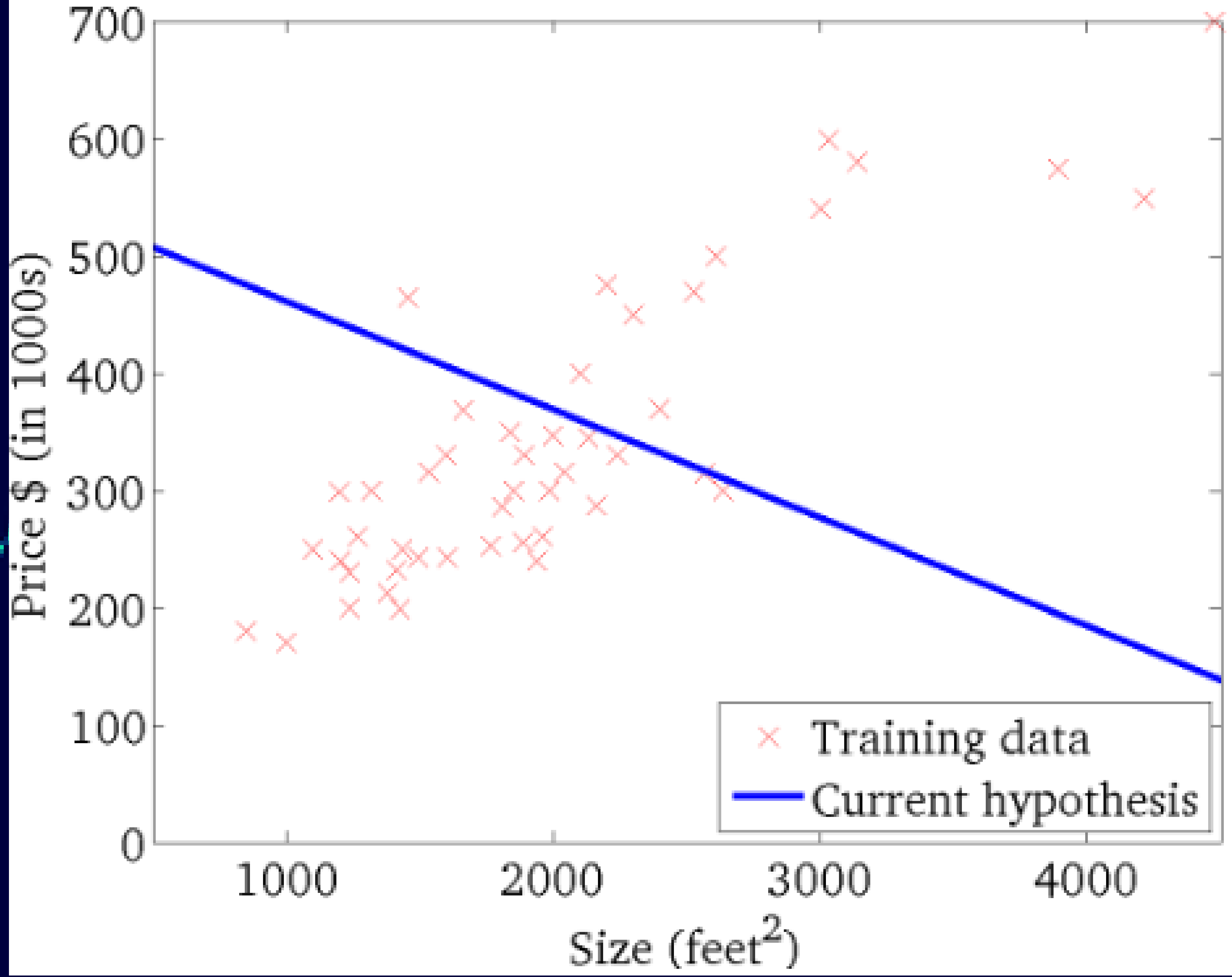
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



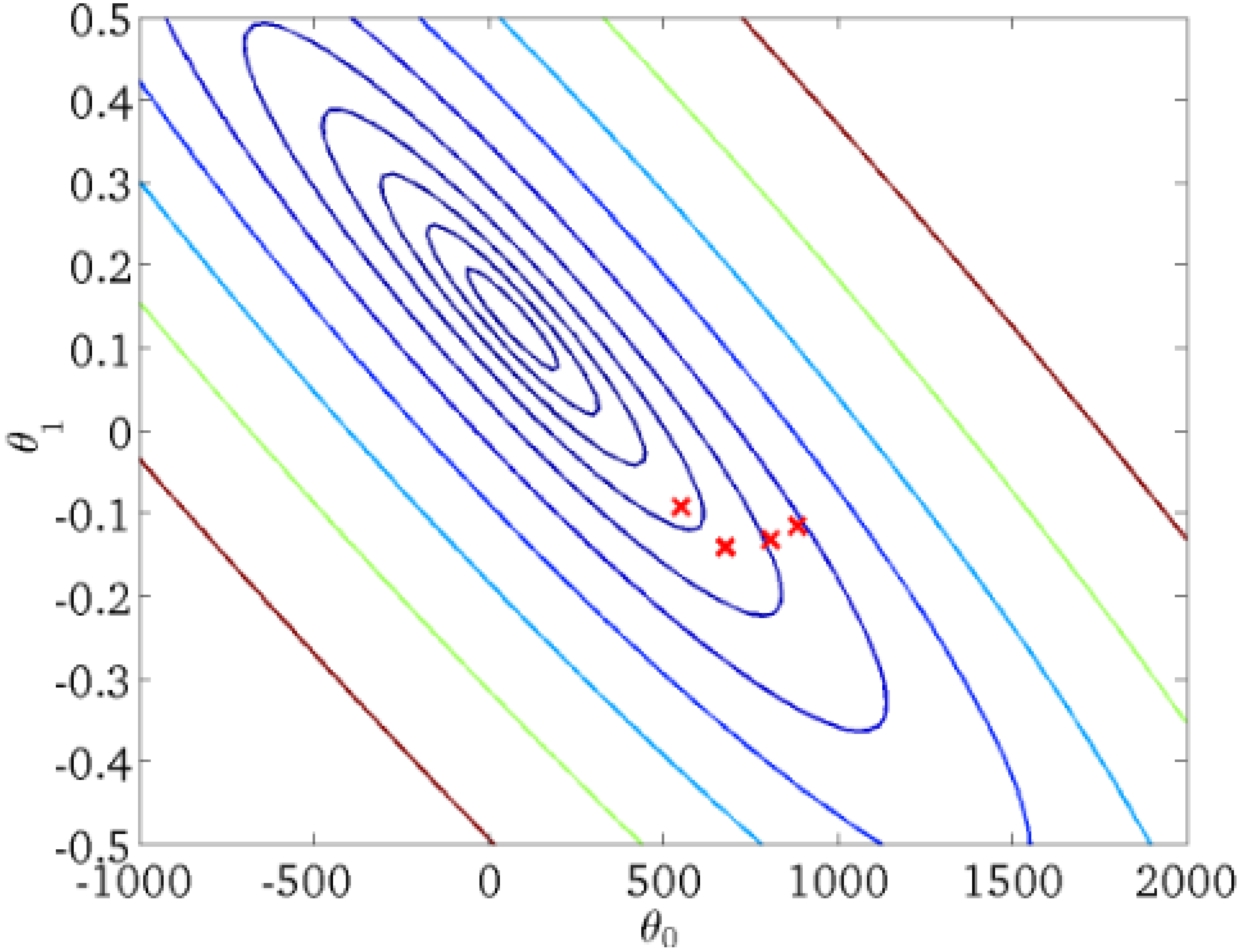
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



