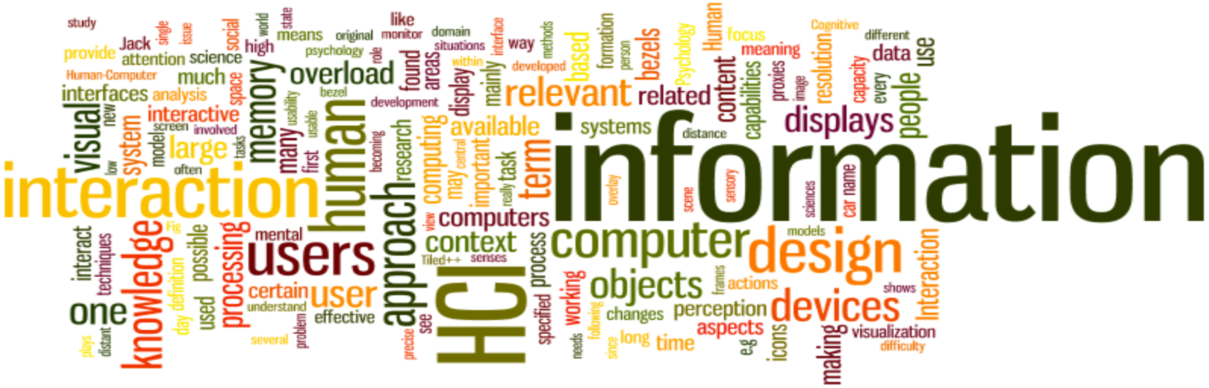
机器学习

—— 第2章 线性回归——



授课：李洁　倪张凯

nijanice@163.com　　zkni@tongji.edu.cn

https://eezkni.github.io/

学习任务的类型 Types of learning task

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

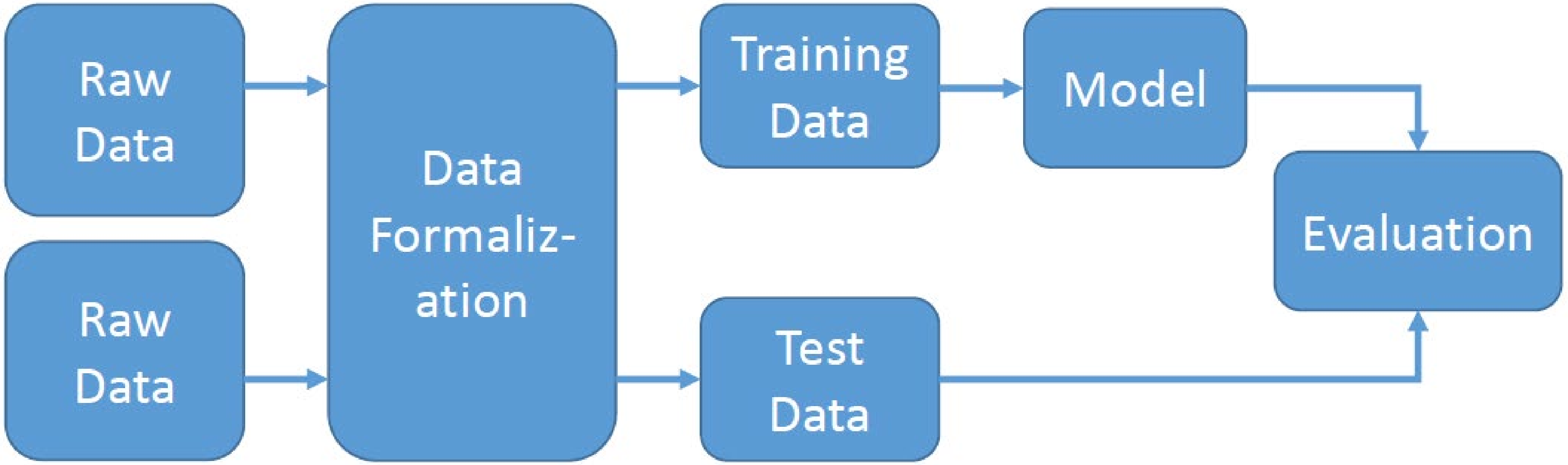
- Unsupervised learning
  – try to find hidden structure in unlabeled training data
  – clustering

