Supervised Learning

Regression Models

Diego Campos Sobrino

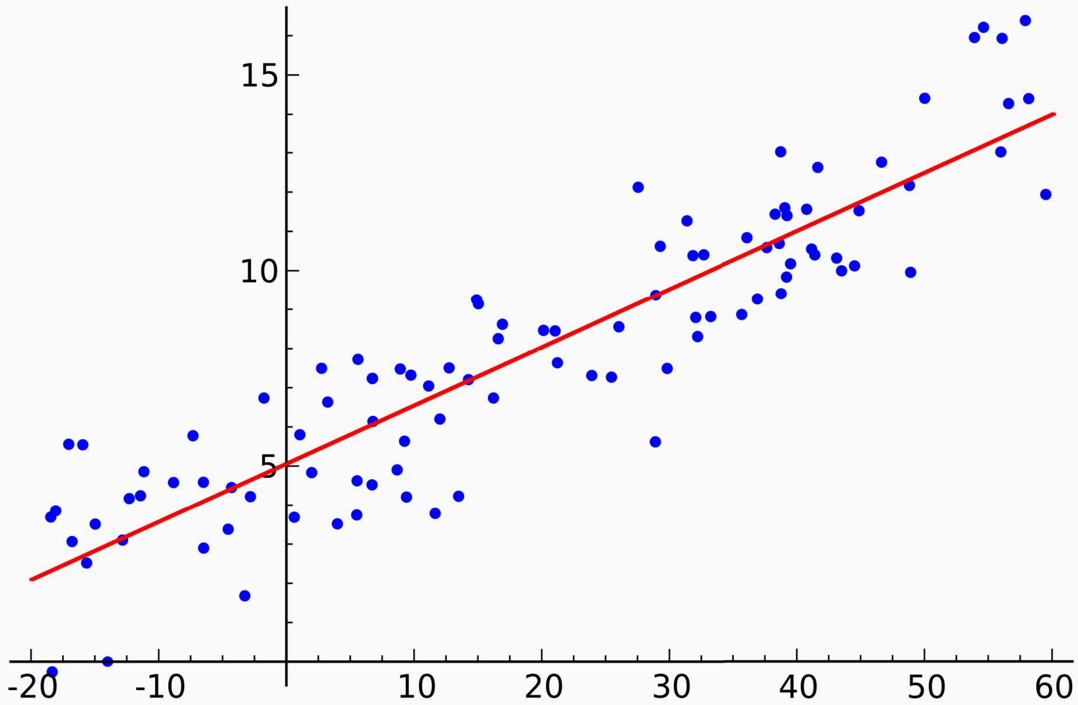
UNIVERSIDAD POLITÉCNICA DE YUCATÁN

BIS Universities

Table of contents

In regression problems, we take input variables and try to fit the output onto a continuous expected result function.

$$\boldsymbol{x} \in \mathbb{R}^n \qquad y \in \mathbb{R} \qquad f : \mathbb{R}^n \to \mathbb{R}$$



A linear model makes a prediction by simply computing a weighted sum of the input features, plus a constant called the *bias* or *intercept*.

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

where:

- $\hat{y}$ is the predicted value.
- n is the number of features.
- x_i is the i^{th} feature value.
- θ_j is the j^{th} model parameter (weights).

In Machine Learning we usually use the vectorized form of the equation

$$\hat{y} = h_\theta(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}$$

where:

- $\boldsymbol{\theta}$ is the parameter vector containing θ_0 to θ_n
- $\boldsymbol{x}$ is the feature vector containing x_0 to x_n with $x_0 = 1$
- h_θ is the hypothesis function using model parameters $\boldsymbol{\theta}$

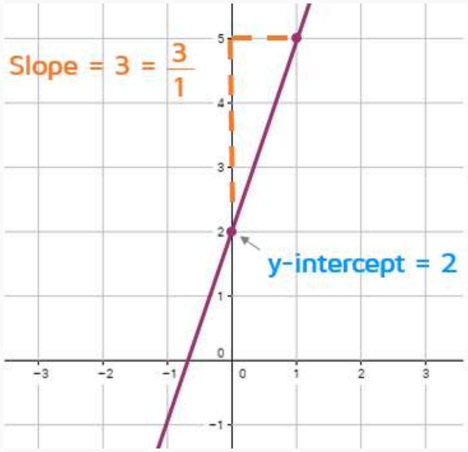
$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\boldsymbol{\theta}^T \boldsymbol{x} = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \dots & \theta_n \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\boldsymbol{\theta}^T \boldsymbol{x} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- Predict a single output value y from a single input value x.
- The input can be seen as the cause and the output the effect.
- We are looking for a function called h_θ that tries to map the input data (the x's) to the output data (the y's).

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



- This is like the equation of a straight line with θ_0 as the intercept (bias) and θ_1 as the slope.
- The values of θ_0 and θ_1 define the specific line used to make predictions.
- We give to $h_\theta(x)$ the value of x and receive and estimated output $\hat{y}$.

Suppose we have the following set of training data:

x	y
0	4
1	7
2	7
3	8

We can make a random guess about our h_θ function with $\theta_0 = 2$ and $\theta_1 = 2$, then the hypothesis function becomes $h_\theta(x) = 2 + 2x$.

- What would be the predictions $\hat{y}$'s for our x's with this model?
- How far off are our predictions $\hat{y}$'s from the actual y's?
- Can we do better?

We can try different combinations of θ_0 and θ_1 to find the straight line that "fit" better our data points.

- Measures the accuracy of our hypothesis function.
- Average-ish of all the results of the hypothesis $\hat{y}$'s compared to actual y's.

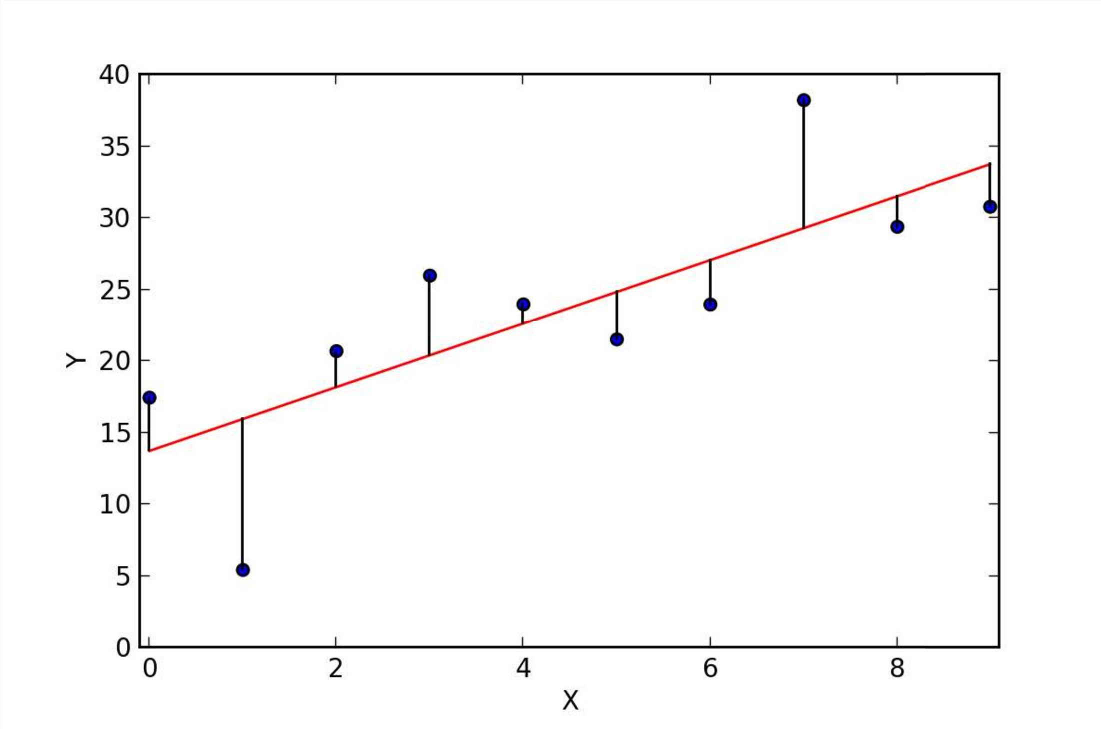
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

  - θ_0 and θ_1 are the parameters of the model.
  - m is the number of examples (observations) in the data.
  - $\hat{y}_i$ is the predicted value of the i-th example.
  - y_i is the actual value of the i-th example.

