Practical Machine Learning

Module 1: Modelling and Linear Regression

intelligent layer

asi

Content and objectives

Content

- Review of modelling
- Linear Regression
- Gradient Descent

Objectives

- Understand where the model fits into the machine learning workflow
- Understand why we need a good model
- Understand types of modelling

Content and objectives

Content

- Review of modelling
- Linear Regression
- Gradient Descent

Objectives

- Understand when we use regression
- Introduce a simple 1D linear model
- Understand how to evaluate the cost of predictions against observations

Content and objectives

Content

- Review of modelling
- Linear Regression
- Gradient Descent

Objectives

- Understand why we use gradient descent
- Understand the operation of gradient descent
- Understand how gradient descent can go wrong, and why

Modelling

We want to find a RELATIONSHIP between INPUTS and TARGET

$$y = f(\mathbf{x}) + \epsilon$$

target function inputs noise

Modelling

Why do we need a GOOD MODEL?

A PREDICTION

$$y^* = f_{est}(\mathbf{x}^*)$$

B INFERENCE

- Which input features are most strongly related to y?
- Are relationships +ve or –ve?
- What is the nature of f? (Linear / non-linear etc.)

Modelling

PARAMETRIC and NON-PARAMETRIC modelling

A PARAMETRIC

$$f(\mathbf{x}) = \theta_0 + x_1\theta_1 + x_2\theta_2 \cdots$$

B NON-PARAMETRIC

- f is not fixed
- complexity of f adapts to the complexity of the data

Modelling

SUPERVISED vs. UNSUPERVISED learning

A SUPERVISED

- Access to target variable
- e.g. linear regression or classification

B UNSUPERVISED

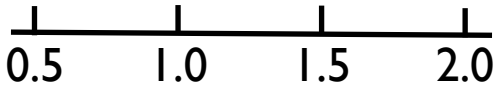
- No access to target variable
- e.g. clustering or dimensionality reduction

Linear Regression with one variable

Regression

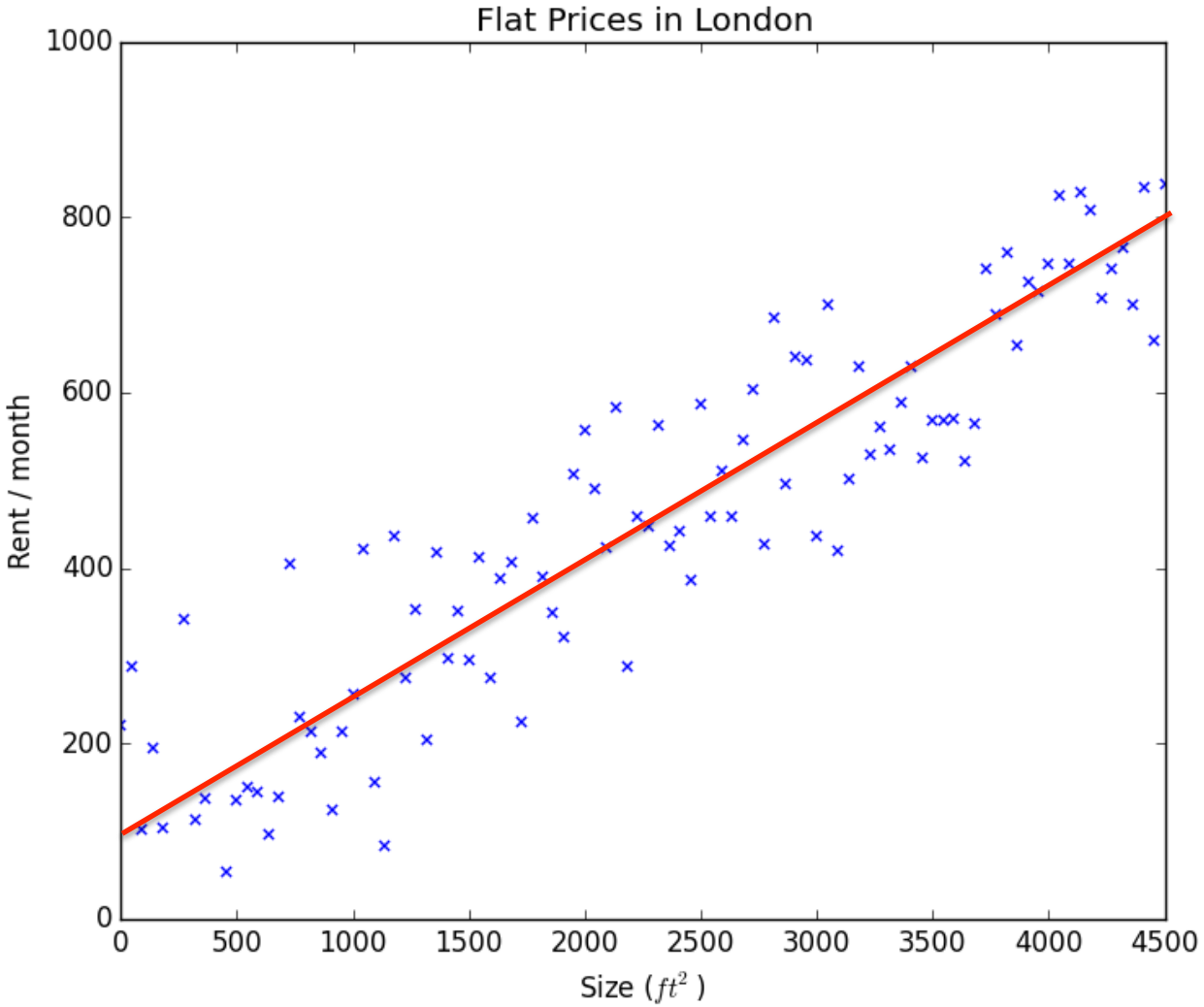
TARGET variable takes on NUMERICAL VALUES

$$f(\mathbf{x}) \longrightarrow$$

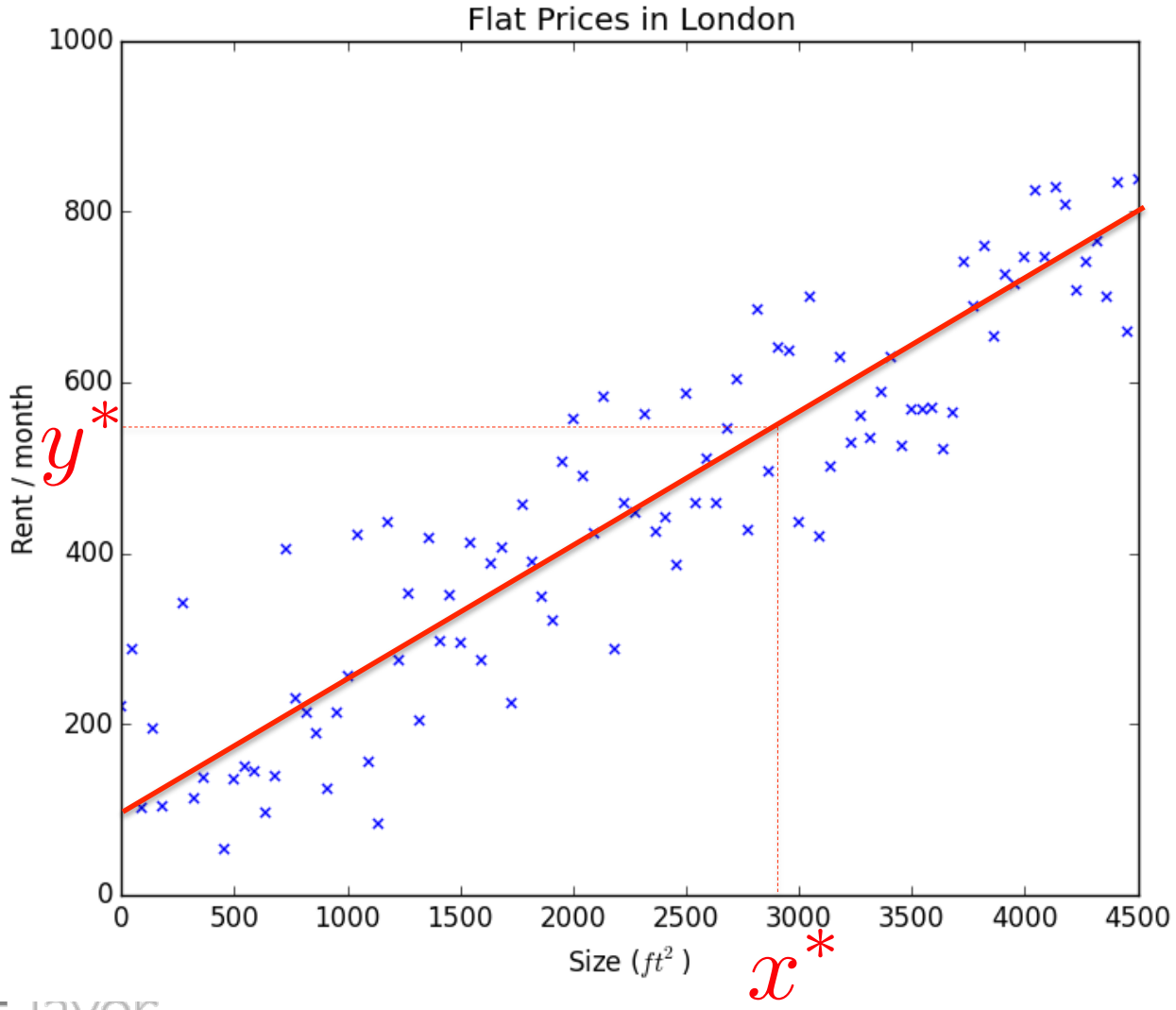


	mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	origin
0	18	8	307	130	3504	12.0	70	1
1	15	8	350	165	3693	11.5	70	1
2	18	8	318	150	3436	11.0	70	1
3	16	8	304	150	3433	12.0	70	1
4	17	8	302	140	3449	10.5	70	1

Linear Regression with 1 variable



Linear Regression with 1 variable



Linear Regression with one variable

Size in ft^2 **(x)**	Rent / month (£) **(y)**
3121	450
1746	415
2443	398
…	…

Notation

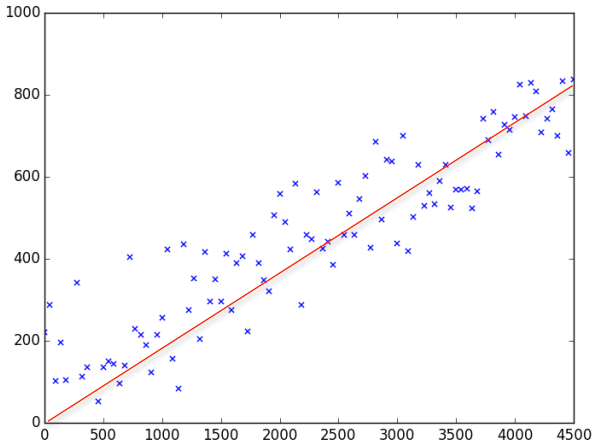
m = No. of training examples
x = input variable (feature)
y = output variable (target)

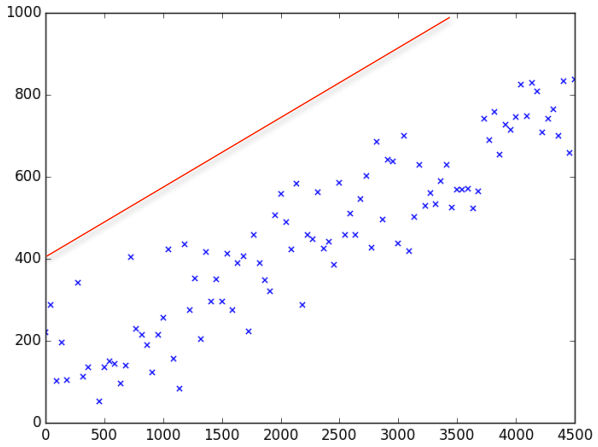
(x,y) = one training example
$(x^{(i)},y^{(i)})$ = i^{th} example

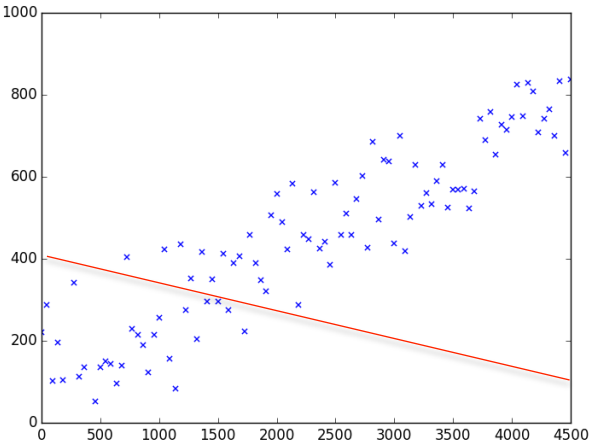
Linear Regression with 1 variable

The Model

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



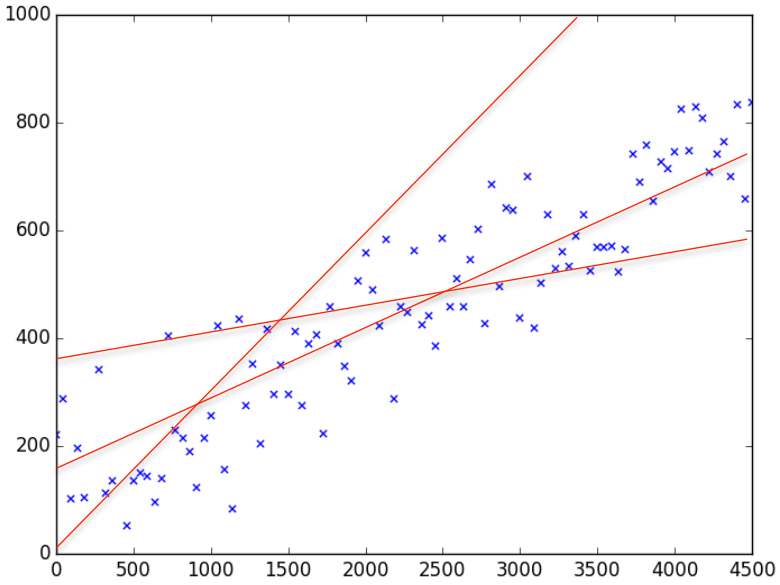




How to choose θ ?