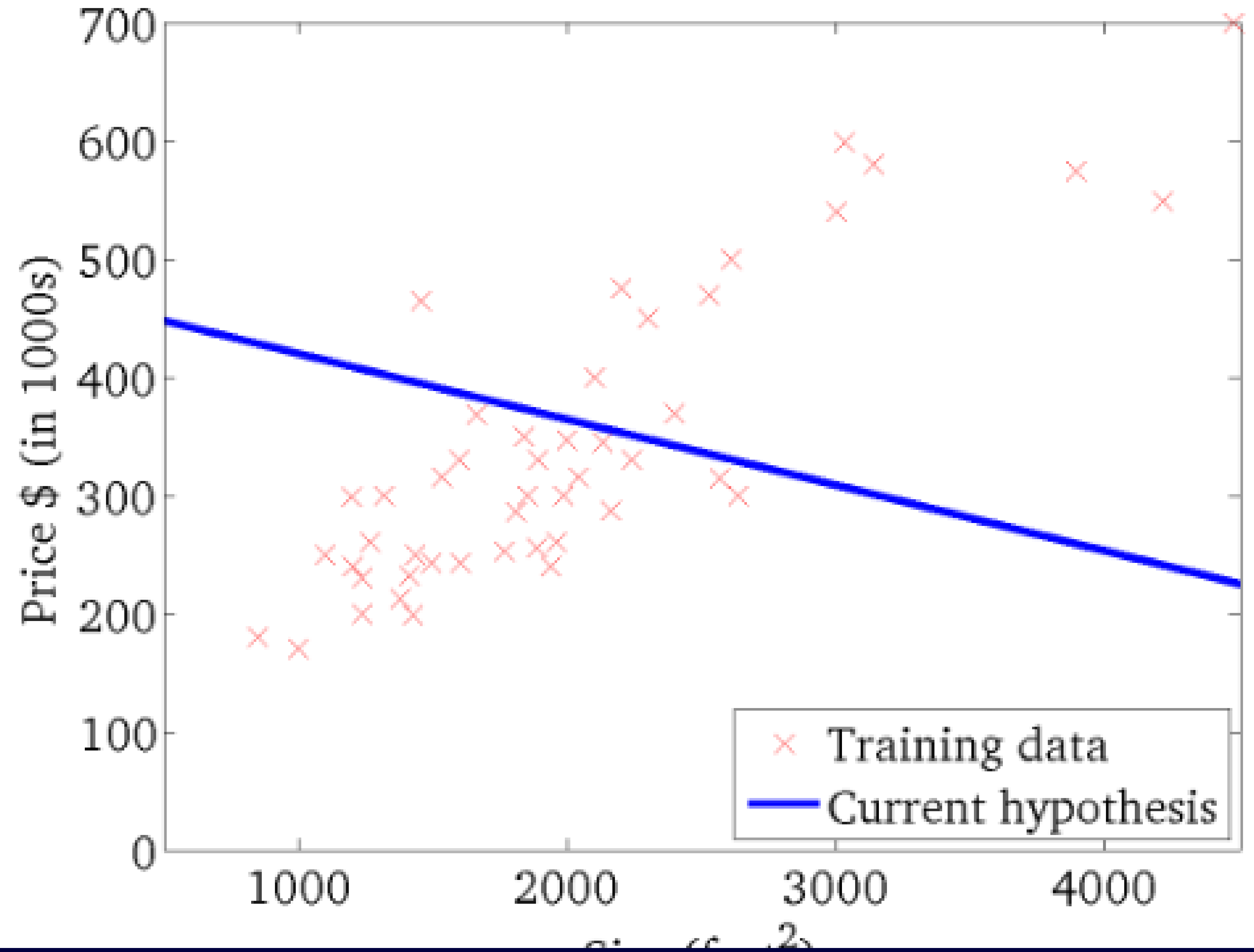
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