- Mean Squared Error (MSE) halved as convenience for computation (more on that later).
- Since $\hat{y}_i = h_\theta(x_i)$ sometimes is written as:

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

Ordinary Least Squares (OLS) method to estimate the unknown parameters.



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

- 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.
- The differences between predicted and actual values are called residuals $e_i = \hat{y}_i - y_i$.
- If the line pass through all points of the training set then $J(\theta_0, \theta_1) = 0$.

To find the value of θ_0 and θ_1 that minimizes J there is a closed-form solution.

For univariate linear regression:

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$$

$$S_{xx} = \frac{1}{m}\sum_{i=1}^{m} (x_i - \bar{x})(x_i - \bar{x})$$

$$S_{xy} = \frac{1}{m}\sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y})$$

$$\theta_1 = \frac{S_{xy}}{S_{xx}}$$

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

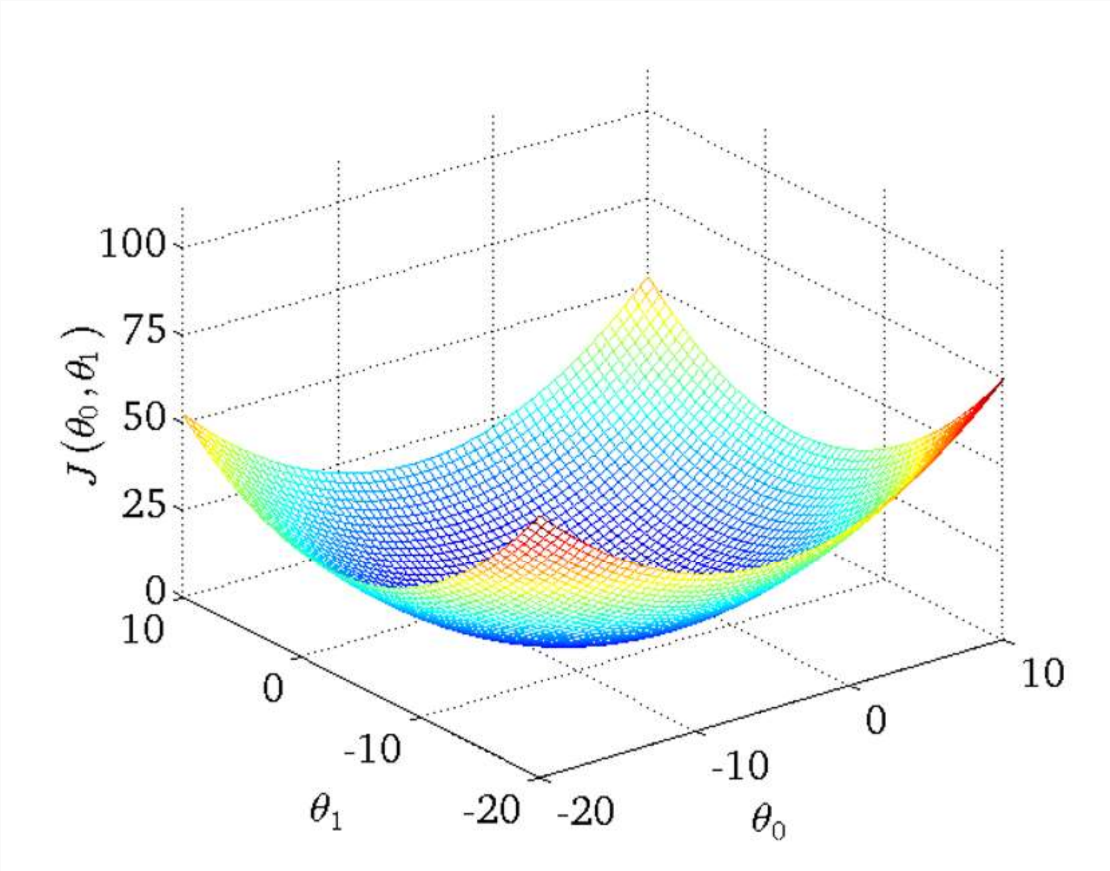
$$\hat{y} = \theta_0 + \theta_1 x$$

Later we'll see how to solve $\boldsymbol{\theta}$ for multivariate data (normal equation).

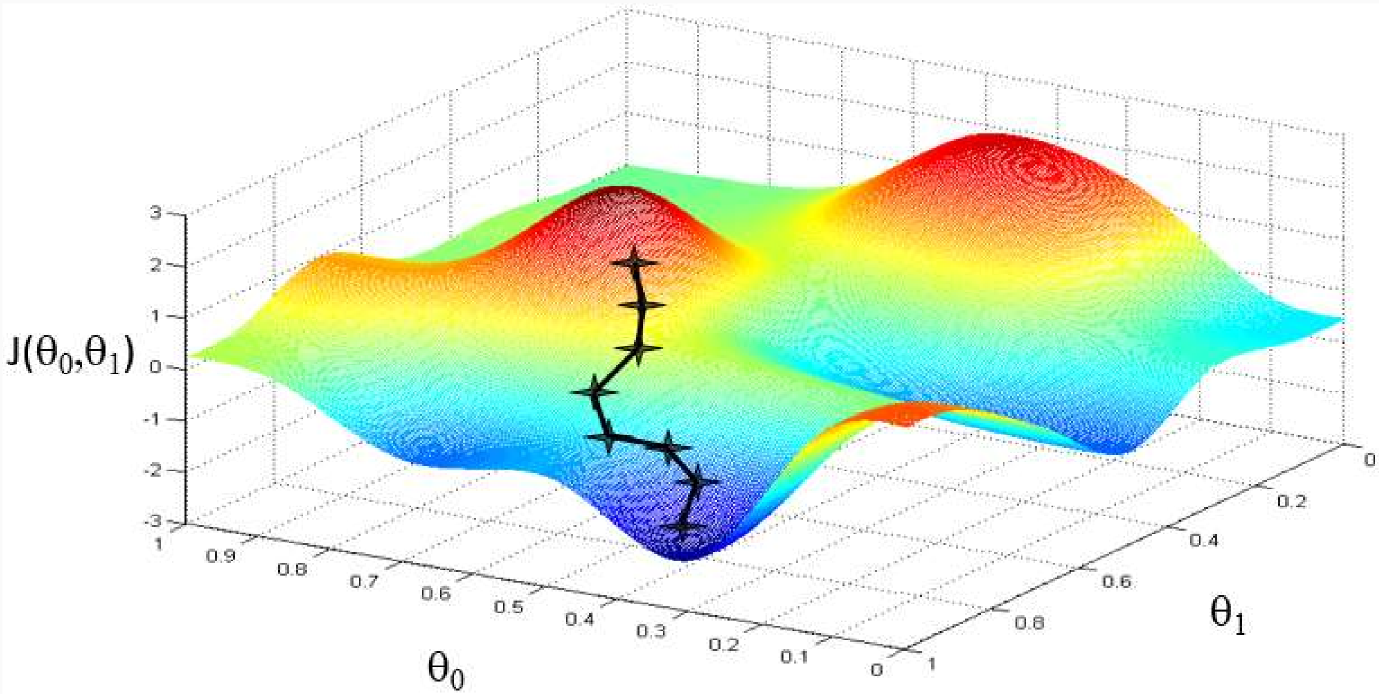
Linear Regression Exercise

Let's code.

- Generic optimization algorithm capable of finding solutions to a wide range of problems.
- Now we don't think about h_θ as a function of x, instead we think of J as a function of θ_0 and θ_1.
- Tweak the parameters iteratively in order to minimize the cost function (find θ_0 and θ_1 at the lower point of J).