Choose θ_1, θ_0 so that f(x) is close to y for all our (x,y)

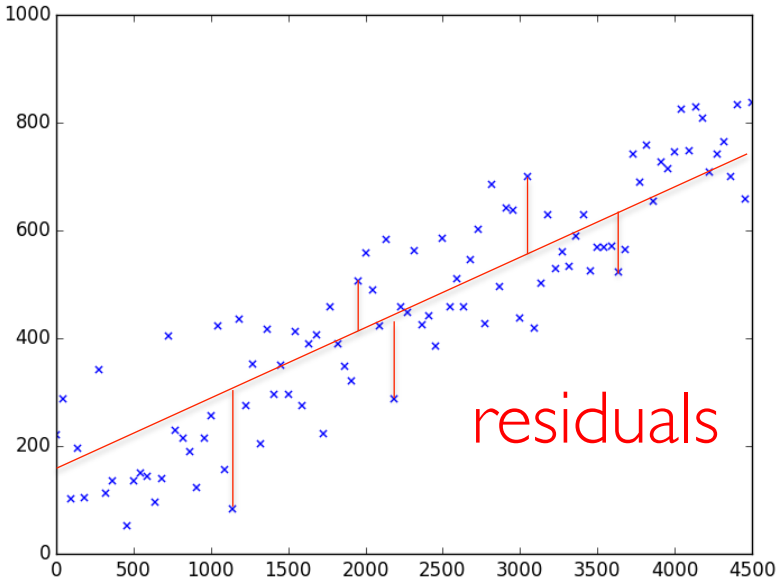
Size in ft^2 **(x)**	Rent / month (£) **(y)**
3121	450
1746	415
2443	398
…	…



How to choose θ ?

Introduce a COST FUNCTION

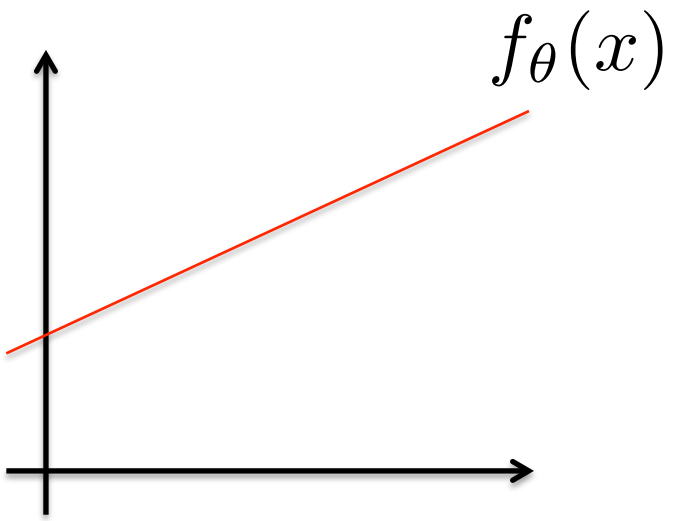
$$C = \frac{1}{2m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i)^2$$



Cost Function

Function:

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



Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$C(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i)^2$$

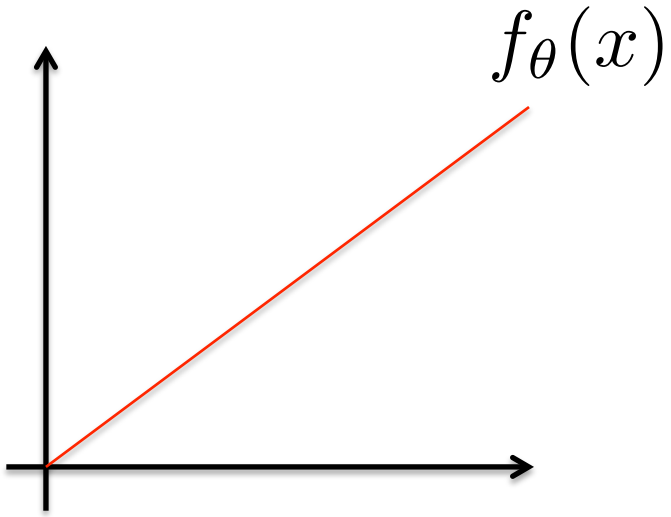
Goal:

$$\underset{\theta_0, \theta_1}{\text{minimize}}\ C(\theta_0, \theta_1)$$

Simplified Cost Function

Function:

$$f_\theta(x) = \theta_1 x$$



Parameters:

$$\theta_1$$

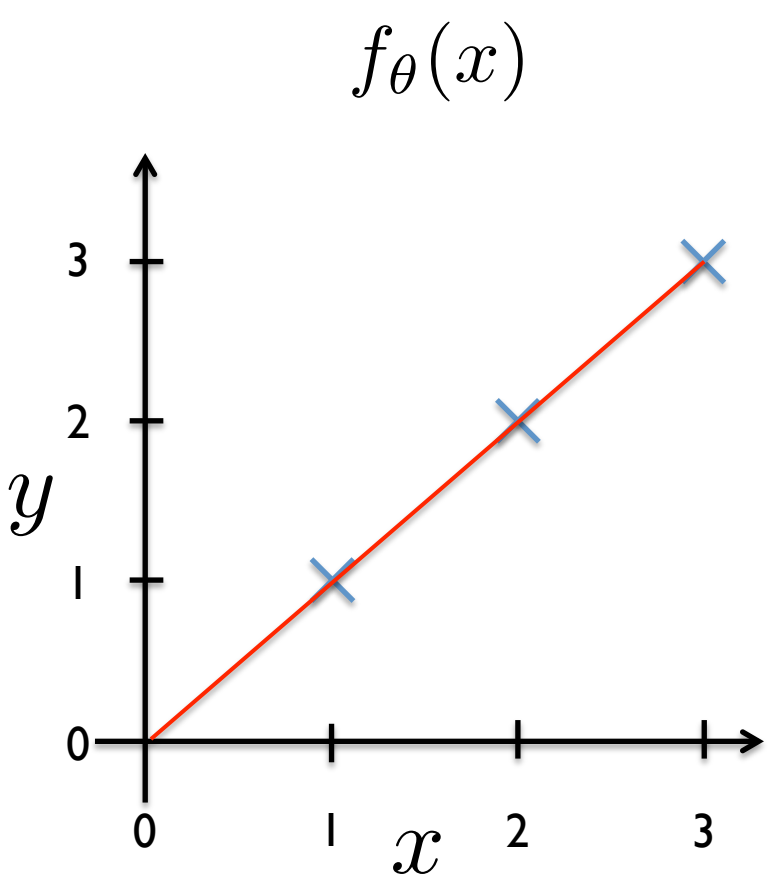
Cost Function:

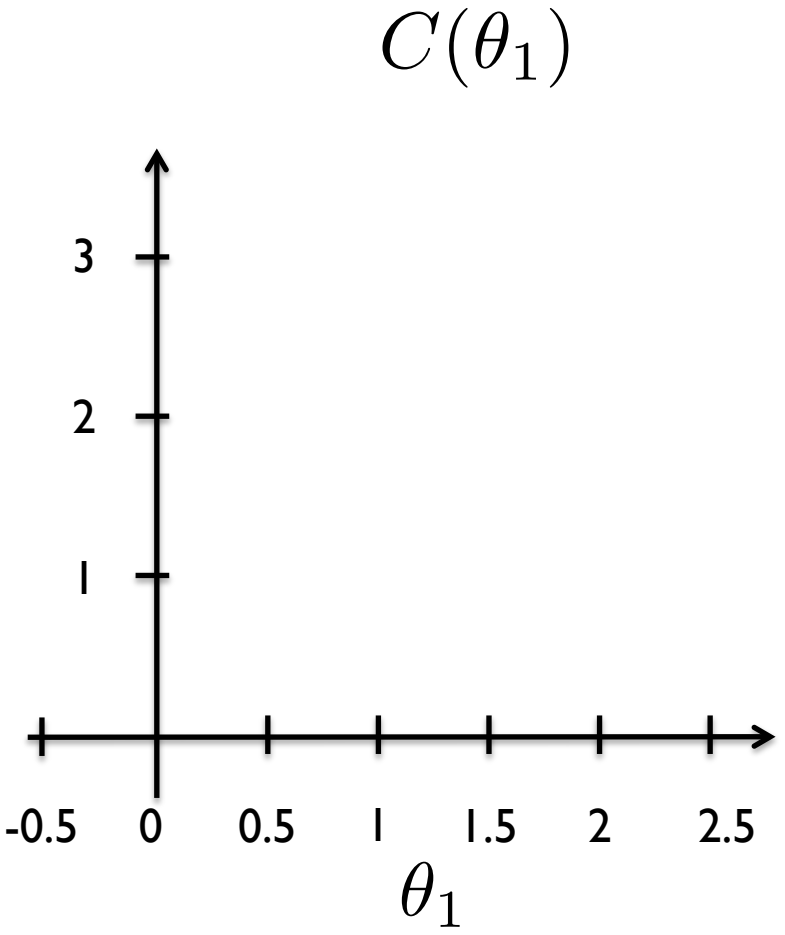
$$C(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i)^2$$

Goal:

$$\underset{\theta_1}{\text{minimize}}\ C(\theta_1)$$

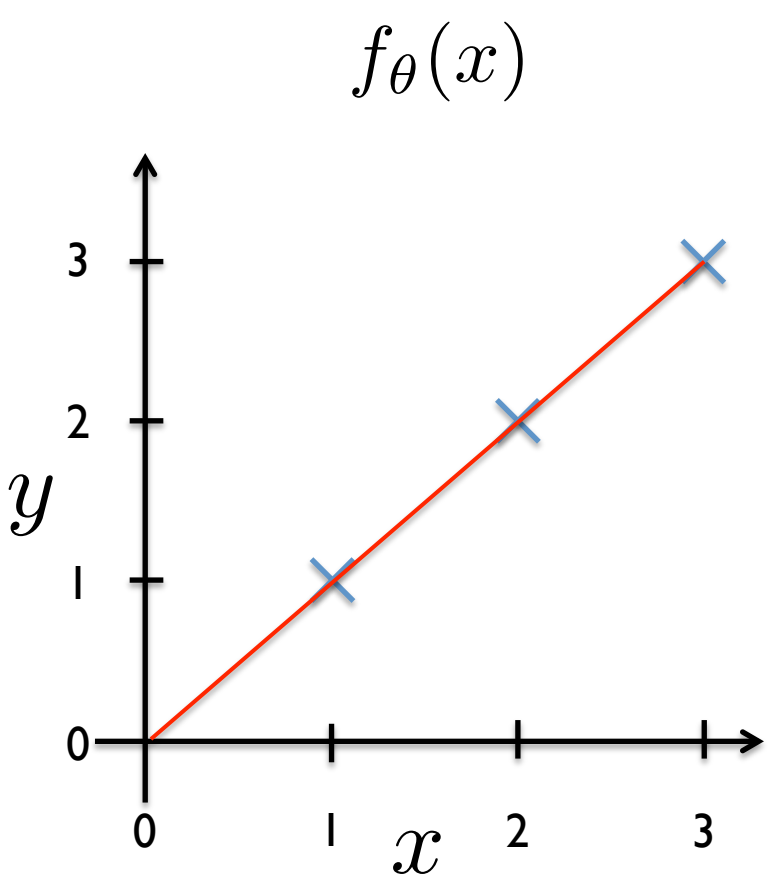
Simplified Cost Function

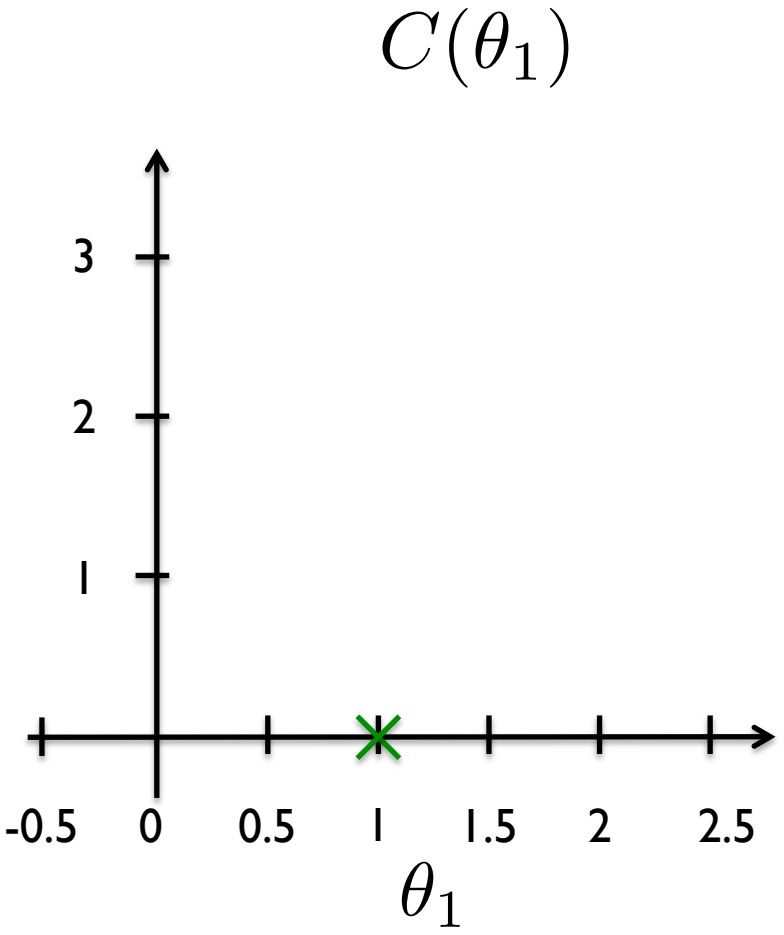




$$C(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i)^2$$

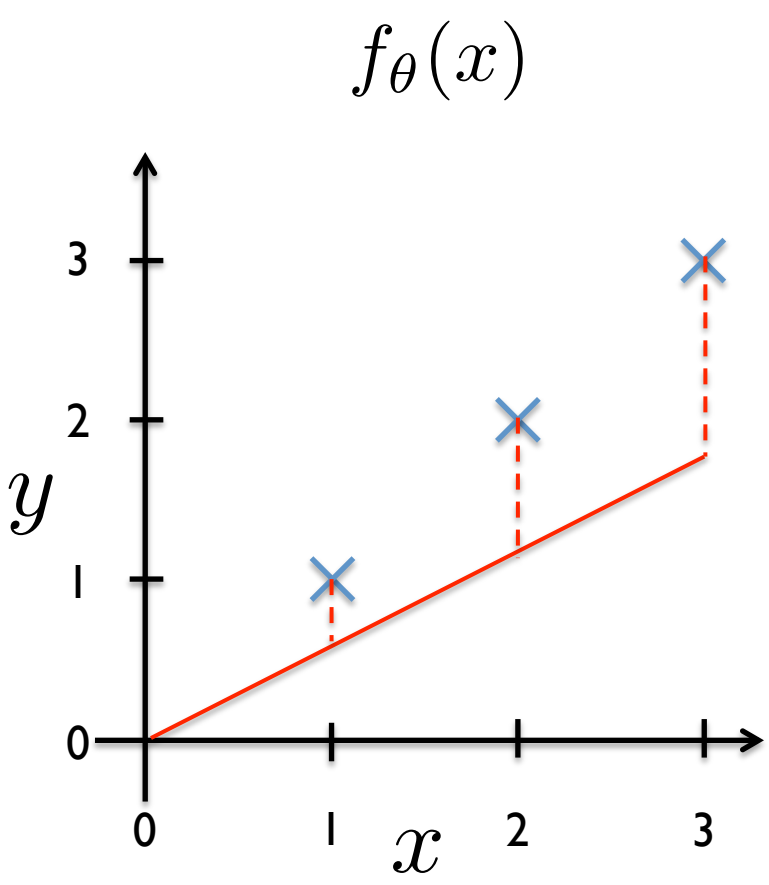
Simplified Cost Function

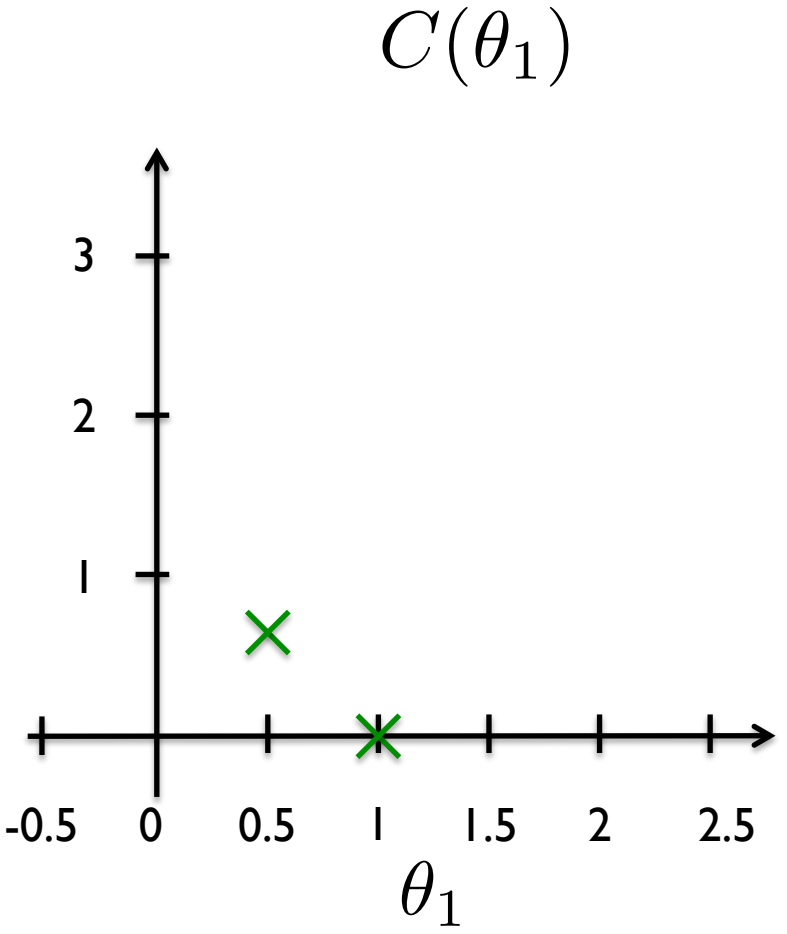




$$C(\theta_1 = 1) = \frac{1}{2m}(0^2 + 0^2 + 0^2)$$

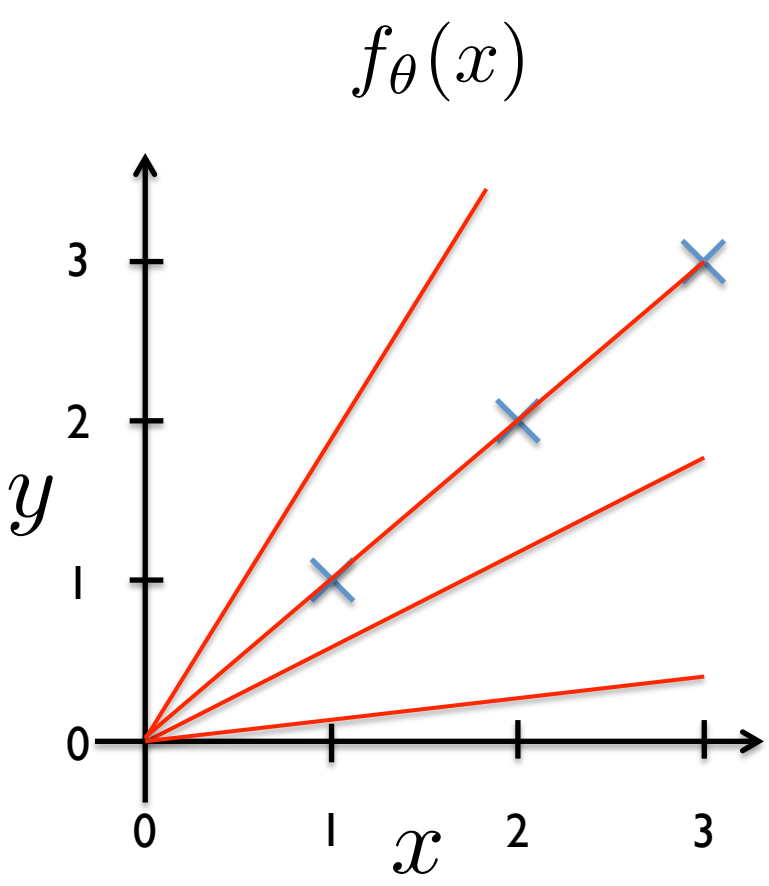
Simplified Cost Function

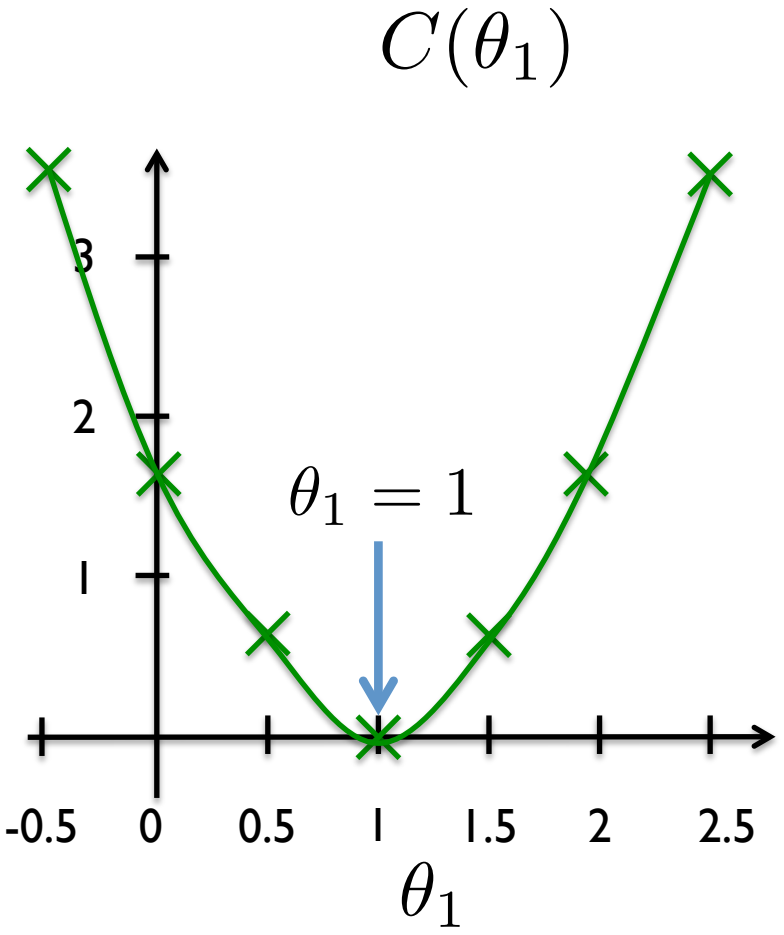




$$C(\theta_1 = 0.5) = \frac{1}{2m}(0.5^2 + 1^2 + 1.5^2) \approx 0.58$$

Simplified Cost Function

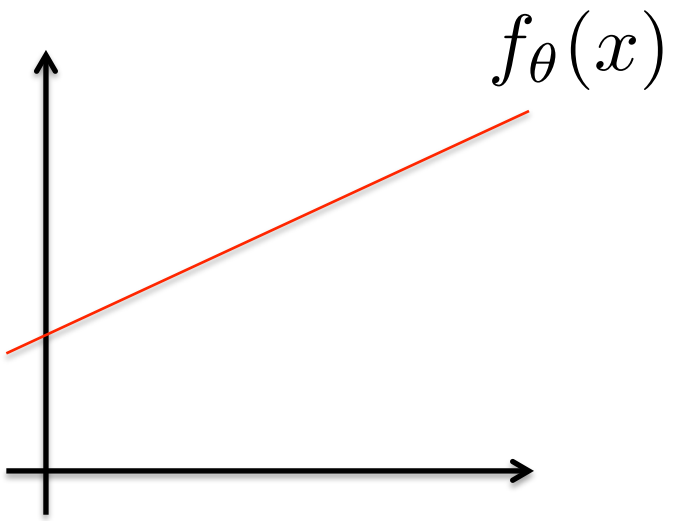




Cost Function

Function:

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



Parameters:

$$\theta_0, \theta_1$$

Cost Function:

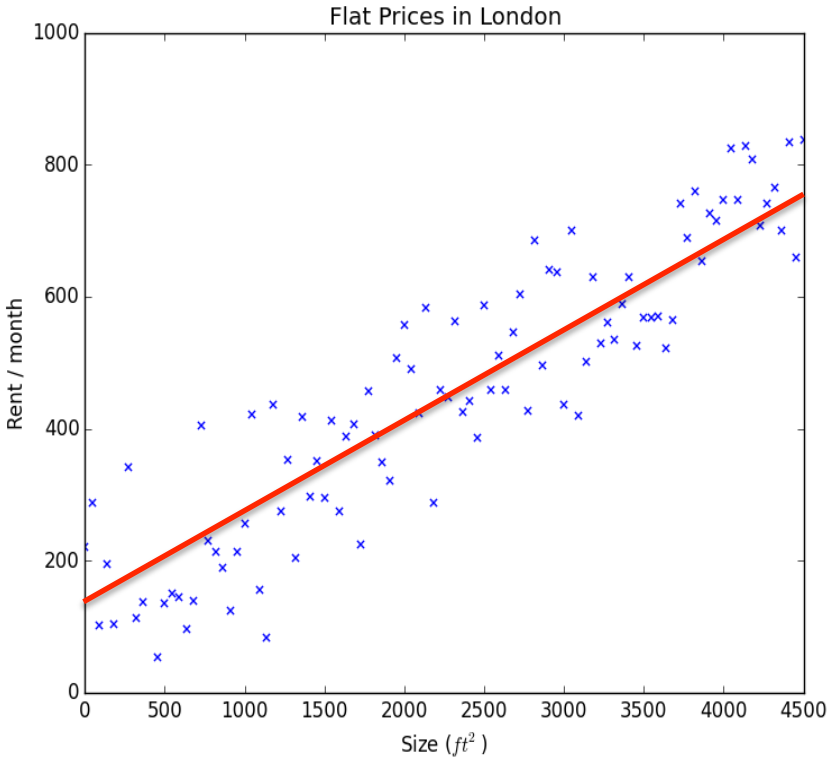
$$C(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i)^2$$

Goal:

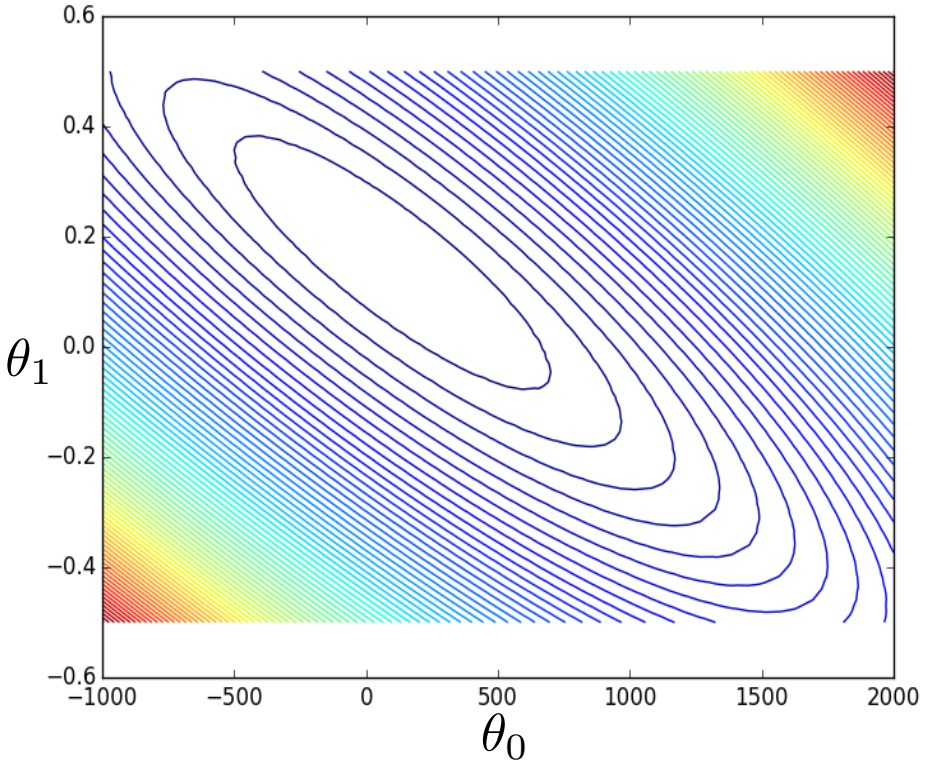
$$\underset{\theta_0, \theta_1}{\text{minimize}}\ C(\theta_0, \theta_1)$$