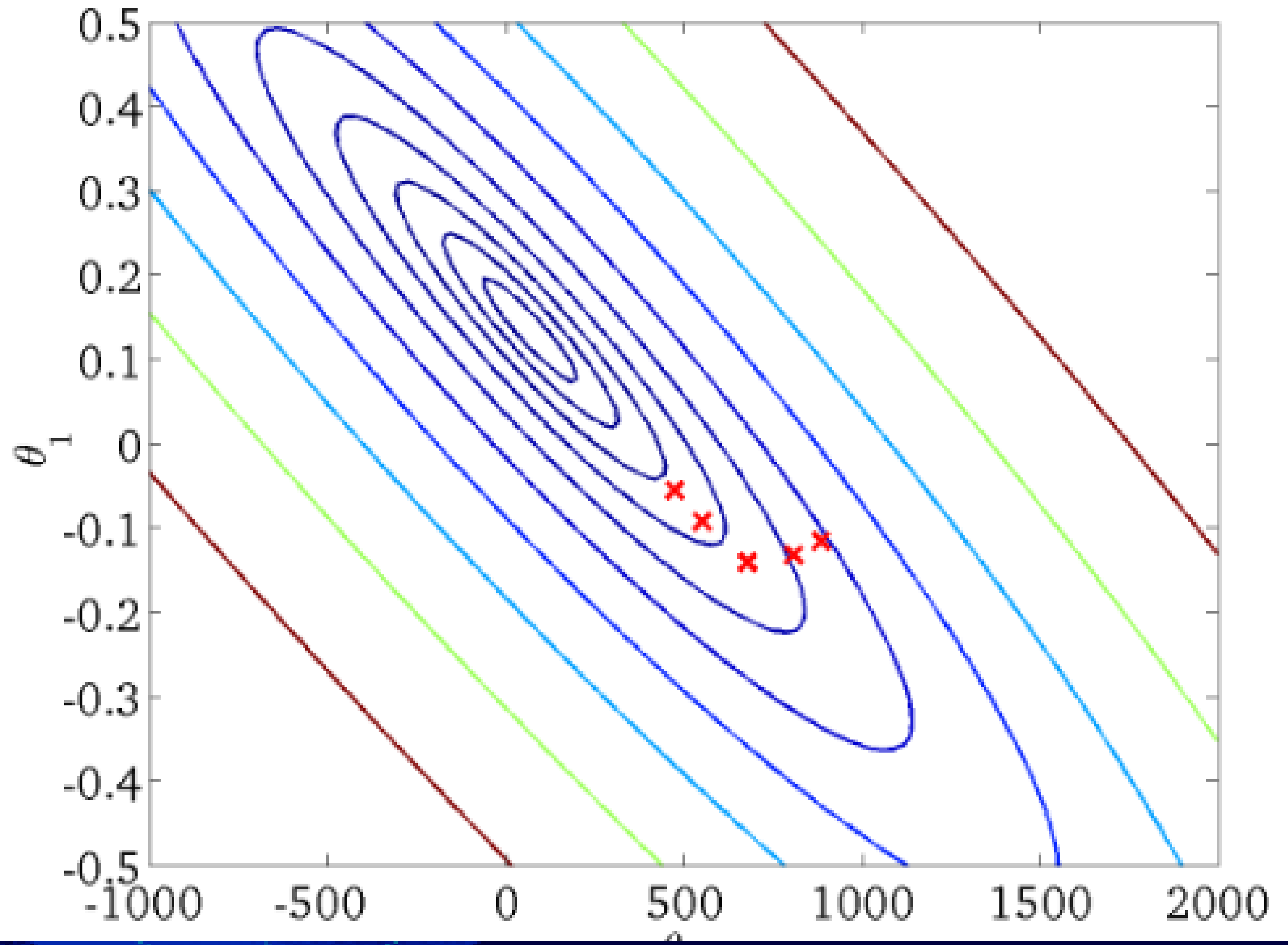
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



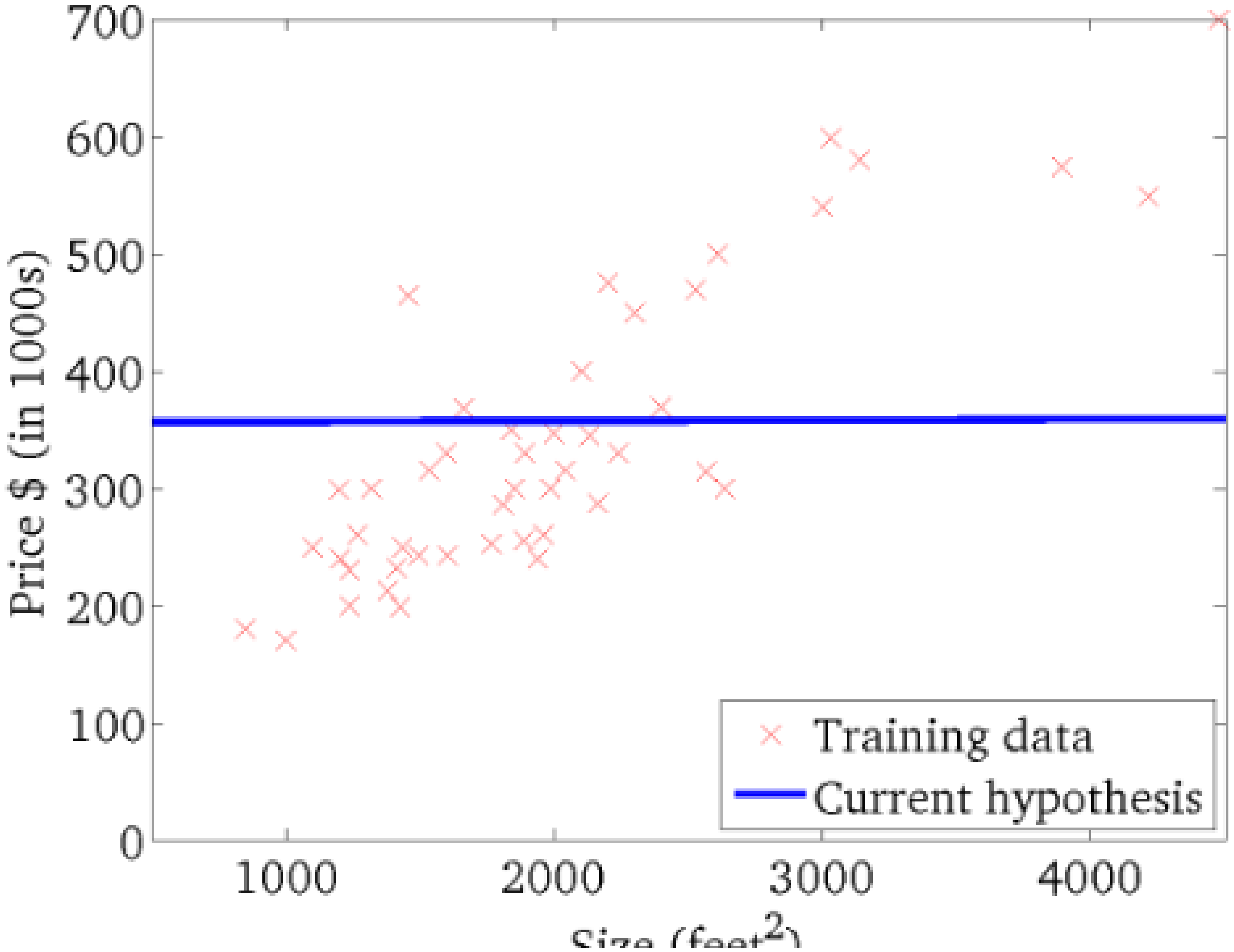
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