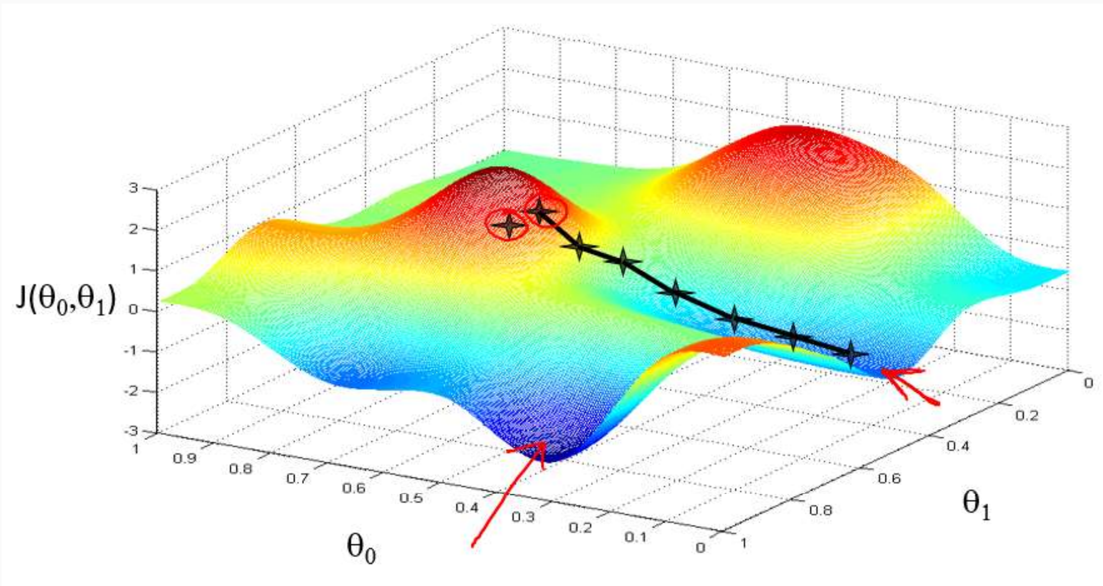
How to get to the bottom of this?



Outline:

- Start with some θ_0 and θ_1
- Keep changing θ_0 and θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum.

- Take the derivative (the tangent) of the cost function.
- The derivative give us the direction of steepest descent to move towards.
- Make steps down the cost function with the step size determined by the derivative and the learning rate α.



Algorithm:

repeat until convergence {

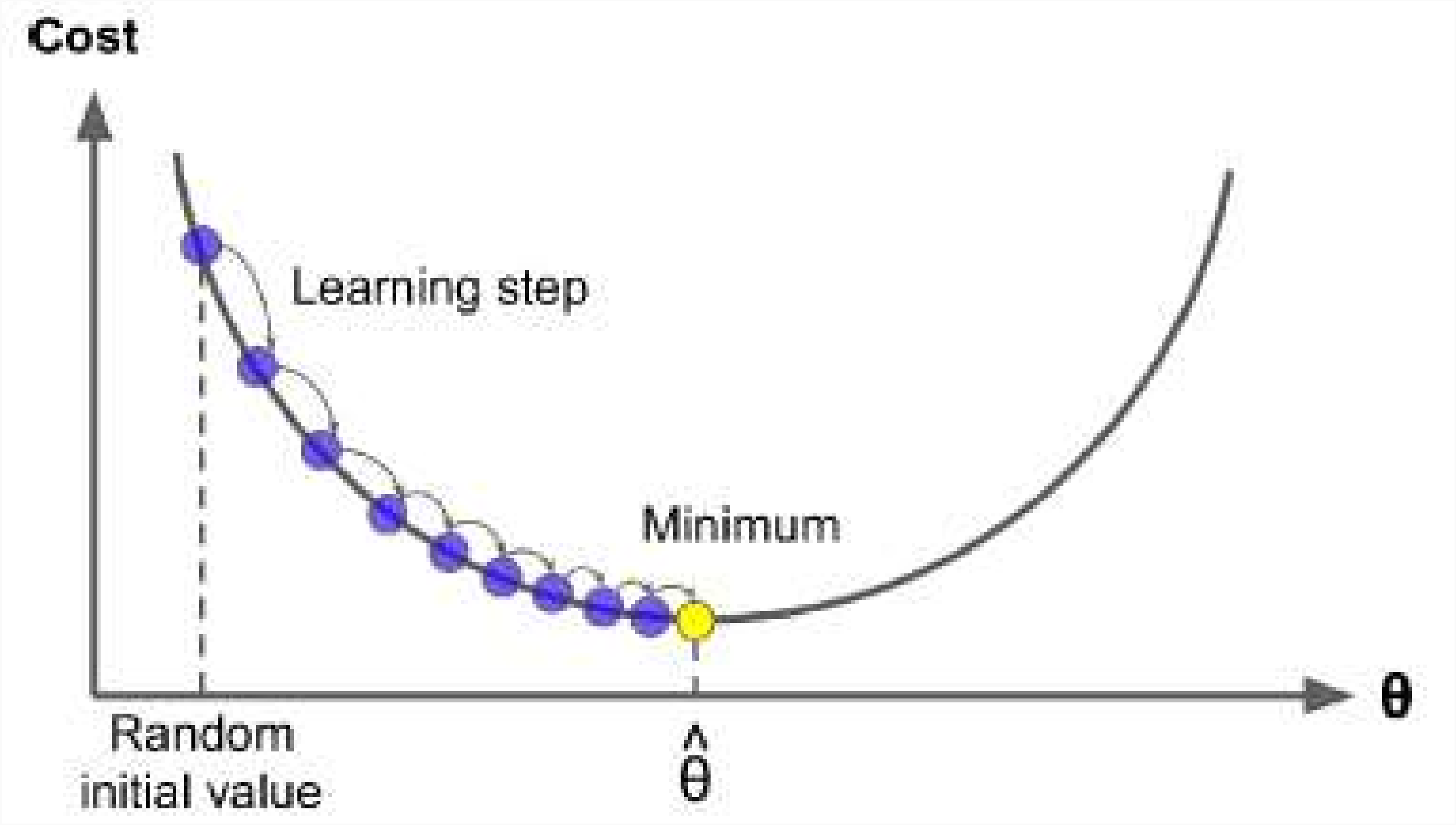
$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

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

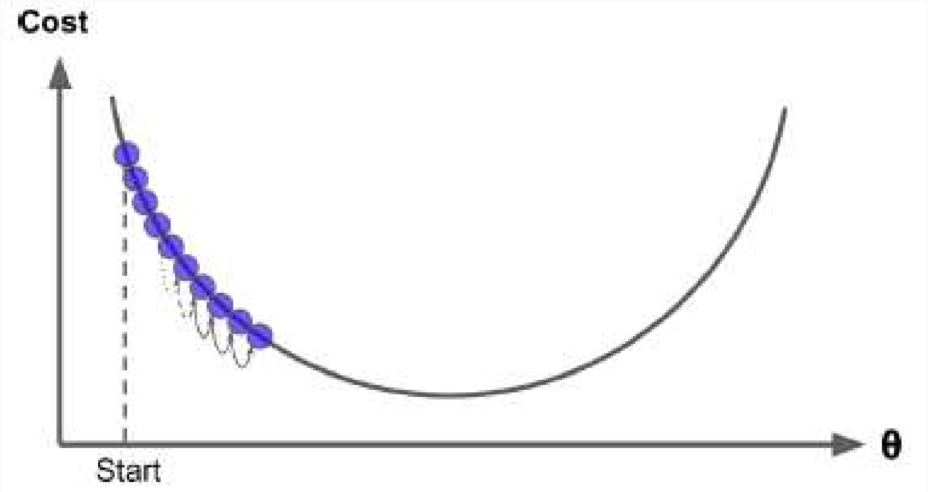
}

- We don't have guarantee to end up in the global minimum.
- Maybe we end up in local minimum depending on where we start.

