Machine Learning
机器学习

Lecture2：线性回归

李洁
nijanice@163.com

学习任务的类型
Types of learning task

- Supervised learning
  - infer a function from labeled training data.

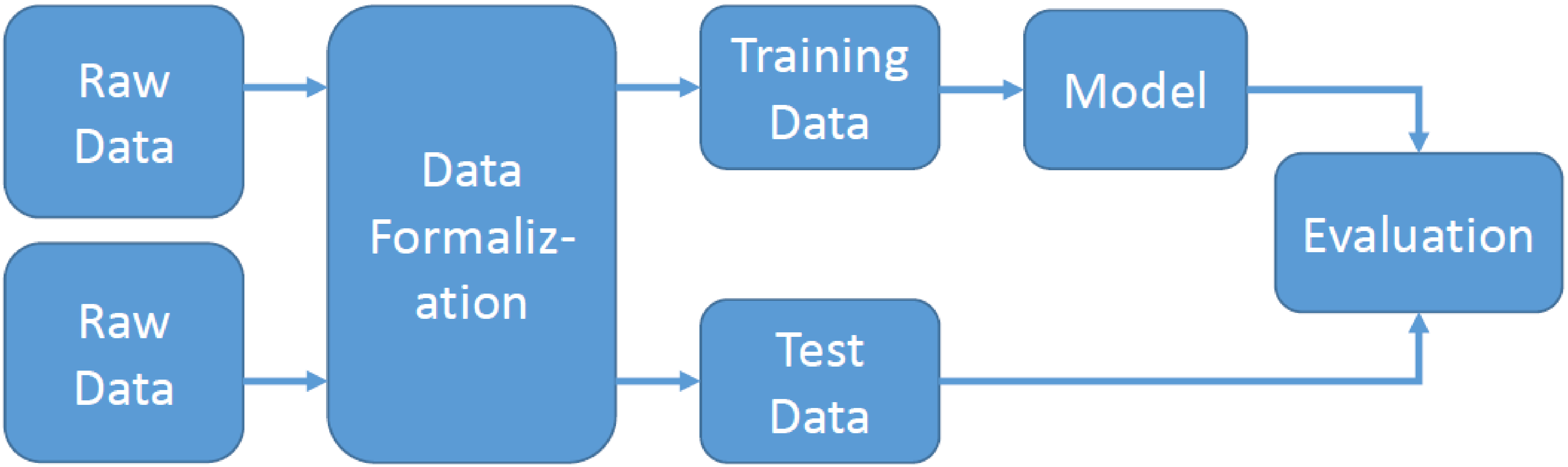
- Unsupervised learning
  - try to find hidden structure in unlabeled training data
  - clustering

- Reinforcement learning
  - To learn a policy of taking actions in a dynamic environment and acquire rewards

学习任务的类型
Types of learning task

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

机器学习的一般过程
Machine Learning Process



- Basic assumption: there exist the same patterns across training and test data

监督学习
Supervised Learning

- Given the training dataset of (data,label) pairs,

$$D = \left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1.2,\ldots,N}$$

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots x_n^{(i)})^T$

$y^{(i)}$ = output data(label) of i^{th} training example

let the machine learn a function from data to label

$$y^{(i)} \approx f_\theta\left(x^{(i)}\right)$$

- Function set $\{f_\theta(x^{(i)})\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ to make the prediction closed to the corresponding label

线性模型
Linear Model

easily understood and implemented, efficient and scalable

- Linear regression
- Linear classification

线性模型举例
Linear model example

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

周志华. “机器学习” （西瓜书）

线性模型举例
Linear model example

$$f_\theta(x) = \theta_1\, x_1 + \theta_2\, x_2 + \cdots + \theta_n\, x_n + \theta_0$$

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

周志华. “机器学习” （西瓜书）

线性回归模型
Linear Regression Model

$$f_\theta(x) = \theta_1\, x_1 + \theta_2\, x_2 + \cdots + \theta_n\, x_n + \ \theta_0$$

sample x

features/variables: $x_1, x_2, \ldots x_n$

线性回归模型
Linear Regression Model

$$f_\theta(x) = \theta_1\, x_1 + \theta_2\, x_2 + \cdots + \theta_n\, x_n + \theta_0$$

sample x

features/variables: $x_1, x_2, \ldots x_n$

$$x = (x_1, x_2, \ldots x_n)^T$$

Feature vector $(x_1, x_2, \ldots x_n)^T$

监督学习
Supervised Learning

- Given the training dataset of (data,label) pairs,

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1.2,\ldots,N}$$

$x^{(i)}$= input data(features) of i^{th} training example
$y^{(i)}$= output data(label) of i^{th} training example

let the machine learn a function from data to label

$$y^{(i)} \approx f_\theta(x^{(i)})$$

- Function set $\{f_\theta(x^{(i)})\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ to make the prediction closed to the corresponding label

监督学习
Supervised Learning

- Given the training dataset of (data,label) pairs,

$$D = \left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1.2,\ldots,N}$$

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots x_n^{(i)})^T$

$y^{(i)}$ = output data(label) of i^{th} training example

let the machine learn a function from data to label

$$y^{(i)} \approx f_\theta\left(x^{(i)}\right)$$

- Function set $\{f_\theta\left(x^{(i)}\right)\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ to make the prediction closed to the corresponding label

线性回归模型
Linear Regression Model

- Given the training dataset of (data,label) pairs,

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1.2,\dots,N}$$

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots x_n^{(i)})^T$

$y^{(i)}$= output data(label) of i^{th} training example

let the machine learn a function from data to label

$$y^{(i)} \approx \boxed{f_\theta(x^{(i)})} \Rightarrow \boxed{f_\theta(x^{(i)}) = \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_n x_n^{(i)} + \theta_0}$$

- Function set $\{f_\theta(x^{(i)})\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ to make the prediction closed to the corresponding label

线性回归模型
Linear Regression Model

- Given the training dataset of (data,label) pairs,

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1,2,\dots,N}$$

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots x_n^{(i)})^T$

$y^{(i)} =$ output data(label) of i^{th} training example

let the machine learn a function from data to label

$$y \approx \boxed{f_\theta(x)} \Rightarrow \boxed{f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \theta_0}$$

- Function set $\{f_\theta(x^{(i)})\}$ is called hypothesis space
- Learning is referred to as updating the parameter θ to make the prediction closed to the corresponding label

线性回归模型
Linear Regression Model

$$f_\theta(x) = \theta_1\, x_1 + \theta_2\, x_2 + \cdots + \theta_n\, x_n + \theta_0$$

sample x

features/variables: $x_1, x_2, \ldots x_n$

线性回归模型
Linear Regression Model

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

sample x

One feature/variable: x

线性回归模型
Linear Regression Model

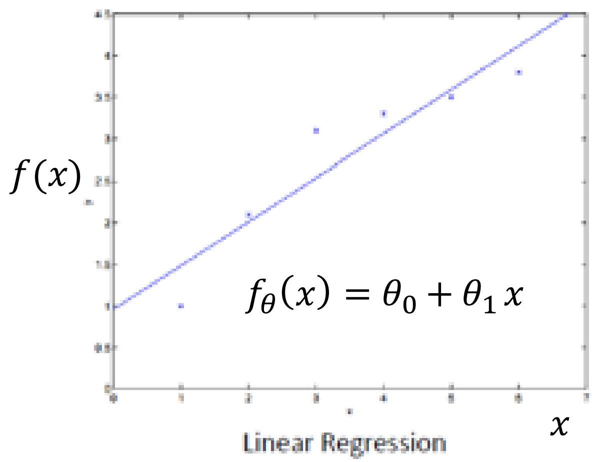
$$f_\theta(x) = \theta_1 x + \theta_0$$

sample x

One feature/variable: x

Linear regression
with one variable

(One-dimensional linear regression)



线性回归模型
Linear Regression Model

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

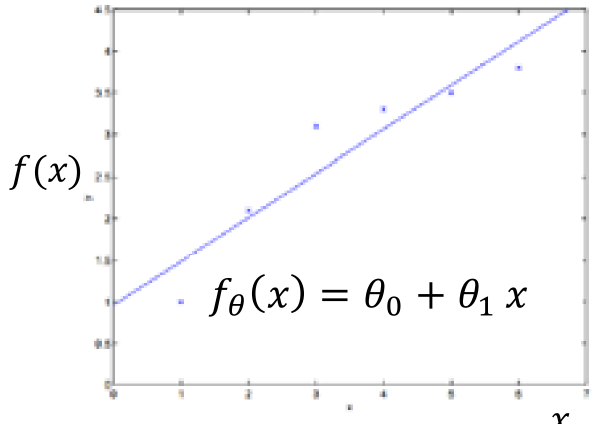
sample x

One feature/variable: x

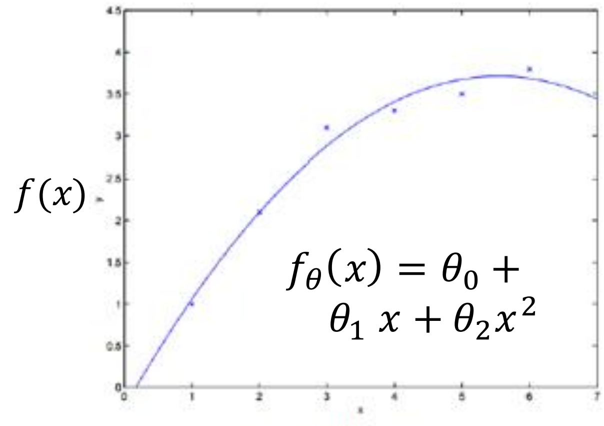
Linear regression with one variable

quadratic regression with one variable

(One-dimensional regression)



Linear Regression



Quadratic Regression
(A kind of generalized linear model)

线性回归模型
Linear Regression Model

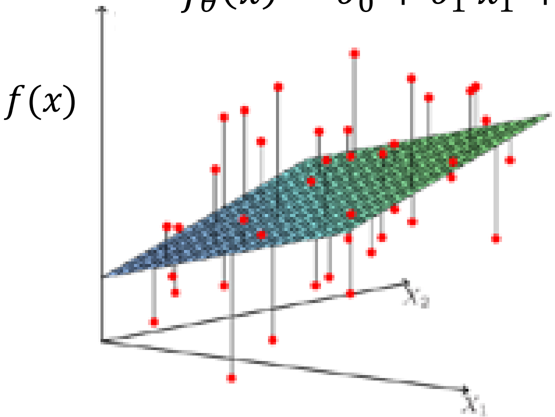
$$f_\theta(x) = \theta_1\, x_1 + \theta_2\, x_2 + \theta_0$$

sample x

Two features/variables: x_1, x_2

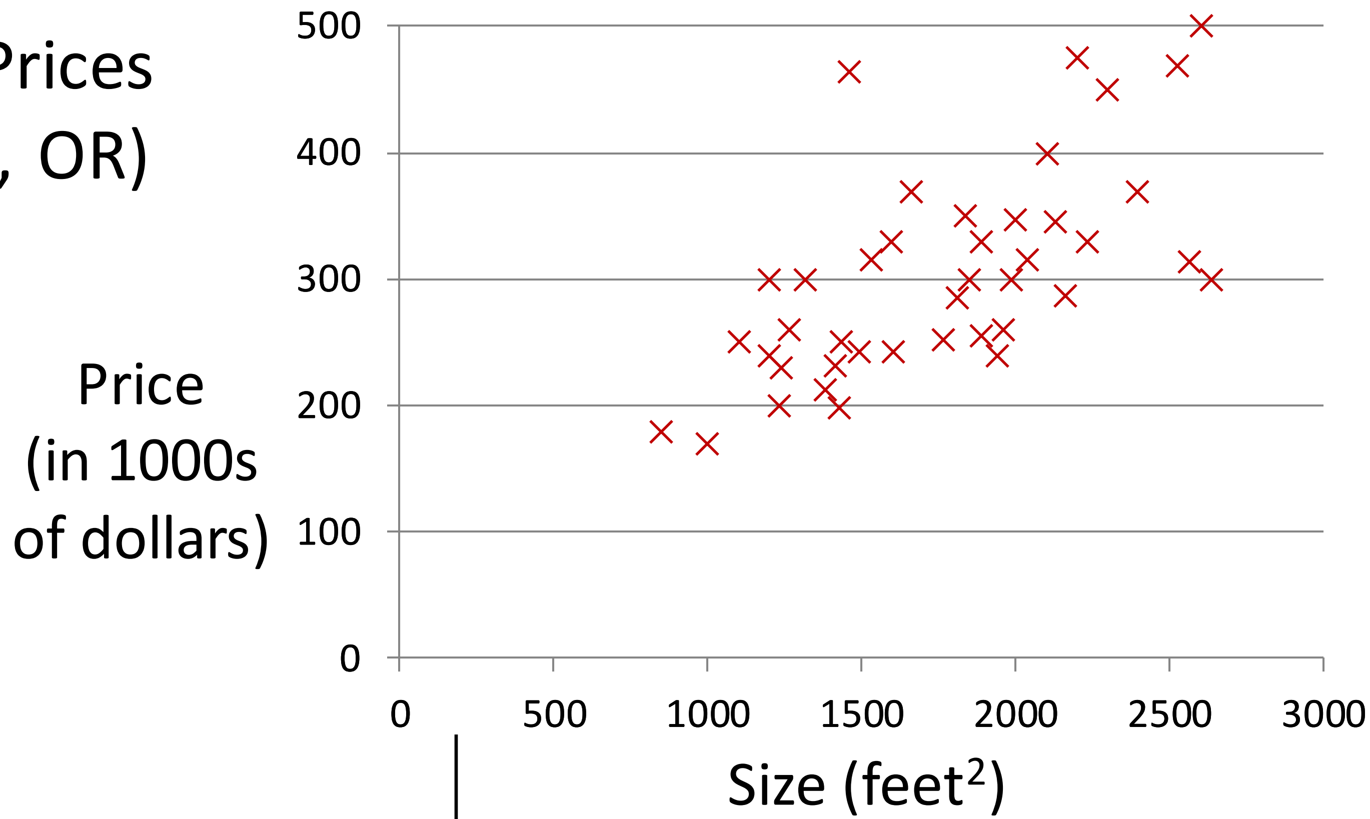
Linear regression
with two variable
(two-dimensional
linear regression)

$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2\, x_2$$



单变量线性回归
Linear regression with one variable

Housing Prices
(Portland, OR)



<u>Supervised Learning</u>

Given the “right answer” for each example in the data.

<u>Regression Problem</u>

Predict real-valued output

单变量线性回归
Linear regression with one variable

Training set of housing prices (Portland, OR)

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

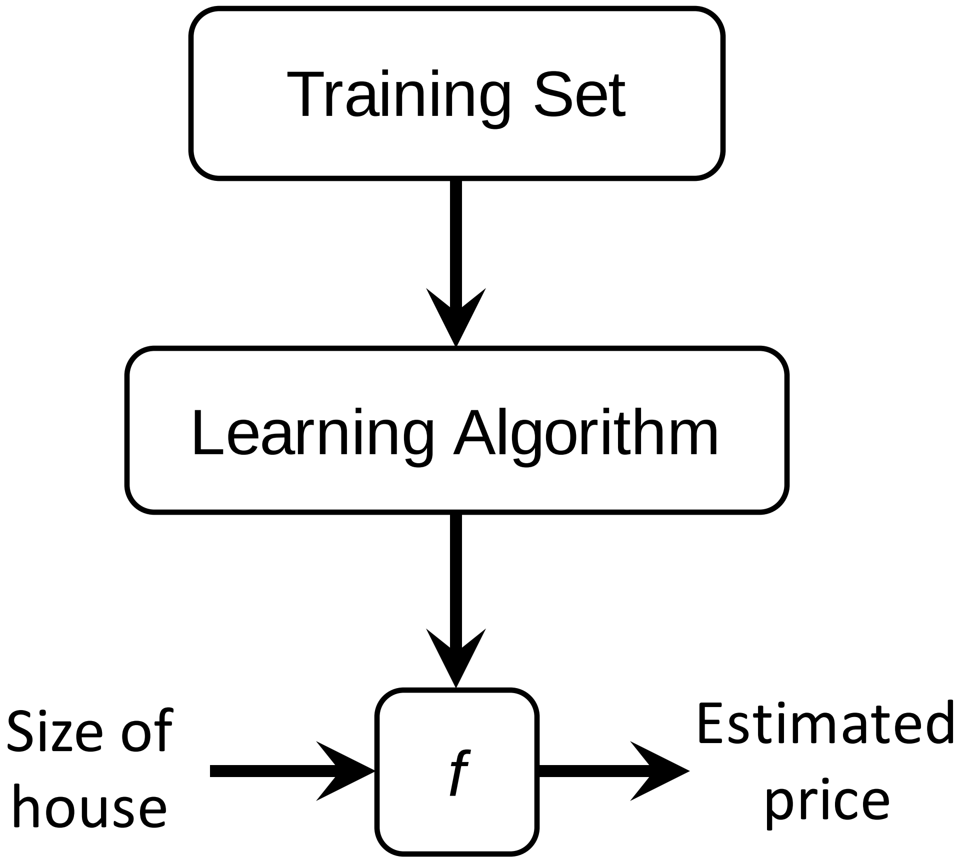
Notation:

N= Number of training examples

x = “input” variable / features

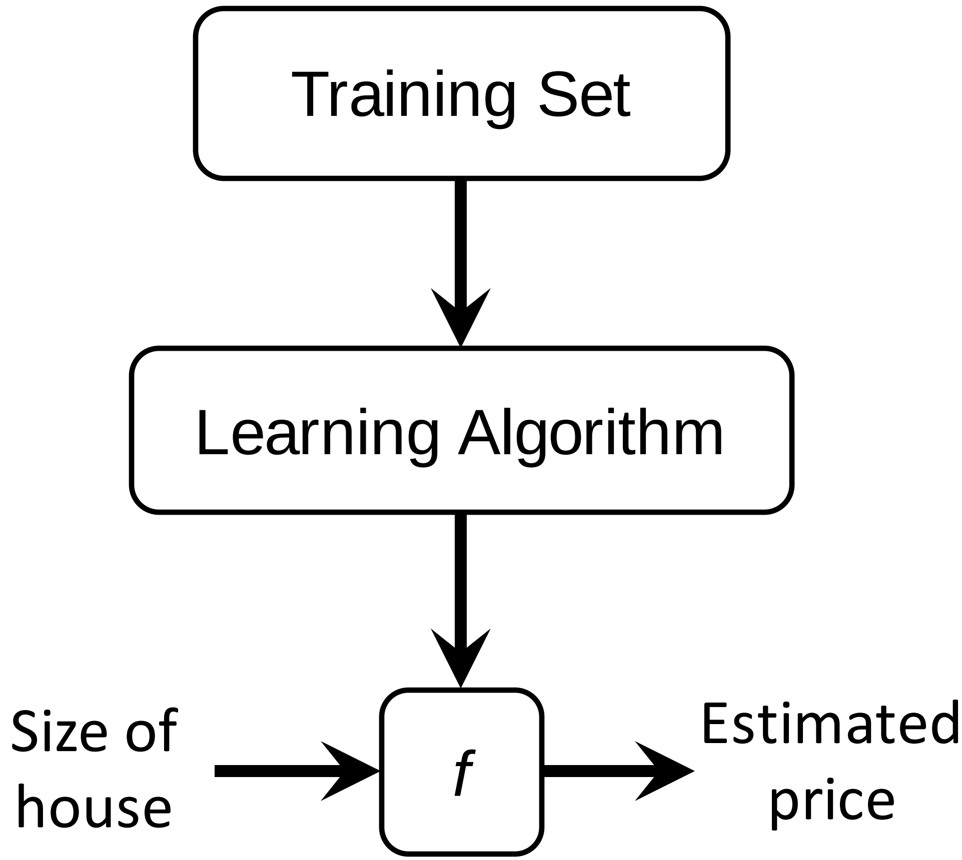
y = “output” variable / “target” variable

单变量线性回归
Linear regression with one variable



How do we represent f?

单变量线性回归
Linear regression with one variable

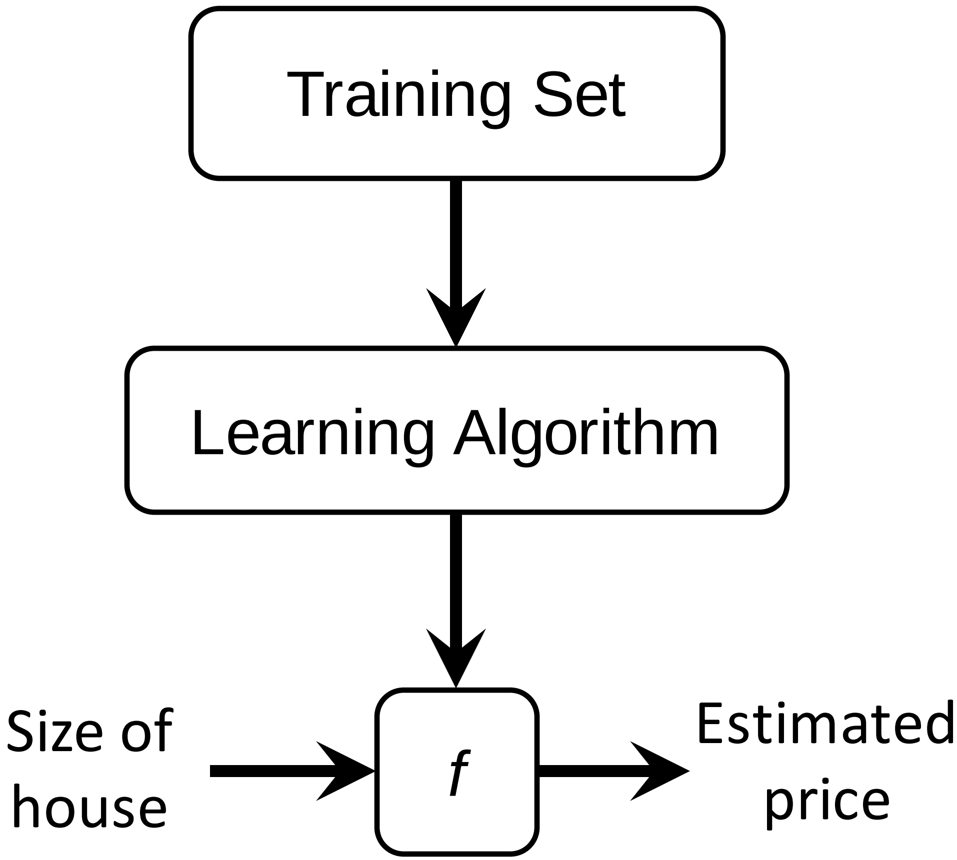


How do we represent f?

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

Linear regression with one variable.
Univariate(one variable) linear regression.