- 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,

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

$x^{(i)}$= 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

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

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

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

给西瓜打分

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,

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

$x^{(i)}$= 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,

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

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

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

let the machine learn a function from data to label

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

- 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

线性回归模型 Linear Regression Model

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

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

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

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

let the machine learn a function from data to label

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

- 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,

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

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

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

let the machine learn a function from data to label

Independent Variable

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

Dependent Variable

- 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

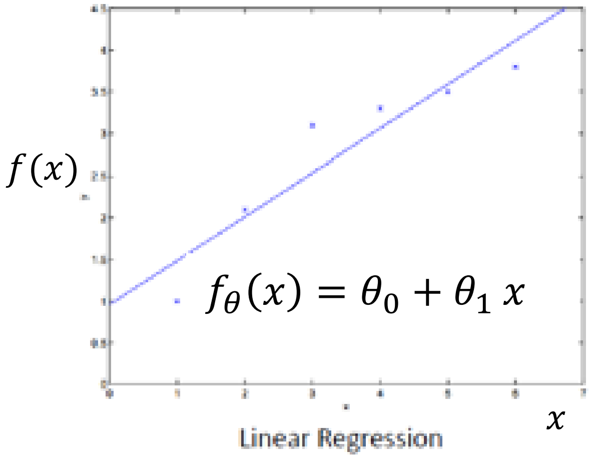
$$f_\theta(x) = \theta_1\, x + \theta_0$$

sample x

One feature/variable: x

Linear regression
with one variable

(One-dimensional regression)



线性回归模型 Linear Regression Model

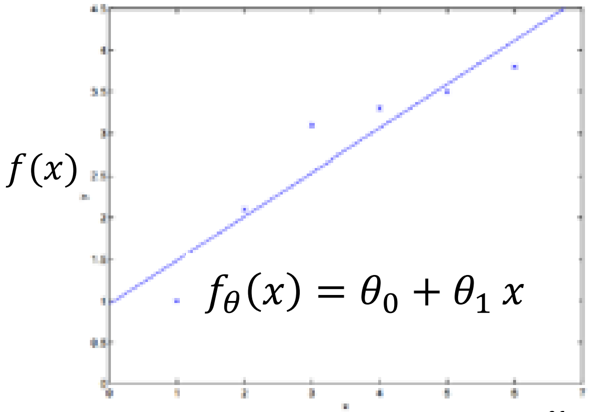
$$f_\theta(x) = \theta_1\, x + \theta_0$$

sample x

One feature/variable: x

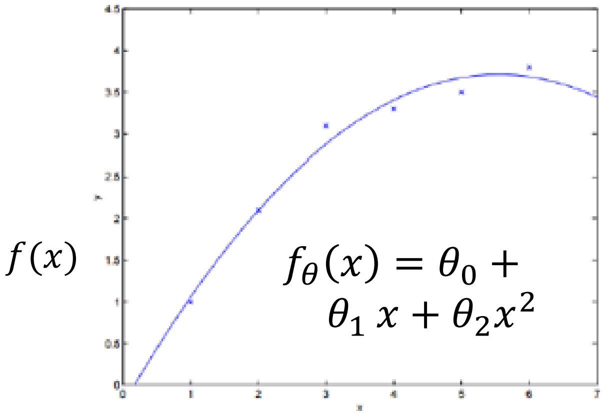
Linear regression with one variable

(One-dimensional regression)



Linear Regression

quadratic regression with one variable



Quadratic Regression
(A kind of generalized linear model)