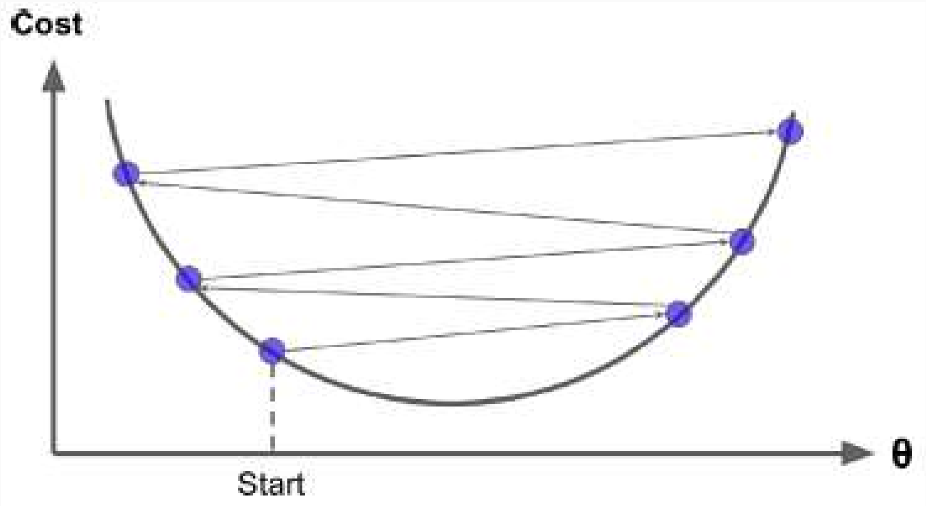
The step size is proportional to the derivative of the cost function, so the steps gradually get smaller as the parameter approach the minimum.



The learning rate hyperparameter α adjust the size of each step.



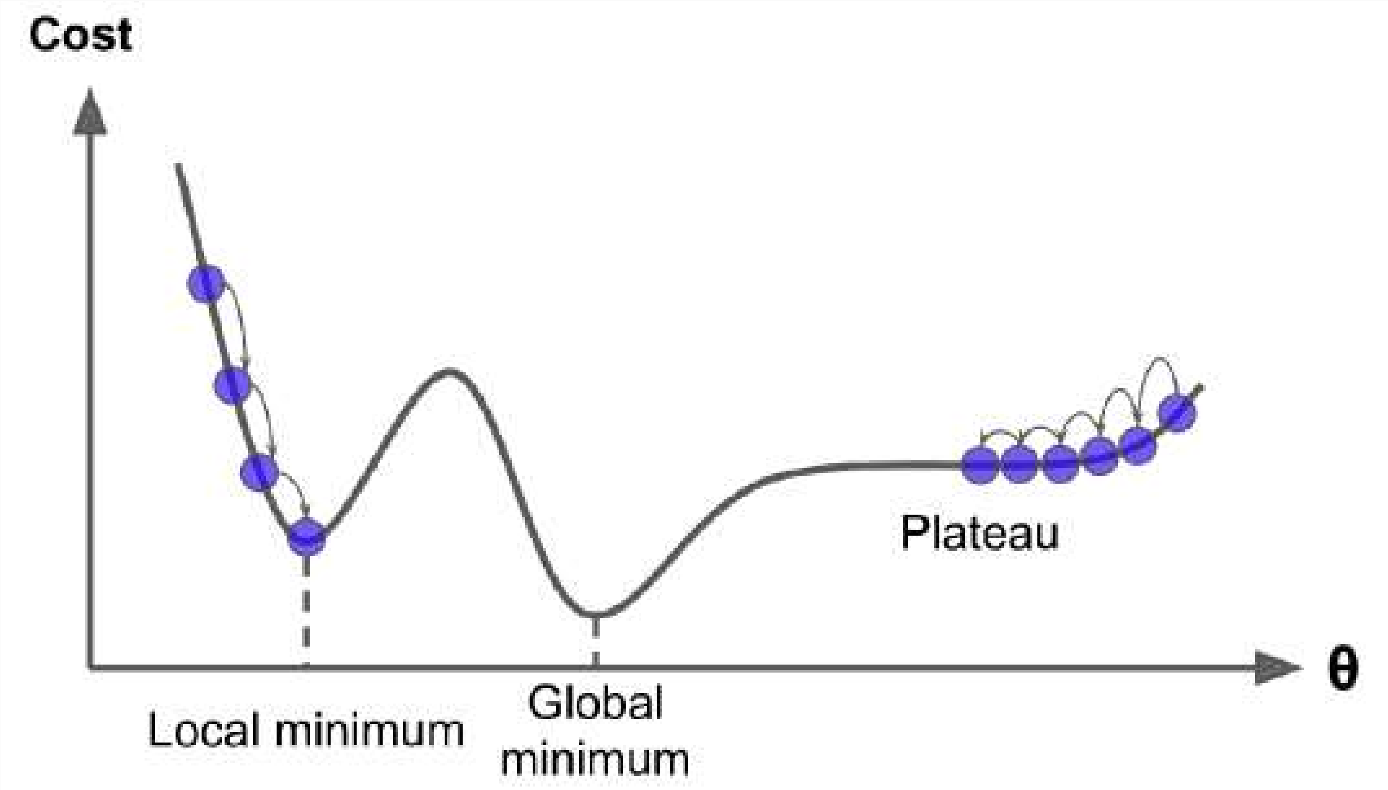
- Gradient descent will be slow.
- Algorithm will go through many iterations to converge.



- Might jump across the valley and end up on the other side.
- May fail to converge or even diverge.

Linear Regression Local and global minimum

- Not all cost functions look nice, regular bowls.
- There may be holes, ridges, plateaus, and irregular terrains.
- Convergence to the minimum may be difficult.



In the case of Linear Regression fortunately the cost function is a convex function, which means there are no local minimum only one global minimum.

Algorithm:

repeat until convergence {

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

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

}

Linear Regression Model:

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

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

But, what is $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ and $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$?

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

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

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

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

When specifically applied to univariate linear regression the algorithm goes like this:

repeat until convergence {

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

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

}

We should update θ_0 and θ_1 "simultaneously", that means that we don't use the new value of one parameter to update the others.