单变量线性回归
Linear regression with one variable



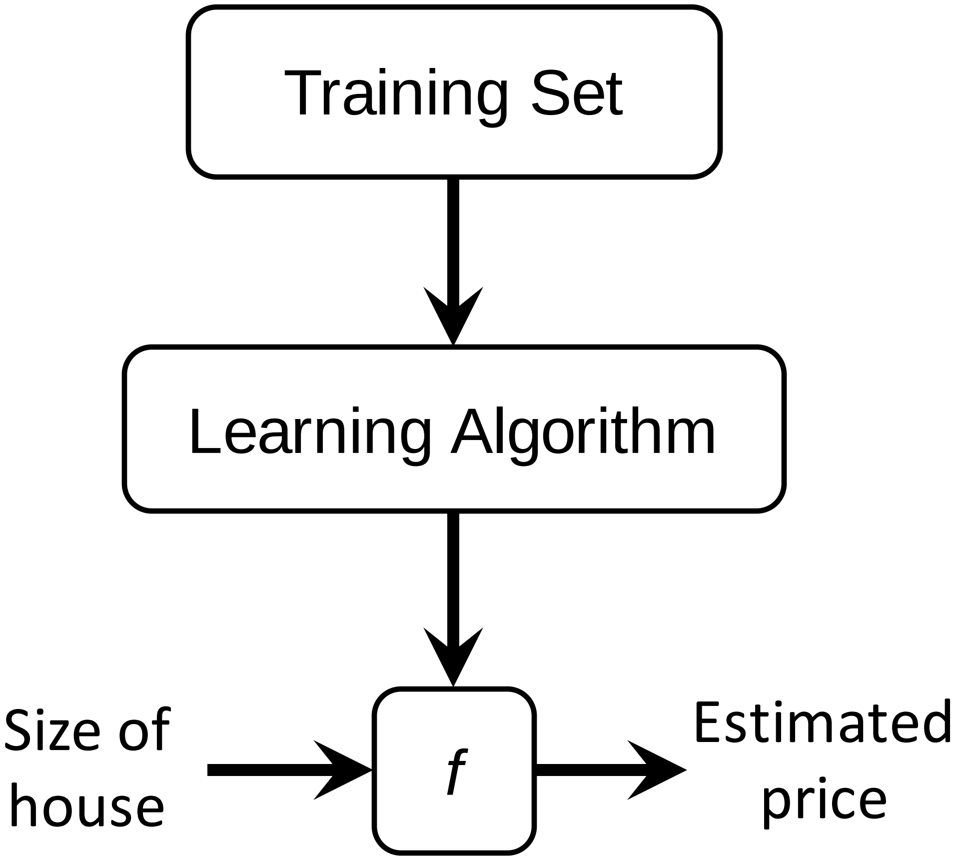
How do we represent f?

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

θ_0, θ_1: Parameters

Linear regression with one variable.
Univariate(one variable) linear regression.

单变量线性回归
Linear regression with one variable



How do we represent f?

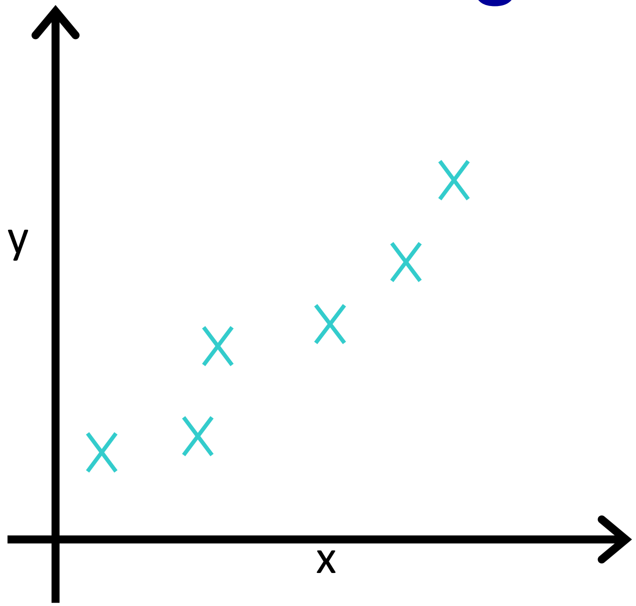
$$f_\theta(x) = \theta_0 + \theta_1 x$$

θ_0, θ_1: Parameters

How to choose θ_0, θ_1 ?

Linear regression with one variable.
Univariate(one variable) linear regression.

单变量线性回归
Linear regression with one variable

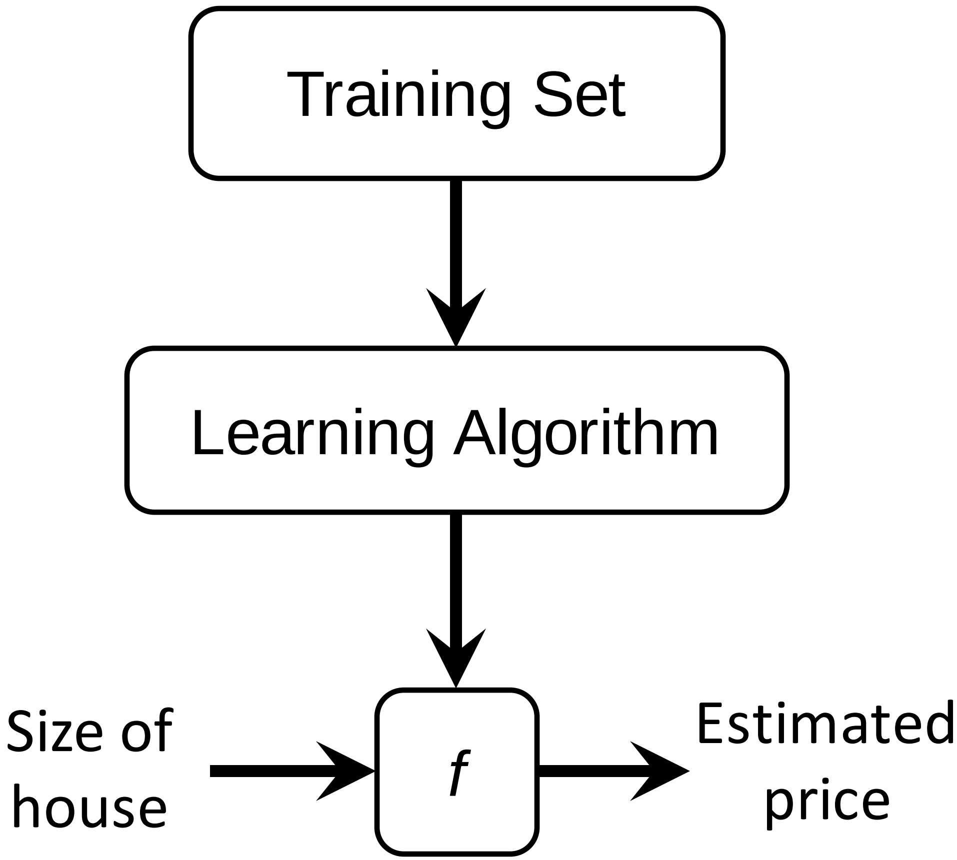


Training Set

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

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

单变量线性回归
Linear regression with one variable



Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

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

Goal:

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

单变量线性回归
Linear regression with one variable

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

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

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

单变量线性回归
Linear regression with one variable

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

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

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

<u>Simplified</u>

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

$$\theta_1$$

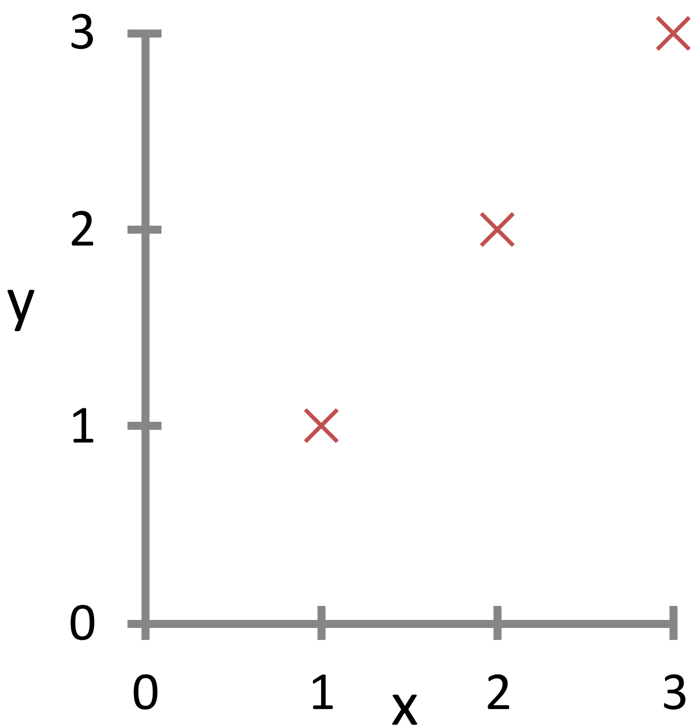
$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

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

单变量线性回归
Linear regression with one variable

$f_\theta(x)$

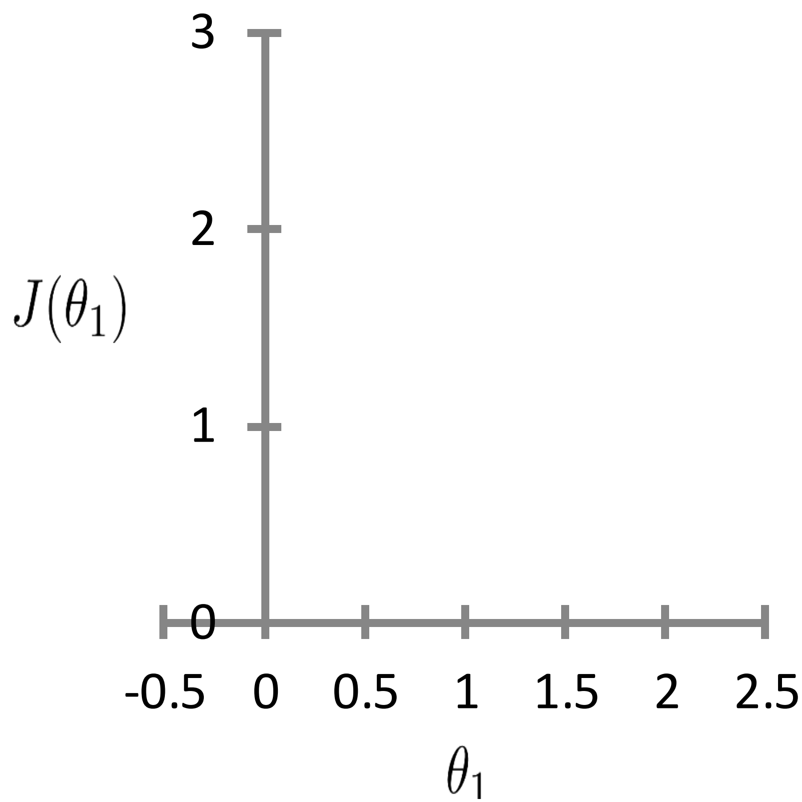
(for fixed θ_1, this is a function of x)



$$f_\theta\left(x^i\right) = \theta_1 x^{(i)}$$

$J(\theta_1)$

(function of the parameter θ_1)

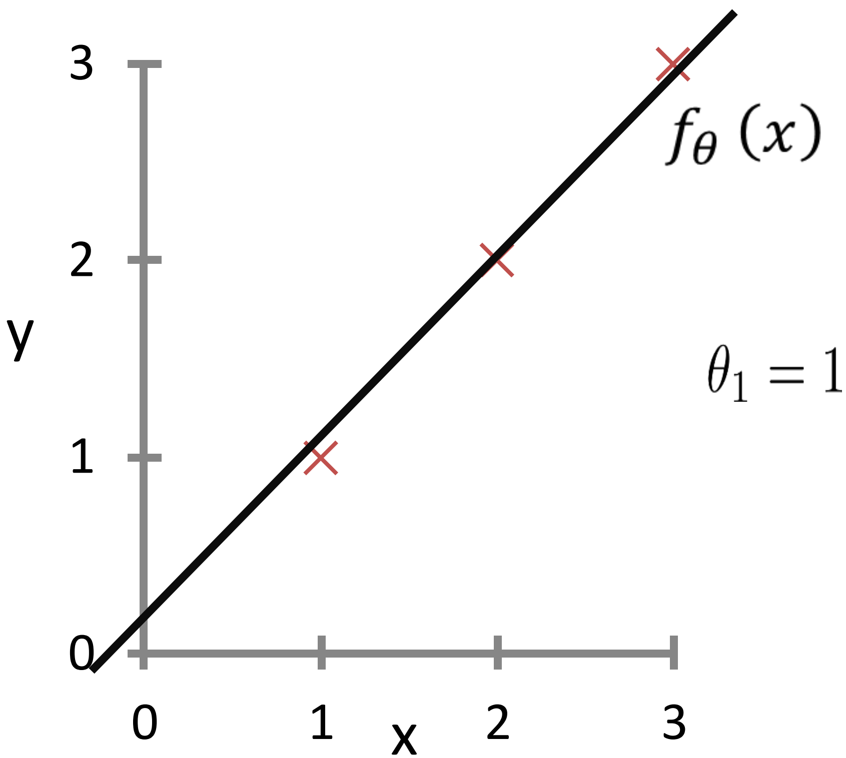


$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归
Linear regression with one variable

$$f_\theta(x)$$

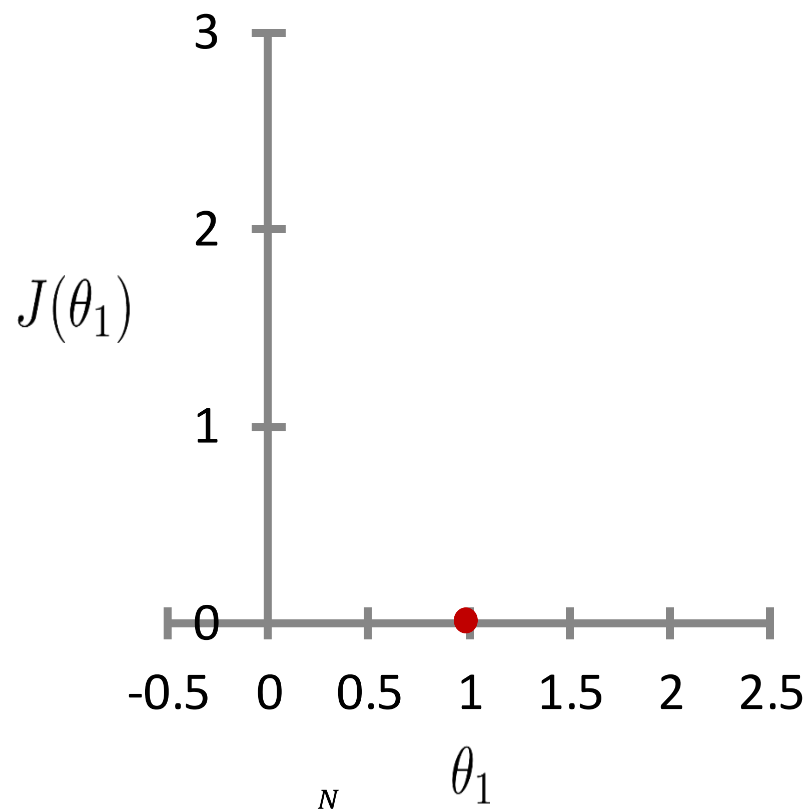
(for fixed θ_1, this is a function of x)



$$f_\theta(x^i) = \theta_1 x^{(i)}$$

$$J(\theta_1)$$

(function of the parameter θ_1)

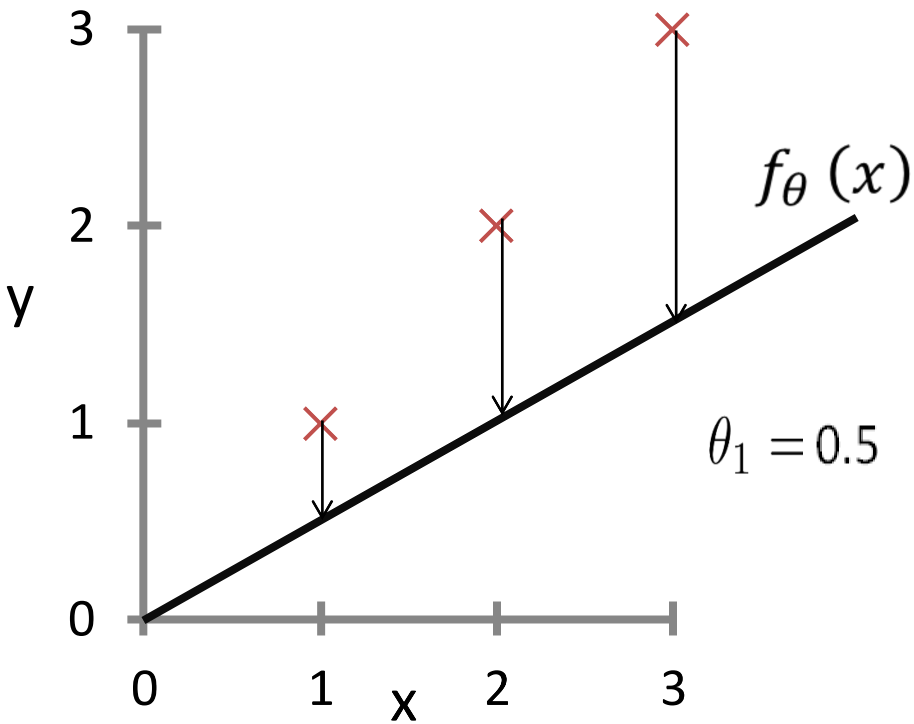


$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归
Linear regression with one variable

$f_\theta(x)$

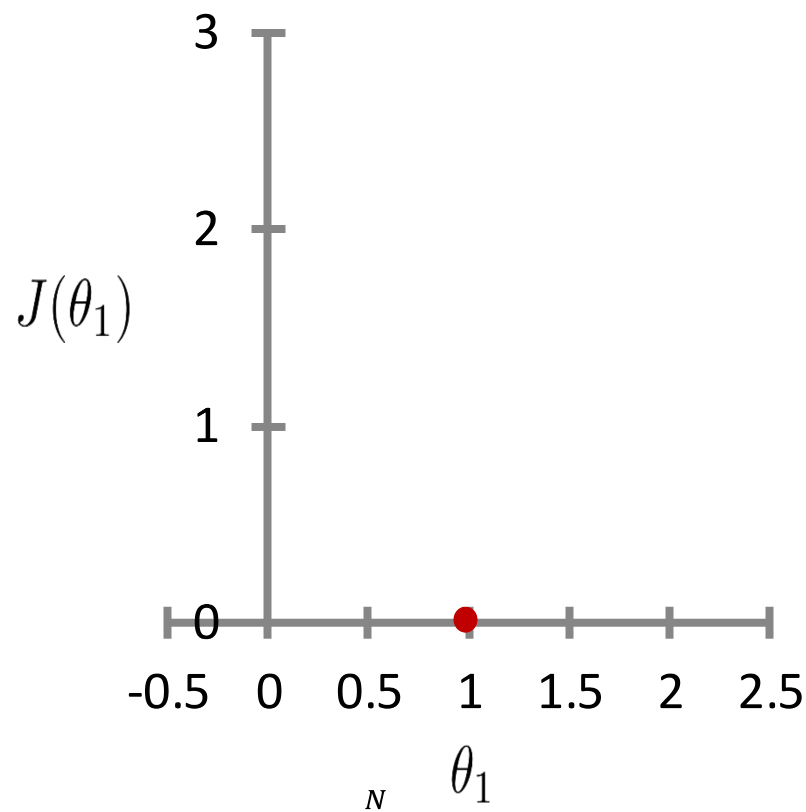
(for fixed θ_1, this is a function of x)



$f_\theta(x^i) = \theta_1 x^{(i)}$

$J(\theta_1)$

(function of the parameter θ_1)

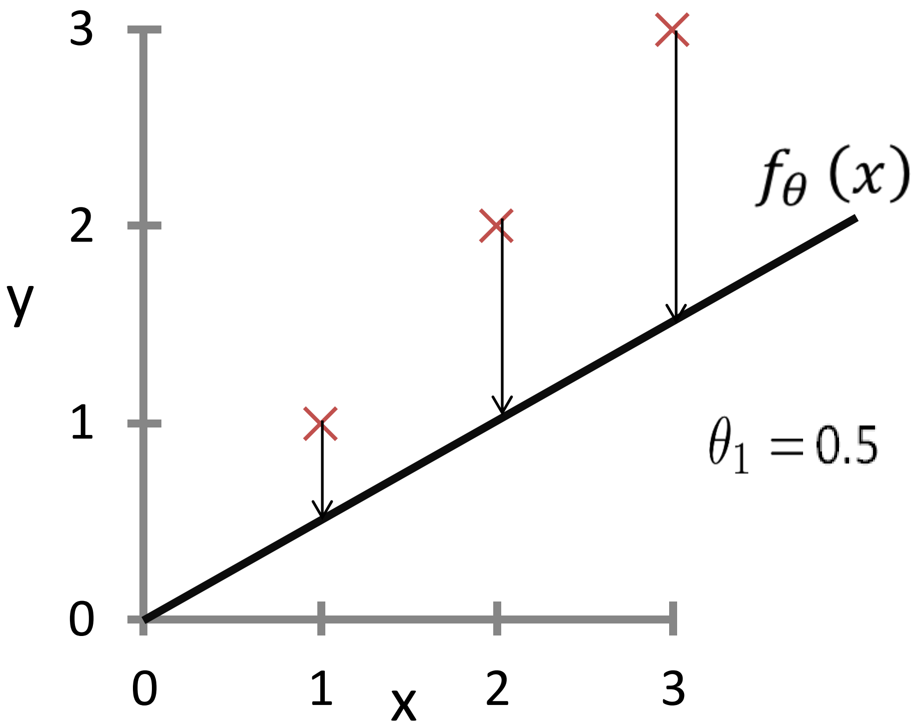


$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归
Linear regression with one variable

$f_\theta(x)$

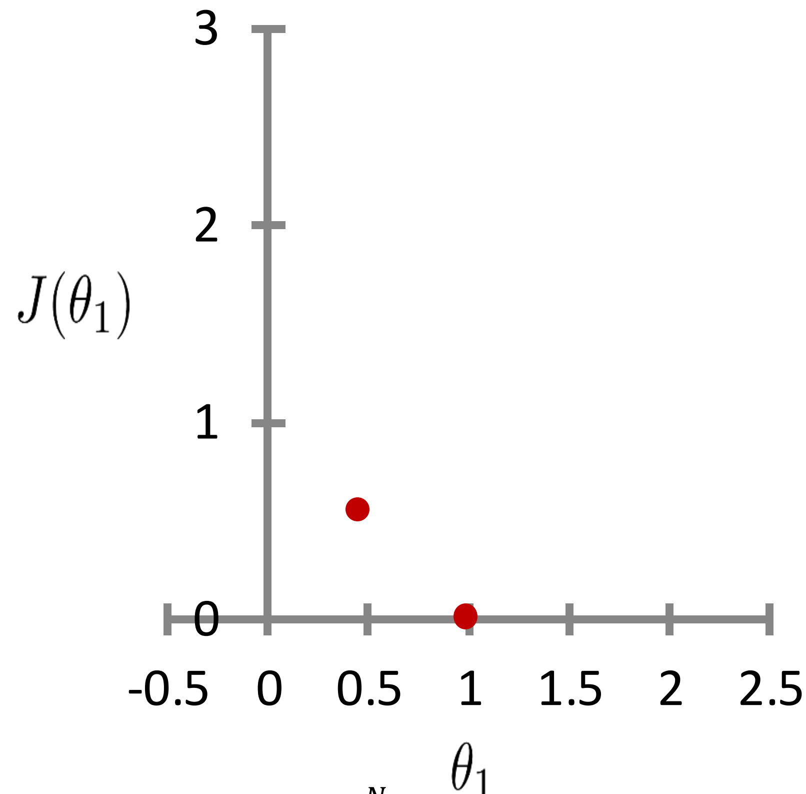
(for fixed θ_1, this is a function of x)



$$f_\theta(x^i) = \theta_1 x^{(i)}$$

$J(\theta_1)$

(function of the parameter θ_1)