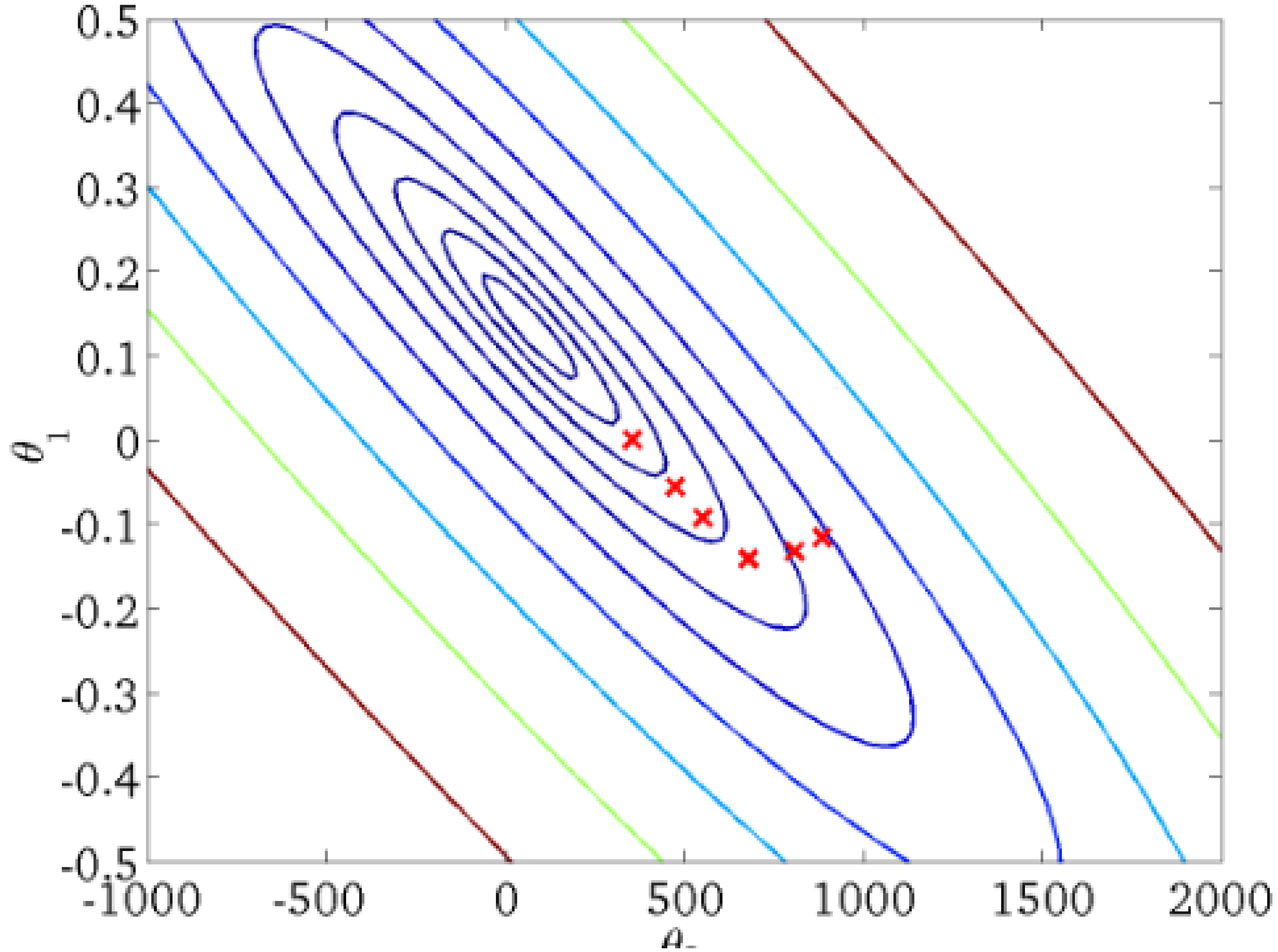
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