线性回归模型 Linear Regression Model

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

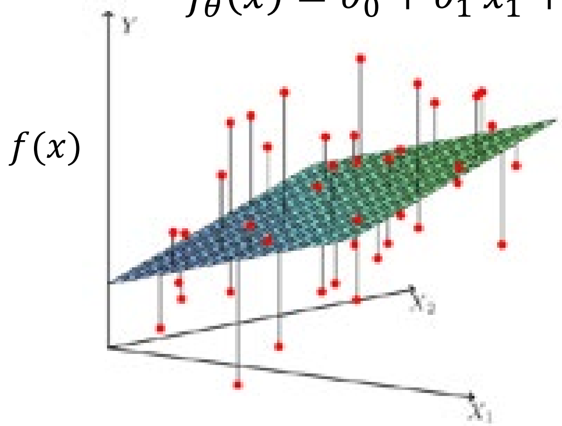
sample x

Two features/variables: x_1, x_2

Linear regression with two variable

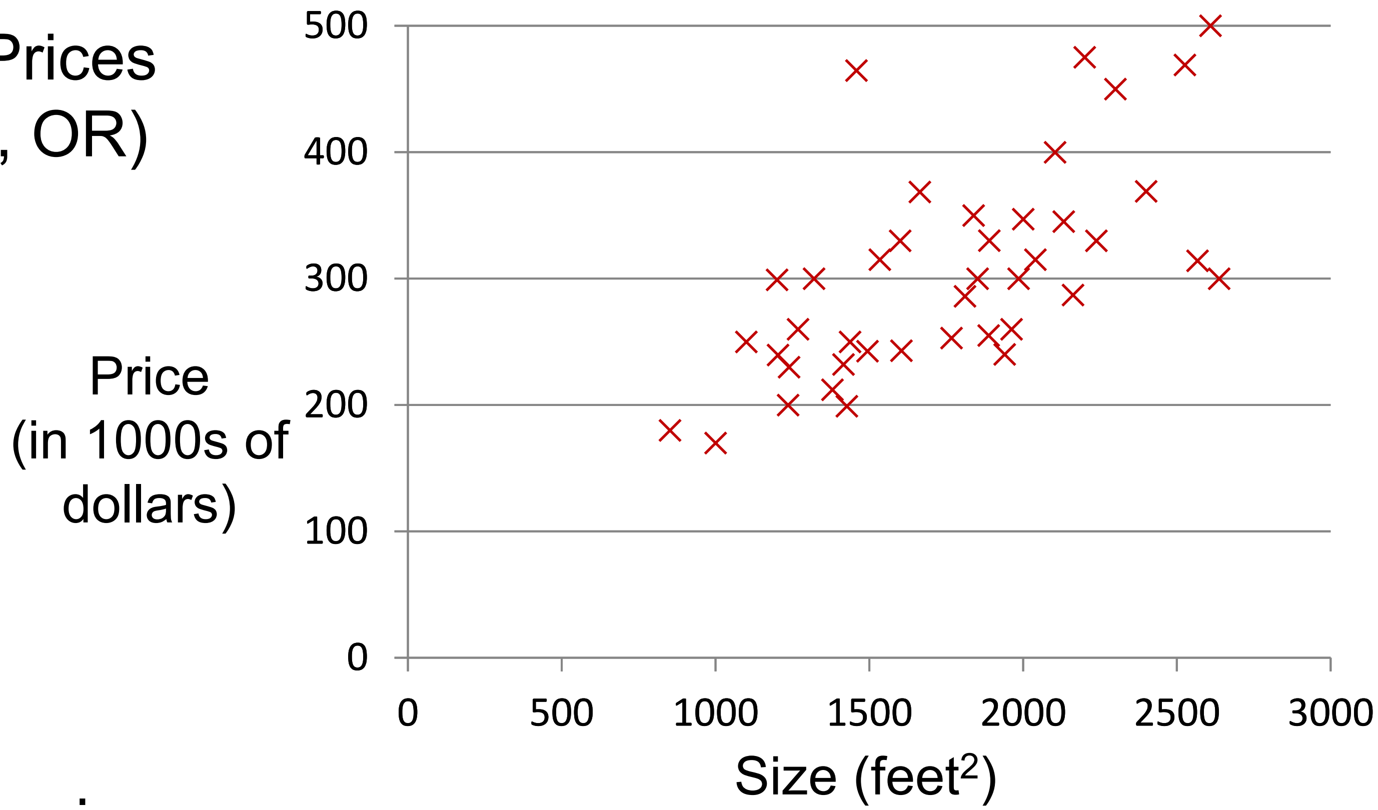
(two-dimensional linear regression)