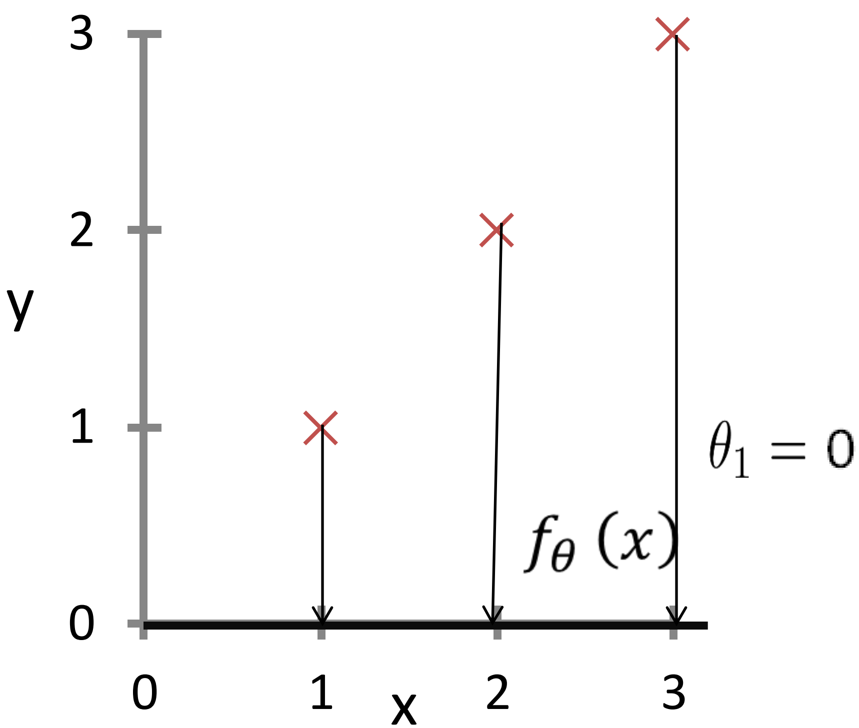


$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归
Linear regression with one variable

$f_\theta\,(x)$

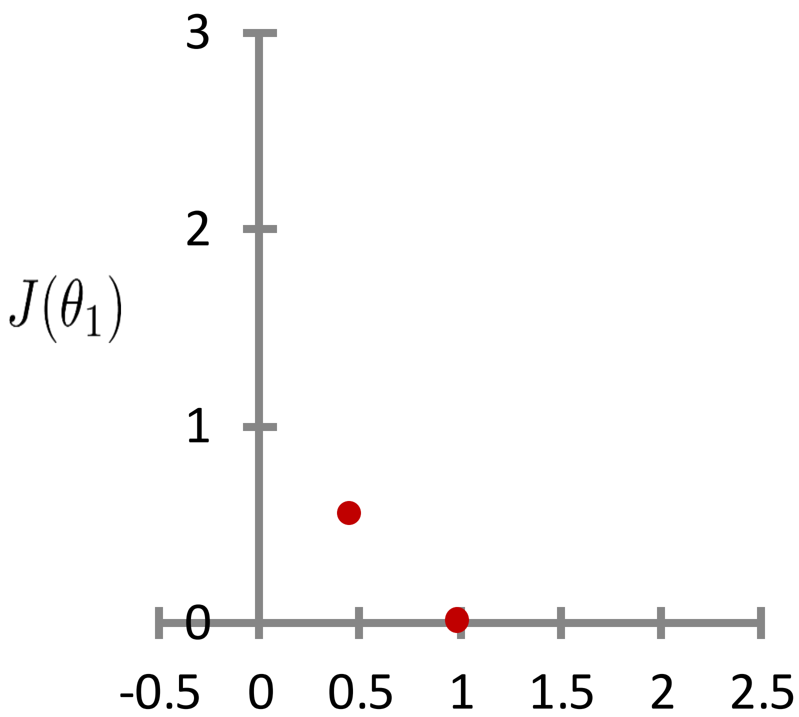
(for fixed θ_1, this is a function of x)



$$f_\theta\left(x^i\right) = \theta_1 x^{(i)}$$

$J(\theta_1)$

(function of the parameter θ_1)

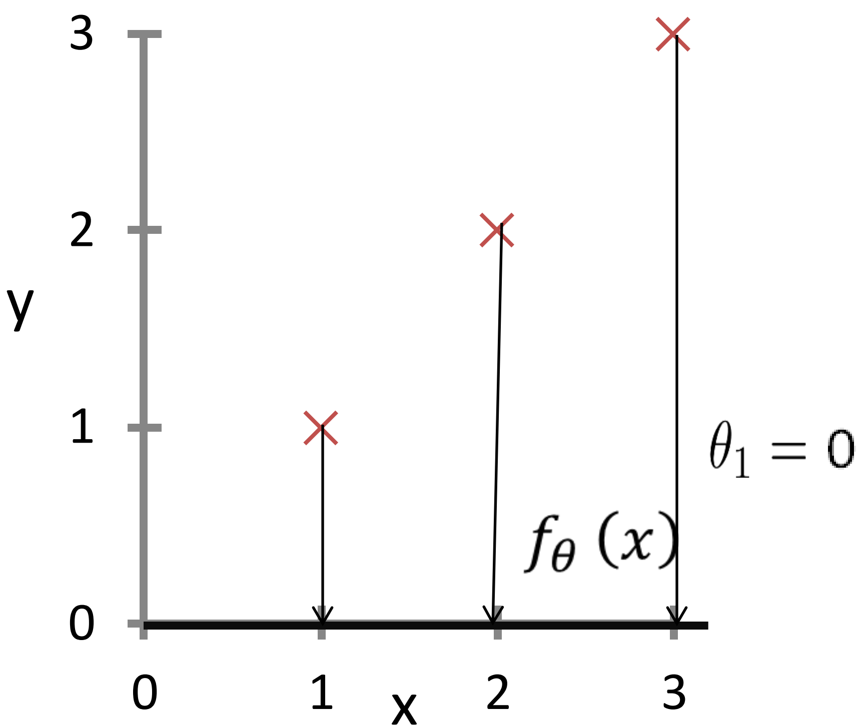


$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta\left(x^{(i)}\right) - y^{(i)})^2$$

单变量线性回归
Linear regression with one variable

$f_\theta(x)$

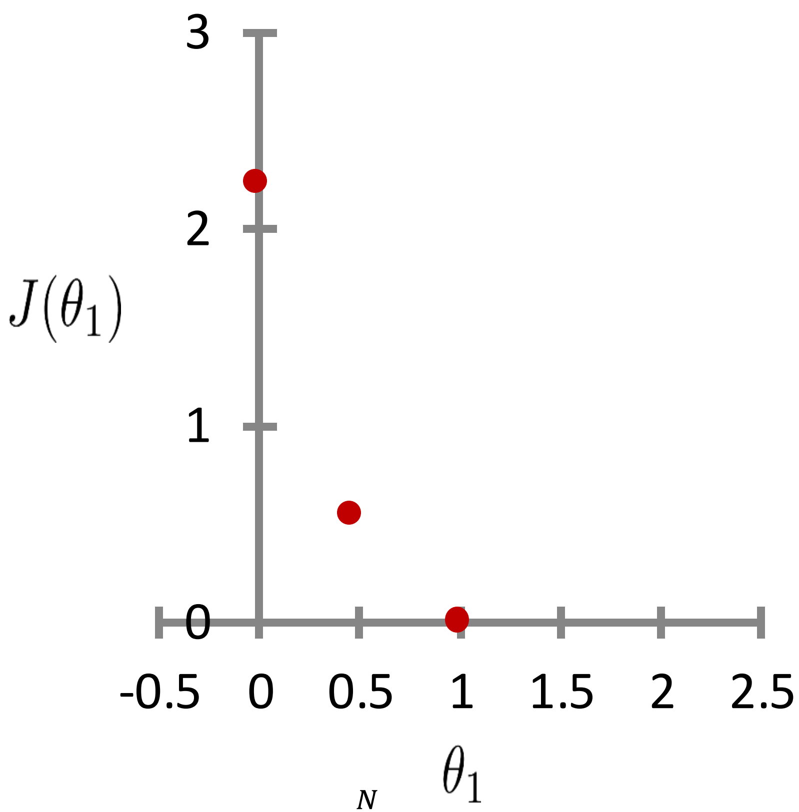
(for fixed θ_1, this is a function of x)



$$f_\theta(x^i) = \theta_1 x^{(i)}$$

$J(\theta_1)$

(function of the parameter θ_1)

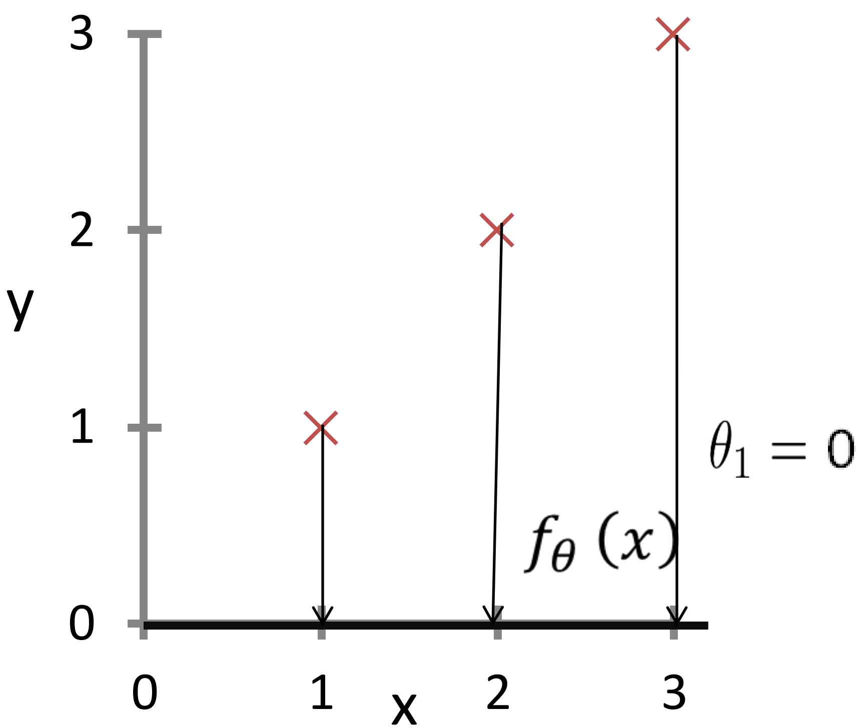


$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归
Linear regression with one variable

$f_\theta(x)$

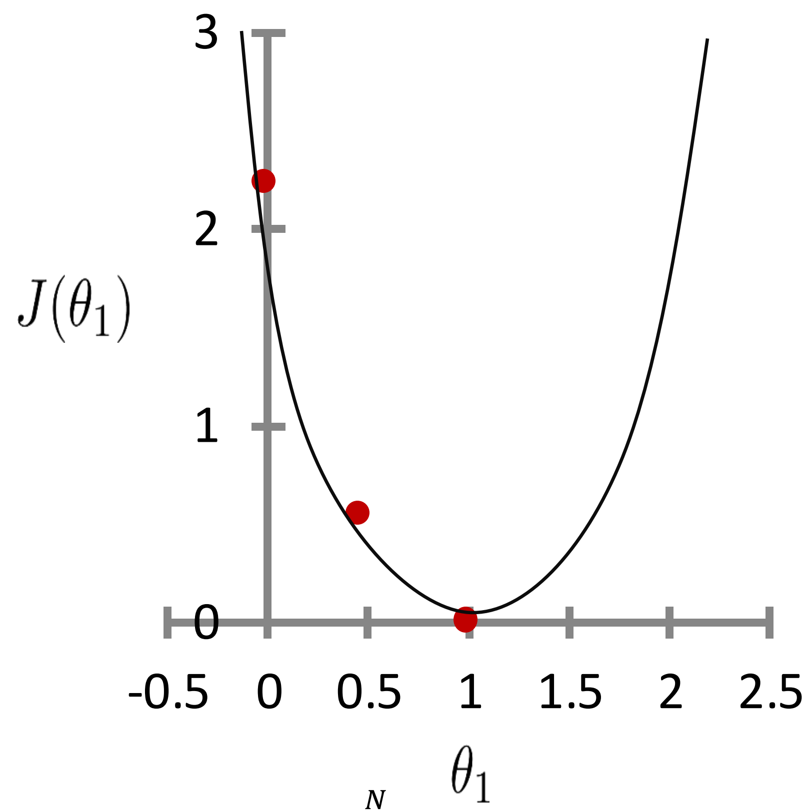
(for fixed θ_1, this is a function of x)



$$f_\theta(x^i) = \theta_1 x^{(i)}$$

$J(\theta_1)$

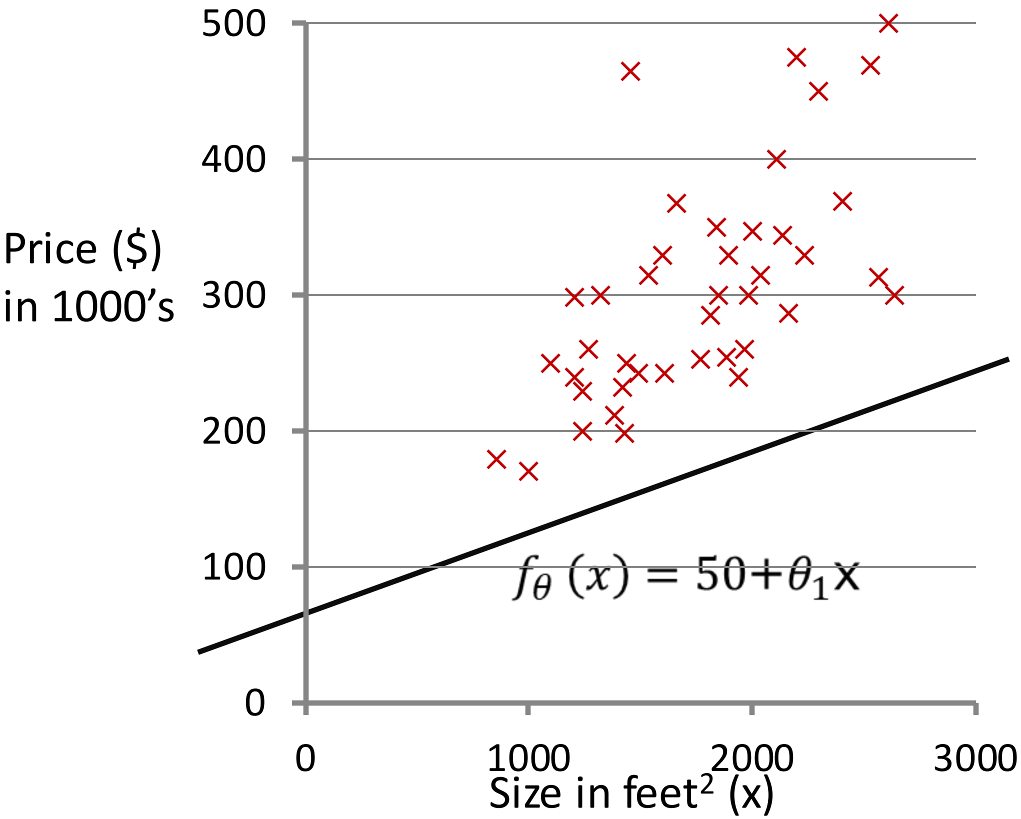
(function of the parameter θ_1)



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

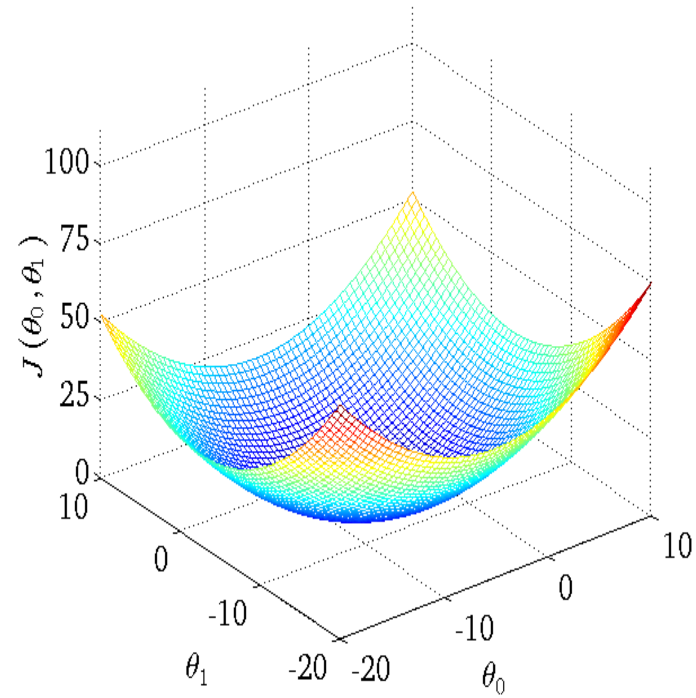
单变量线性回归
Linear regression with one variable

$f_\theta\ (x)$



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

$J(\theta_0, \theta_1)$

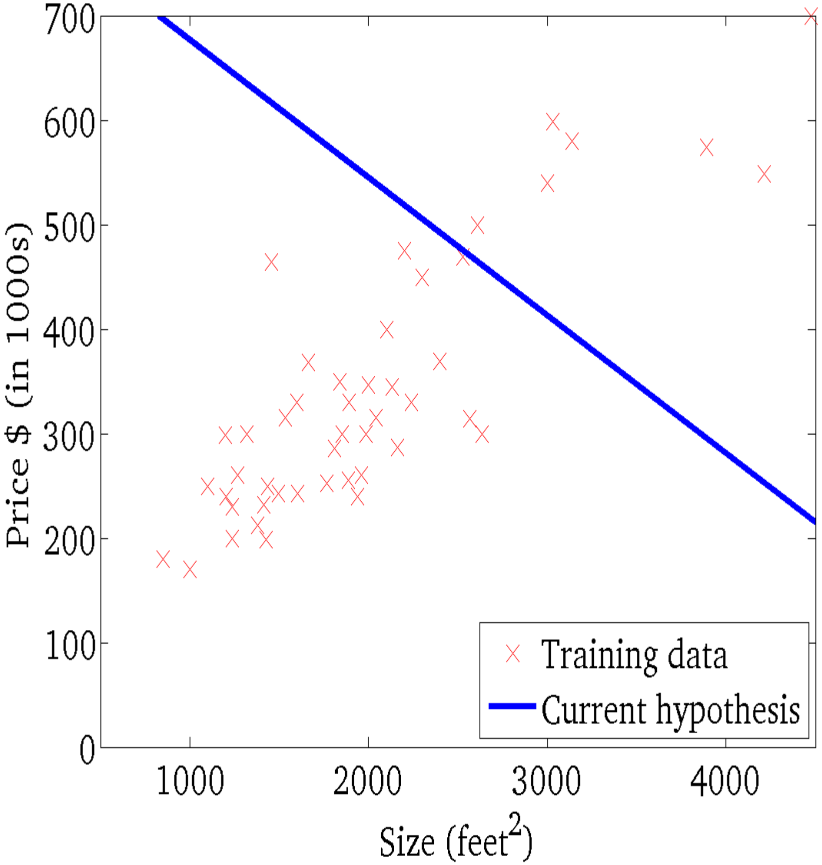


$$J(\theta_0, \theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归
Linear regression with one variable

$f_\theta(x)$

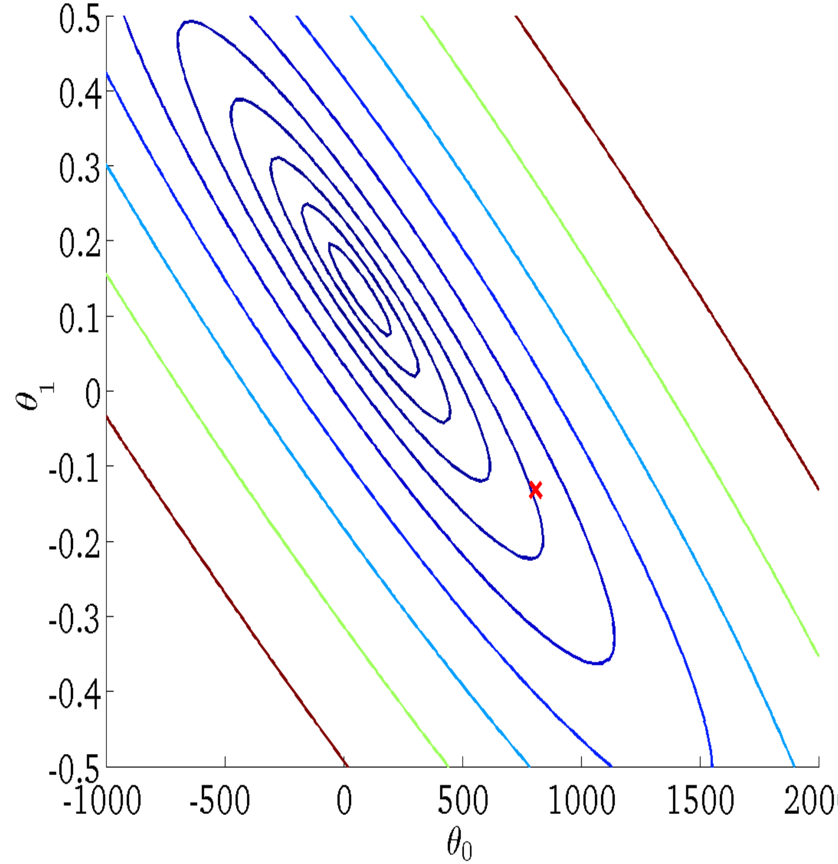
(for fixed θ_0, θ_1, this is a function of x)



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

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)

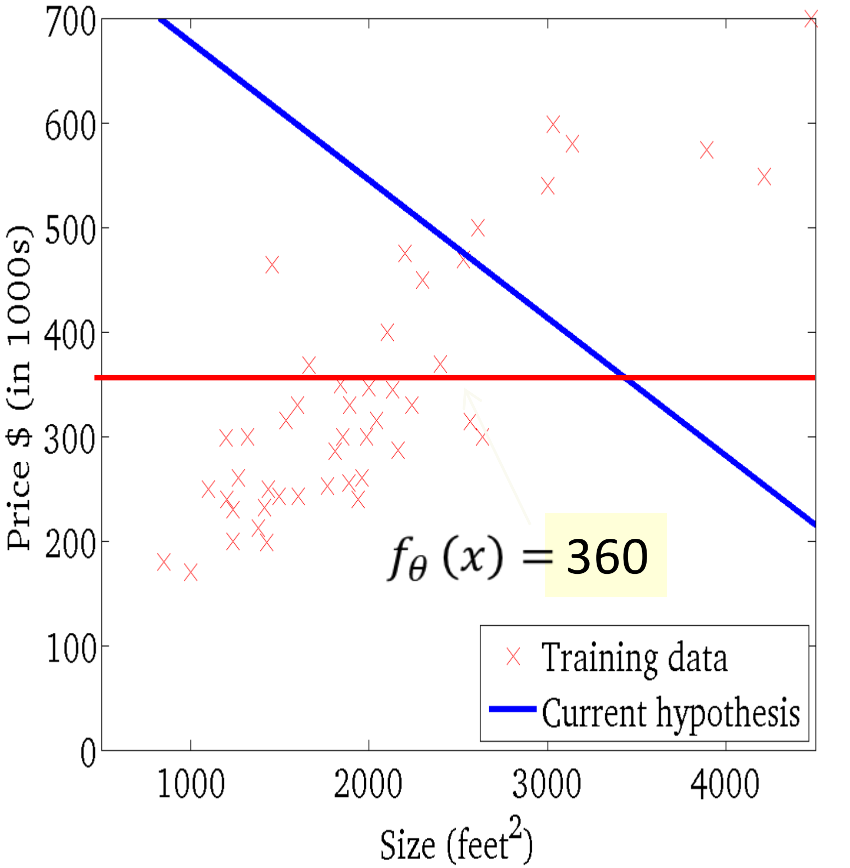


$$J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{i=1}^{N} \left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