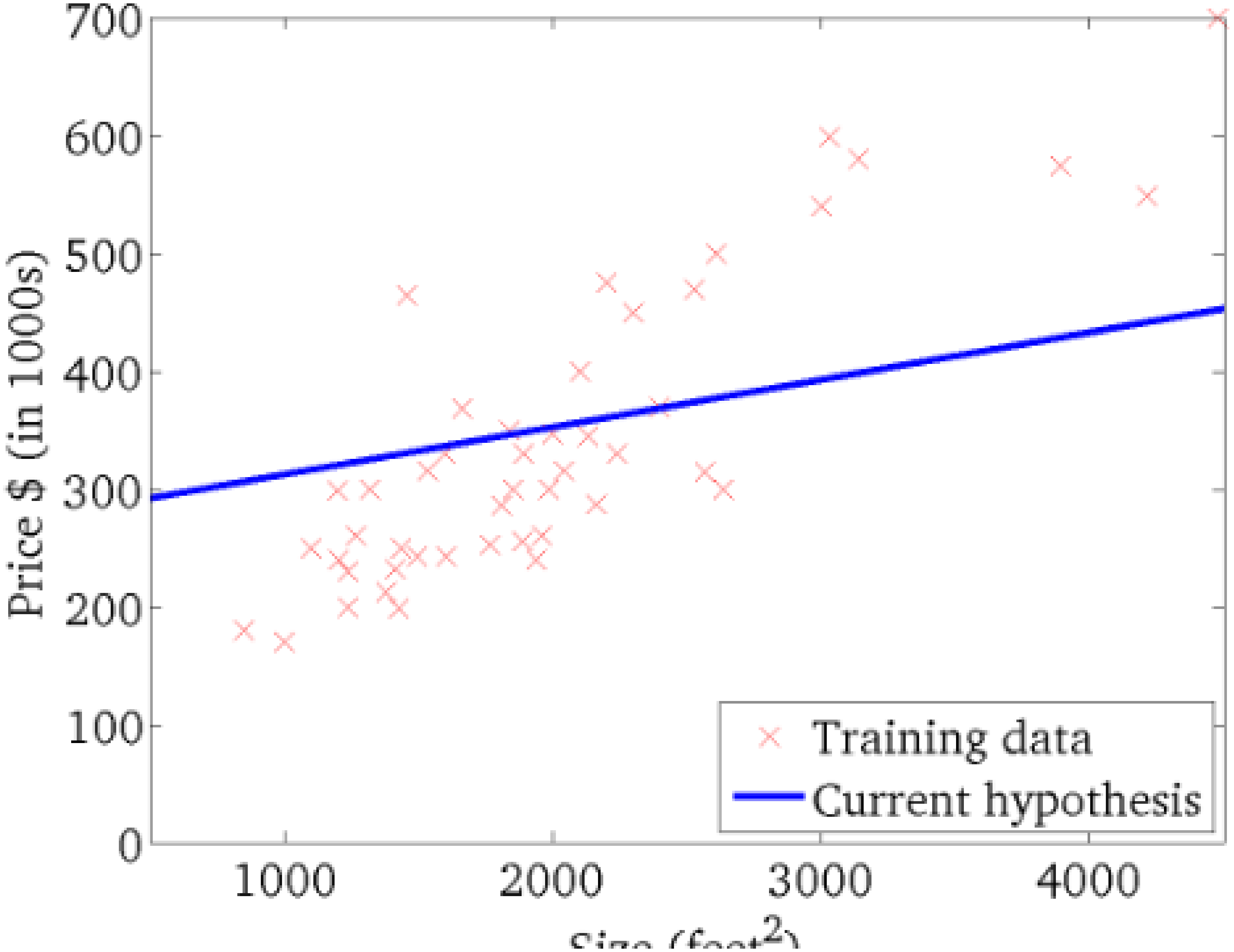
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



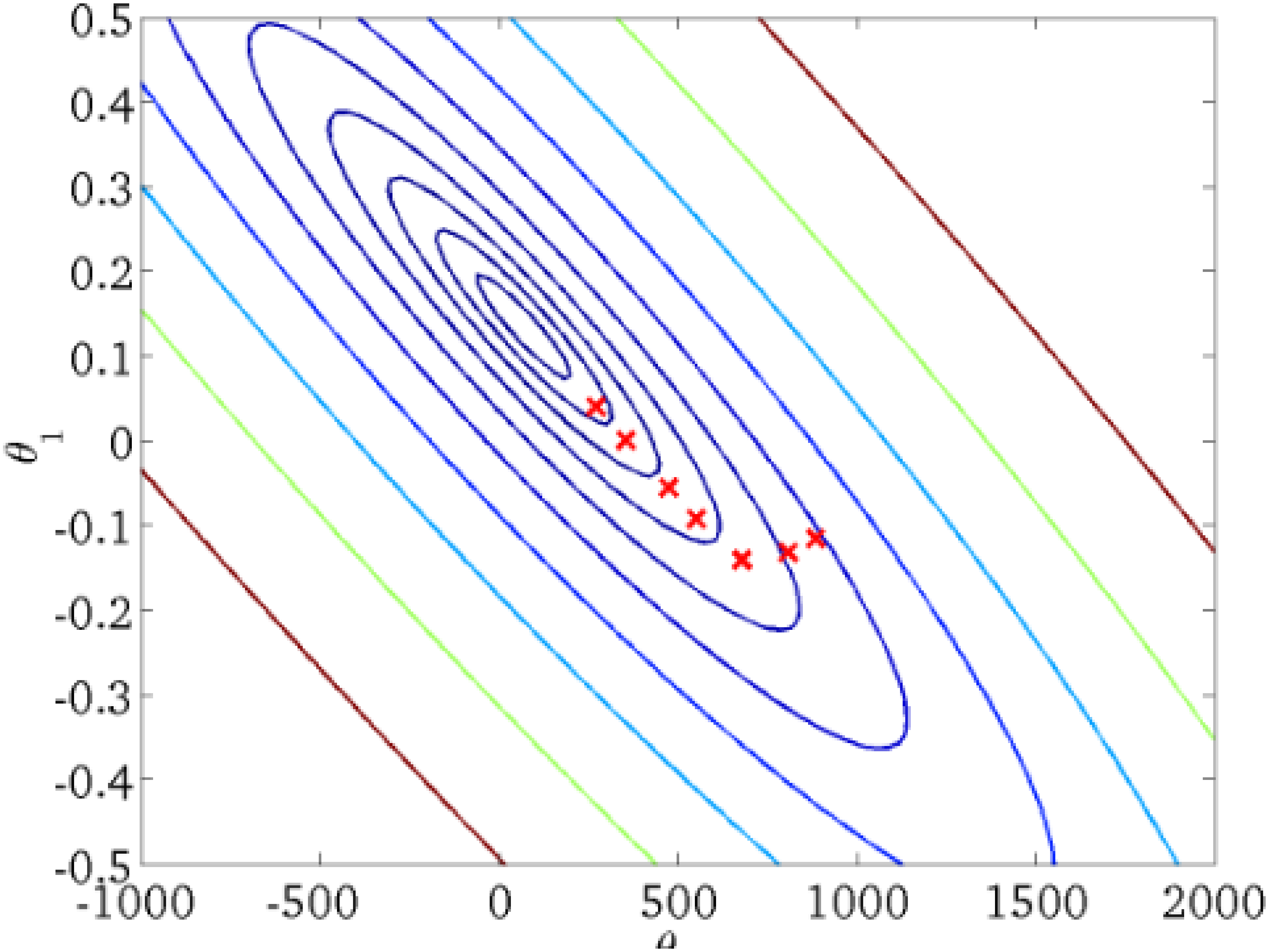
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