Cost Function

$f_\theta(x)$

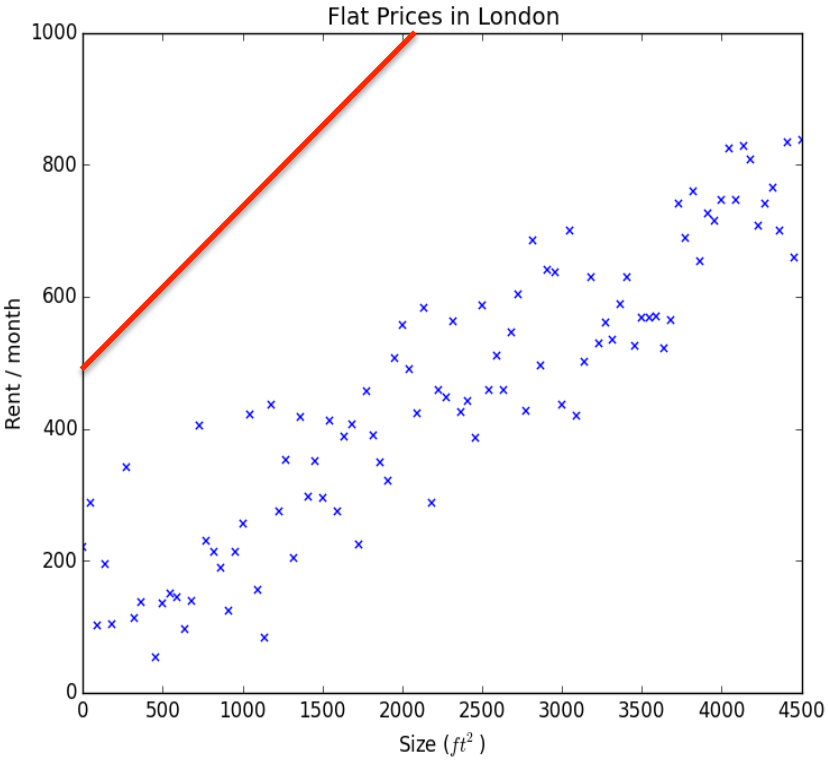


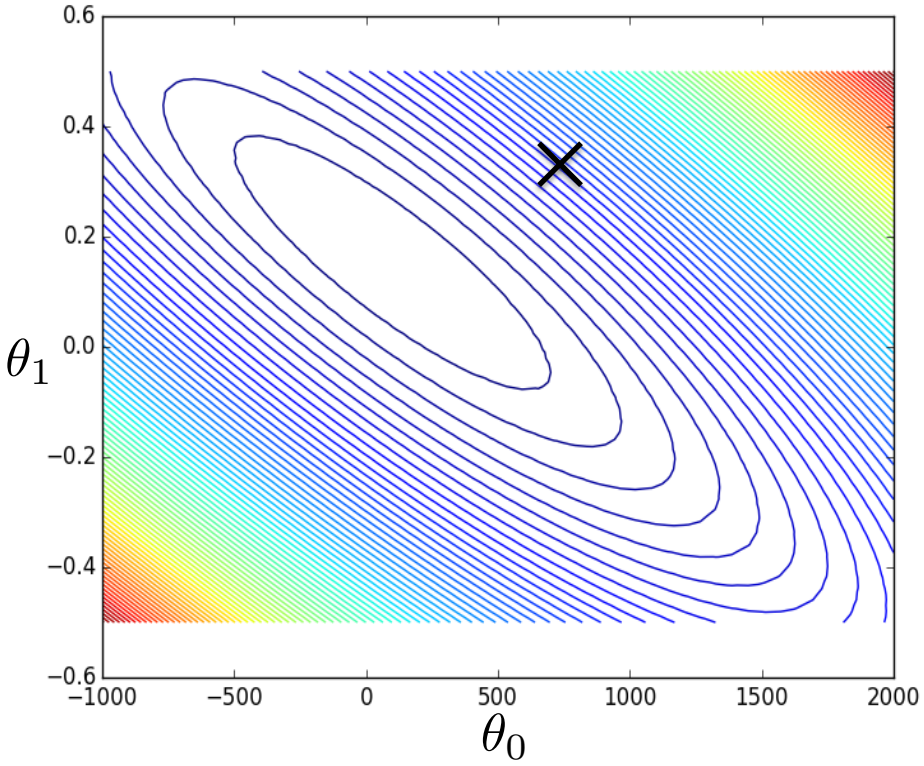
$C(\theta_0, \theta_1)$



Cost Function

Points on cost surface correspond to different functions



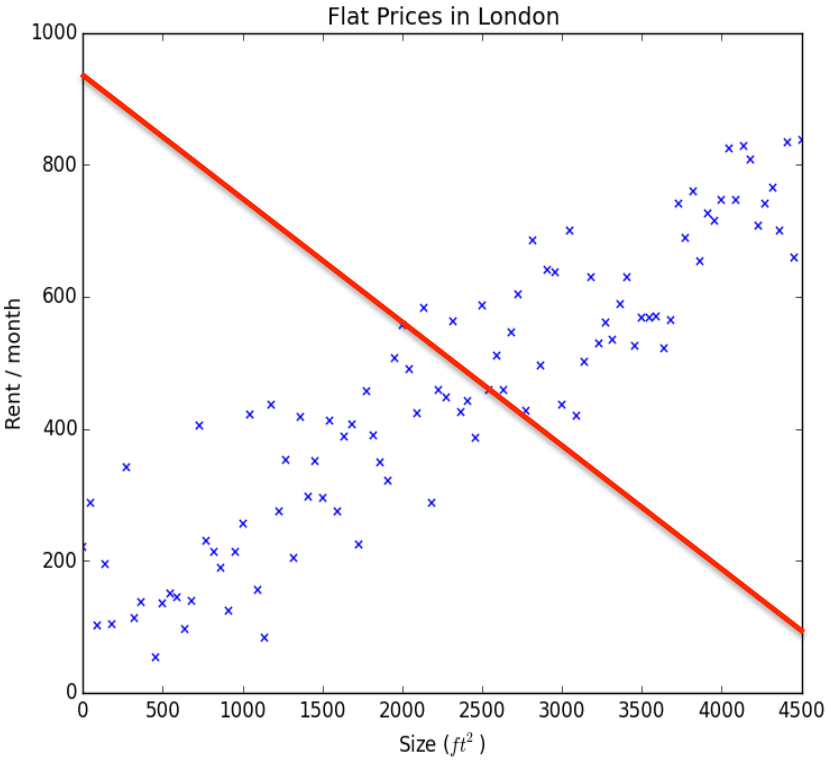


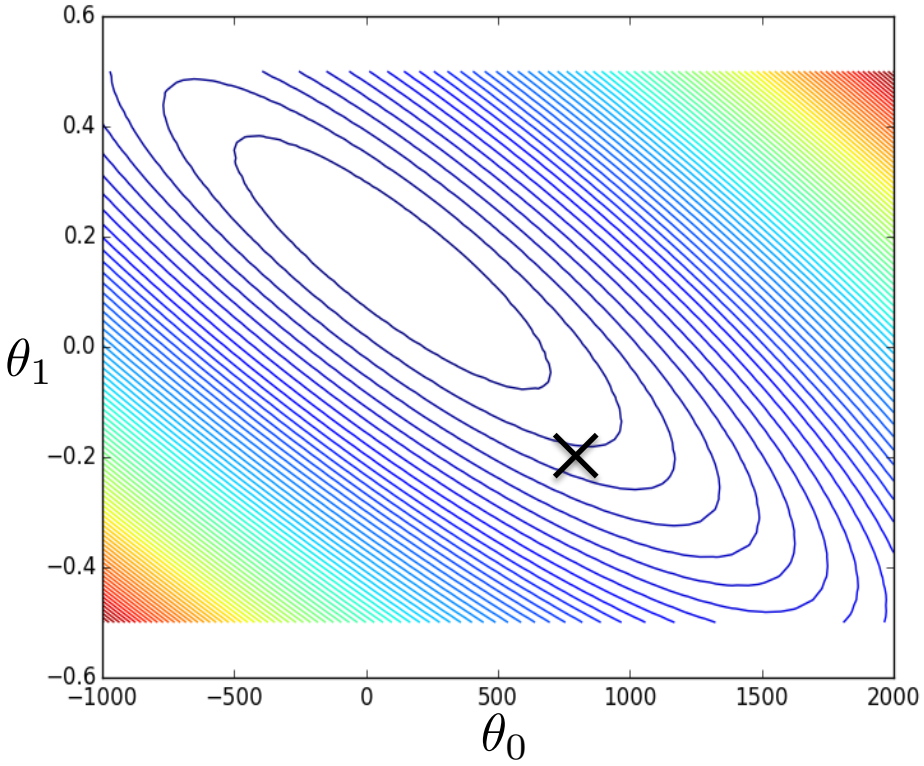
intelligent layer

asi

Cost Function

Points on cost surface correspond to different functions



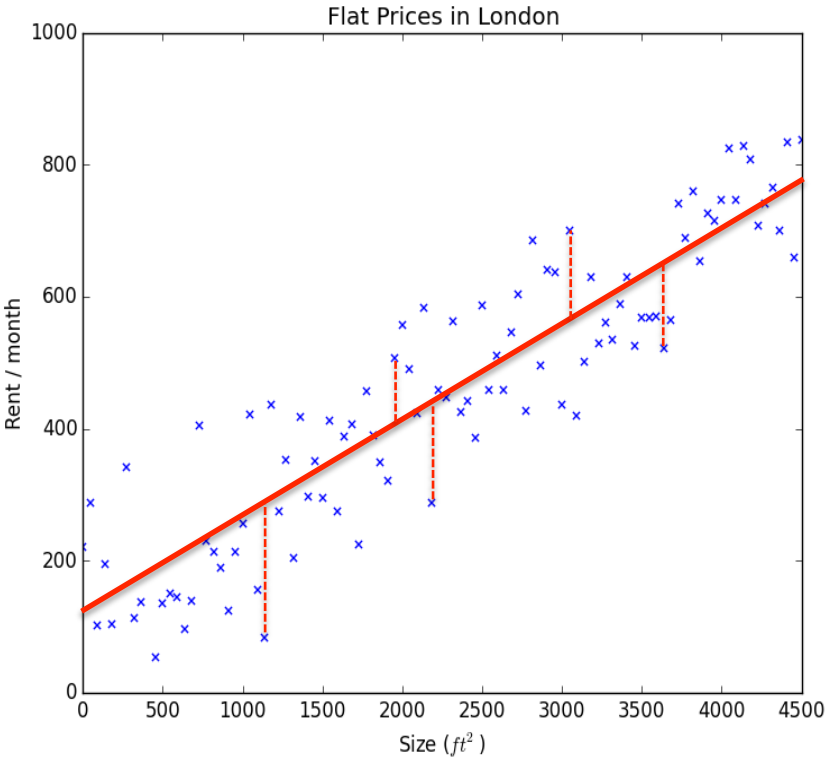


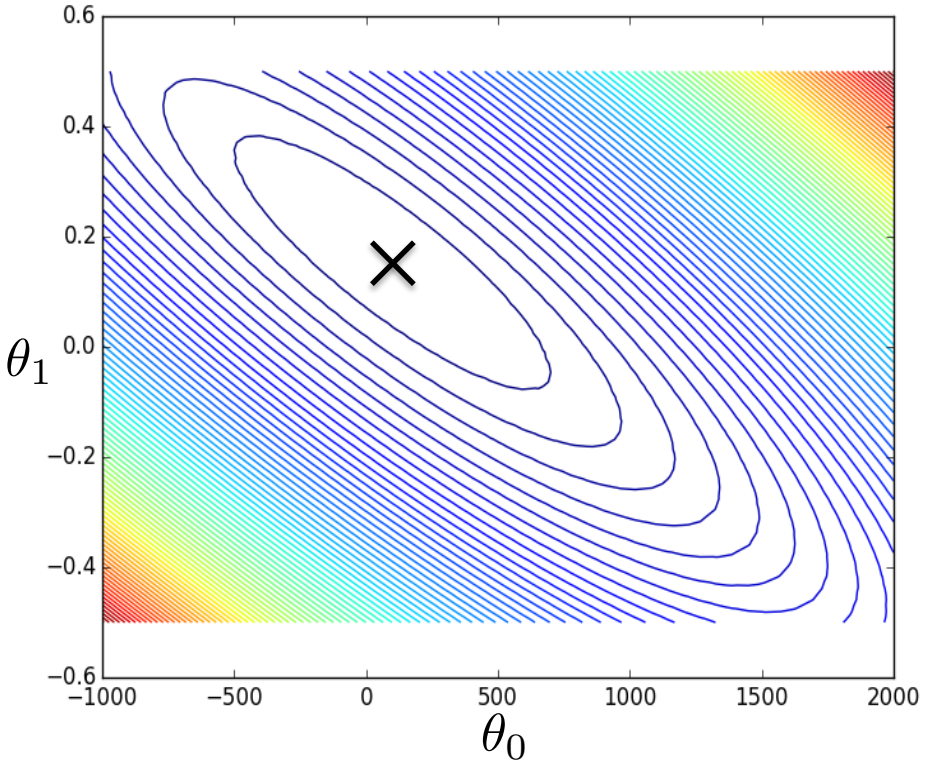
intelligent layer

asi

Cost Function

Points on cost surface correspond to different functions





intelligent layer

asi

EXERCISE

Time: ? mins

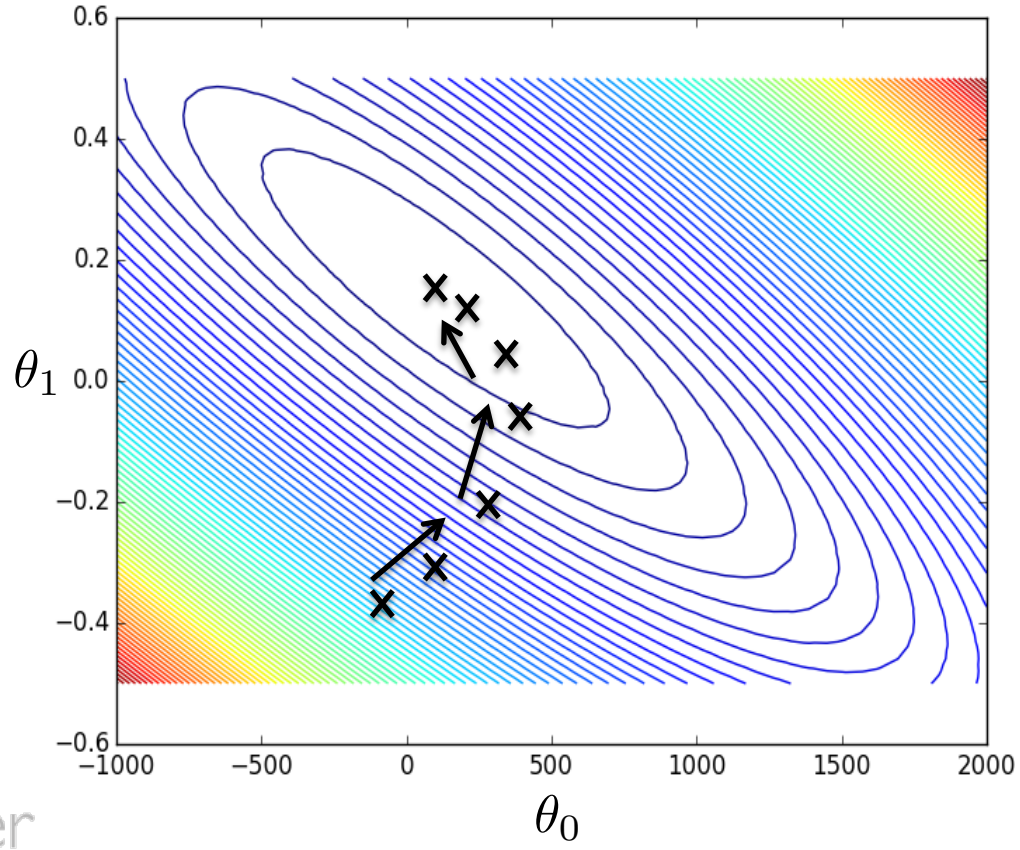
Linear Regression Cost Function

- Open *Linear Regression iPython Notebook* and complete exercises up to Gradient Descent

Gradient Descent

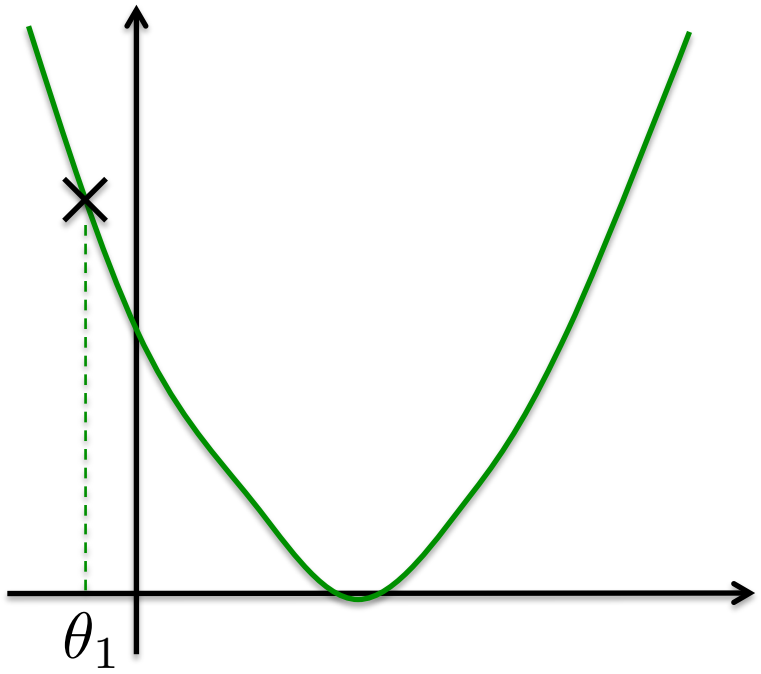
Minimising the Cost Function

Keep changing θ_1 and θ_0 so that C reduces



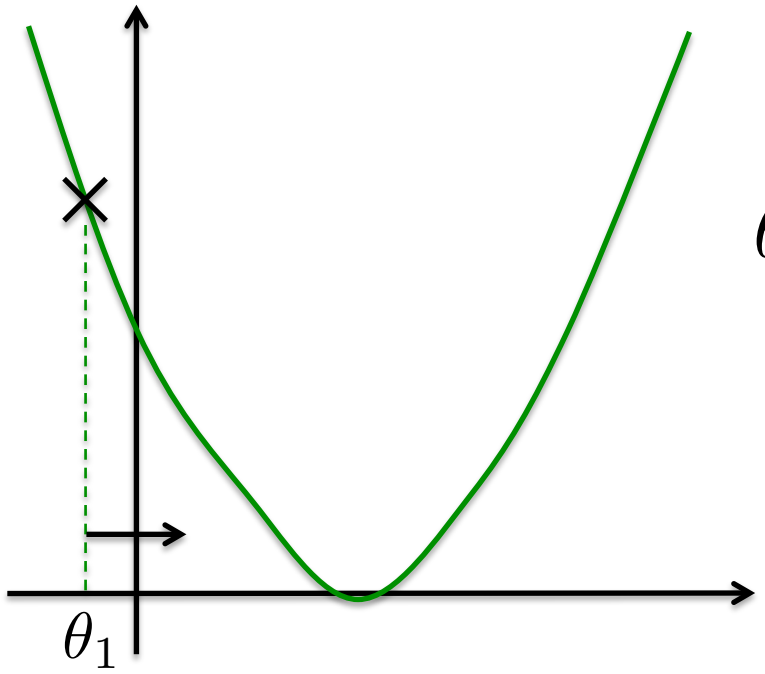
Minimising the Cost Function

Keep changing θ_1 and θ_0 so that C reduces



Minimising the Cost Function

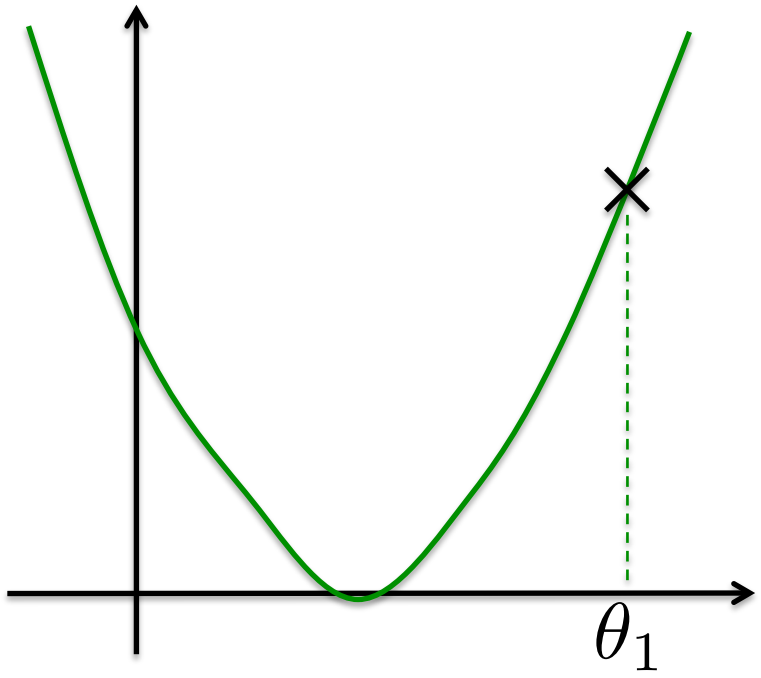
Keep changing θ_1 and θ_0 so that C reduces