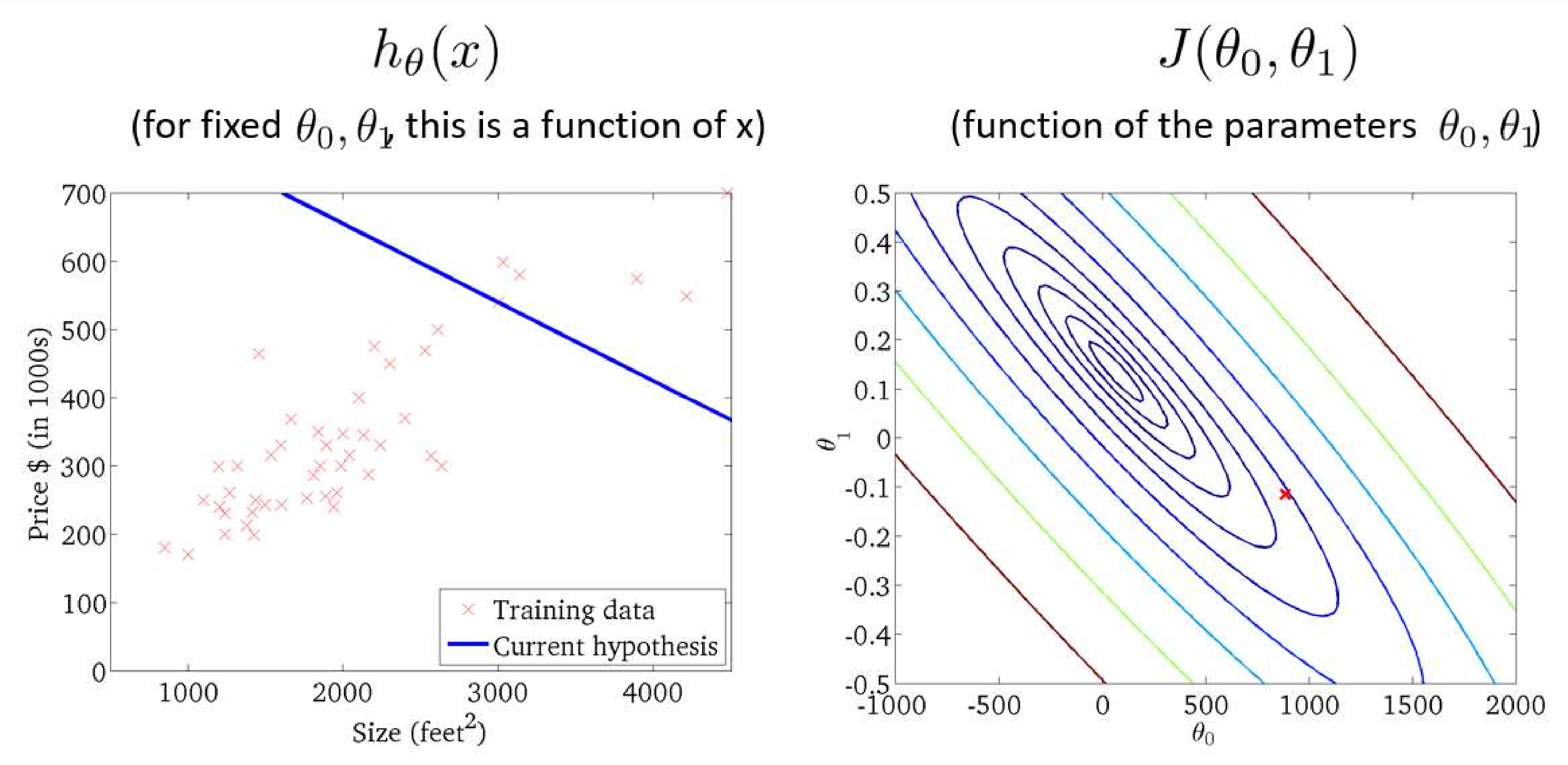
Correct

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

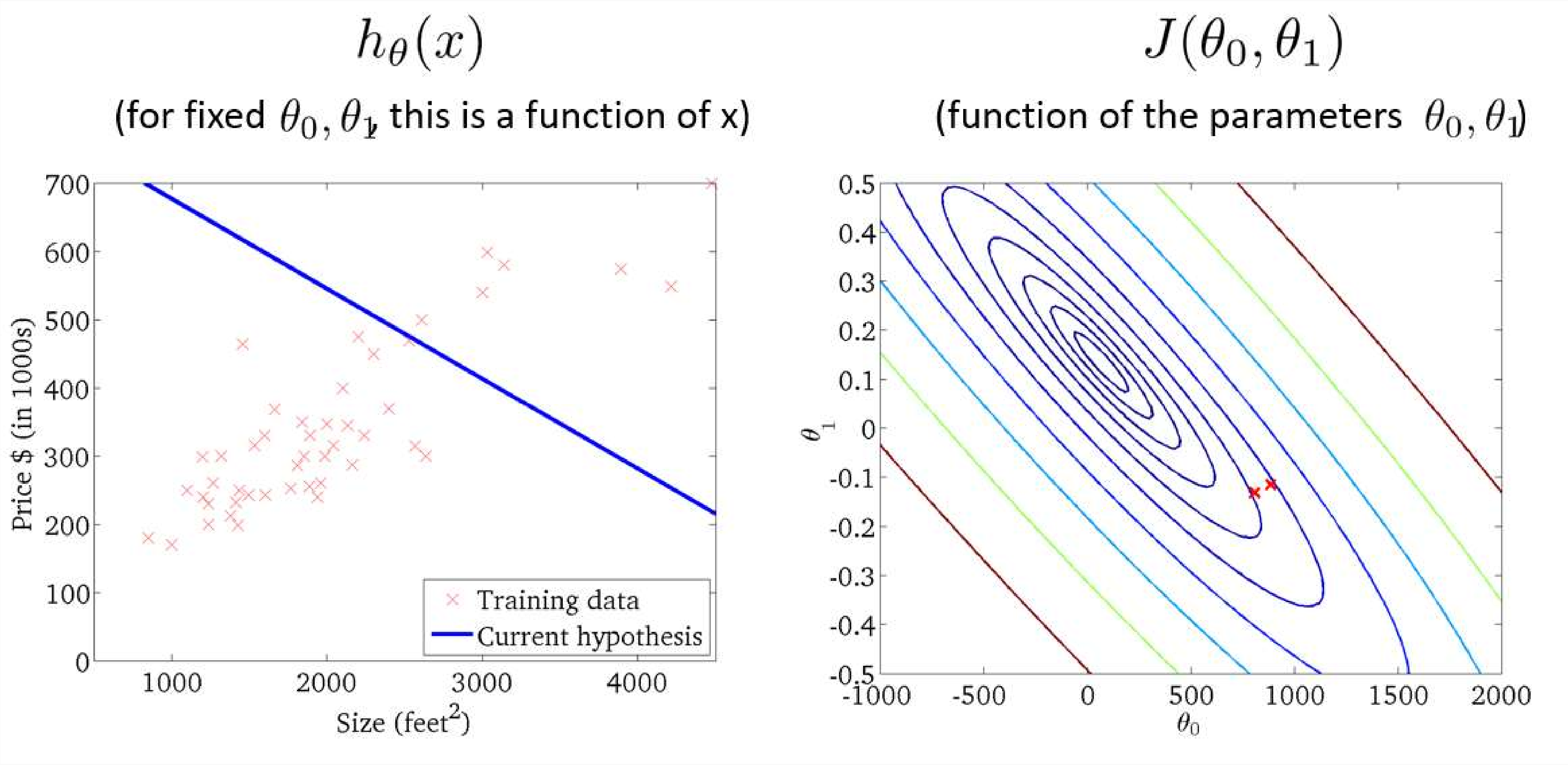
Incorrect

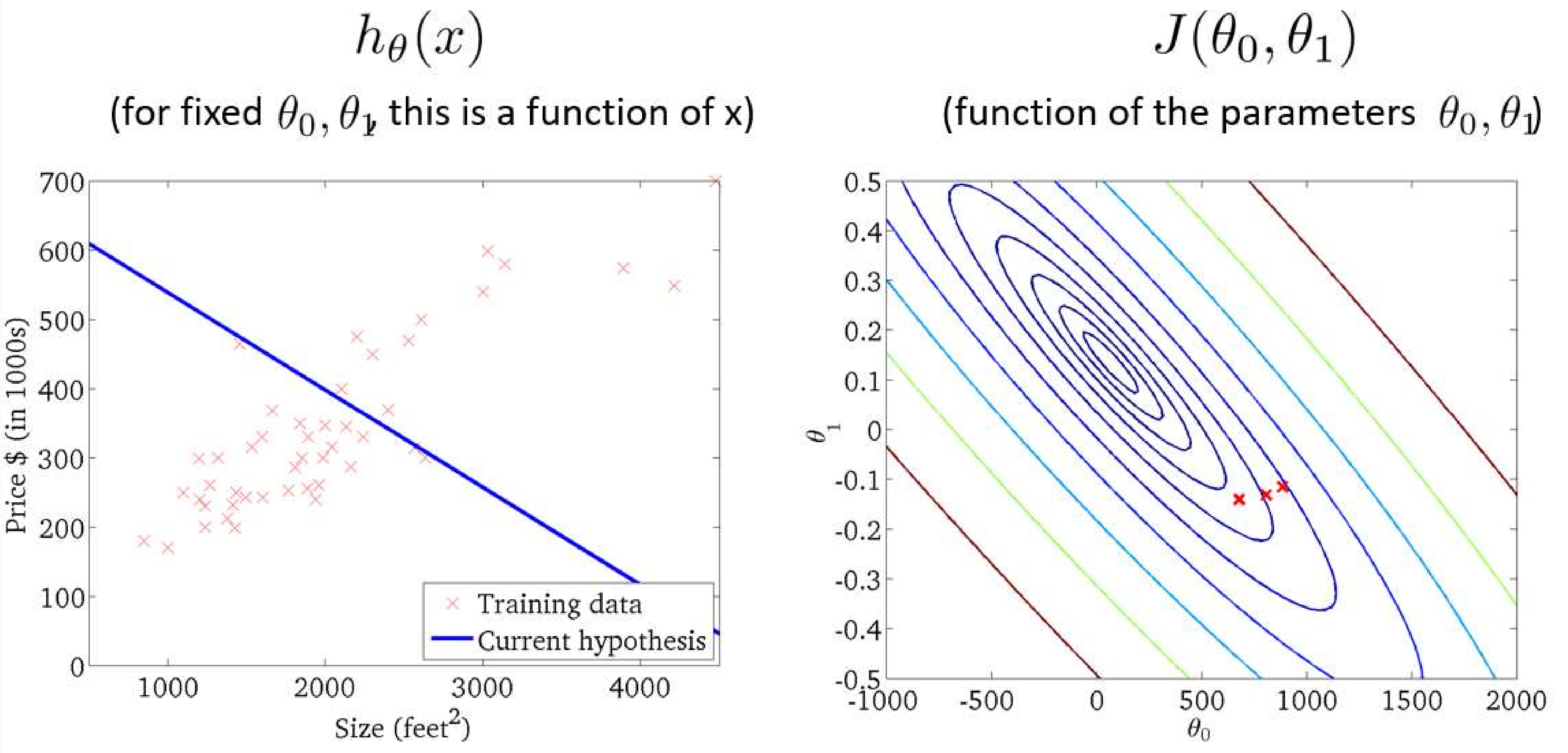
$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$