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



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
…	…

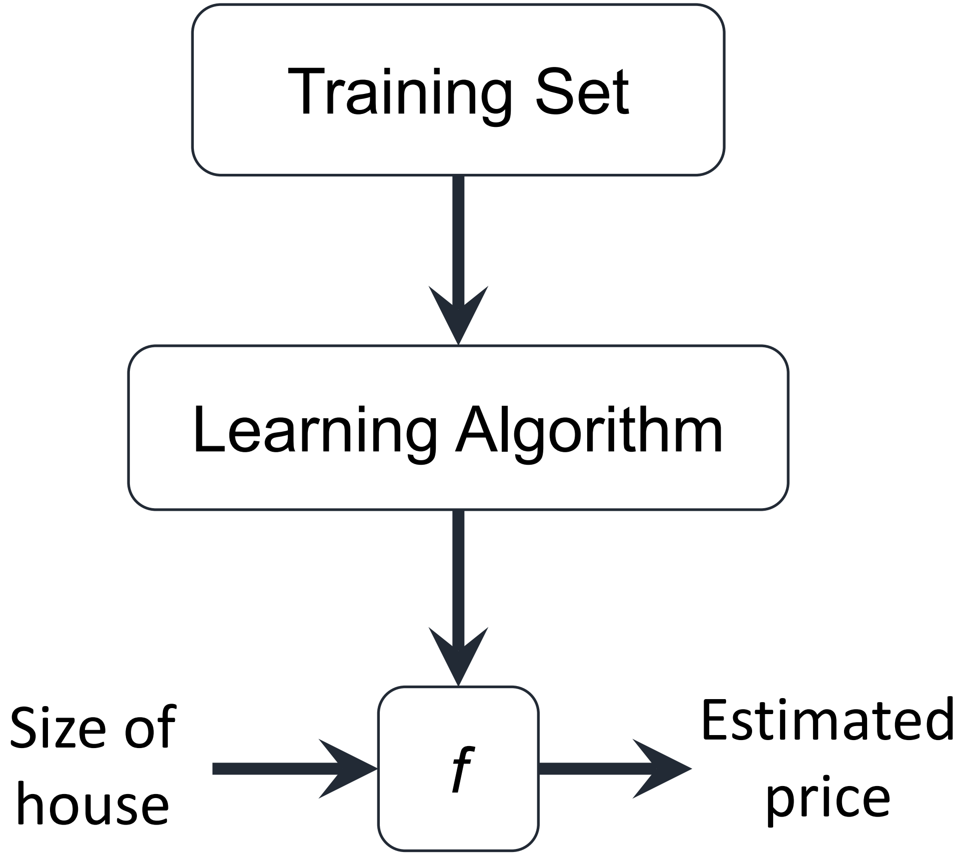
Notation:

N= Number of training examples

x = “input” variable / features

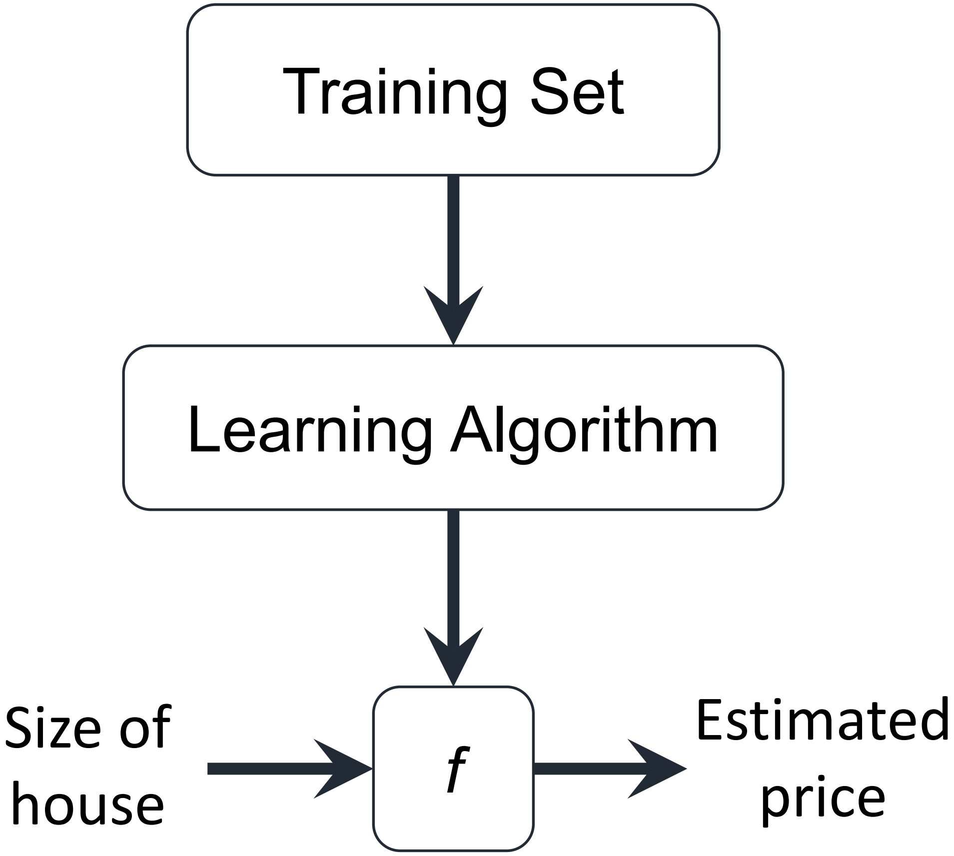
y = “output” variable / “target” variable