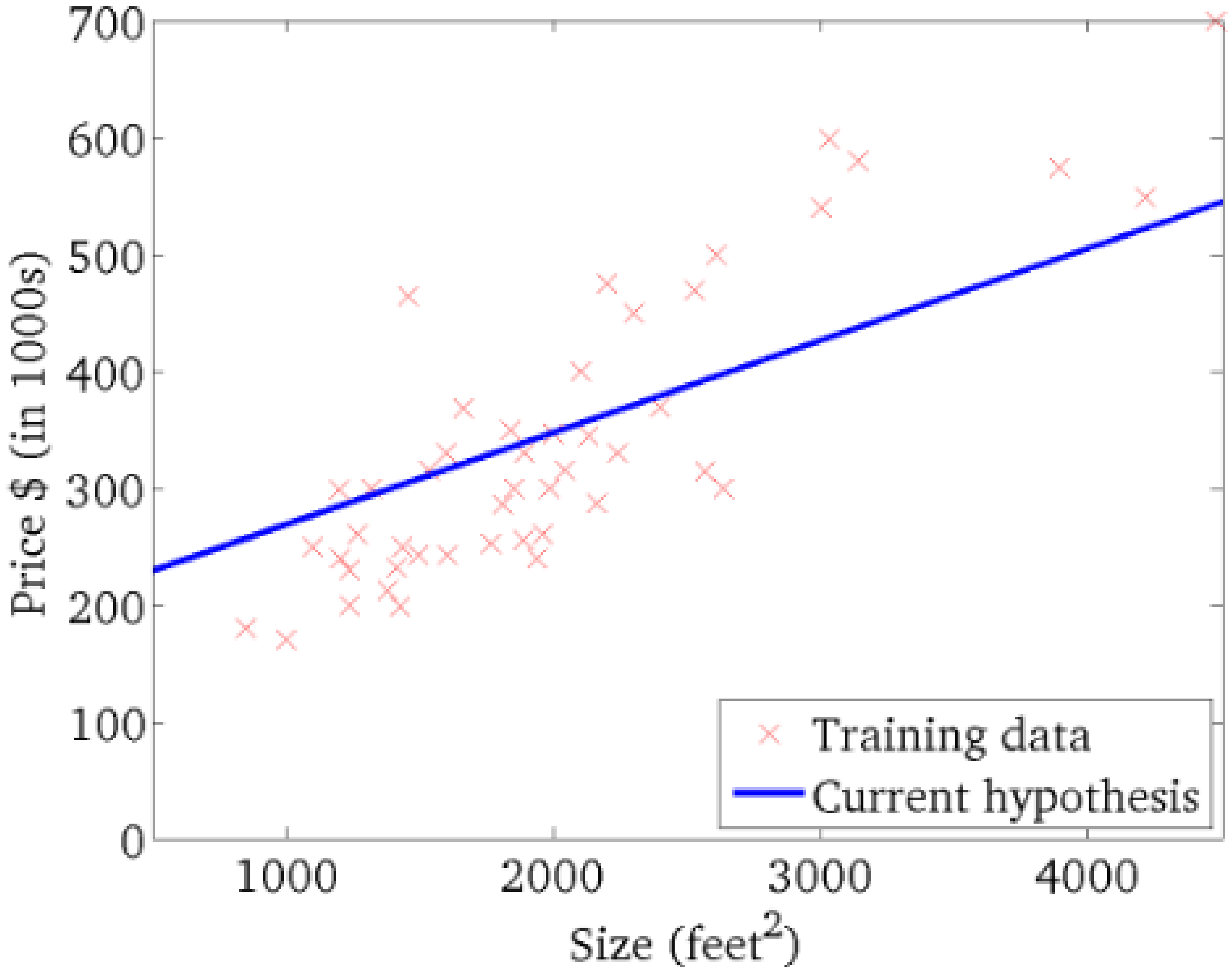
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