Linear Regression Gradient Descent in action



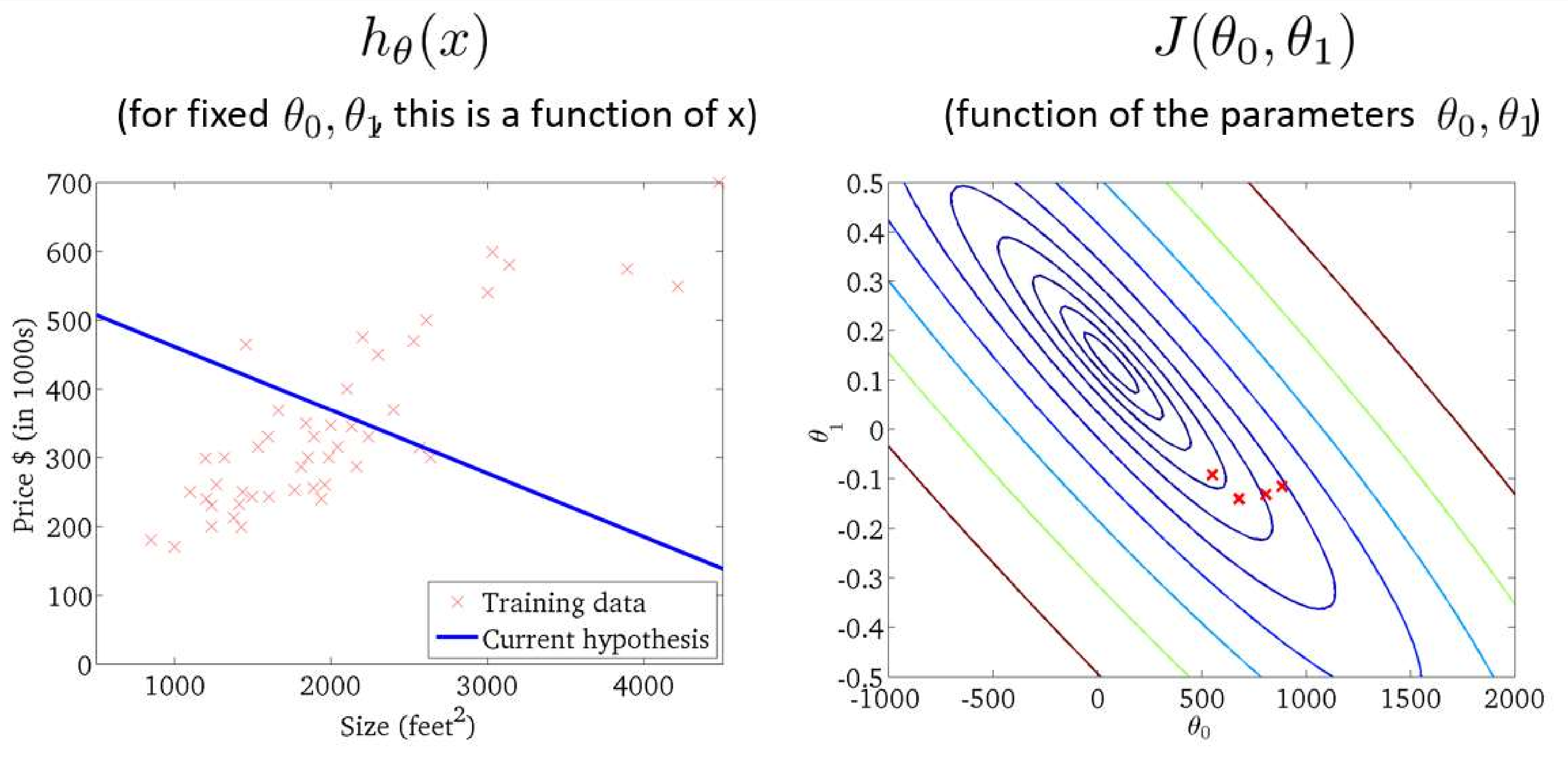
Linear Regression Gradient Descent in action

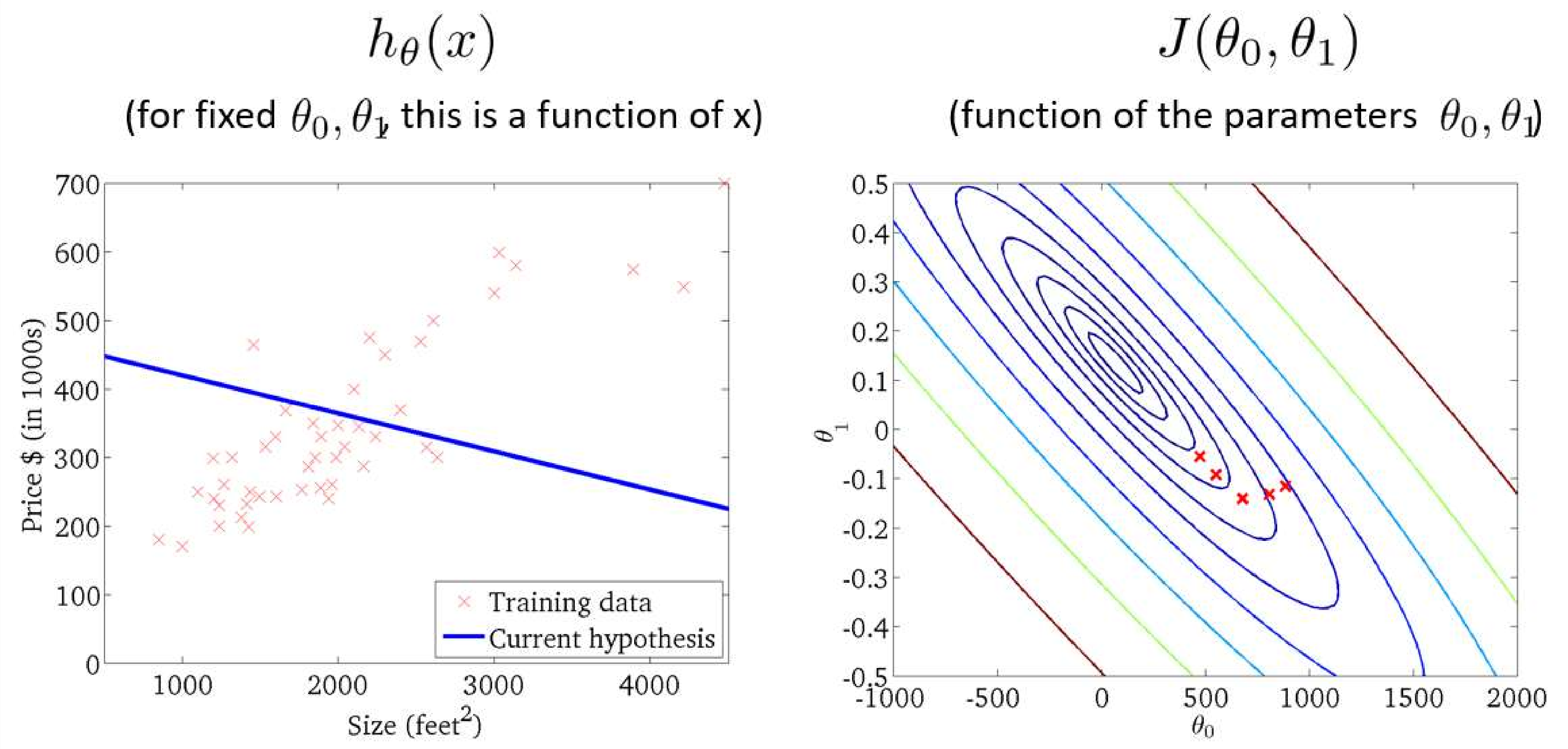




$h_\theta(x)$
(for fixed θ_0, θ_1, this is a function of x)
700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis
$J(\theta_0, \theta_1)$
(function of the parameters θ_0, θ_1)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
θ_1
-1000
-500
0
500
1000
1500
2000
θ_0