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

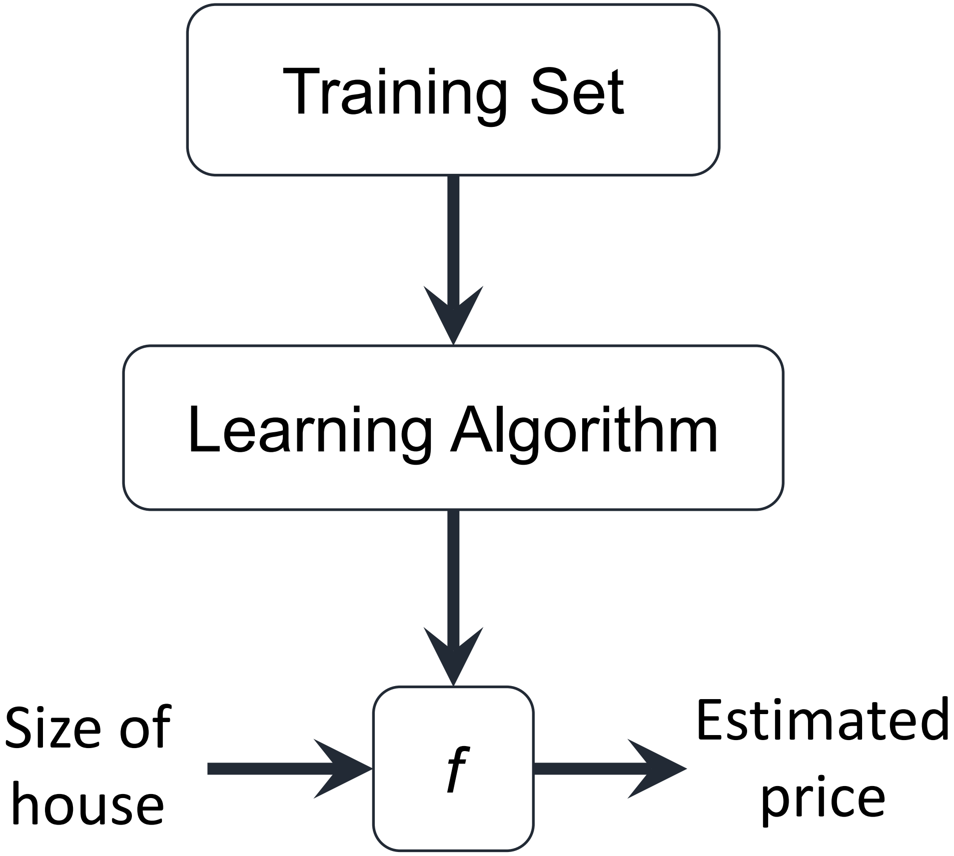


How do we represent f?