单变量线性回归
Linear regression with one variable

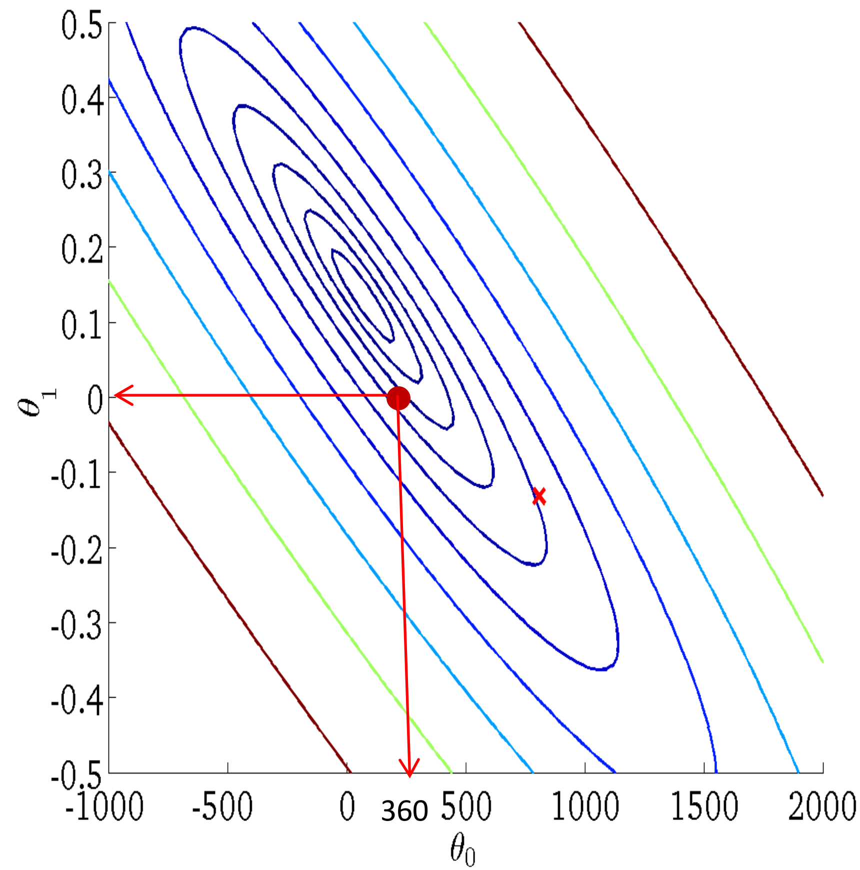
$f_\theta(x)$

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



$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



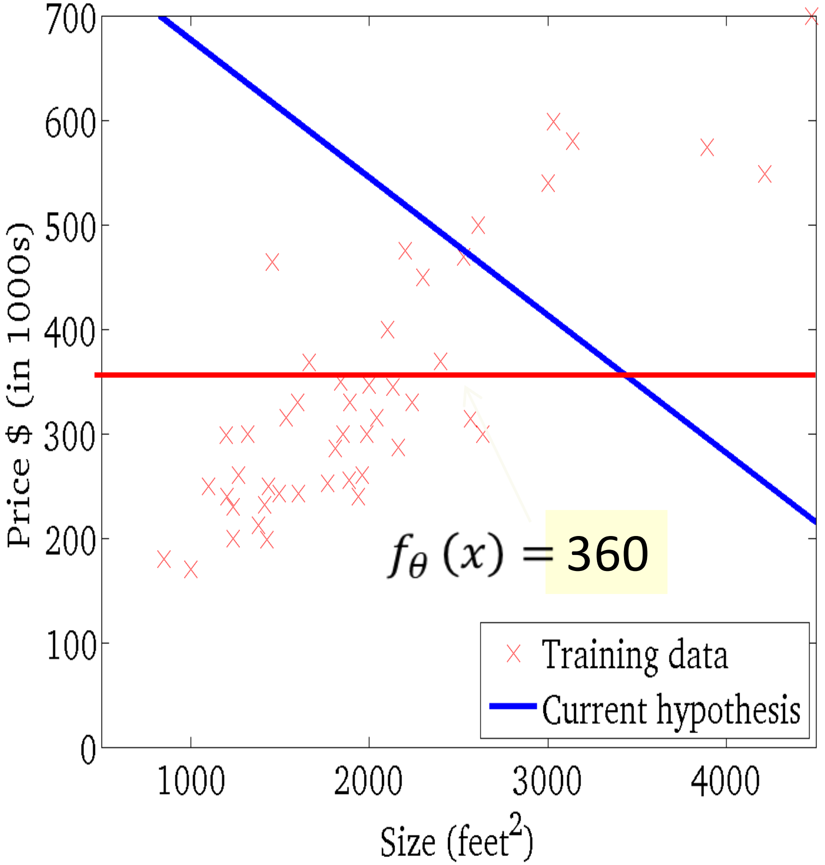
$$f_\theta(x) = \theta_0 + \theta_1 x$$

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

单变量线性回归
Linear regression with one variable

$f_\theta(x)$

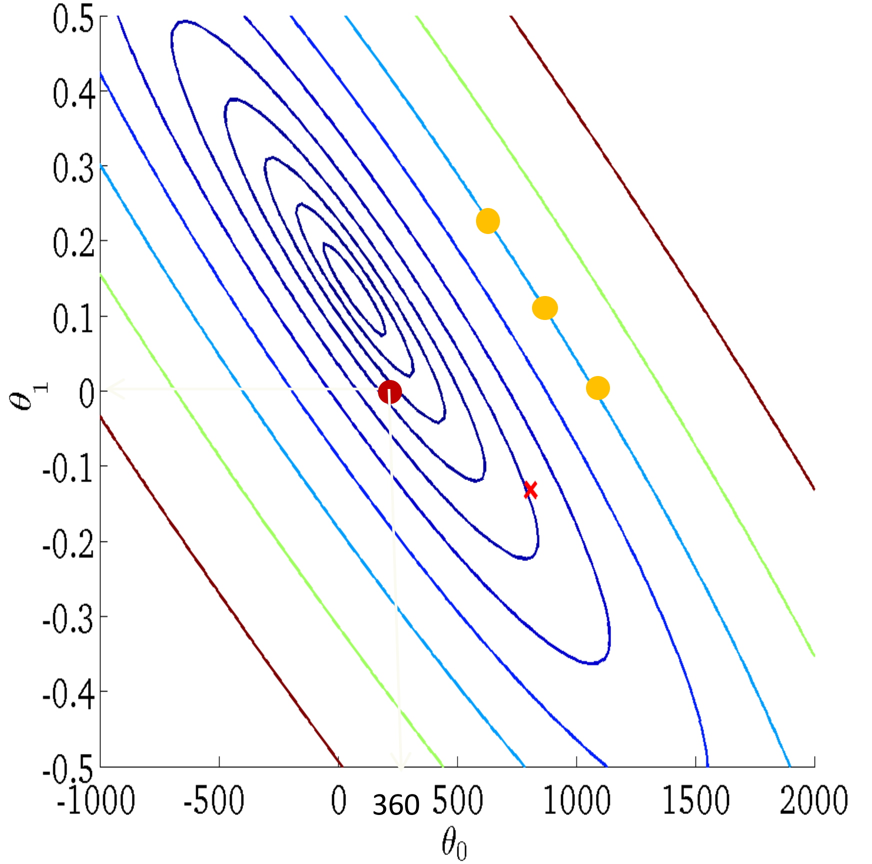
(for fixed θ_0, θ_1, this is a function of x)



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

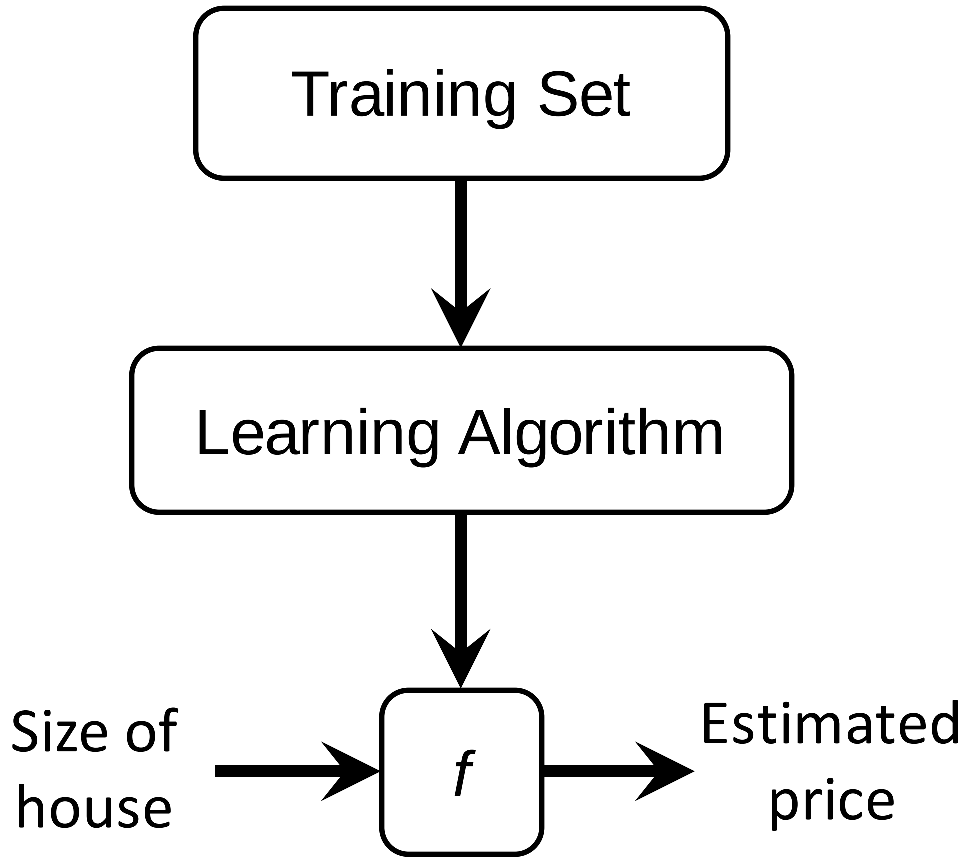
$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



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

单变量线性回归
Linear regression with one variable



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

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

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

Goal:

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

梯度下降法
Gradient descent algorithm

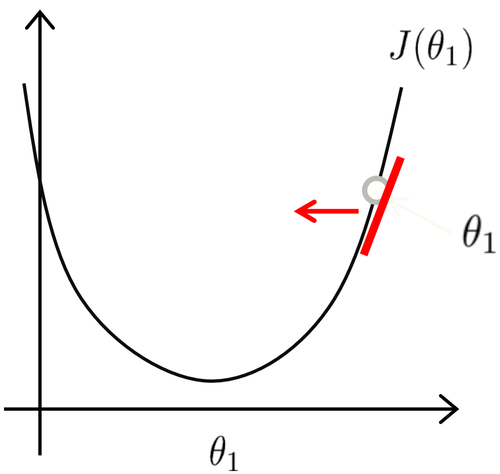
repeat until convergence {

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

}

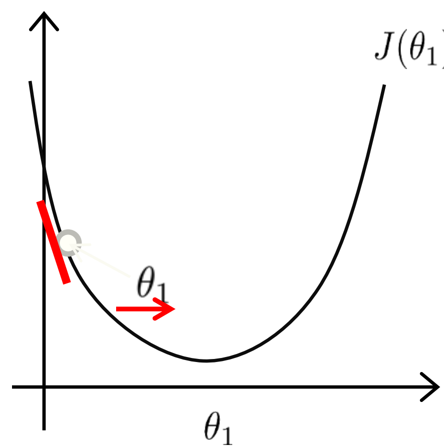
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} \underline{J(\theta_0, \theta_1)}$$

≥ 0



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

≤ 0



梯度下降法
Gradient descent algorithm

repeat until convergence {

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

(simultaneously update $j = 0$ and $j = 1$)

}

梯度下降法
Gradient descent algorithm

repeat until convergence {

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

(simultaneously update $j = 0$ and $j = 1$)

}

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

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

梯度下降法
Gradient descent algorithm

repeat until convergence {

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

}

Correct: Simultaneous update

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

Incorrect:

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

梯度下降法
Gradient descent algorithm

Gradient descent algorithm

repeat until convergence {

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

(for $j = 1$ and $j = 0$)

}

Linear Regression Model

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

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

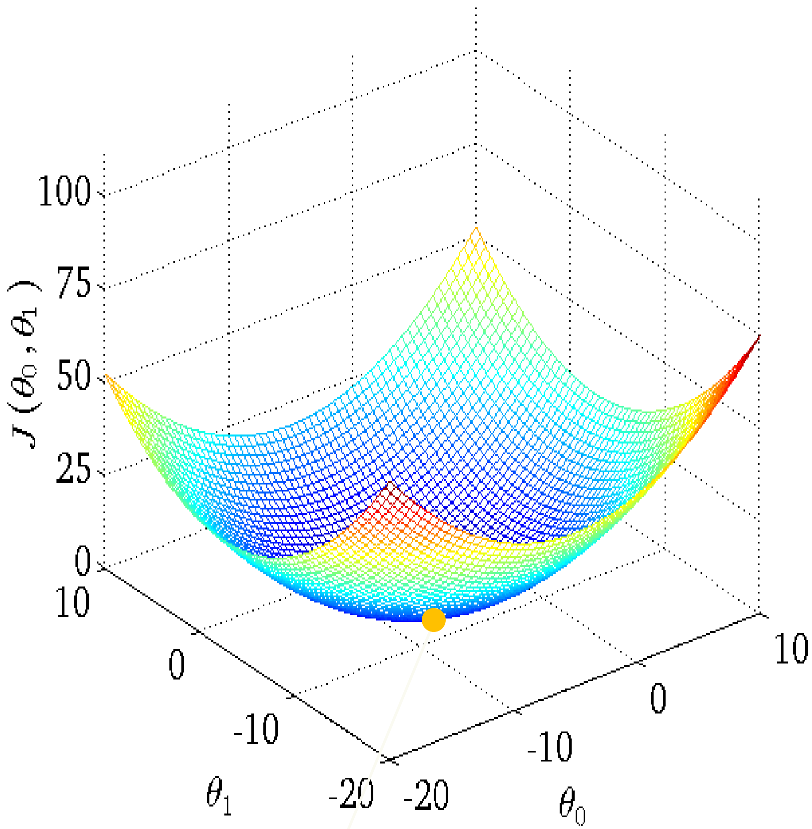
Repeat until convergece

$$\theta_0 := \theta_0 - a \frac{1}{N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)})$$

$$\theta_1 := \theta_1 - a \frac{1}{N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$

update θ_0 and θ_1 simultaneously

凸函数
Bowled shape Convex Function

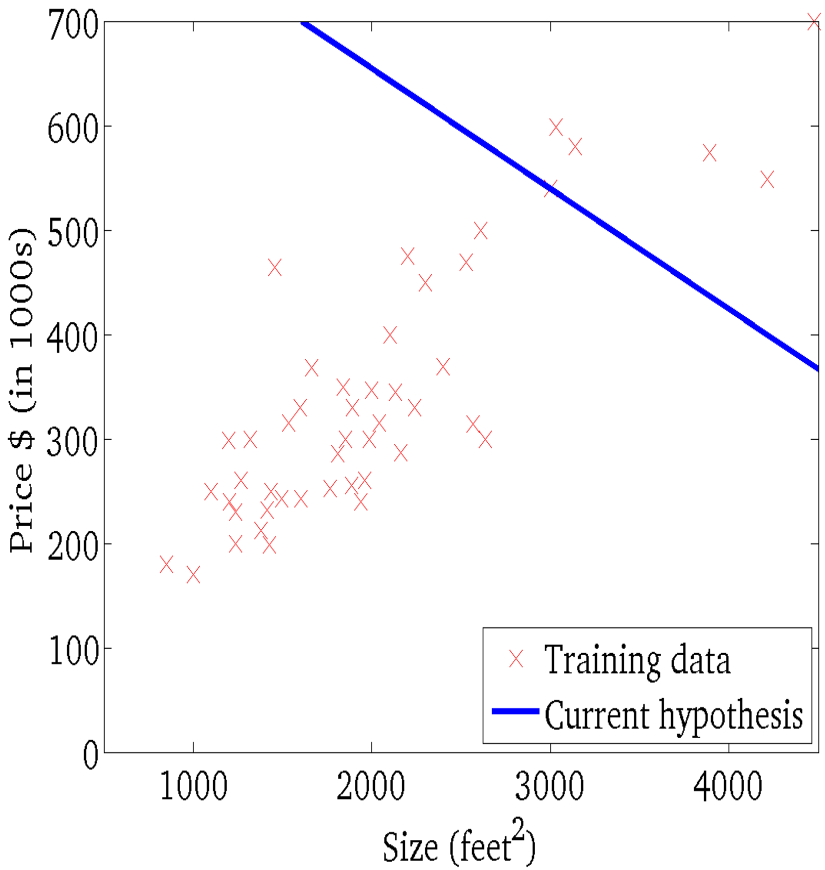


$$J(\theta_0, \theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta\left(x^{(i)}\right) - y^{(i)})^2$$

Unique Minimum

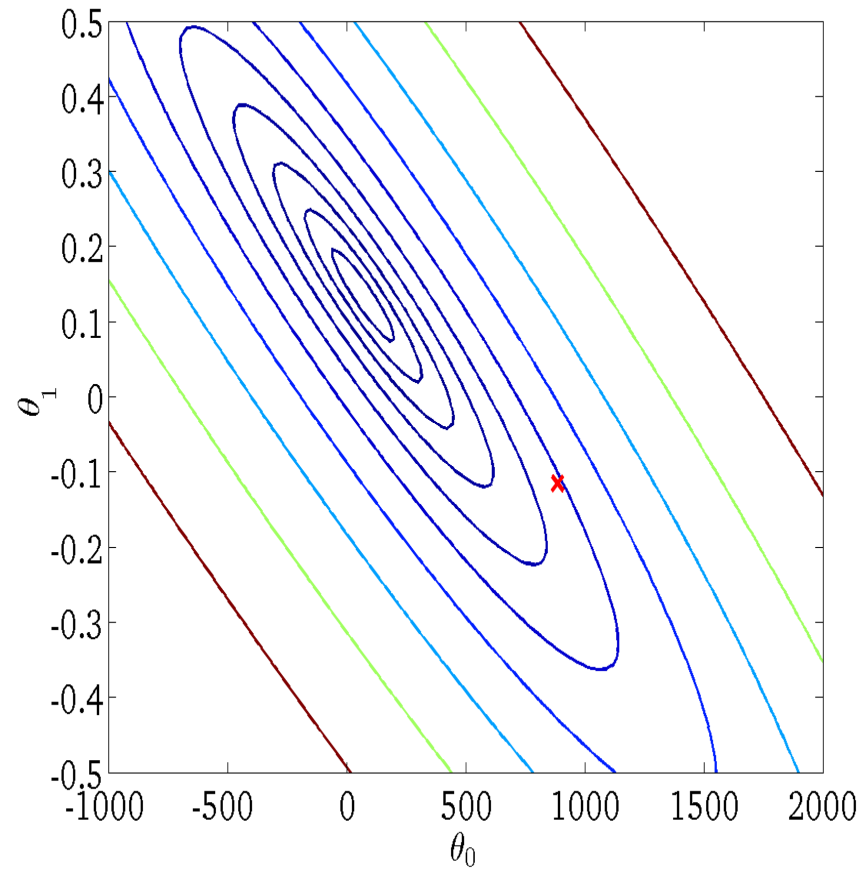
Different initial lead to the same optimum

搜索过程
Search Procedure



$f_{\theta}(x)$

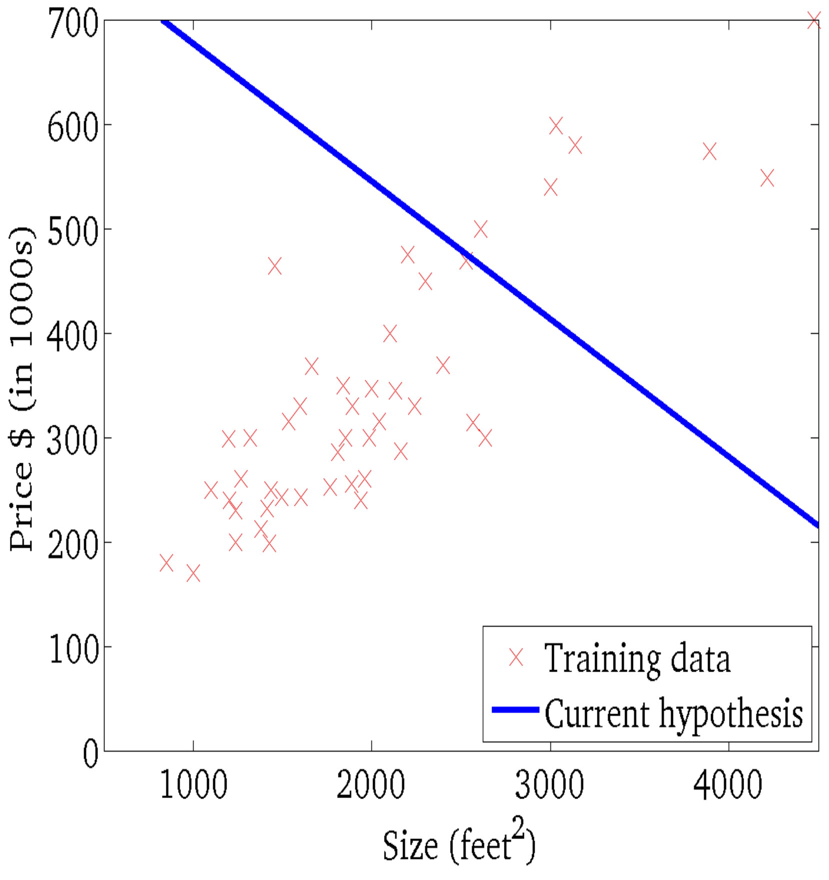
(for fixed θ_0, θ_1, this is a function of x)