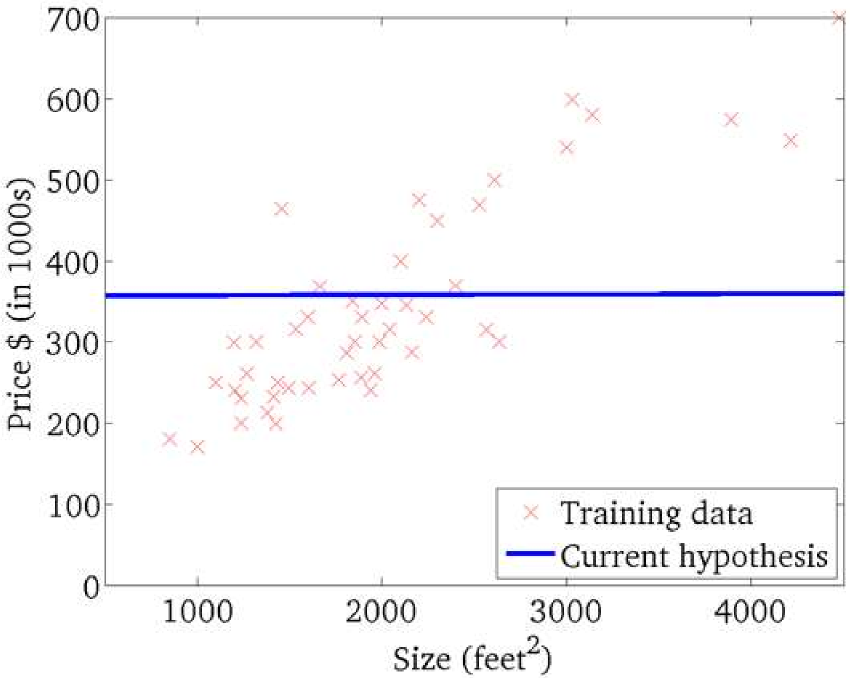
Linear Regression Gradient Descent in action




$h_\theta(x)$
(for fixed θ_0, θ_1, this is a function of x)
700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis
$J(\theta_0, \theta_1)$
(function of the parameters θ_0, θ_1)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
θ_1
-1000
-500
0
500
1000
1500
2000
θ_0

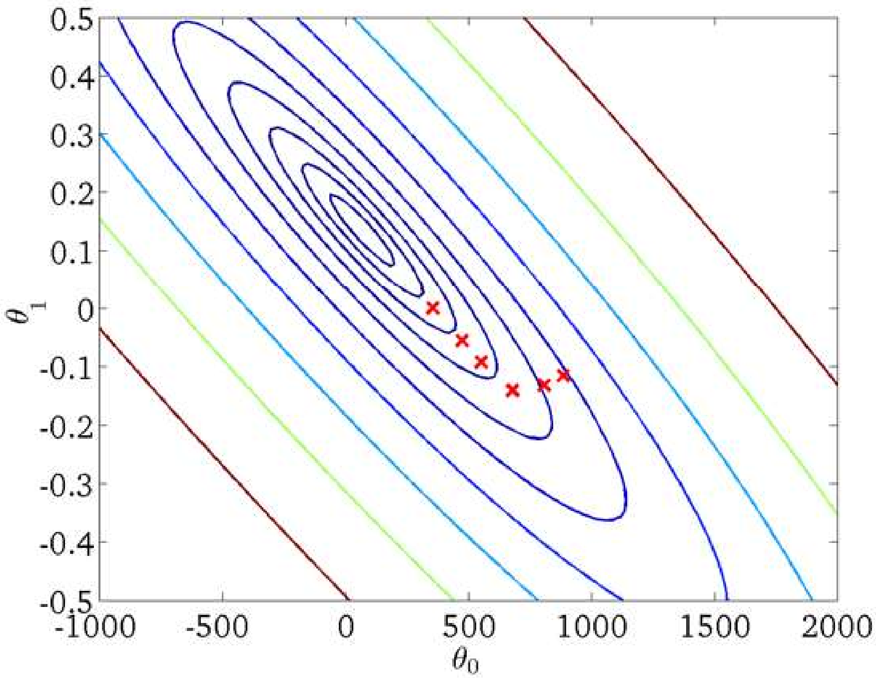
$h_\theta(x)$

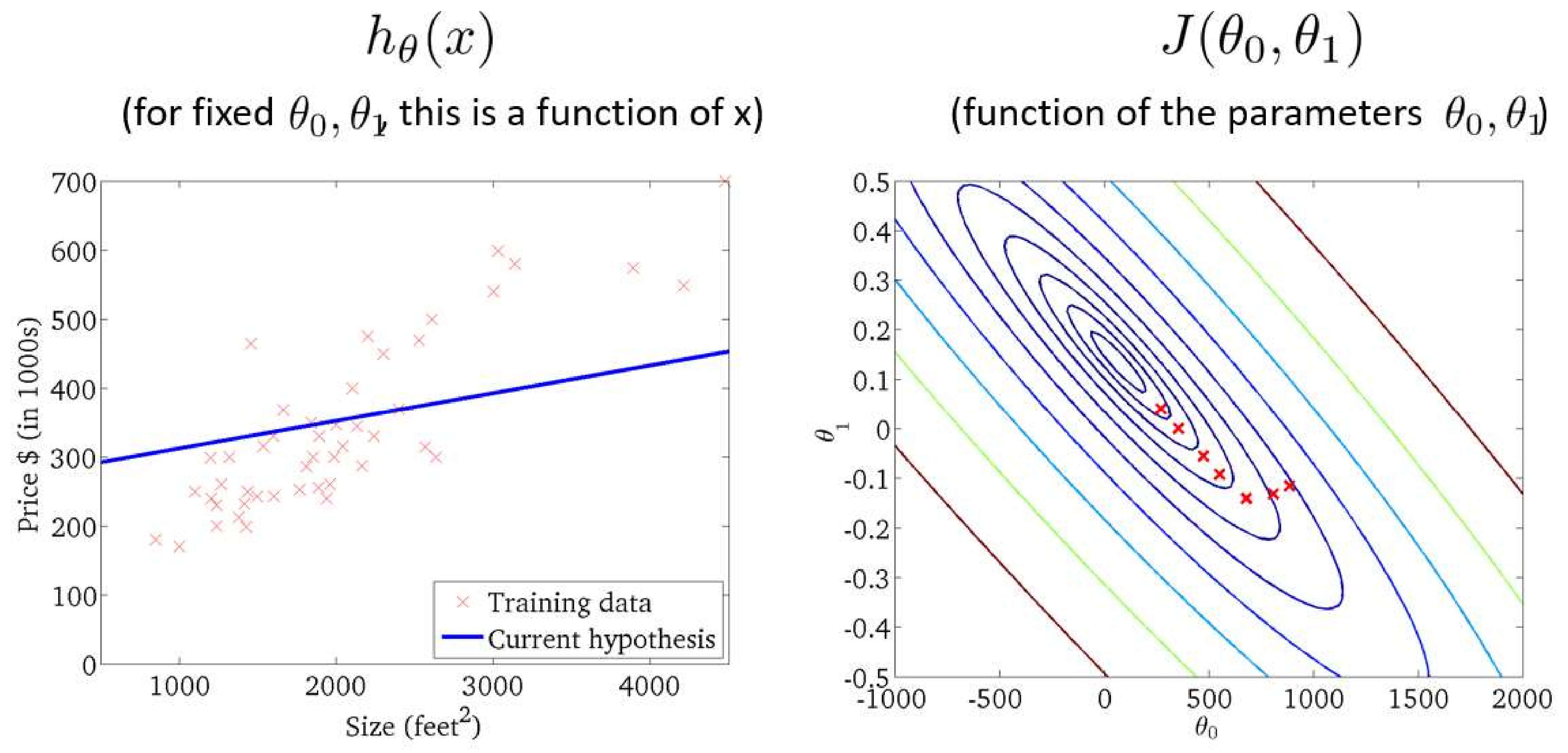
(for fixed θ_0, θ_1, this is a function of x)