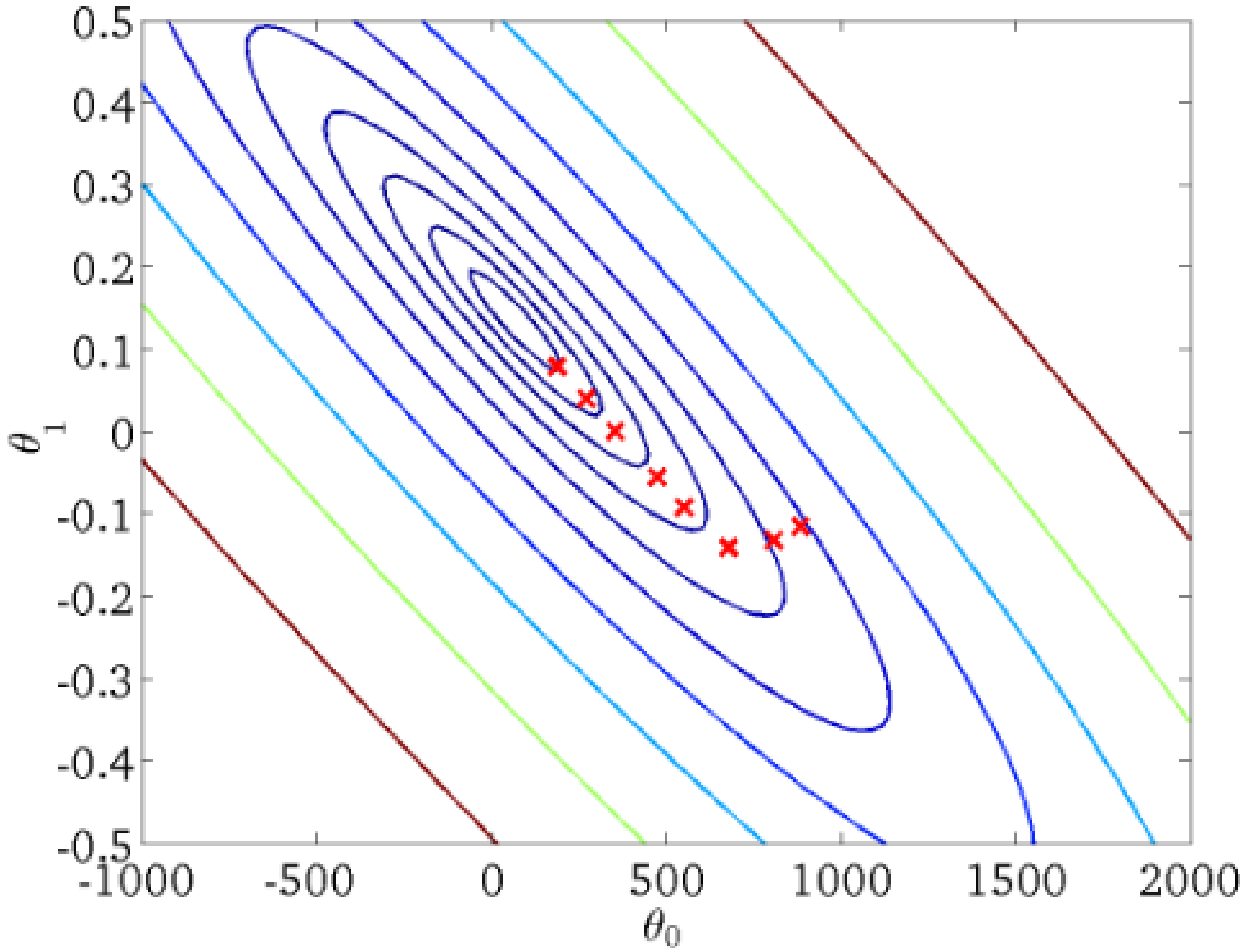
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



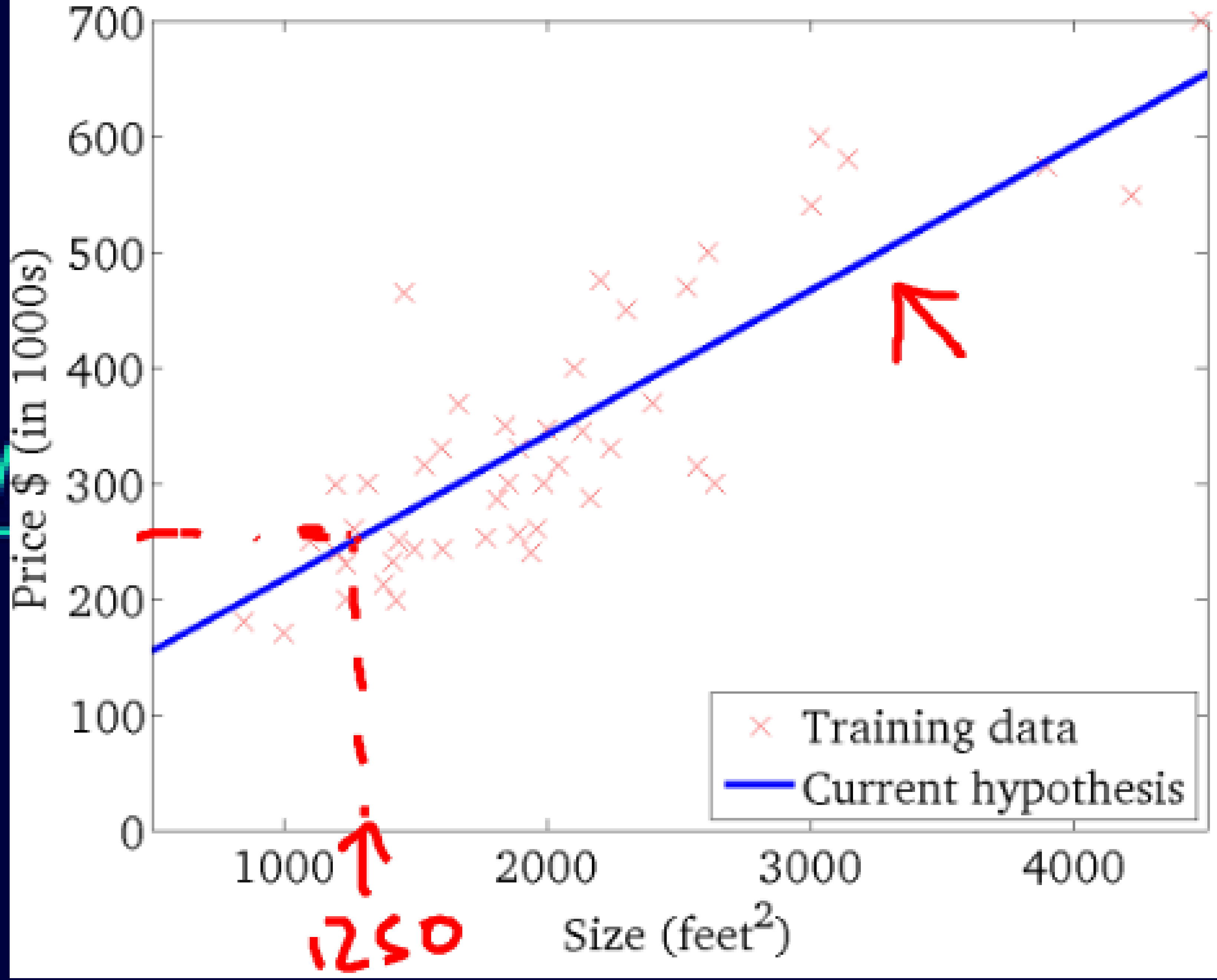
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