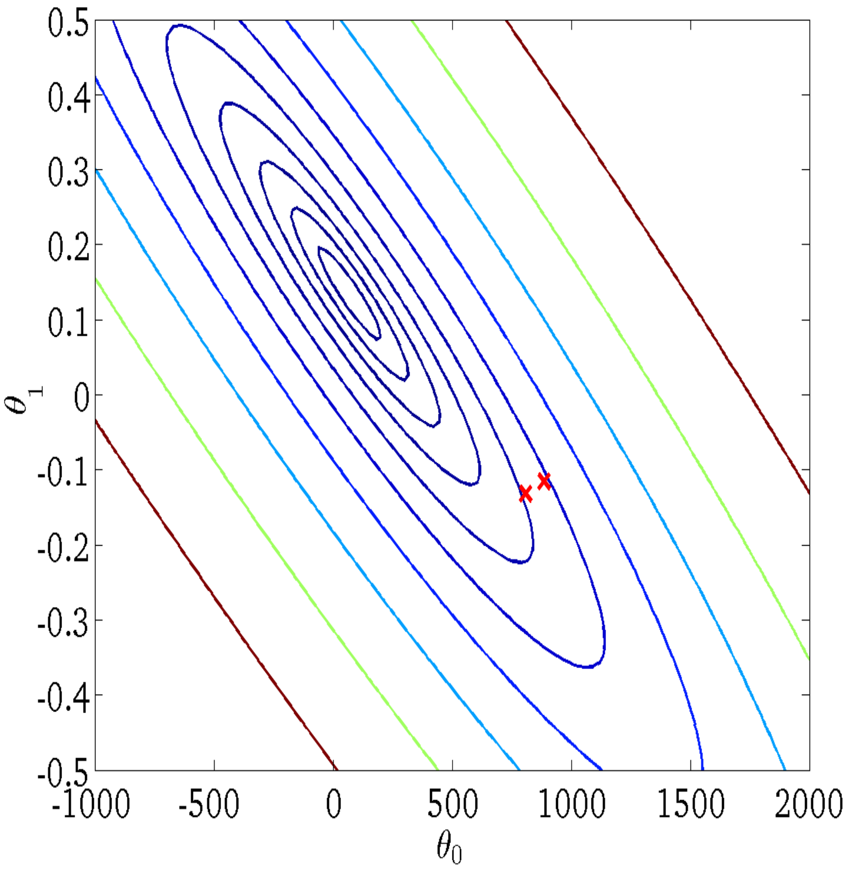


$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure





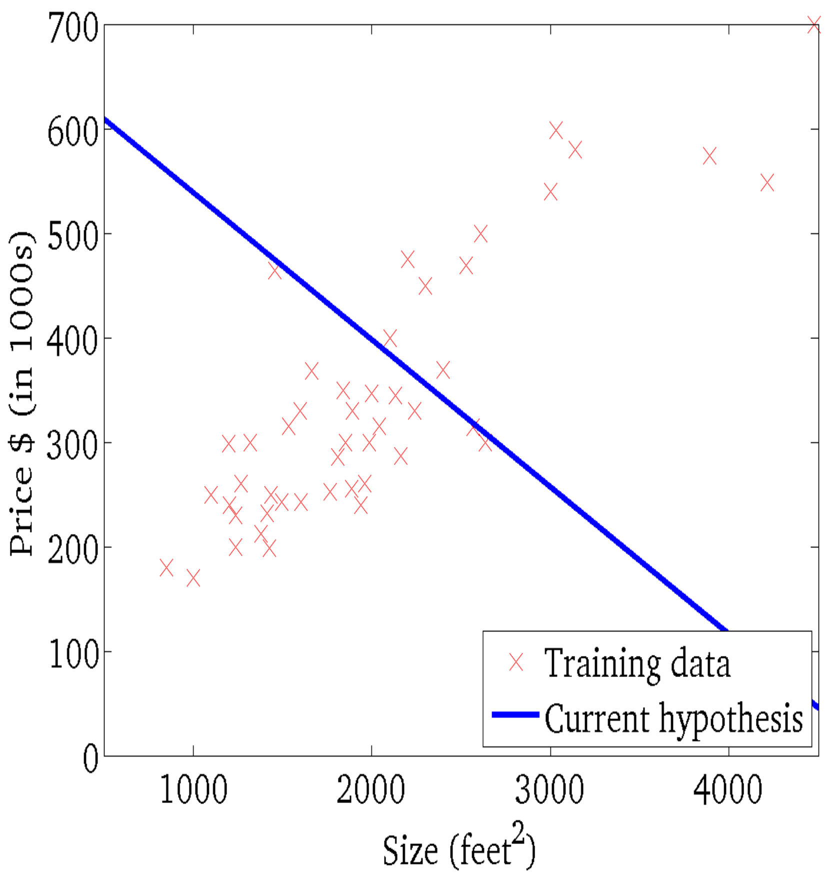
$$f_\theta(x)$$

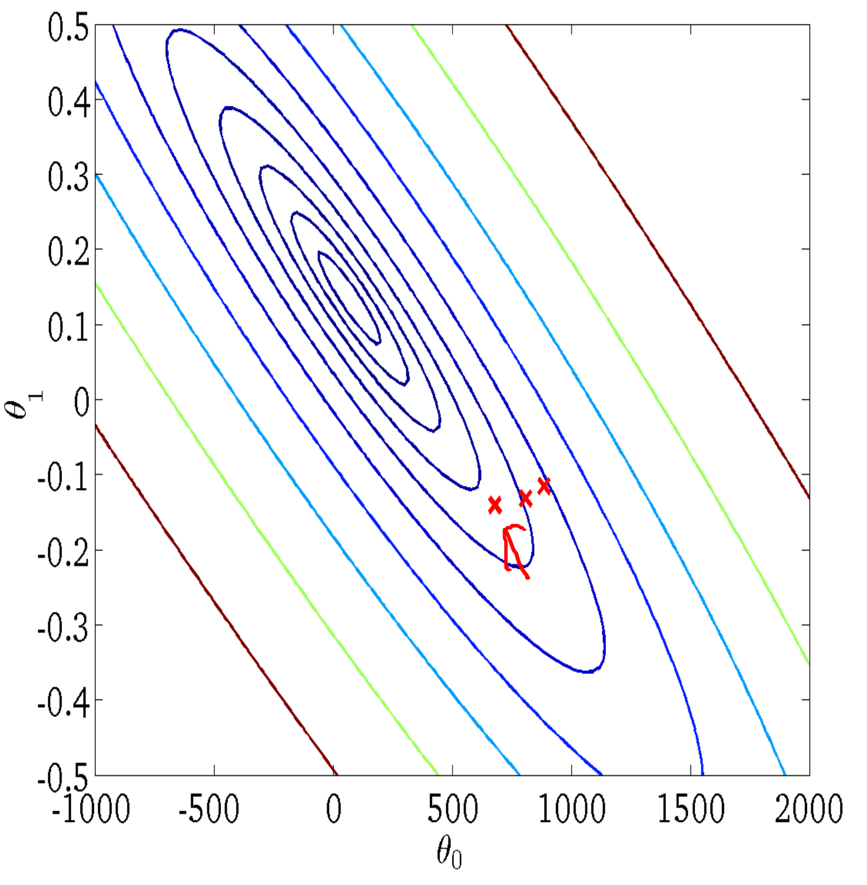
(for fixed θ_0, θ_1, this is a function of x)

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

(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure





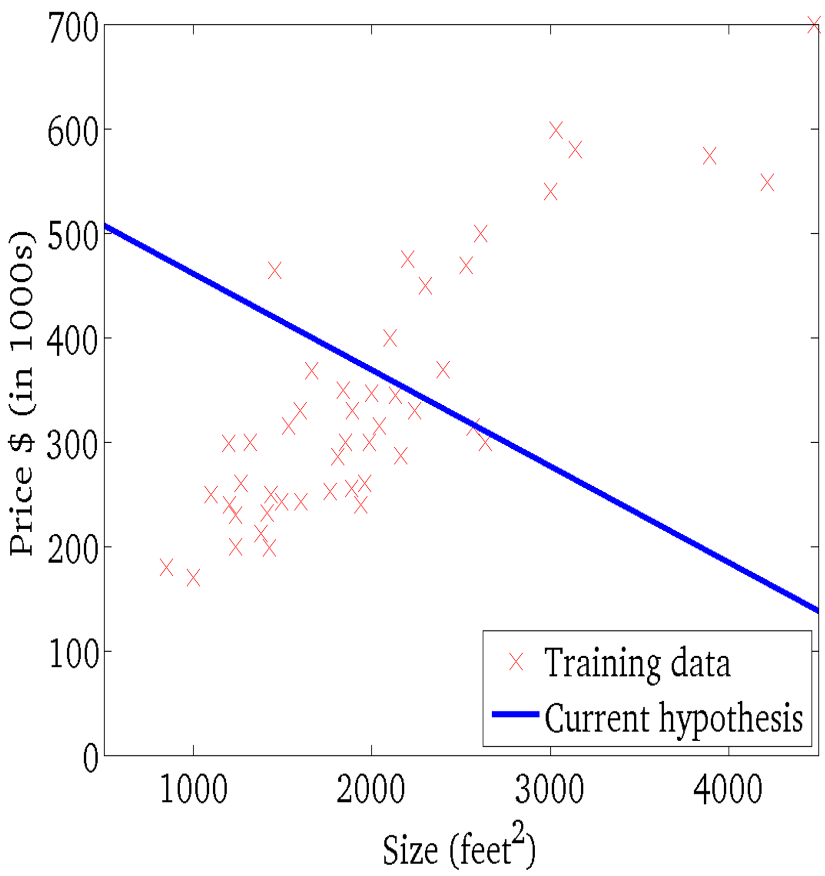
$f_\theta(x)$

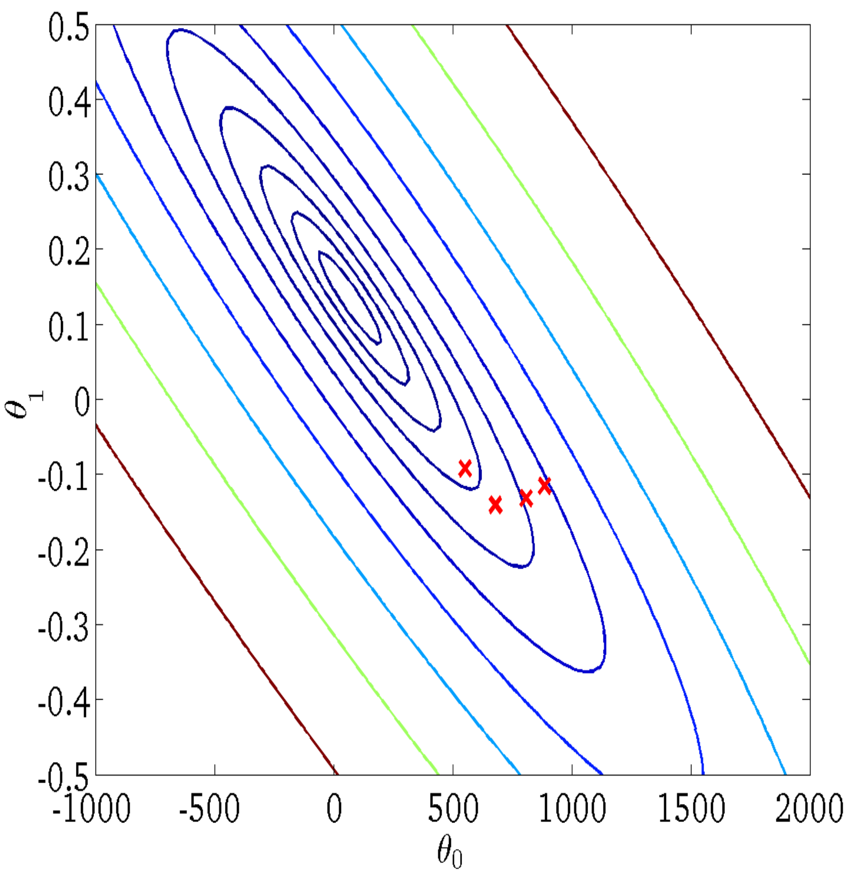
(for fixed θ_0, θ_1 , this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure





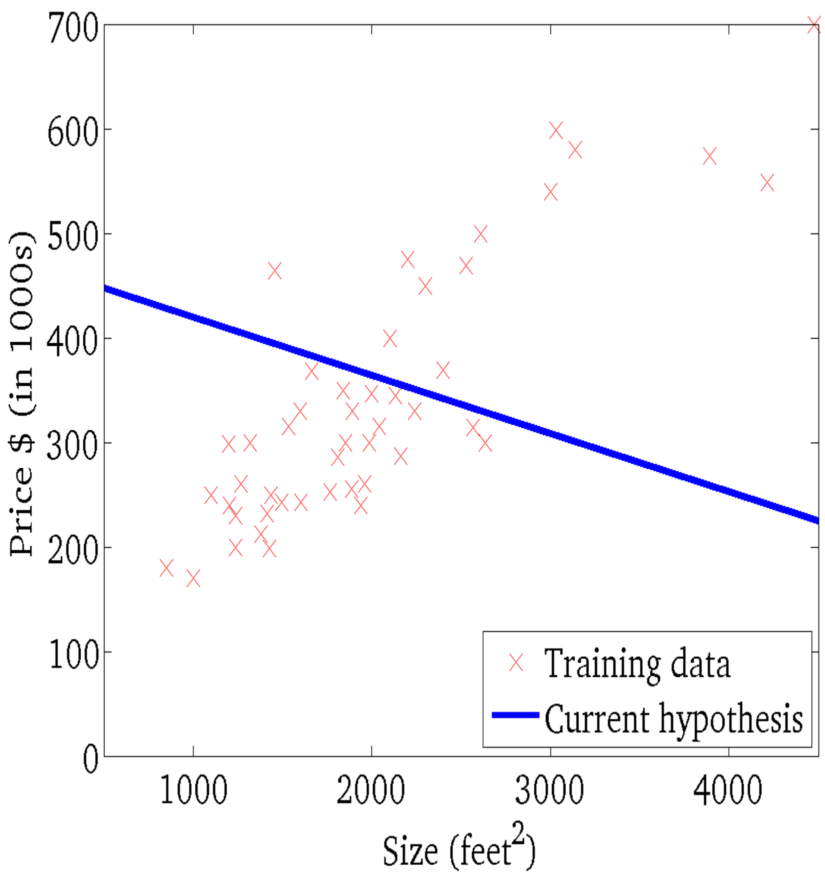
$$f_\theta(x)$$

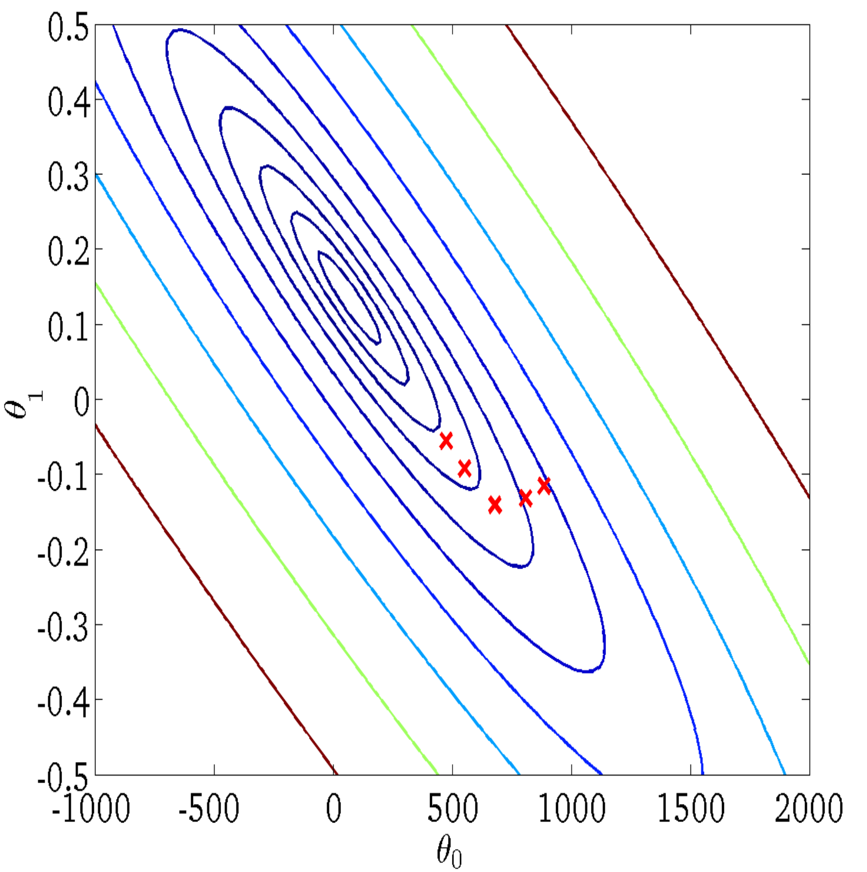
(for fixed θ_0, θ_1 , this is a function of x)

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

(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure





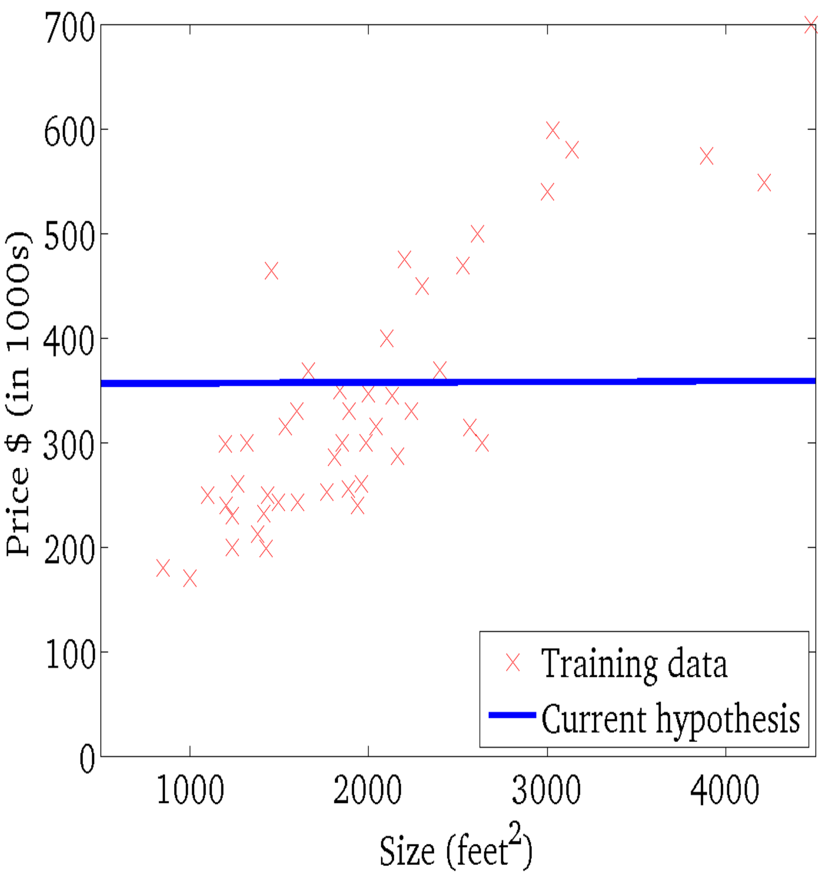
$f_\theta(x)$

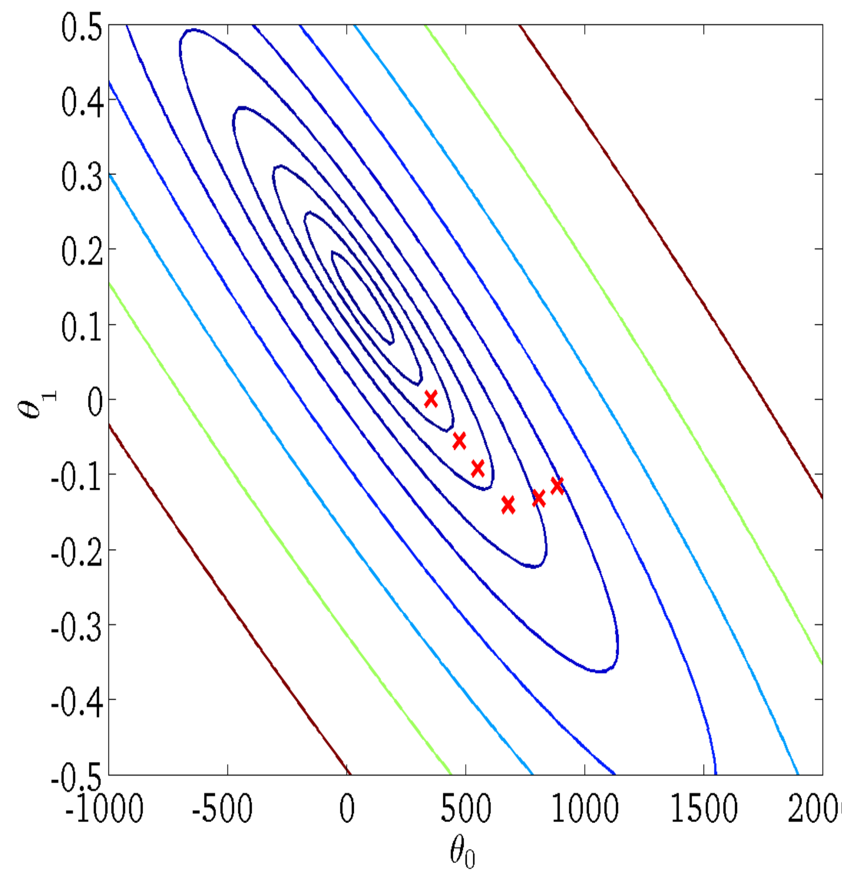
(for fixed θ_0, θ_1 , this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure





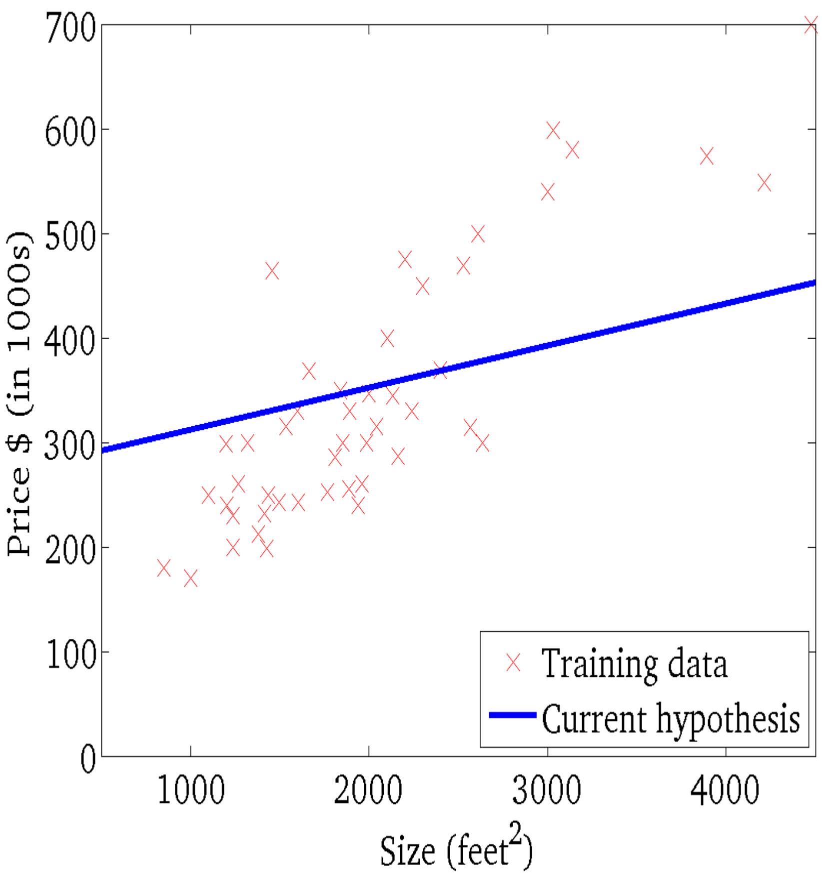
$f_\theta(x)$

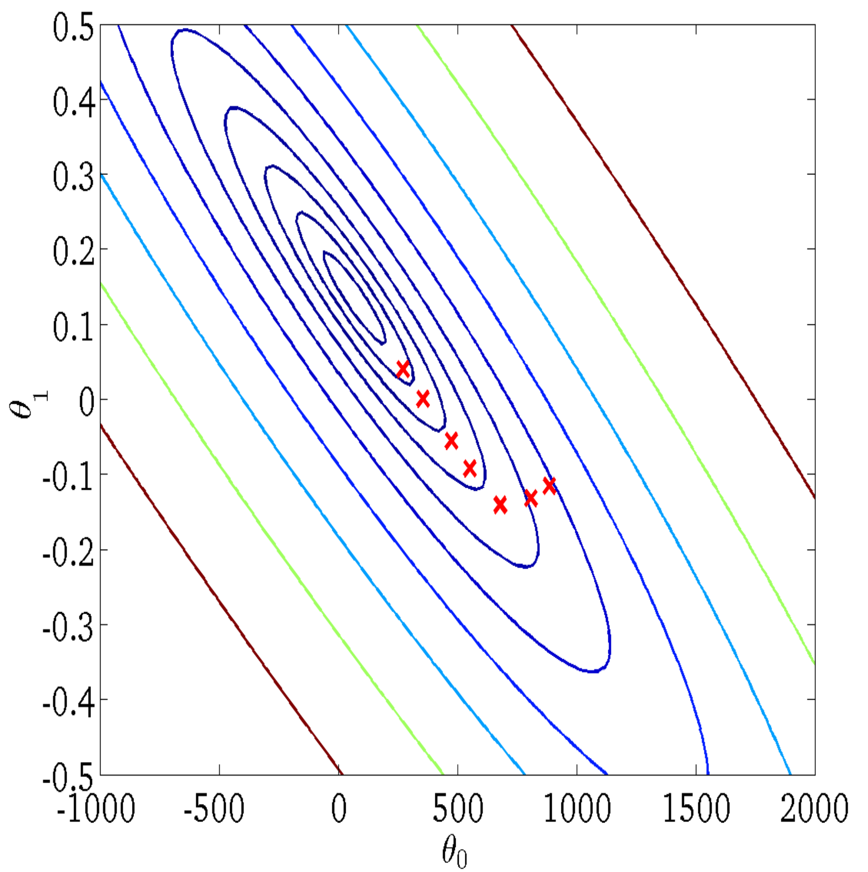
(for fixed θ_0, θ_1 , this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure





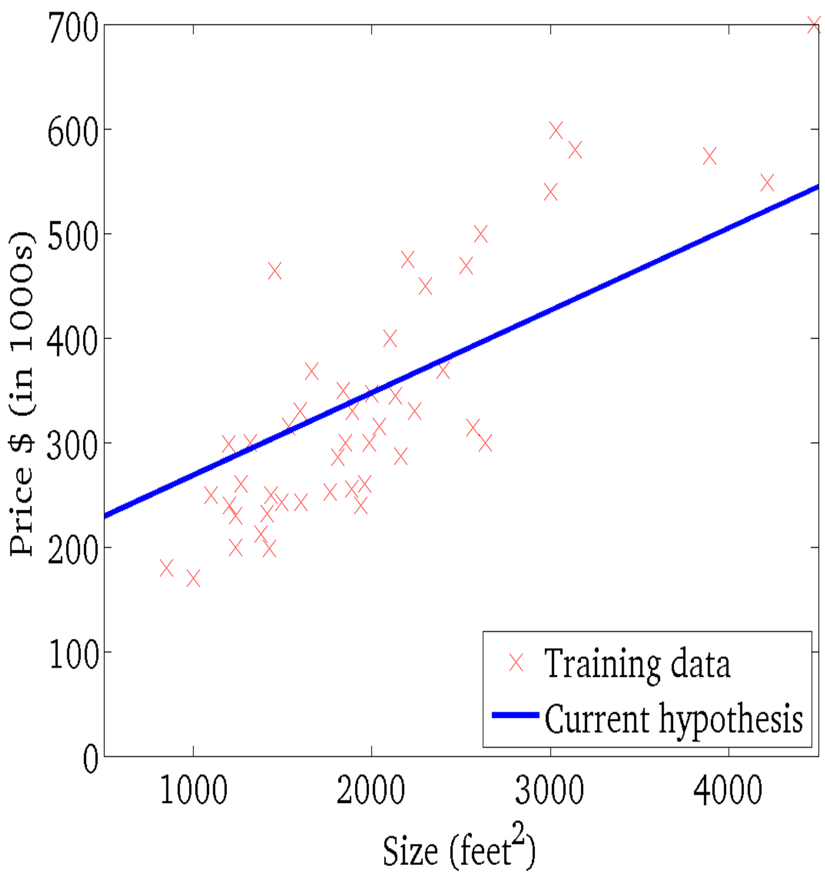
$f_\theta(x)$

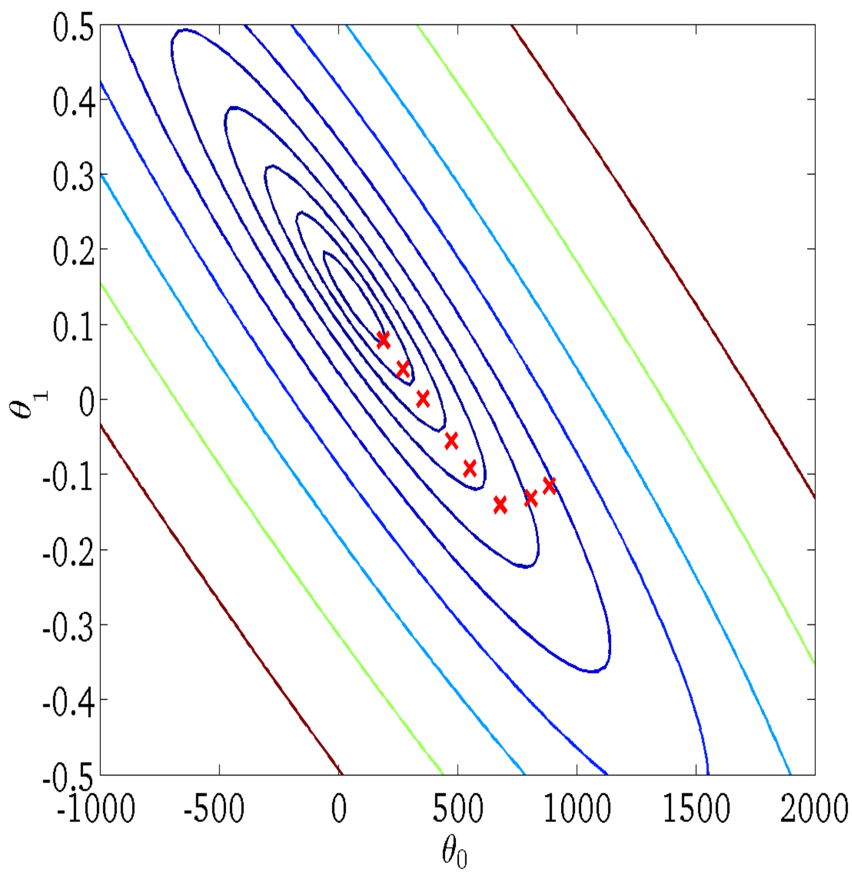
(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure





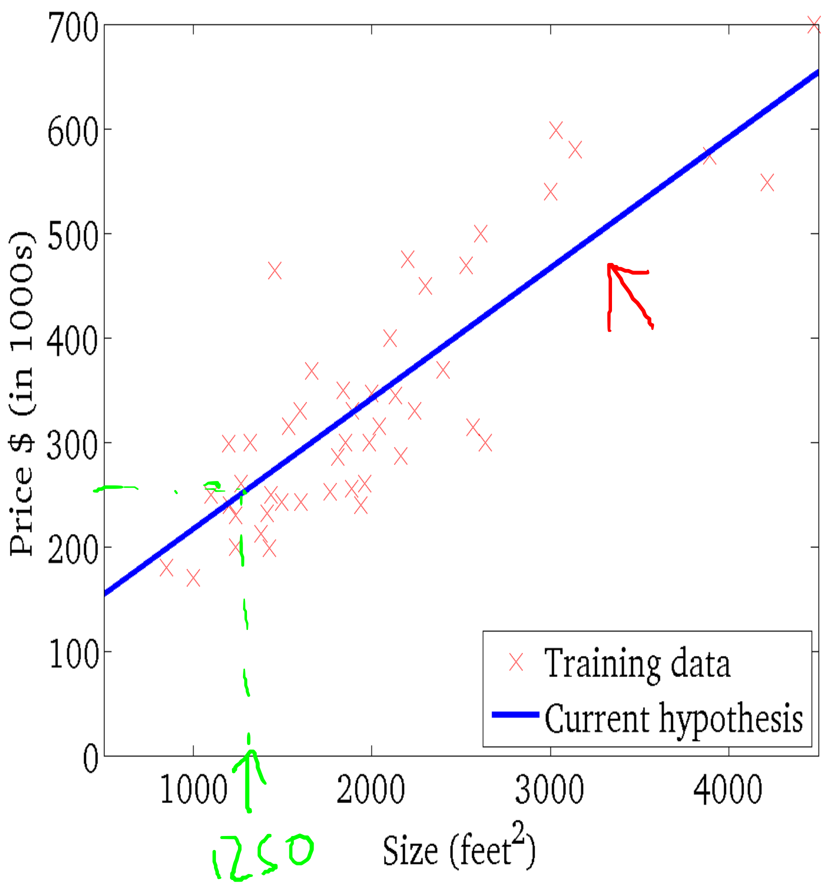
$$f_\theta(x)$$

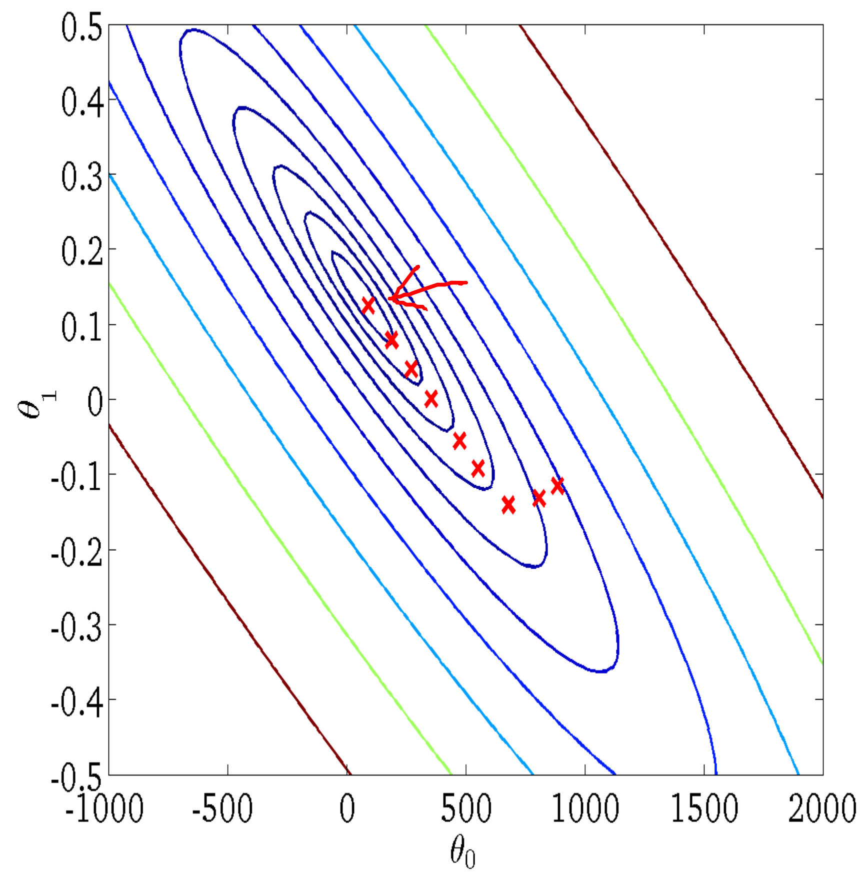
(for fixed θ_0, θ_1 , this is a function of x)

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

(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure





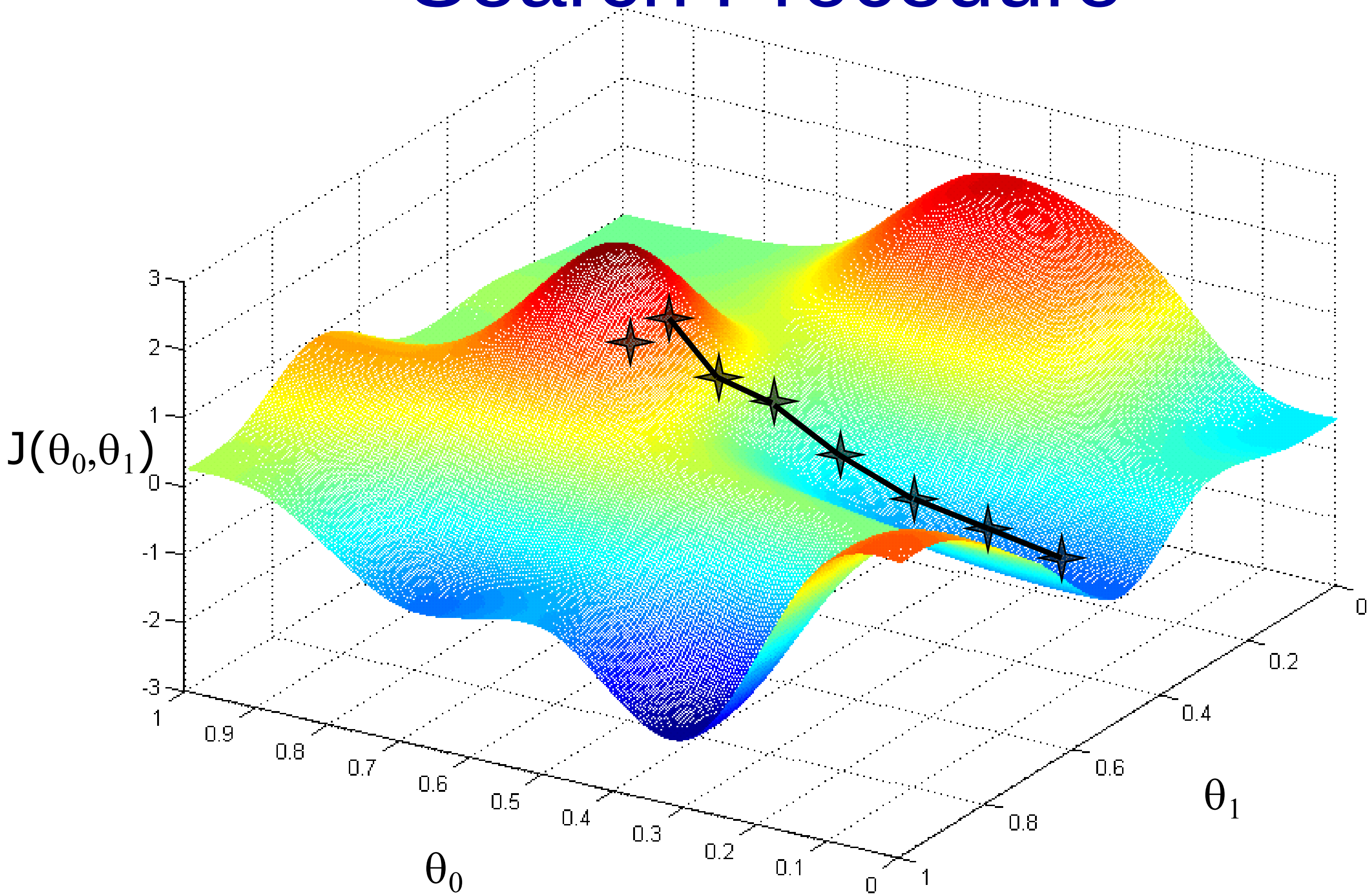
$f_\theta(x)$

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

$J(\theta_0, \theta_1)$

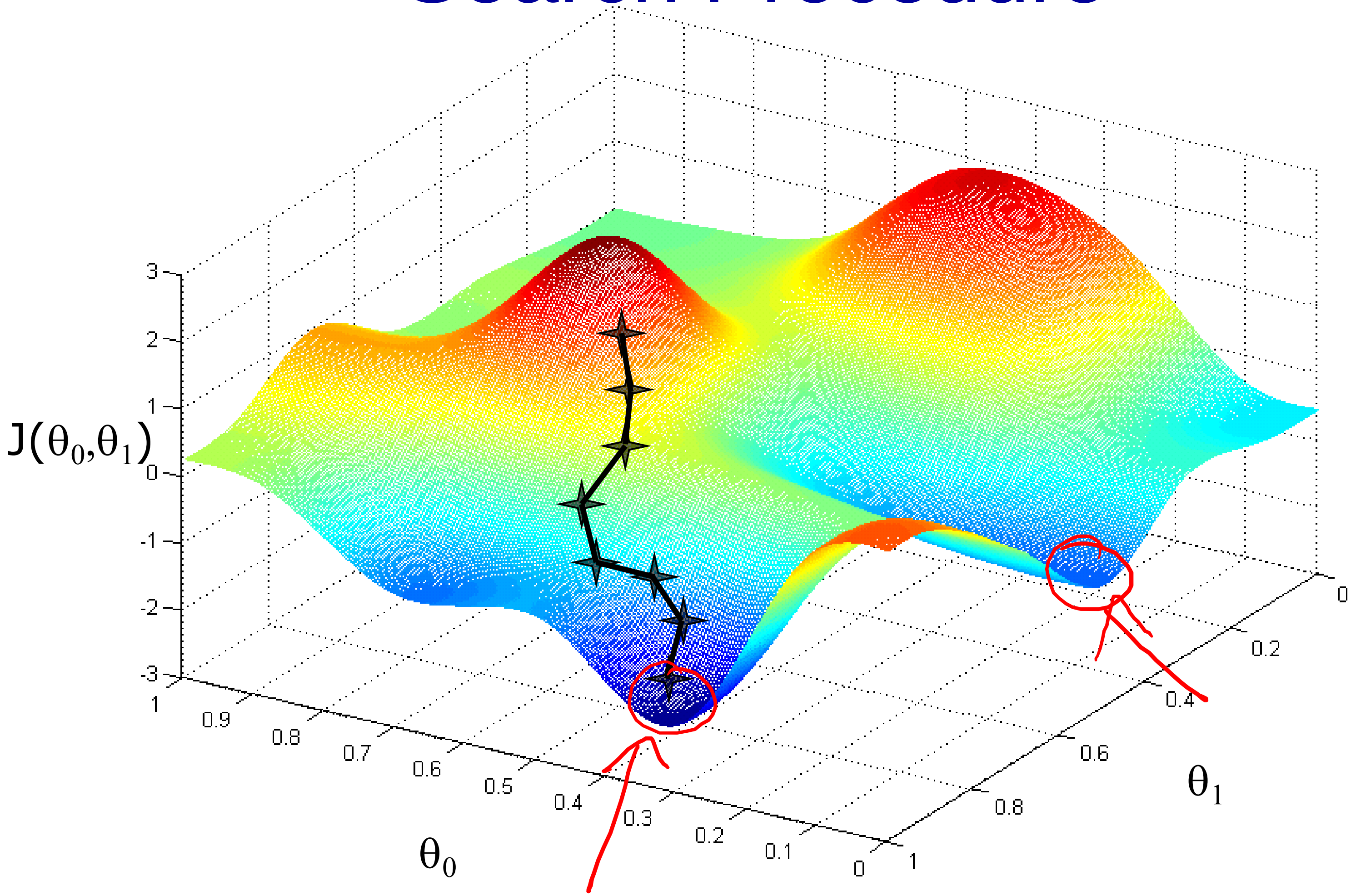
(function of the parameters θ_0, θ_1)

搜索过程
Search Procedure



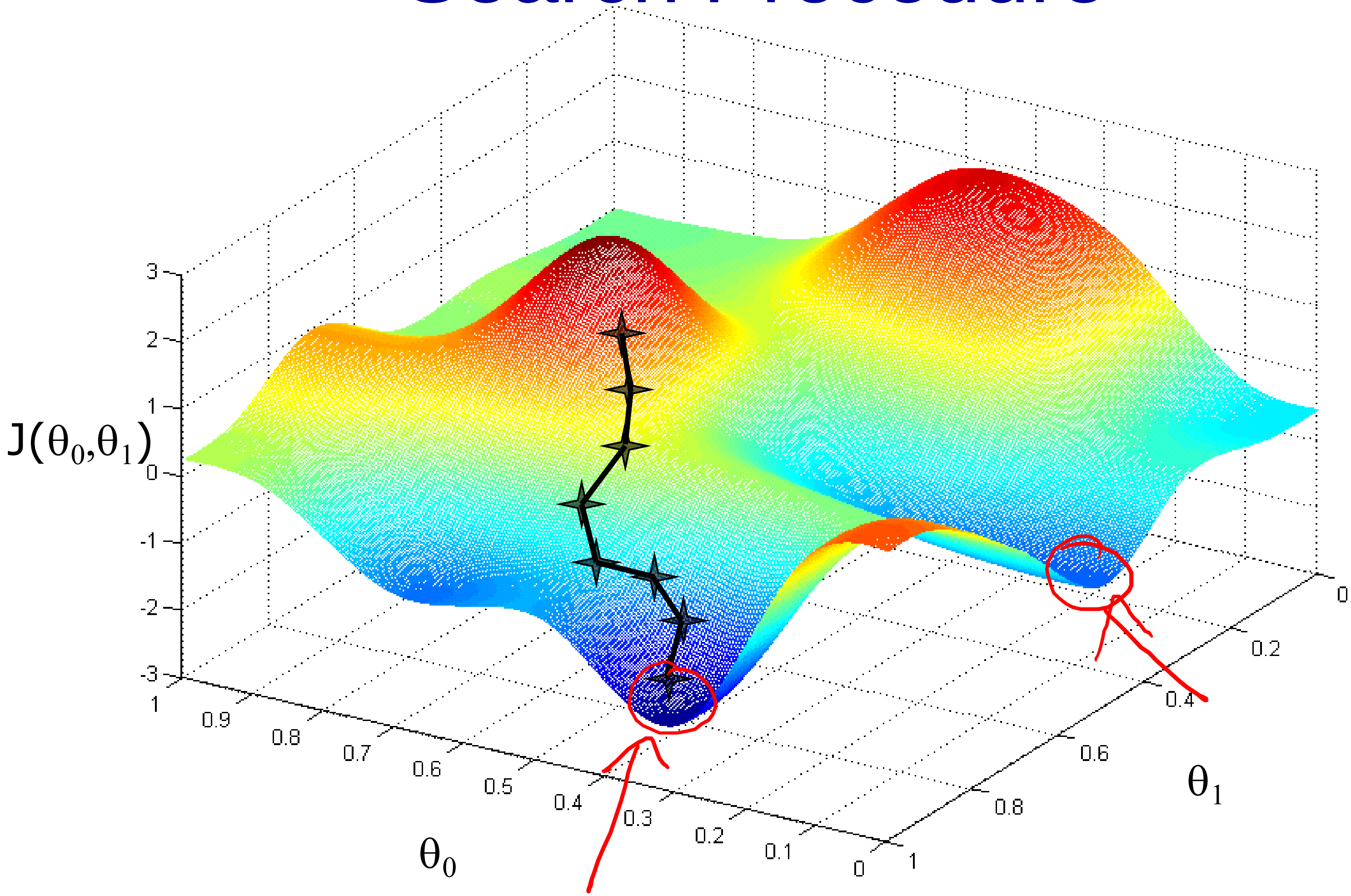
- Choose an initial value for θ
- Update θ iteratively with the data
- Until we research a minimum

搜索过程
Search Procedure



- Choose a new initial value for θ
- Update θ iteratively with the data
- Until we research a minimum

搜索过程
Search Procedure



- Choose a new initial value for θ
- Update θ iteratively with the data
- Until we research a minimum

In linear regression, the loss function L is convex. Different initial lead to the same optimum.