$$\theta_1 := \theta_1 + \text{positive number}$$

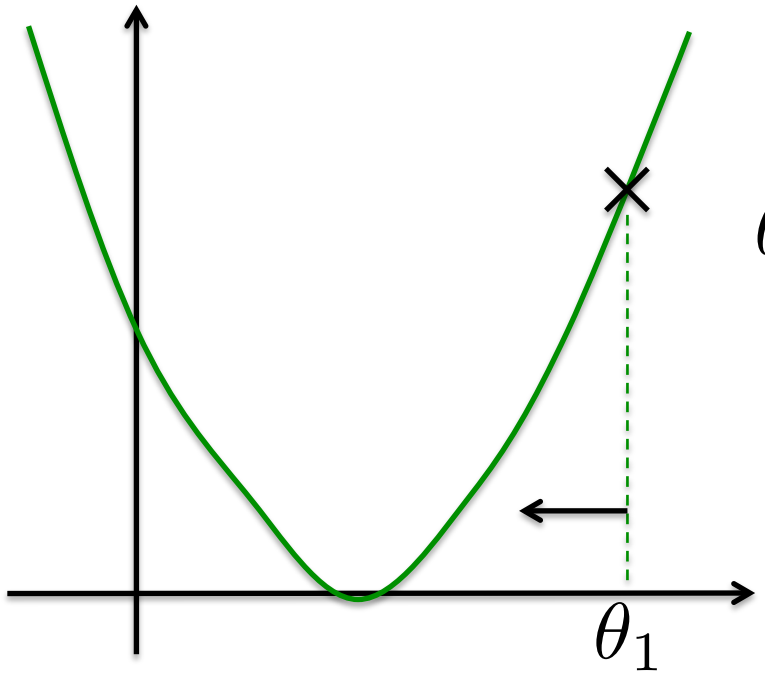
Minimising the Cost Function

Keep changing θ_1 and θ_0 so that C reduces



Minimising the Cost Function

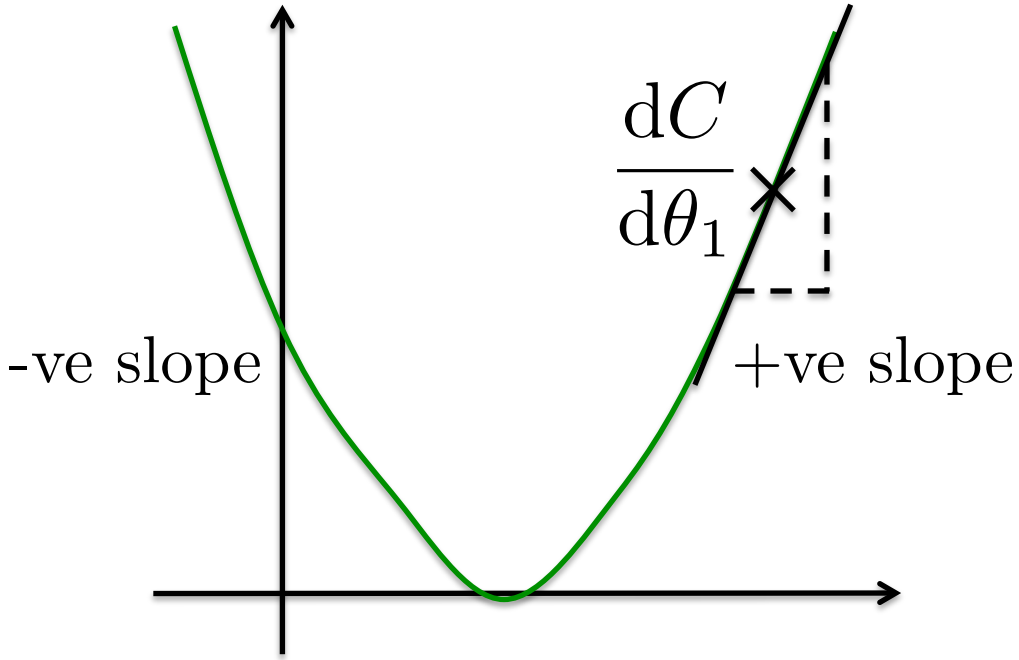
Keep changing θ_1 and θ_0 so that C reduces



$$\theta_1 := \theta_1 + \text{negative number}$$

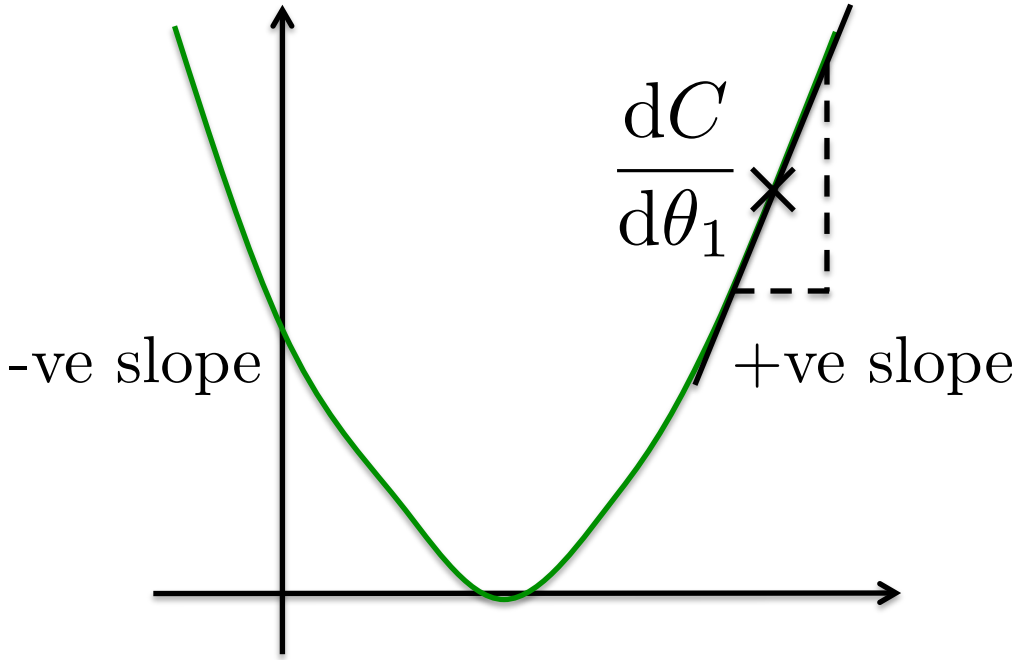
Minimising the Cost Function

Gradient of line tells us which direction to move



Minimising the Cost Function

Gradient of line tells us which direction to move

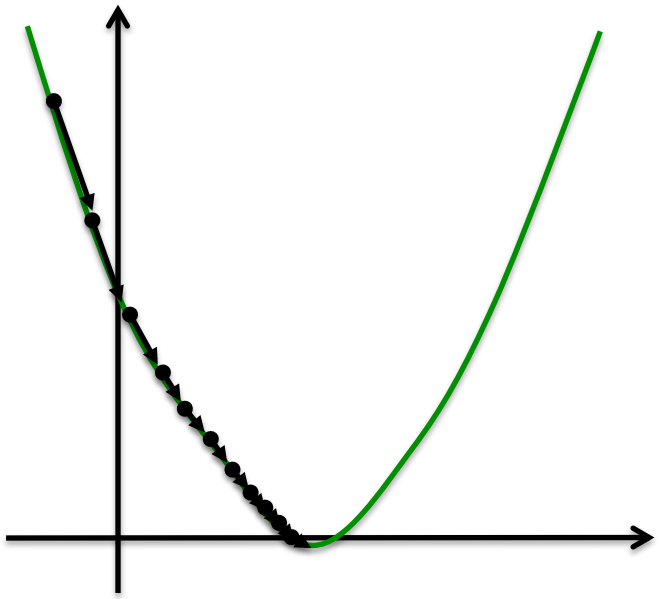


$$\theta_1 := \theta_1 - \alpha \frac{\mathrm{d}C}{\mathrm{d}\theta_1}$$

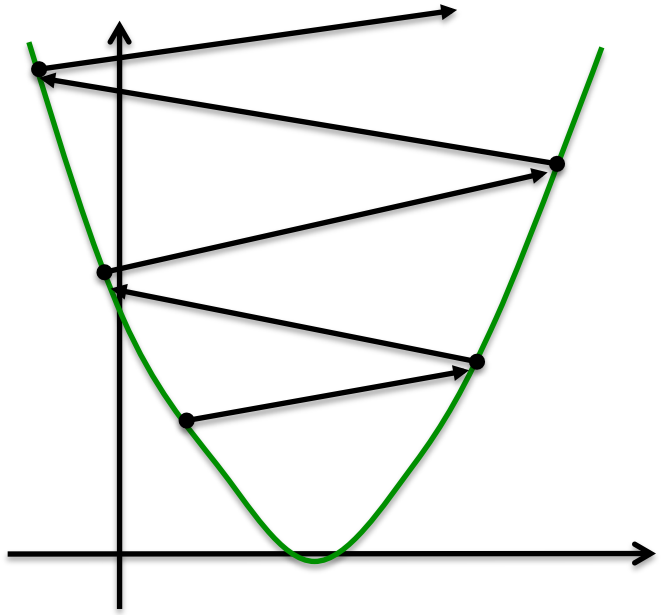
-ve α!

Choosing rate parameter

Convergence of algorithm depends on α



Too small = slow convergence

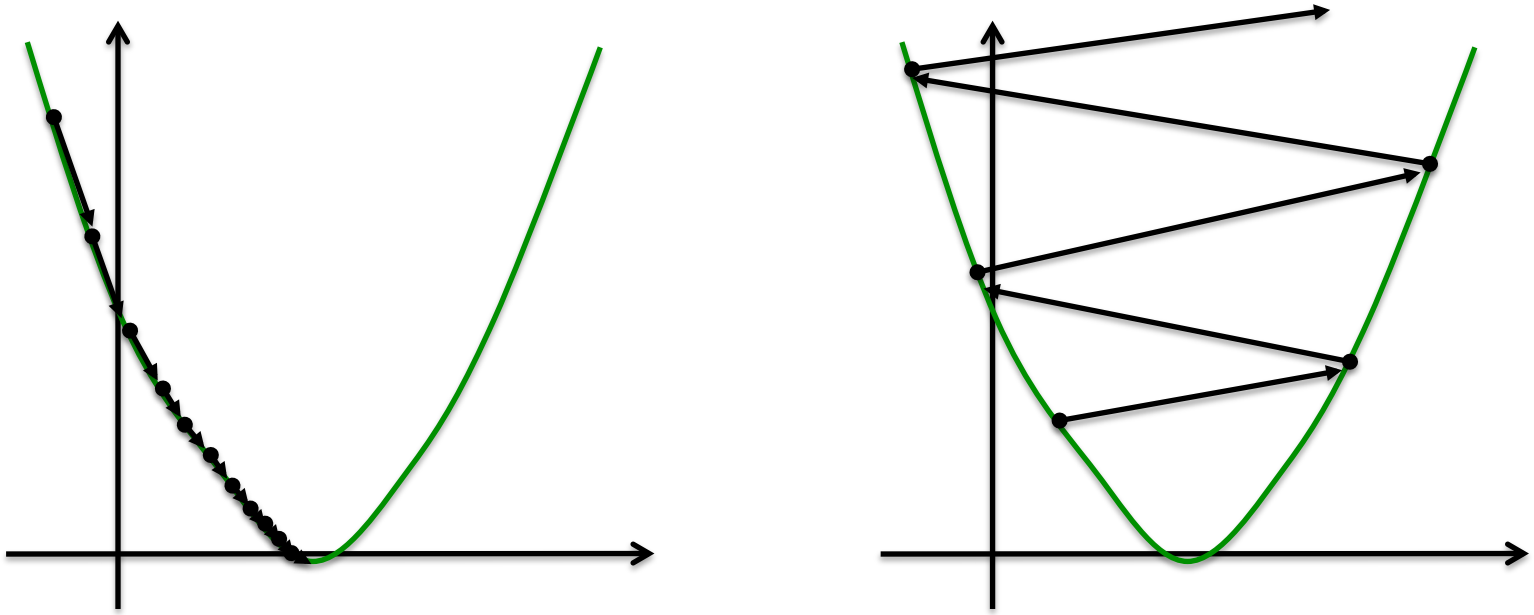


Too big = divergent

$$\theta_1 := \theta_1 - \alpha \frac{\mathrm{d}C}{\mathrm{d}\theta_1}$$

Choosing rate parameter

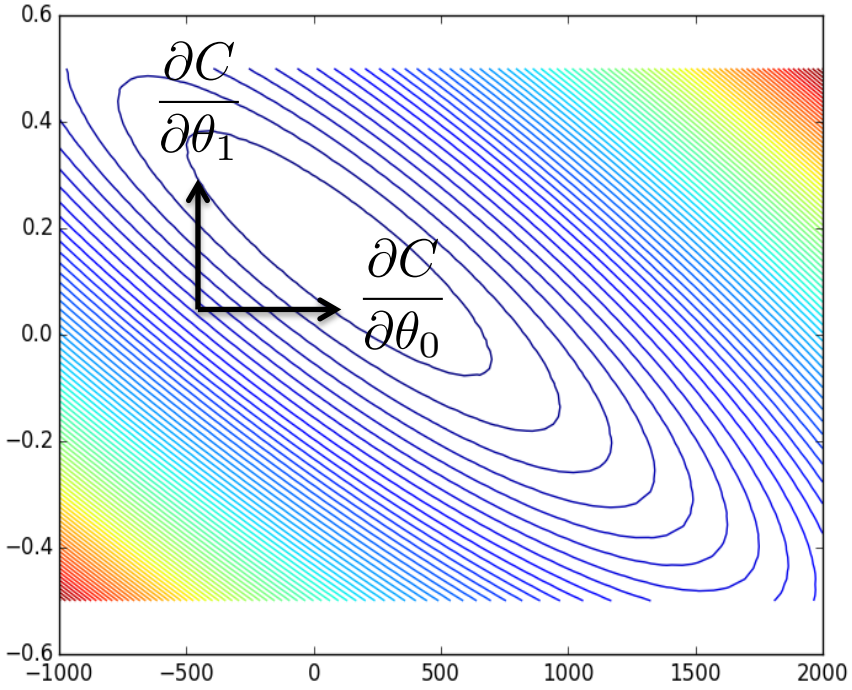
Convergence of algorithm depends on α



NOTE - Gradient Descent will naturally take smaller steps as we approach a local minimum