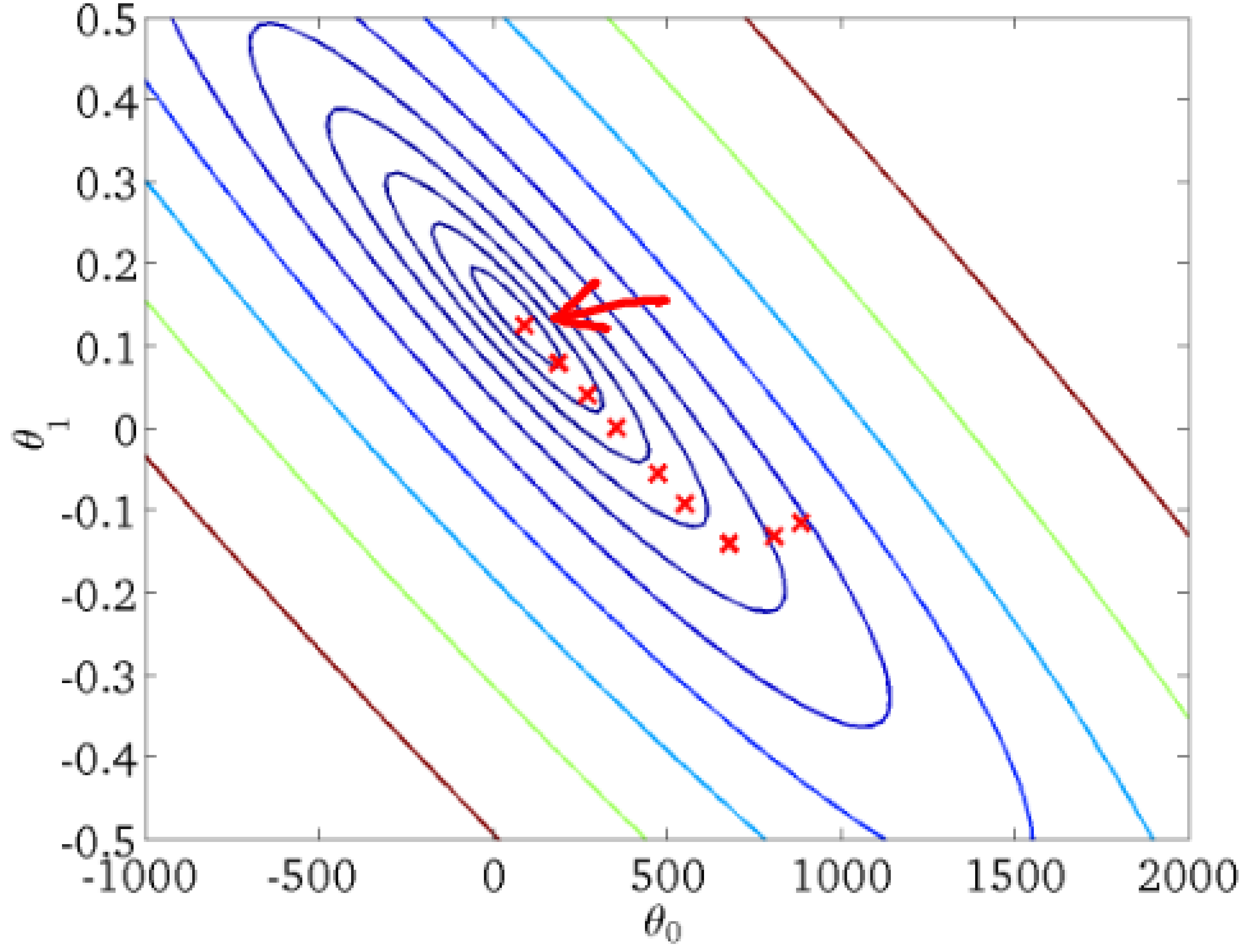
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)




YOU DID MATH?
GOOD JOB!
A = πr²
C = 2πr
V = πr²h

GNU Octave

GNU Octave is software featuring a high-level programming language, primarily intended for numerical computations. Octave helps in solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB.

Basic Operations

Moving Data Around

Computing on Data

Plotting Data

Control statements: for, while, if statements

Functions

Vectorization

NEXT SECTION ON

ARTIFICIAL NEURAL NETWORKS