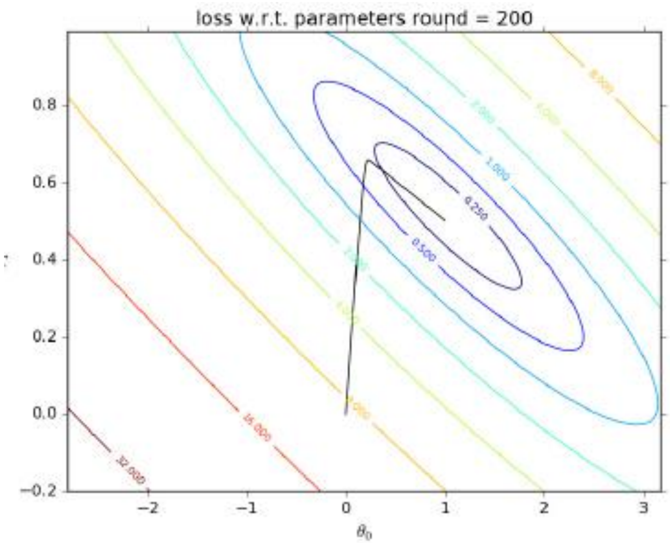
批量梯度下降
Batch Gradient descent

"Batch": Each step of gradient descent uses all the training examples.

Repeat until convergence

$$\theta_0 := \theta_0 - a\frac{1}{N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - a\frac{1}{N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$



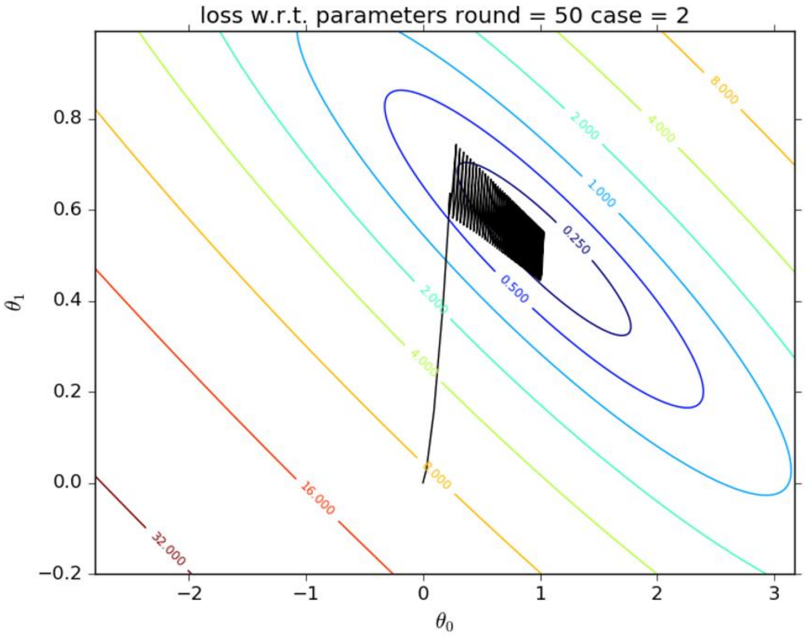
随机梯度下降
Stochastic Gradient descent

“stochastic”: Each step of gradient descent uses single training example.

Repeat until convergence

$$\theta_0 := \theta_0 - a\cancel{\frac{1}{N}\sum_{i=1}^{N}}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - a\cancel{\frac{1}{N}\sum_{i=1}^{N}}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$



Compare with BGD
- Faster learning
- Uncertainty or fluctuation in learning

小批量梯度下降
Mini-Batch Gradient descent

- A combination of batch GD and stochastic GD

- Split the whole dataset into *K* mini-batches

$$\{1, 2, 3, \ldots, K\}$$

- For each mini-batch *k*, perform one-step BGD toward

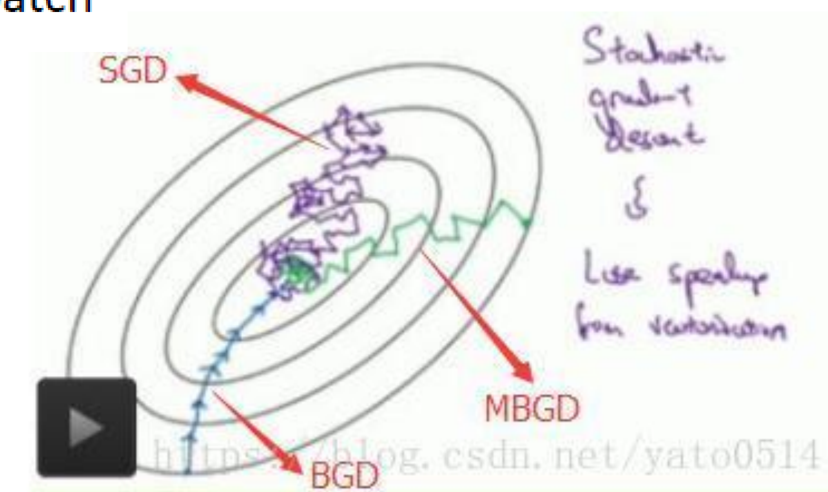
$$J^k(\theta) := \frac{1}{2N_k}\sum_{i=1}^{N_k}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- Good learning stability (BGD)
- Good convergence rate (SGD)

- Update $\theta_{\text{new}} = \theta_{\text{old}} - \eta \frac{\partial J^{(k)}(\theta)}{\partial \theta}$ for each mini-batch

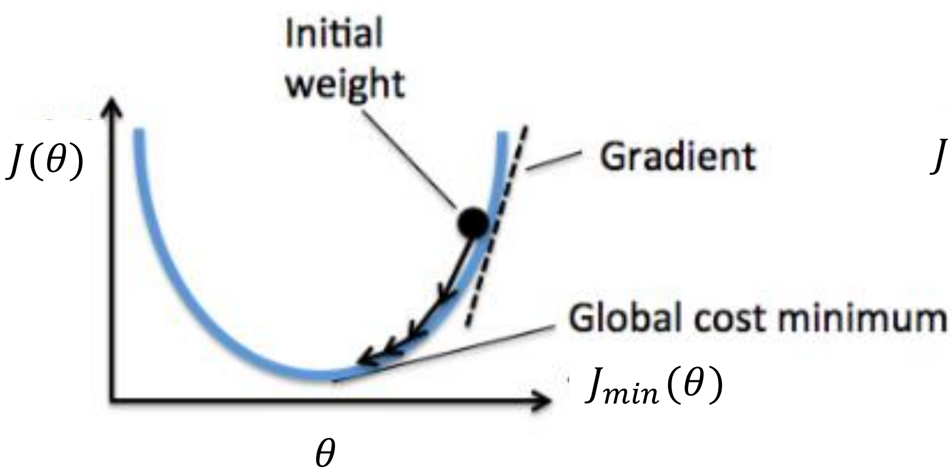
$$\theta_0 := \theta_0 - a\frac{1}{N_k}\sum_{i=1}^{N_k}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - a\frac{1}{N_k}\sum_{i=1}^{N_k}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

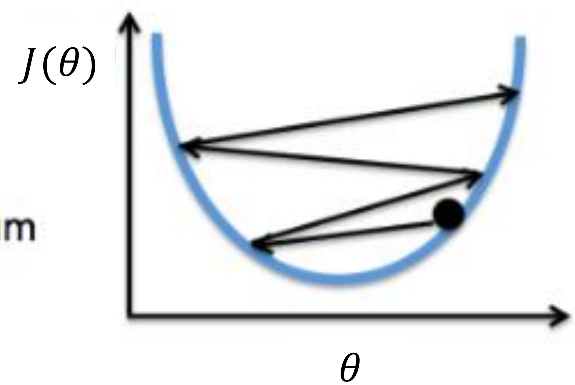


学习率选择
Choose learning rate

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

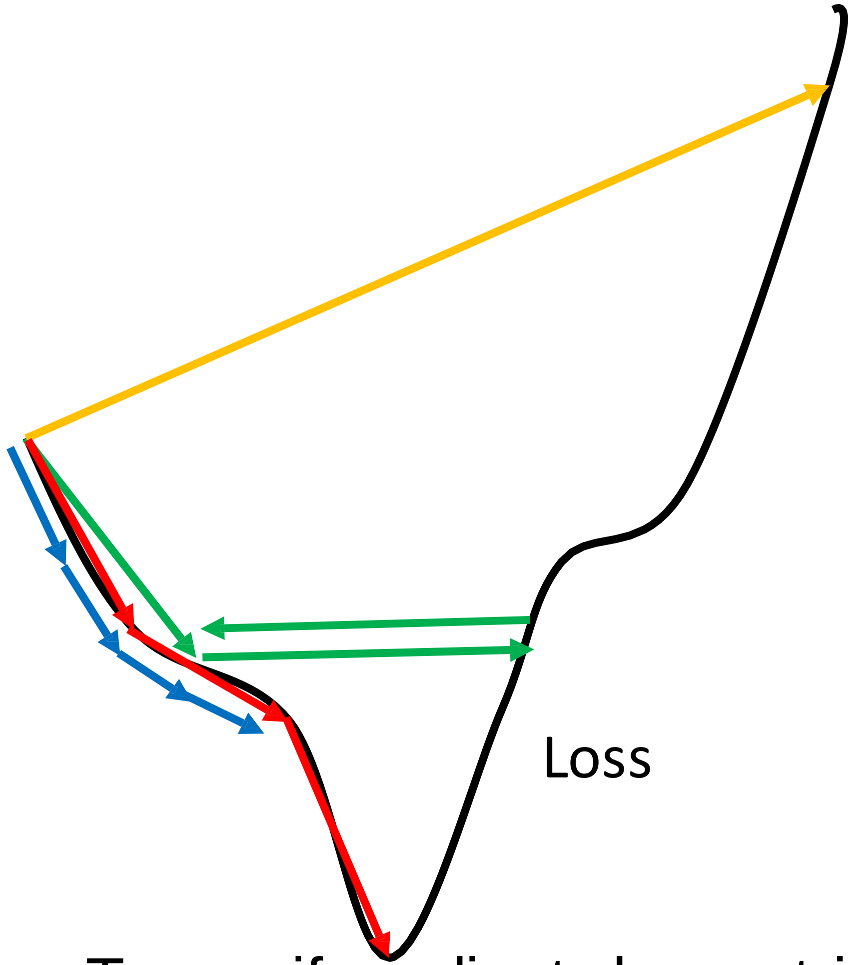


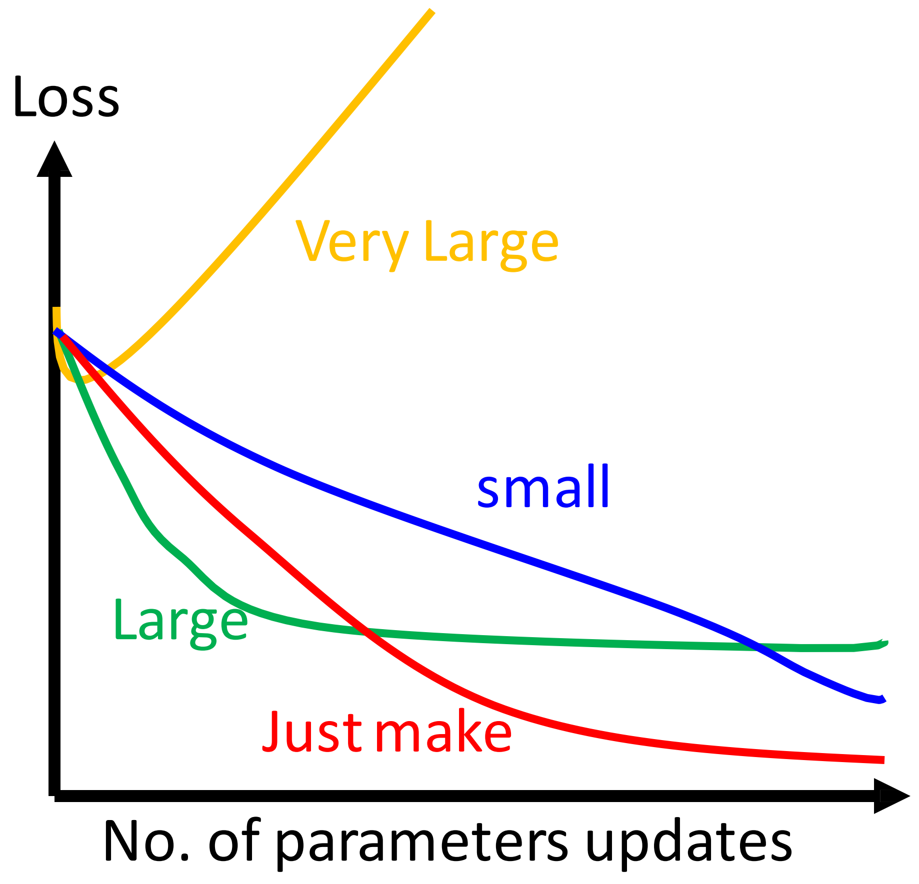
If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



To see if gradient descent is working, print out for each or every $J(\theta)$ several iterations. If $J(\theta)$ does not drop properly, adjust the learning rate!

学习率选择
Choose learning rate





To see if gradient descent is working, print out for each or every $J(\theta)$ several iterations. If $J(\theta)$ does not drop properly, adjust the learning rate!

多变量线性回归
Linear regression with multiple variable

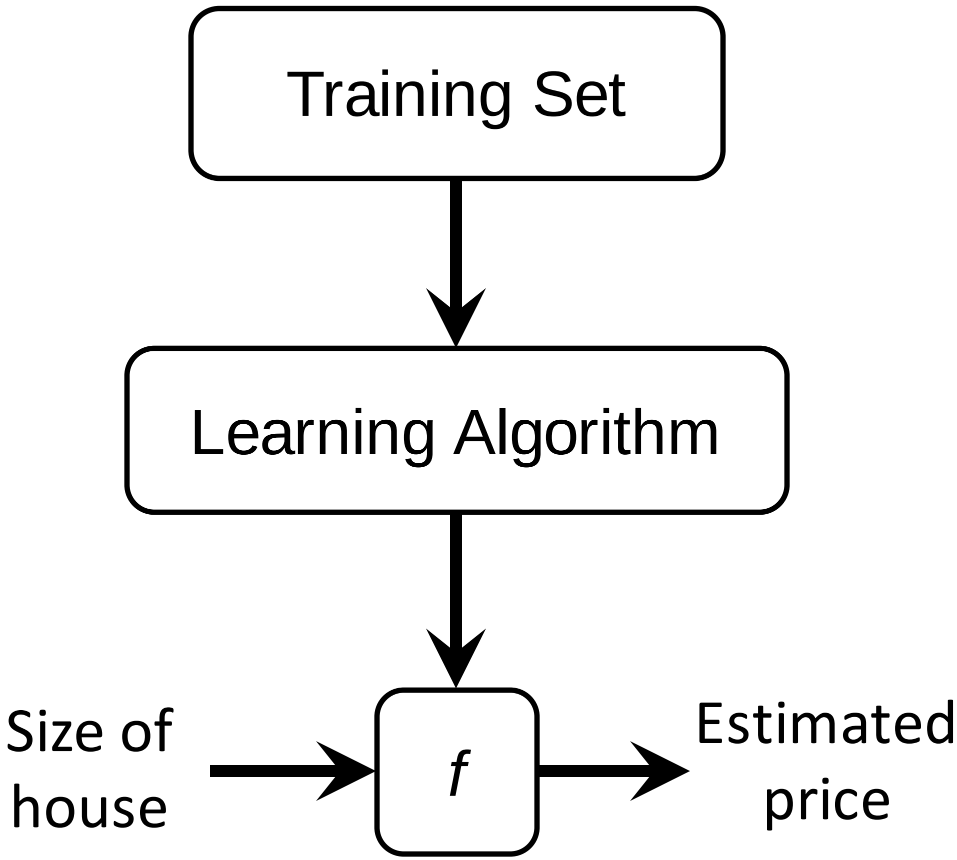
Size (feet²)	Price ($1000)
2104	460
1416	232
1534	315
852	178
…	…

Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

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

$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2\, x_2 + \cdots + \theta_n\, x_n$$

单变量线性回归
Linear regression with one variable



Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

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

Goal:

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

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

多变量线性回归
Linear regression with multiple variable

Hypothesis: $f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

Cost function: $J(\theta_0, \theta_1, \ldots \theta_n) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$

Notation:

n = number of features

$x^{(i)}$ = input (features) of i^{th} training example.

$x_j^{(i)}$ = value of feature j in i^{th} training example.

Gradient descent:

Repeat $\{$

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

$\}$ (simultaneously update for every $j = 0, \ldots, n$)

多变量线性回归

Linear regression with multiple variable

Previously (n=1):

Repeat {

$$\theta_0 := \theta_0 - a\underbrace{\frac{1}{N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)}_{\frac{\partial}{\partial\theta_0}J(\theta)}$$

$$\theta_1 := \theta_1 - a\frac{1}{N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

(simultaneously update θ_0, θ_1)

}

New algorithm $(n \geq 1)$:

Repeat {

$$\theta_j := \theta_j - a\frac{1}{N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, \ldots, n$)

}

$$\theta_0 := \theta_0 - a\frac{1}{N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_0^{(i)}$$

$$\theta_1 := \theta_1 - a\frac{1}{N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_1^{(i)}$$

$\ldots$

多变量线性回归
Linear regression with multiple variable

Size (feet²)	Price ($1000)
2104	460
1416	232
1534	315
852	178
…	…

Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

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

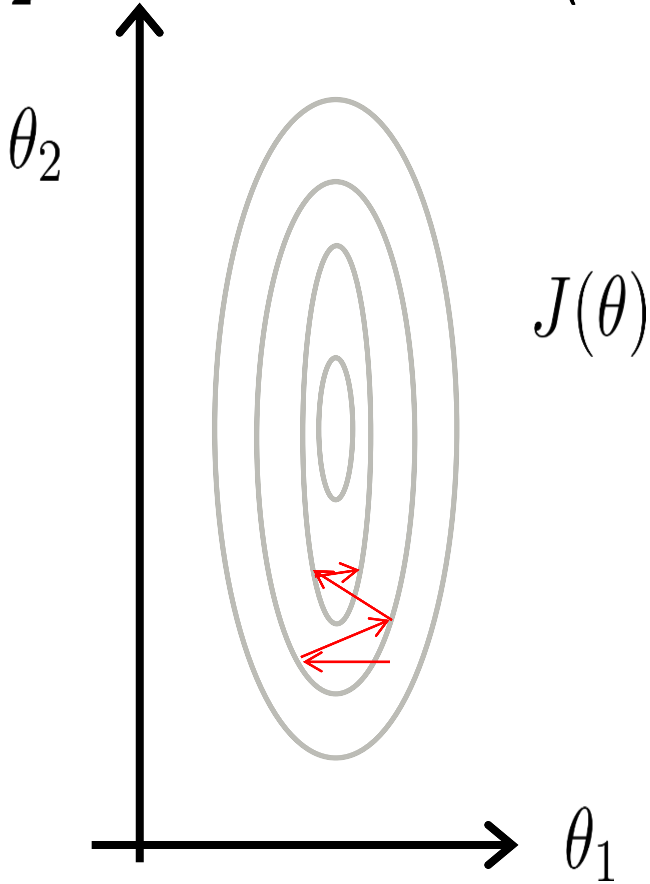
$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2\, x_2 + \cdots + \theta_n\, x_n$$

特征归一化
Feature Scaling

$$f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)

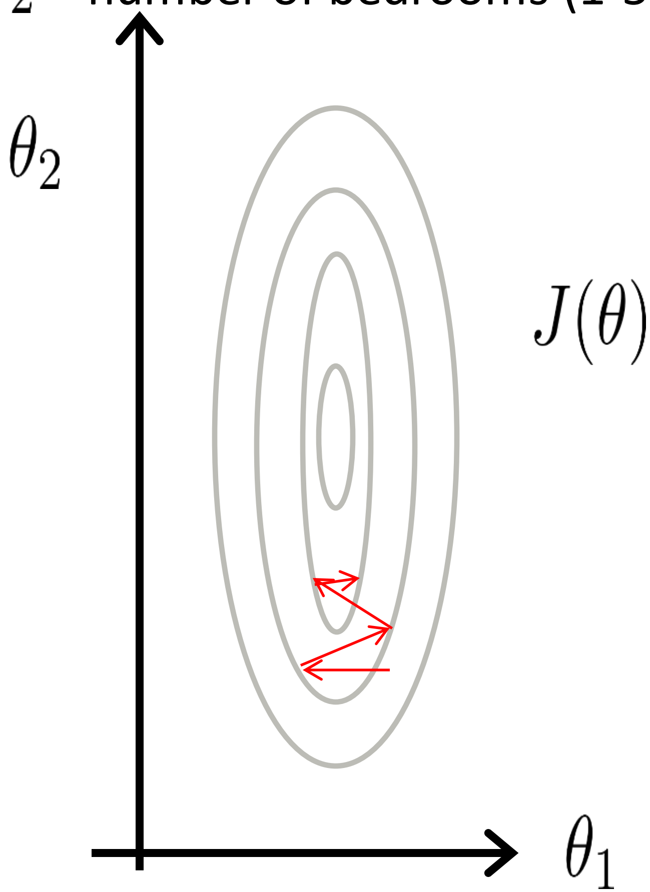


特征归一化
Feature Scaling

Idea: Make sure features are on a similar scale.