Full Problem

We must take PARTIAL DERIVATIVES



$$\theta_0 := \theta_0 - \alpha \frac{\partial C}{\partial \theta_0}$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial C}{\partial \theta_1}$$

Gradient Descent Algorithm

Linear Model:

$$f_\theta(x) = \theta_0 + \theta_1 x \qquad C(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i)^2$$

Gradient Descent Algorithm:

repeat until convergence {

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

}

Formula for the update equations

$$C(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i)^2$$

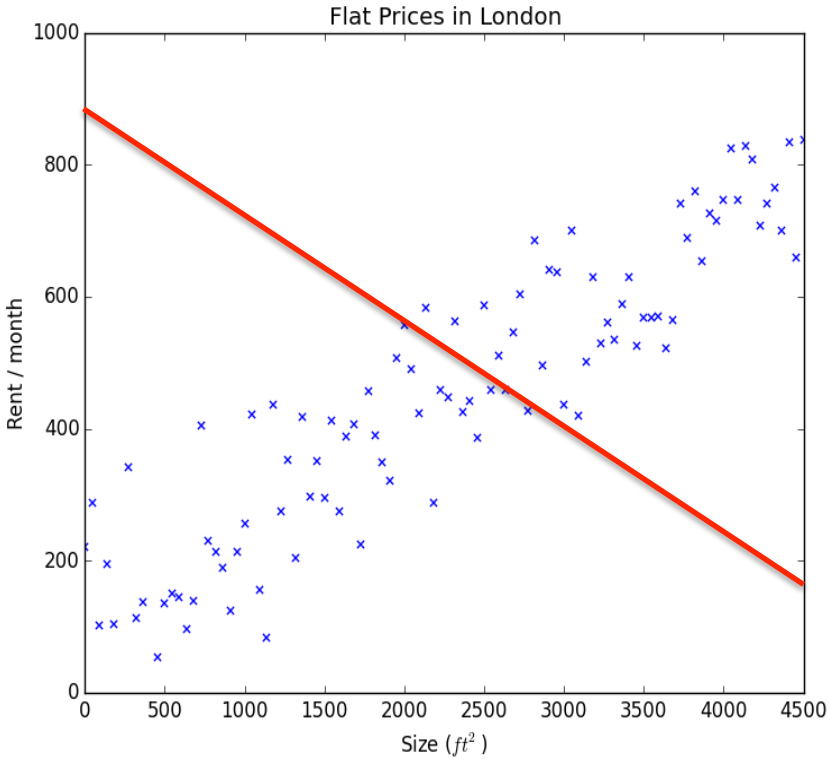
$$= \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^i - y^i)^2$$

$$\theta_0 : \quad \frac{\partial C}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i)$$

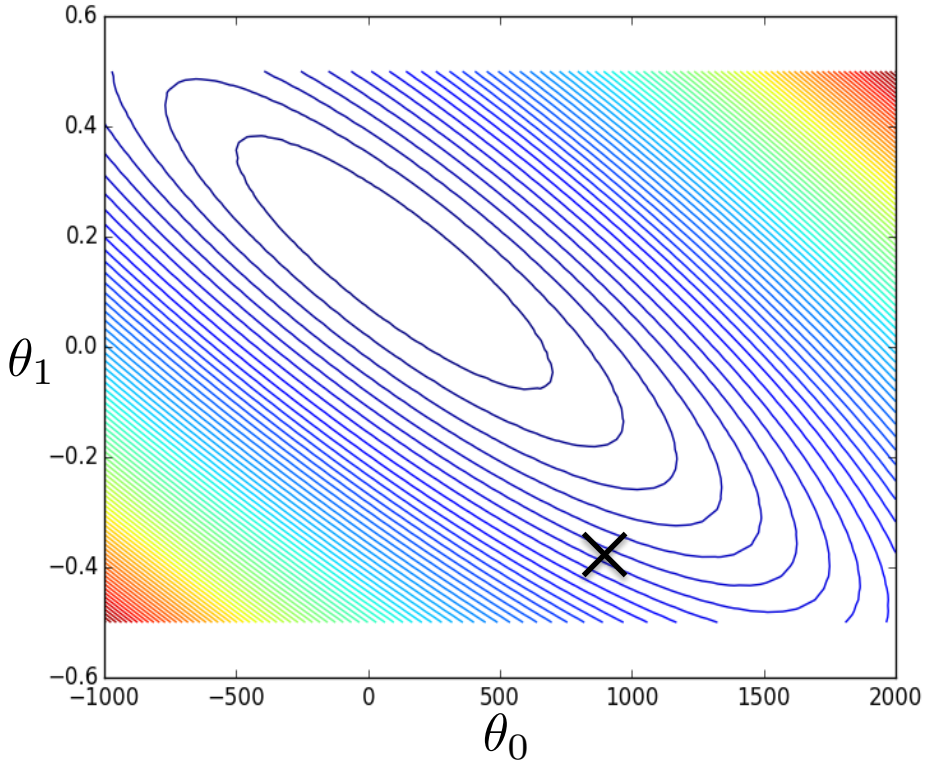
$$\theta_1 : \quad \frac{\partial C}{\partial \theta_1} = \frac{1}{m} \sum_{i=1}^{m} (f_\theta(x^i) - y^i) \cdot x^i$$