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

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

单变量线性回归 Linear regression with one variable



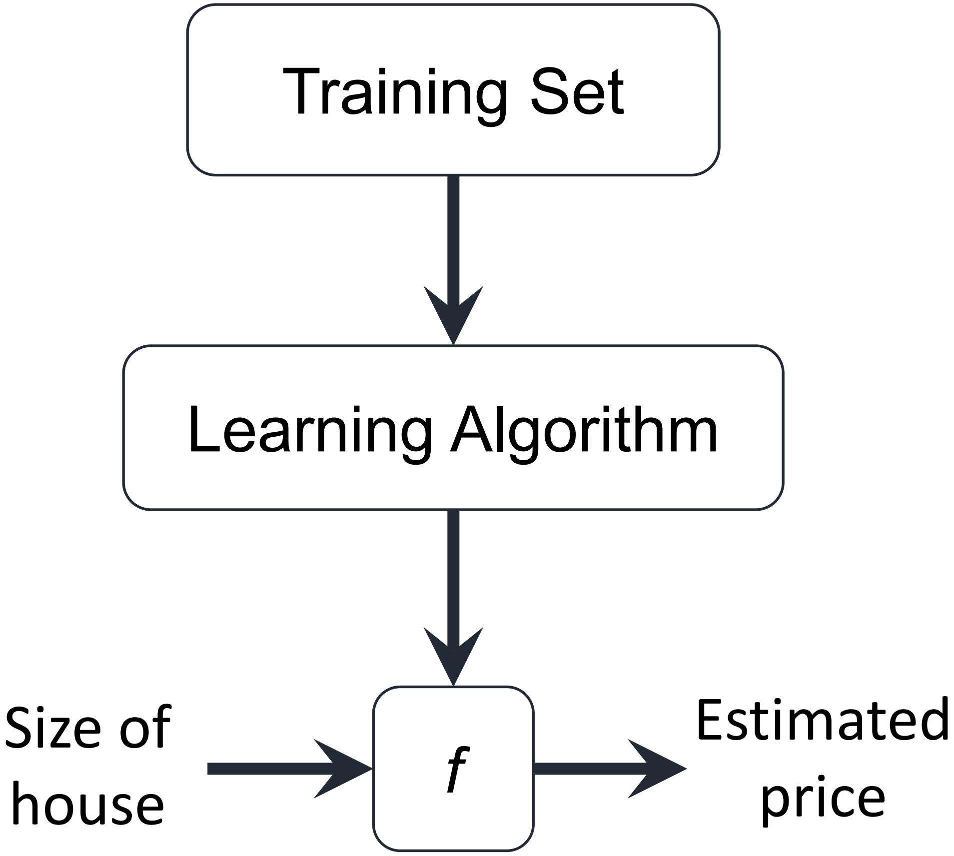
How do we represent f?

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

θ_i ‘s: 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?

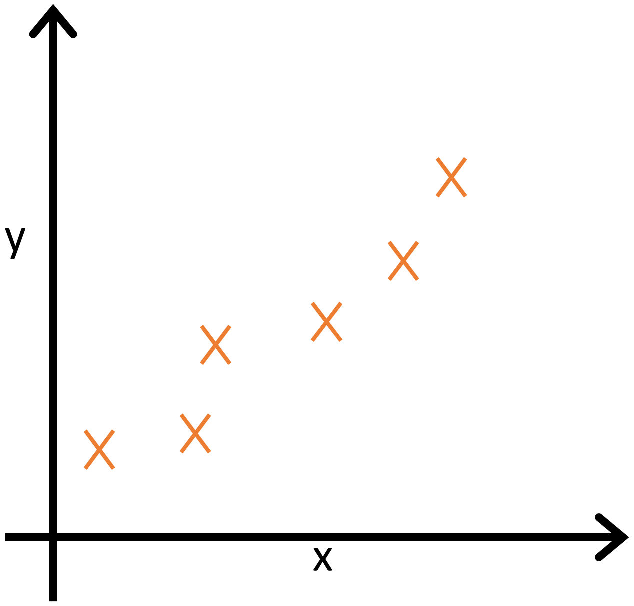
$$f_\theta(x) = \theta_0 + \theta_1 x$$

θ_i's: Parameters

How to choose θ_i's ?

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

单变量线性回归 Linear regression with one variable



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

Training Set

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

损失函数 Loss function

0-1 Loss Function

$$L\left(y^{(i)}, f\left(x^{(i)}\right)\right) = \begin{cases} 1, & \text{if } y^{(i)} \neq f\left(x^{(i)}\right) \\ 0, & \text{if } y^{(i)} = f\left(x^{(i)}\right) \end{cases}$$

Mean Squared Error, MSE

$$L\left(y^{(i)}, f\left(x^{(i)}\right)\right) = \left(y^{(i)} - f\left(x^{(i)}\right)\right)^2$$