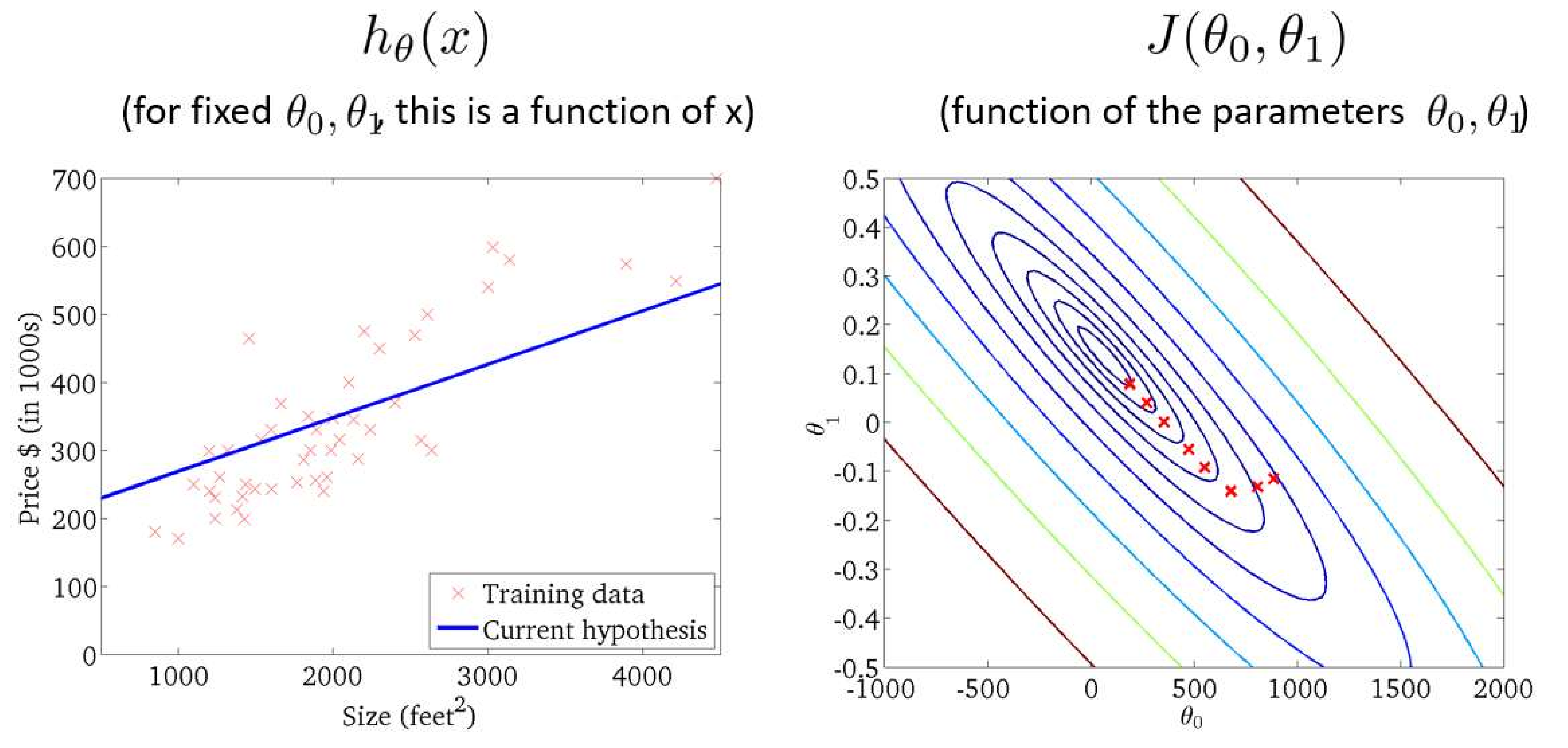


$J(\theta_0, \theta_1)$

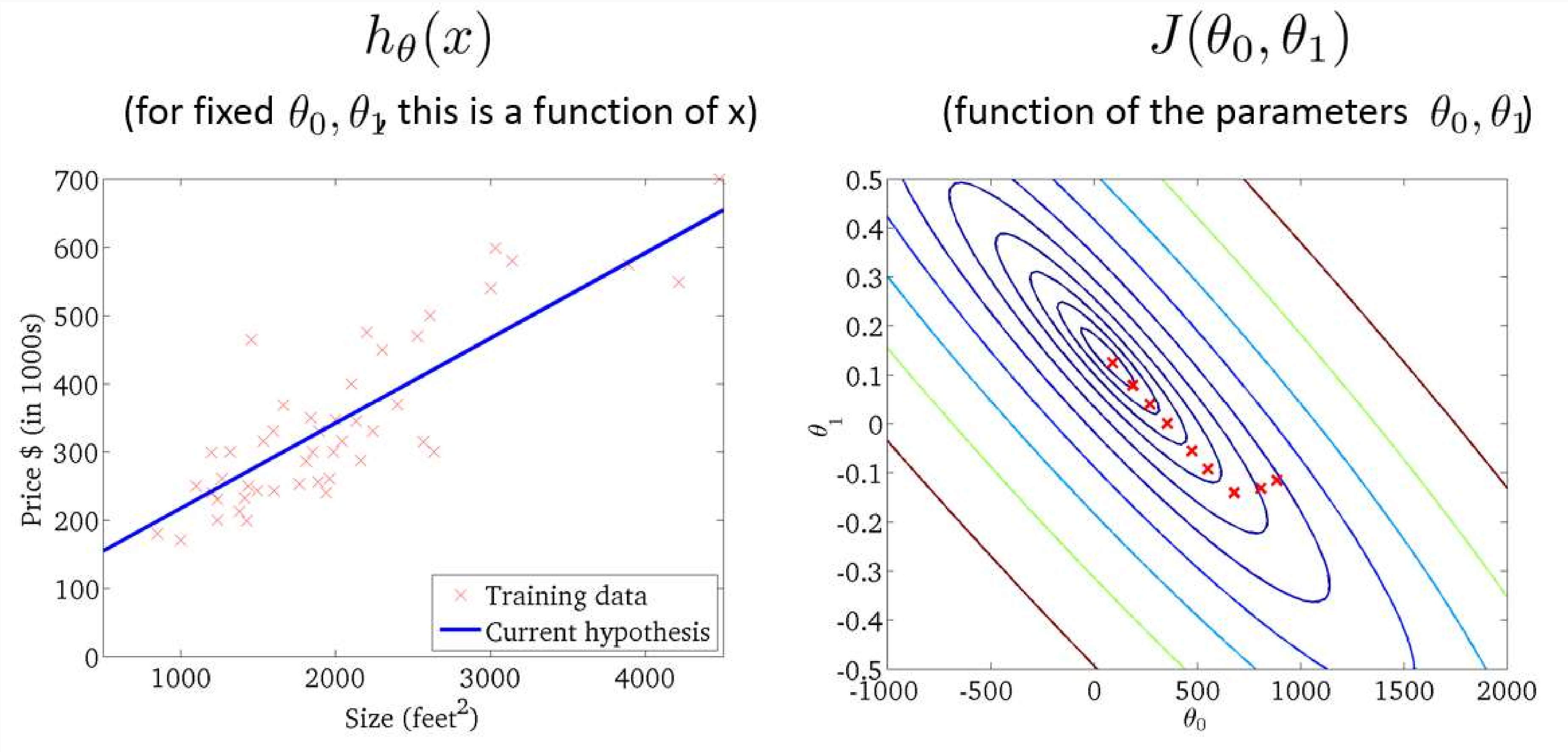
(function of the parameters θ_0, θ_1)




$h_\theta(x)$
(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)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
θ_1
θ_0



$h_\theta(x)$
(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)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
700
600
500
400
300
200
100
0
1000
2000
3000
4000
θ_1
θ_0
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000


$h_\theta(x)$
(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)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
0
100
200
300
400
500
600
700
1000
2000
3000
4000
θ_1
θ_0
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000

- Batch (goes over all training data before updating parameters)
- Mini-batch (goes over subsets of training data and update parameters)
- Stochastic (update parameters after every training example)