Using Gradient Descent

$f_\theta(x)$

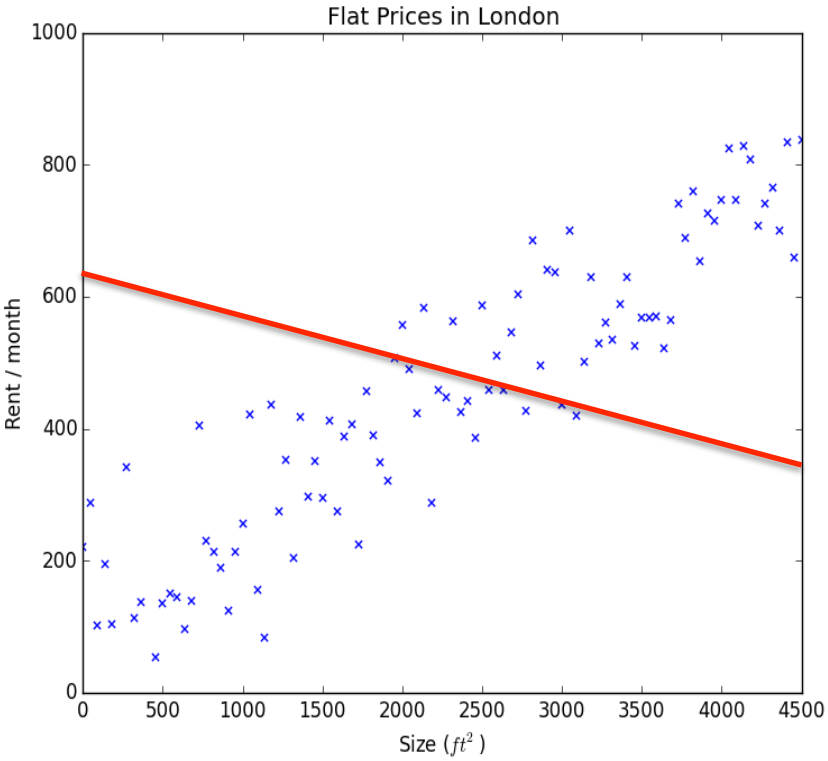


$C(\theta_0, \theta_1)$

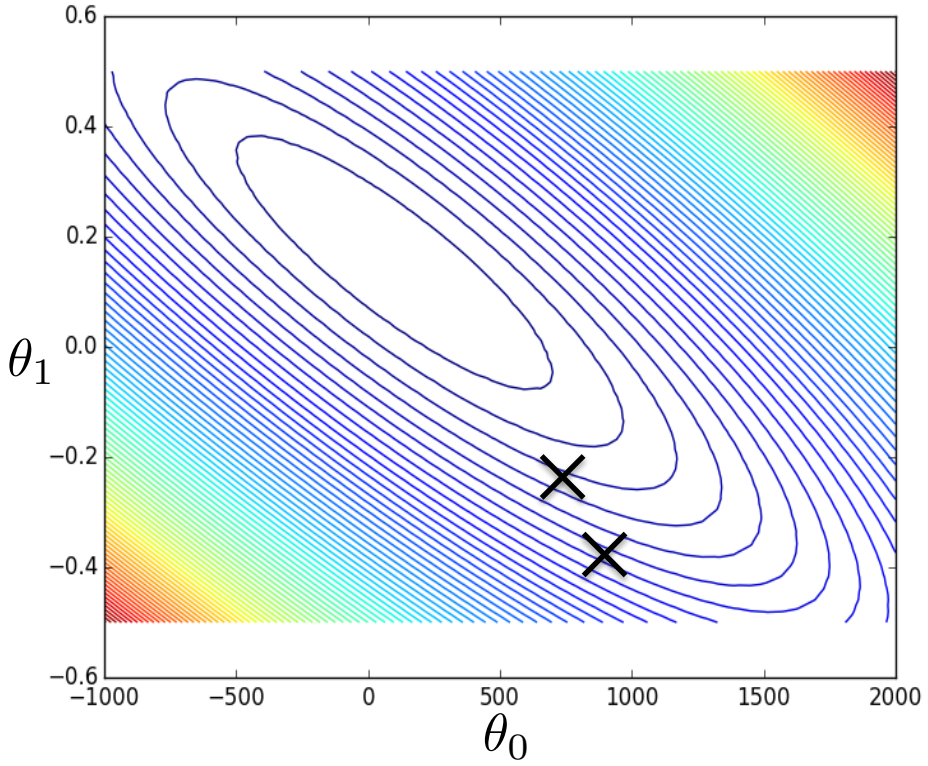


Using Gradient Descent

$f_\theta(x)$

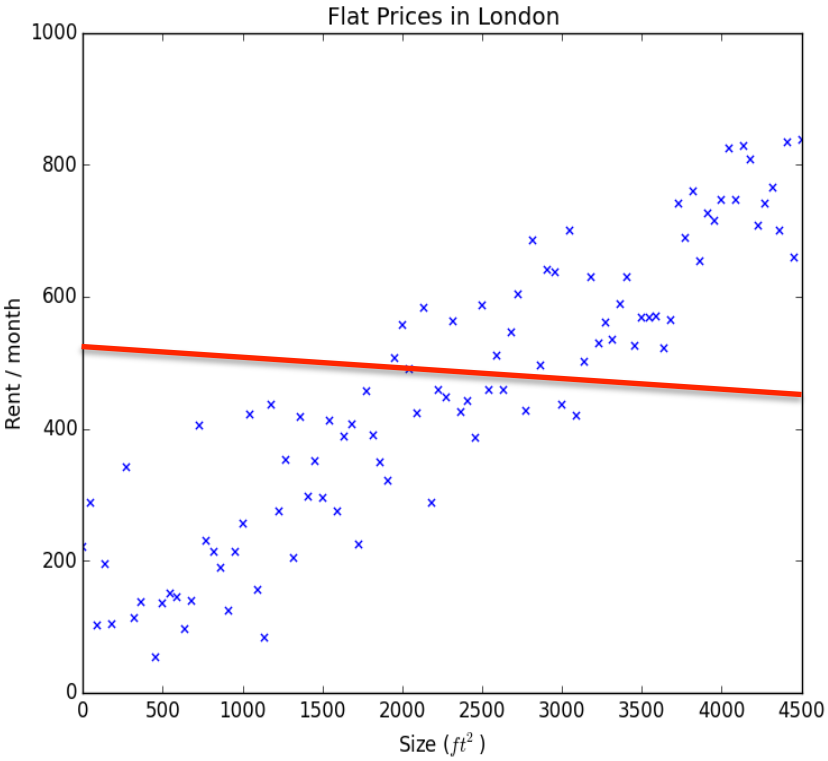


$C(\theta_0, \theta_1)$

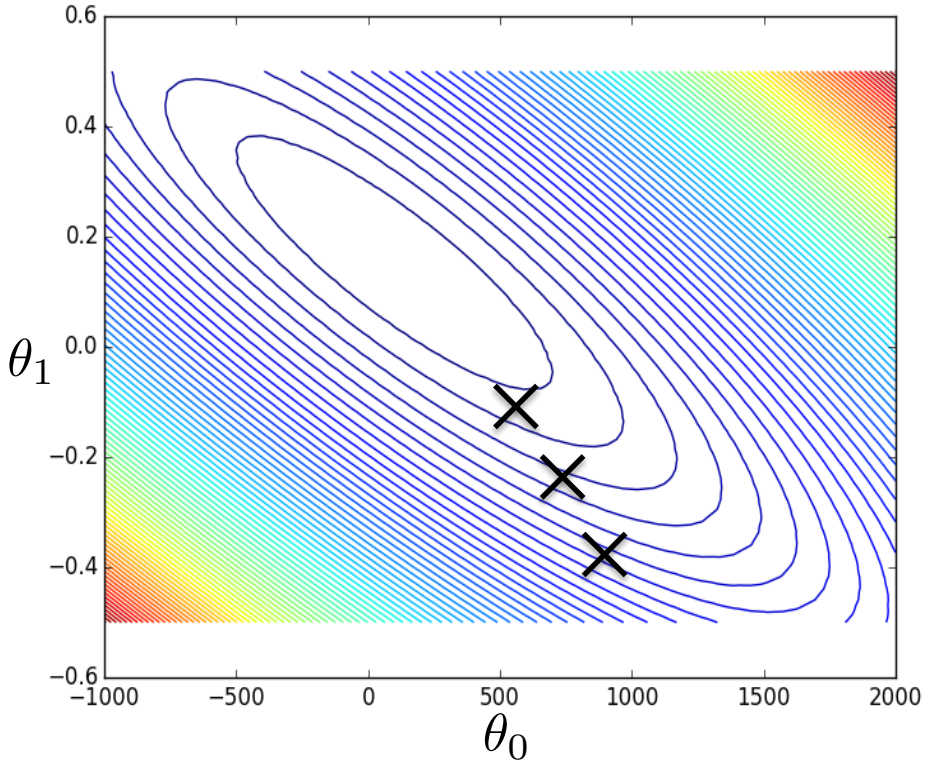


Using Gradient Descent

$f_\theta(x)$



$C(\theta_0, \theta_1)$



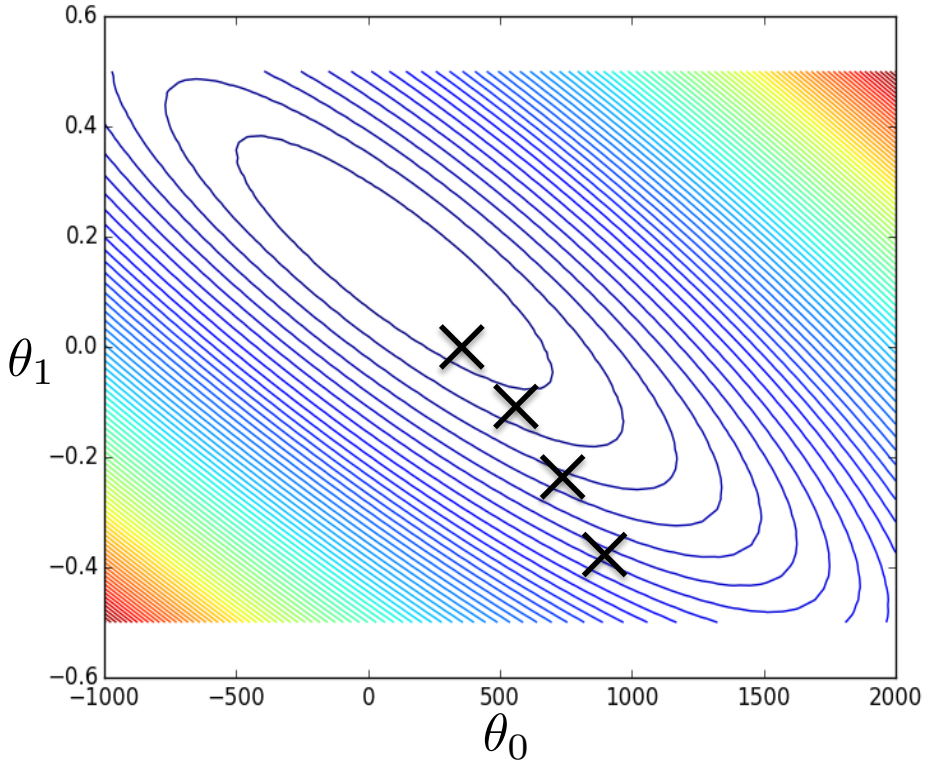
intelligent layer

asi

Using Gradient Descent

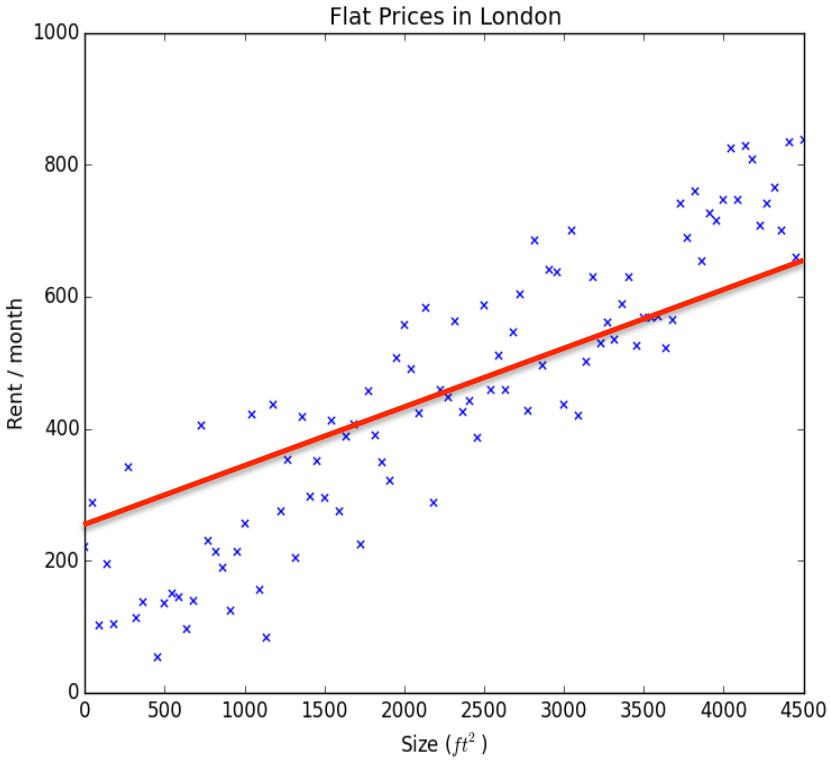
$f_\theta(x)$

$C(\theta_0, \theta_1)$

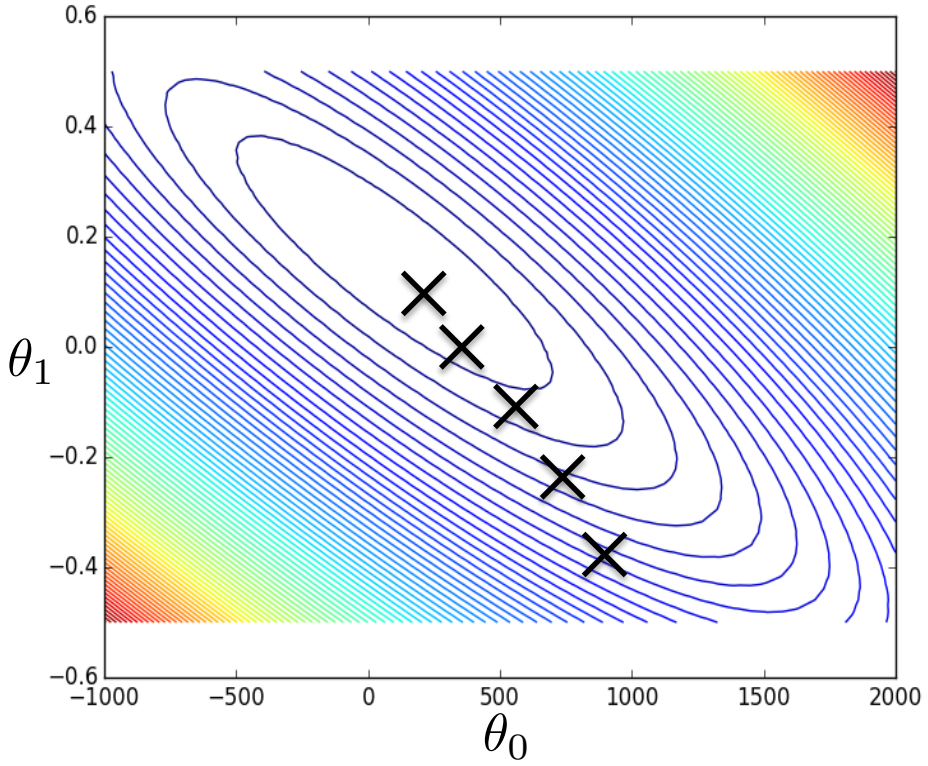


Using Gradient Descent

$f_\theta(x)$



$C(\theta_0, \theta_1)$

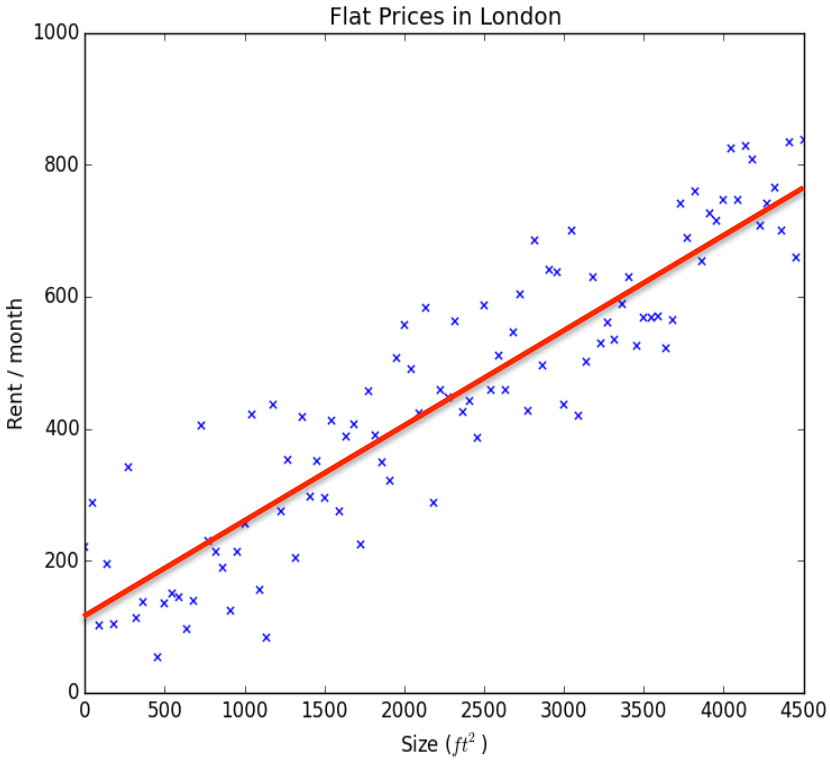


intelligent layer

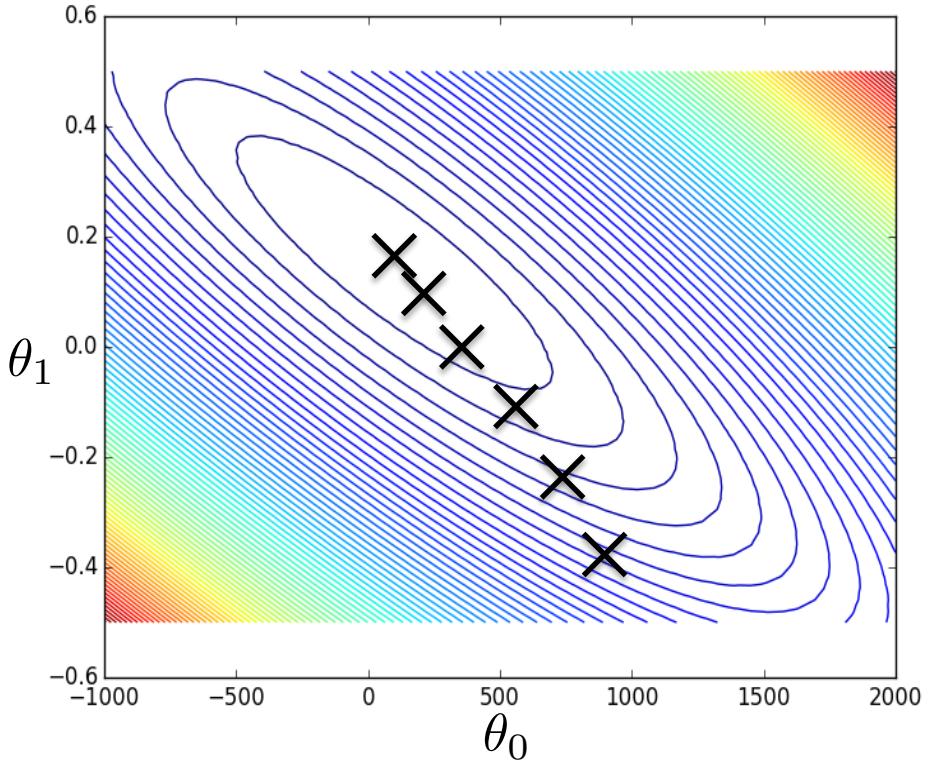
asi

Using Gradient Descent

$f_\theta(x)$



$C(\theta_0, \theta_1)$



EXERCISE

Time: ? mins

Linear Regression Gradient Descent

- Finish exercises in *Linear Regression iPython Notebook*