Absolute Loss Function

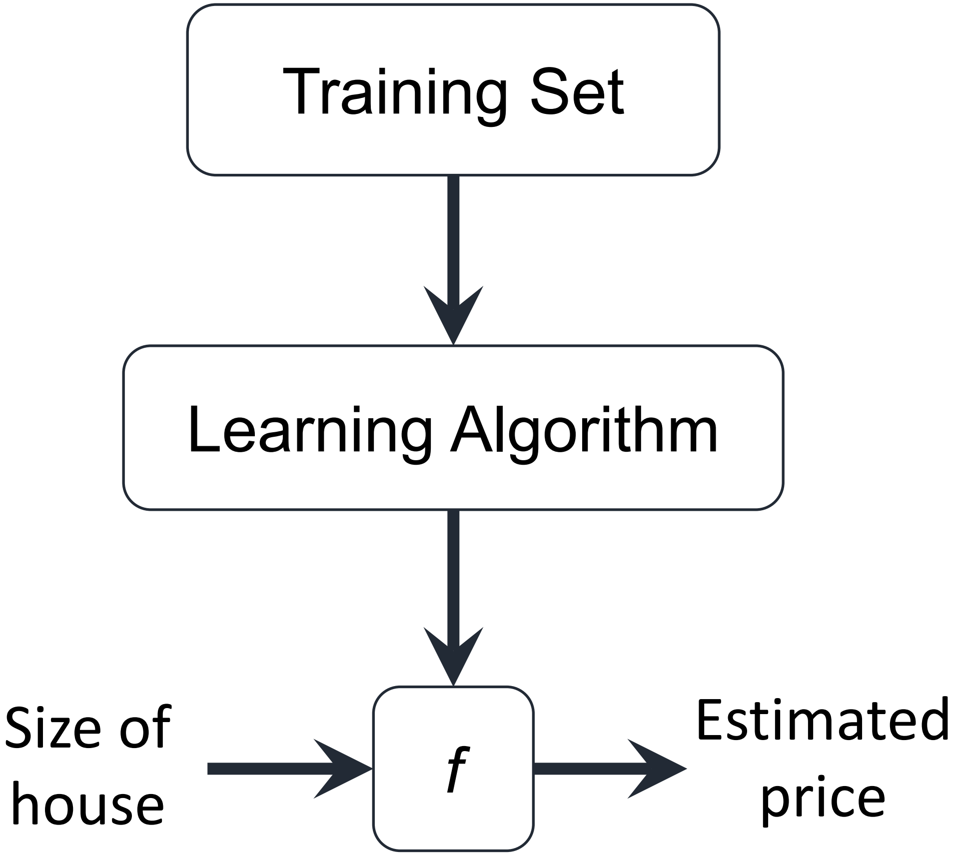
$$L\left(y^{(i)}, f\left(x^{(i)}\right)\right) = \left|y^{(i)} - f\left(x^{(i)}\right)\right|$$

Logarithmic Loss Function (Cross-Entropy Loss Function)

$$L\left(y^{(i)}, p^{(i)}\right) = -\left[y^{(i)} \log\left(p^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - p^{(i)}\right) \right]$$

$y^{(i)} \in \{0,1\}$, $p^{(i)}=f\left(x^{(i)}\right)$ is the predicted probability that the i th sample belongs

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:

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

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

Simplified hypothesis:

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

Parameters:

$$\theta_1$$

Cost Function:

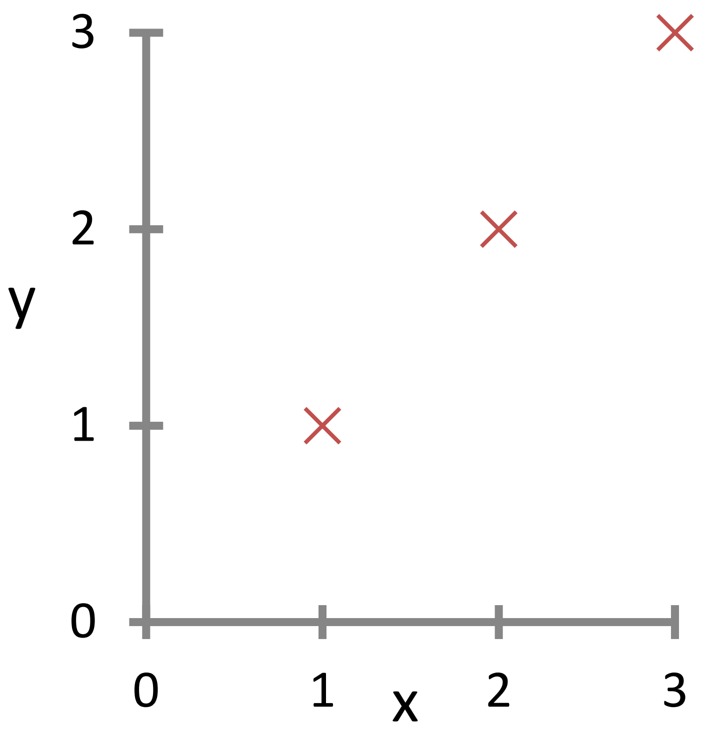
$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