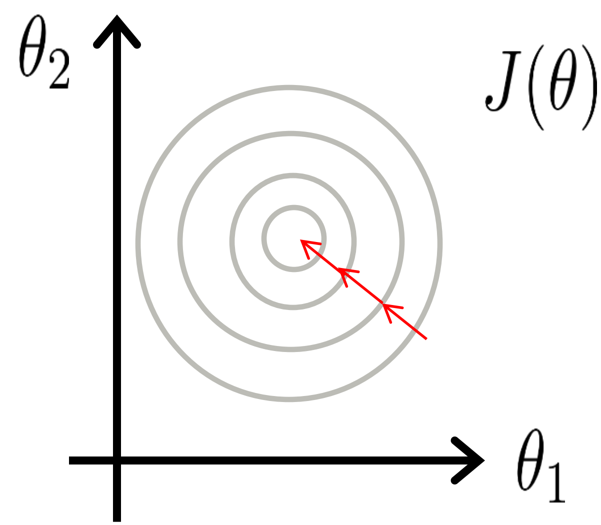
E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)



$$x_1 = \frac{size(feet^2)}{2000}$$

$$x_2 = \frac{\text{number of bedrooms}}{5}$$



特征归一化
Feature Scaling

Get every feature into approximately a similar scale.

Mean Normalization

$$x' = \frac{x - mean(x)}{max(x) - min(x)}$$

Standardization

$$x' = \frac{x - mean(x)}{std(x)}$$

$$std(x) = \sqrt{\frac{\sum (x - mean(x))^2}{n}}$$

e.g. Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean.

E.g.

$$x_1 = \frac{size - 1000}{2000}$$

$$x_2 = \frac{\#bedrooms - 2}{4}$$

$$-0.5 \leq x_1 \leq 0.5, -0.5 \leq x_2 \leq 0.5$$

多变量线性回归

Linear regression with multiple variable

Previously (n=1):

Repeat $\{$

$$\theta_0 := \theta_0 - a\underbrace{\frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial\theta_0}J(\theta)}$$

$$\theta_1 := \theta_1 - a\frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

(simultaneously update θ_0, θ_1)

$\}$

New algorithm $(n \geq 1)$:

Repeat $\{$

$$\theta_j := \theta_j - a\frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, \ldots, n$)

$\}$

$$\theta_0 := \theta_0 - a\frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_1 := \theta_1 - a\frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

$\ldots$

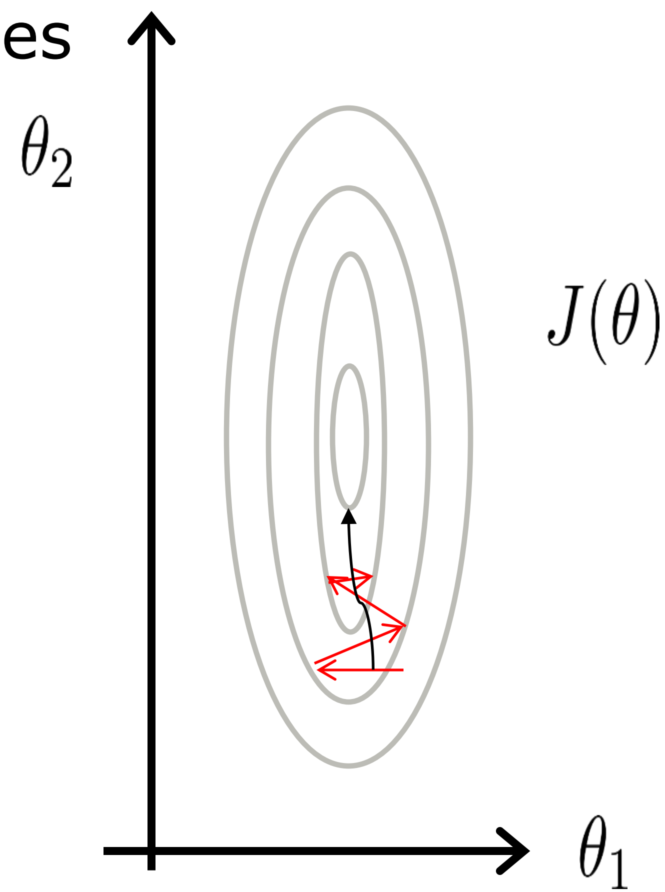
自适应的学习率
Adaptive Learning Rates

Adagrad

Divide the learning rate of each parameter by the root mean square of its previous derivatives

$$\theta^{(t+1)} := \theta^{(t)} - \frac{a}{\sqrt{\sum_{i=0}^{t} (g^{(i)})^2}} g^{(t)}$$

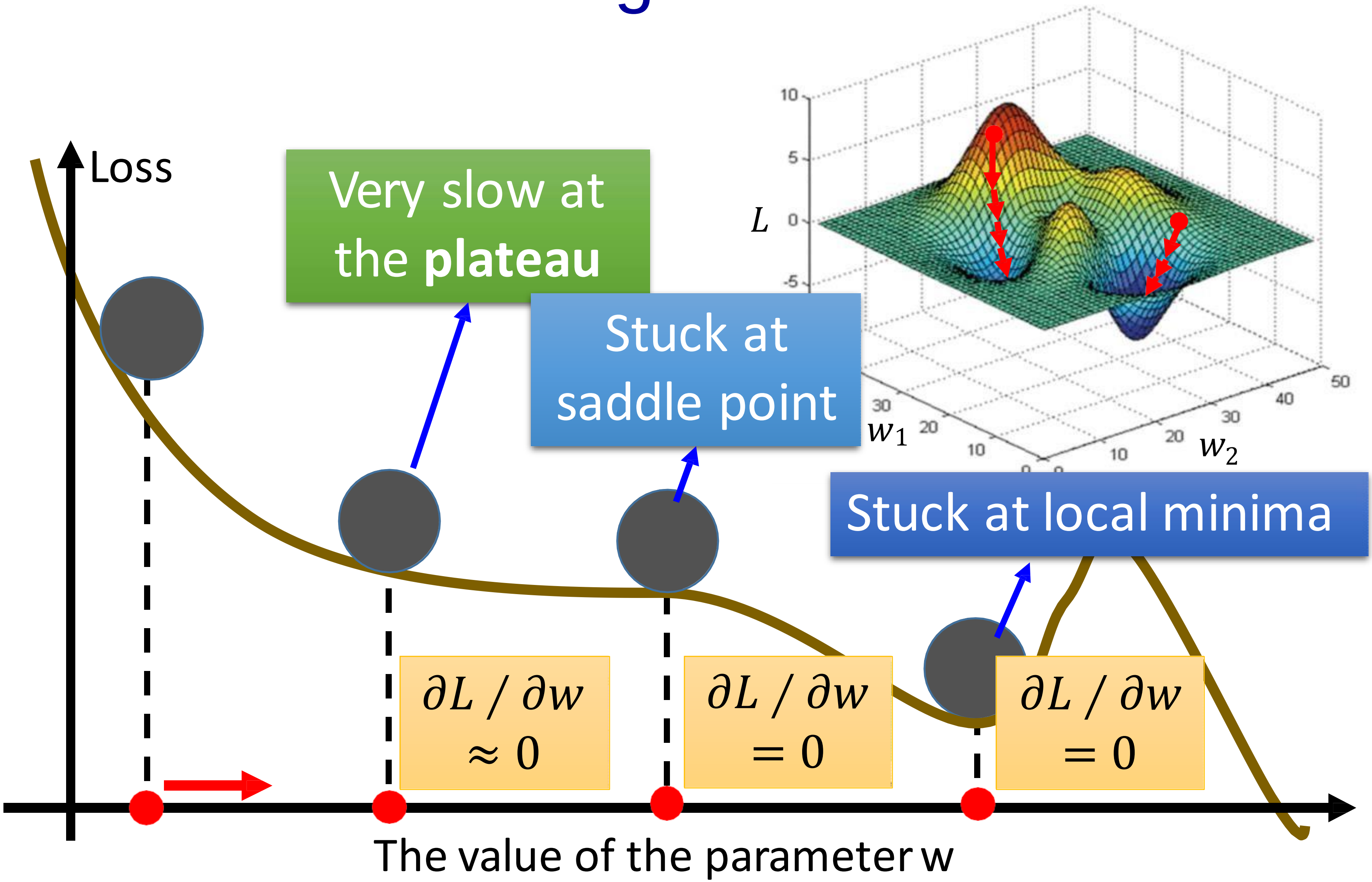
$$g^{(t)} = \frac{\partial J(\theta^{(t)})}{\partial \theta}$$



adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters

梯度的限制

Limitation of gradient descent



梯度下降优化算法
Gradient descent optimization algorithms

- **Momentum**

  helps accelerate SGD in the relevant direction and dampens oscillations

- **Adagrad**

  (Adaptive Gradient)

  adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters

- **RMSProp**

  (Root Mean Square propagation)

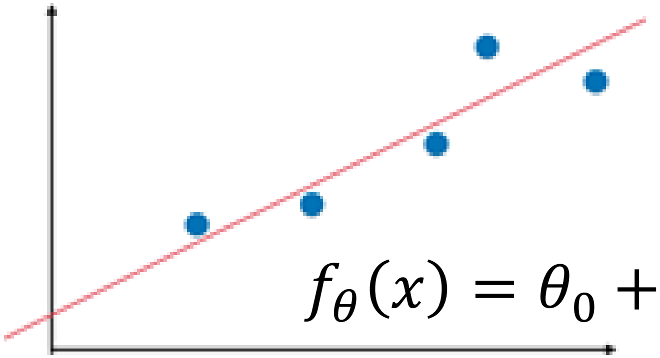
  divides the learning rate by an exponentially decaying average of squared gradients

- **Adam**

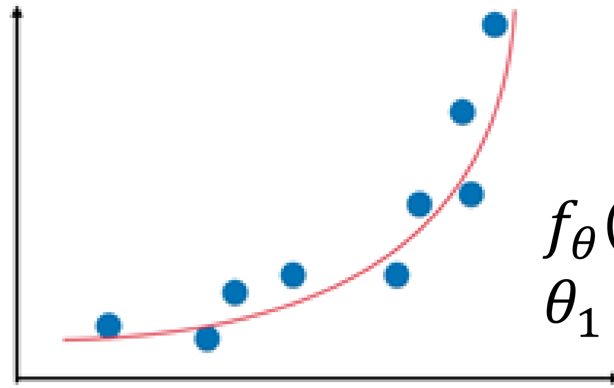
  (Adaptive Moment Estimation)

  stores an exponentially decaying average of past squared gradients like RMSprop, also keeps an exponentially decaying average of past gradients, similar to momentum

多项式回归
Polynomial regression

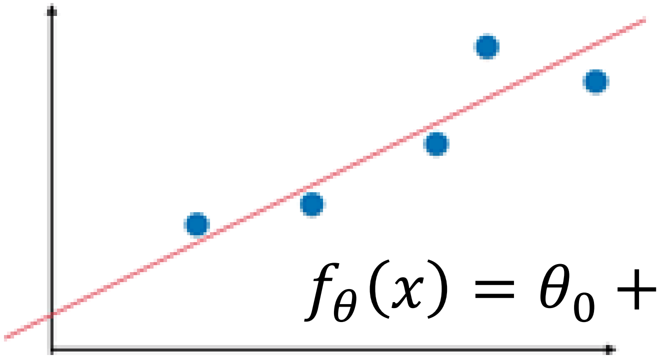


$$f_\theta(x) = \theta_0 + \theta_1\, x_1$$

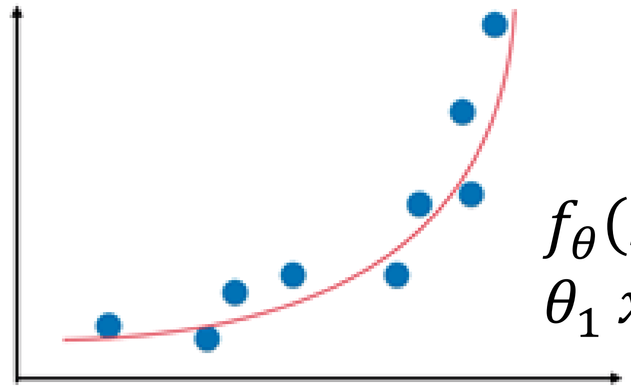


$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 x_1^2$$

多项式回归
Polynomial regression

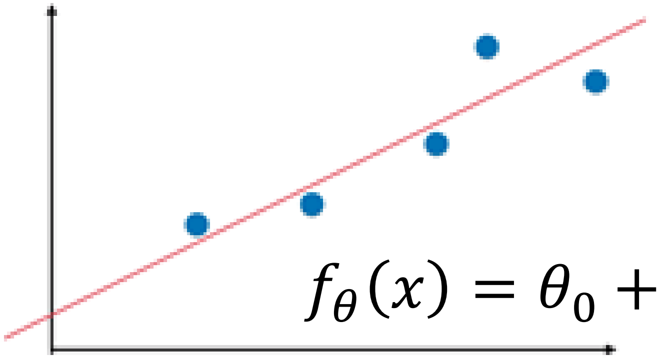


$$f_\theta(x) = \theta_0 + \theta_1\, x_1$$

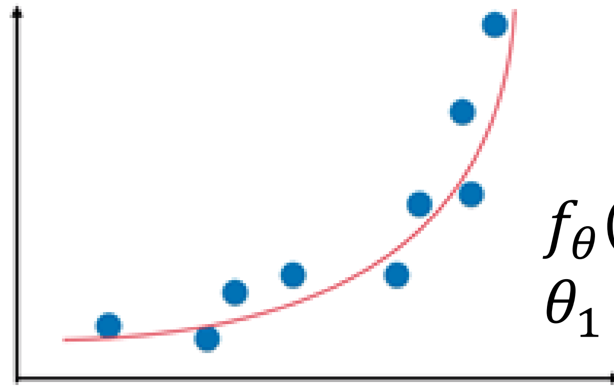


$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 \boxed{x_1^2}$$

多项式回归
Polynomial regression



$$f_\theta(x) = \theta_0 + \theta_1\, x_1$$

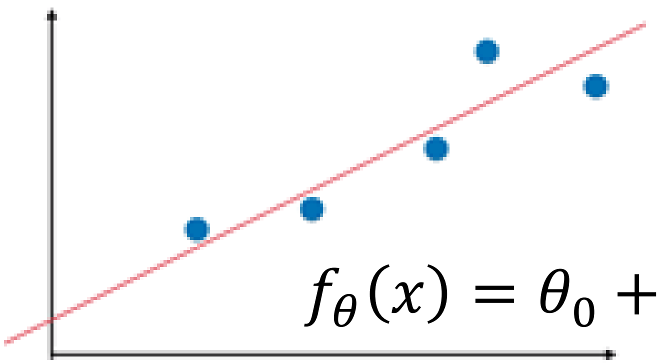


$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 \boxed{x_1^2}$$

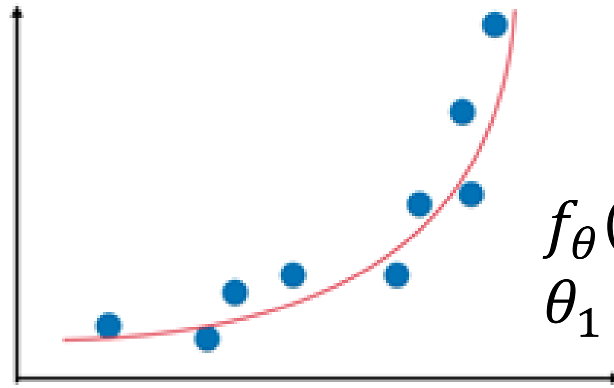
$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 \boxed{x_2}$$

Polynomial feature

多项式回归
Polynomial regression



$$f_\theta(x) = \theta_0 + \theta_1\, x_1$$



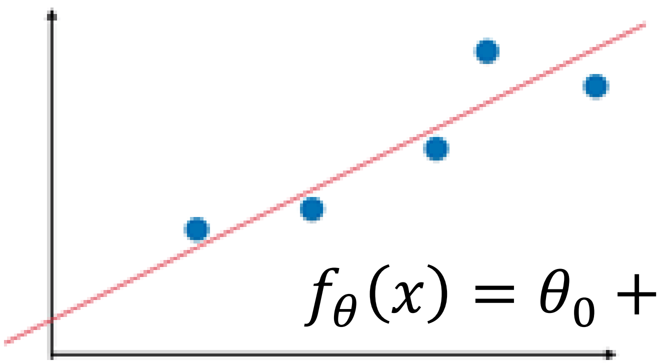
$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 x_1^2$$

$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 x_1^2 + \theta_2 x_1^3\ + \theta_2 x_1^4\ + \theta_2 x_1^5\ + \cdots$$

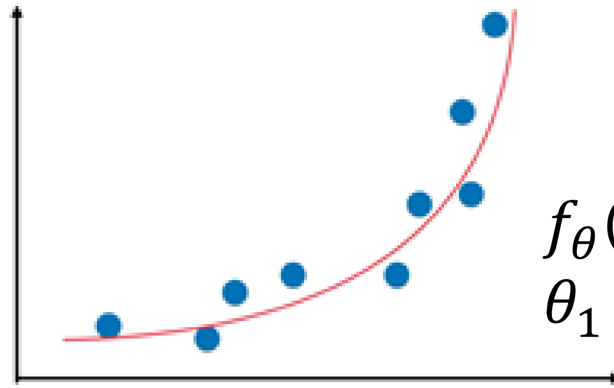
$$\Downarrow$$

$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2\, x_2\ + \theta_2\, x_3\ + \theta_2\, x_4\ + \theta_2\, x_5\ + \ldots$$

多项式回归
Polynomial regression



$$f_\theta(x) = \theta_0 + \theta_1\, x_1$$



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

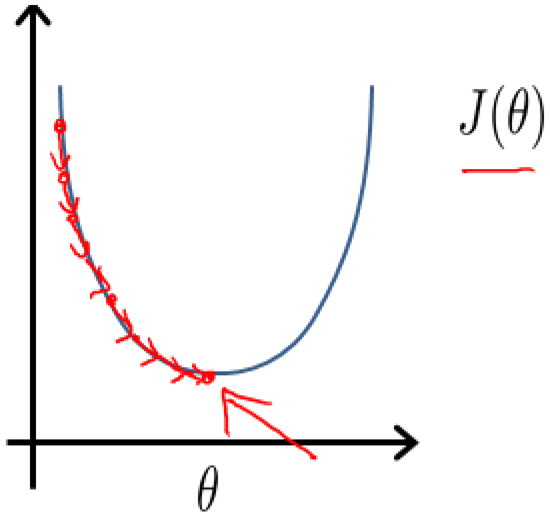
$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 x_1^2 + \theta_2 x_1^3 + \theta_2 x_1^4 + \theta_2 x_1^5 + \cdots$$

$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2\, x_2 + \theta_2\, x_3 + \theta_2\, x_4 + \theta_2\, x_5 + \ldots$$

✓ able to model all sorts of relationships

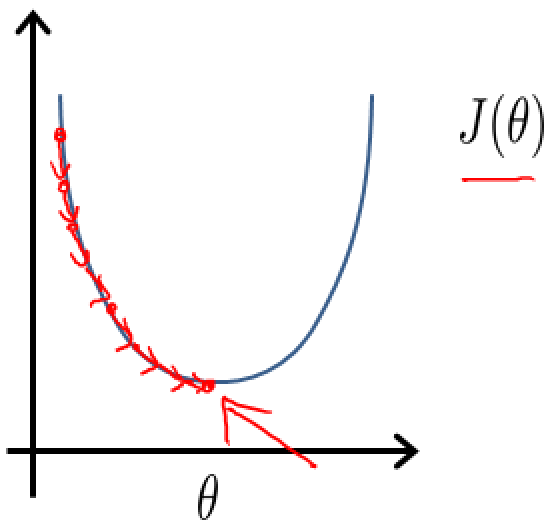
X easy to overfit

最小二乘方线性回归
Least square linear regression

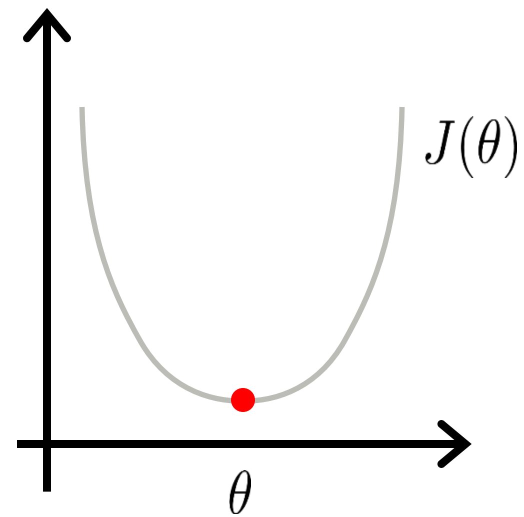


最小二乘方线性回归
Least square linear regression

Gradient Descent



Normal equation



$$\frac{\partial}{\partial \theta_j} J(\theta) = \cdots = 0$$

(for every j)

Solve for $\theta_0, \theta_1, \ldots, \theta_n$

最小二乘法求解
Least square method

	Size (feet²)	Number of bedrooms	number of floors	Age of home (years)	Price ($1000)
x_0	x_1	x_2	x_3	x_4	y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}$$

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 = \frac{1}{2}(X\theta - y)^2 = \frac{1}{2}(X\theta - y)^T(X\theta - y)$$

最小二乘法
Least square method

- Objective $\min_{\theta} J(\theta)$

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 = \frac{1}{2}(X\theta - y)^2 = \frac{1}{2}(X\theta - y)^T(X\theta - y)$$

- Gradient

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{2}\frac{\partial}{\partial \theta}(\theta^T X^T X\theta - y^T X\theta - \theta^T X^T y + y^T y)$$

$$= \frac{1}{2}\frac{\partial}{\partial \theta}(\theta^T X^T X\theta - 2\theta^T X^T y + y^T y)$$

$$= X^T X\theta - X^T y$$

- Solution

$$\frac{\partial J(\theta)}{\partial \theta} = 0 \Rightarrow X^T X\theta - X^T y = 0 \quad \Rightarrow \theta = (X^T X)^{-1} X^T y$$

正规方程求解
Normal equation method

	Size (feet²)	Number of bedrooms	number of floors	Age of home (years)	Price ($1000)
x_0	x_1	x_2	x_3	x_4	y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \quad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}$$

$$\theta = (X^T X)^{-1} X^T y$$

最小二乘法
Least square method

x_0	Size (feet²) x_1	Number of bedrooms x_2	number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}$$

$$\boxed{\theta = (X^T X)^{-1} X^T y} \leftarrow O(\text{number of features})^3$$

梯度下降 VS 最小二乘法
Gradient descent VS Least square method

<u>Gradient Descent</u>

- Need to choose α .
- Needs many iterations.
- Works well even when the number of features is large.

Least square method

- No need to choose α .
- Don’t need to iterate.
- Need to compute $(X^T X)^{-1}$
- Slow if the number of features is very large (>10000) .
- only applicable to linear models
- Sometimes cannot be directly calculated(if $X^T X$ is non-invertible).

评价标准
Evaluation indices

MSE(Mean Squared Error)
均方误差

$$\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - f(x^{(i)}))^2$$

RMSE（Root Mean Squared Error）
均方根误差

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - f(x^{(i)}))^2}$$

MAE (Mean absolute Error)
平均绝对误差

$$\frac{1}{N}\sum_{i=1}^{N}\left|(y^{(i)} - f(x^{(i)}))^2\right|$$

R-Squared（r2score）R方/决定系数

$$= 1 - \frac{\sum_{i=1}^{N}(y^{(i)} - f(x^{(i)}))^2}{\sum_{i=1}^{N}(y^{(i)} - \bar{y}))^2}$$

$$= 1 - \frac{MSE}{Var}$$

思考题

多变量线性回归相比单变量回归，采用标准的梯度下降求解会有什么问题及可能的解决方法?