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

单变量线性回归 Linear regression with one variable

$f_\theta(x)$

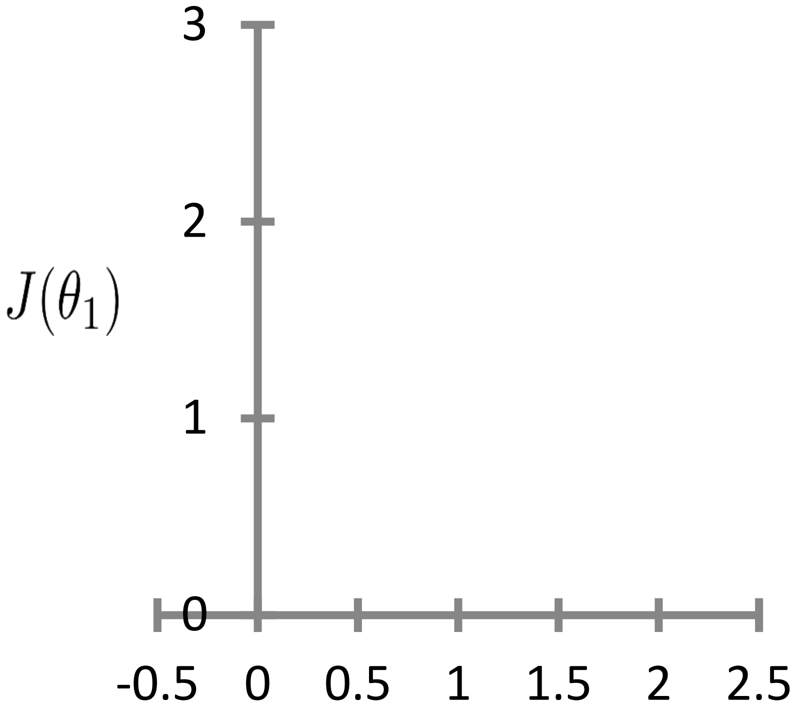
(for fixed θ_1, this is a function of x)



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

$J(\theta_1)$

(function of the parameter θ_1)

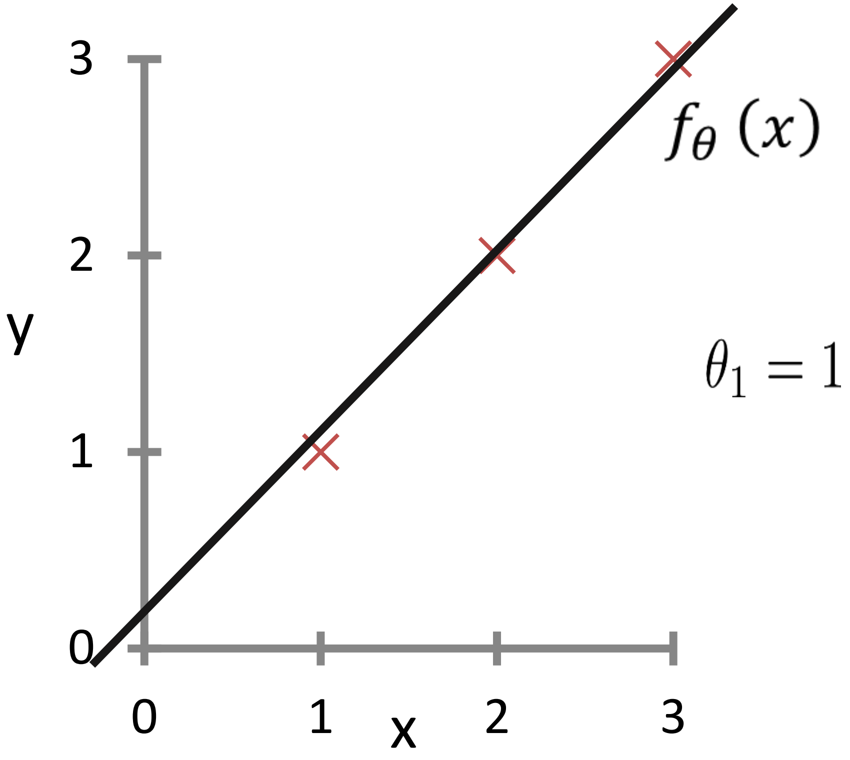


$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归 Linear regression with one variable

$f_\theta(x)$

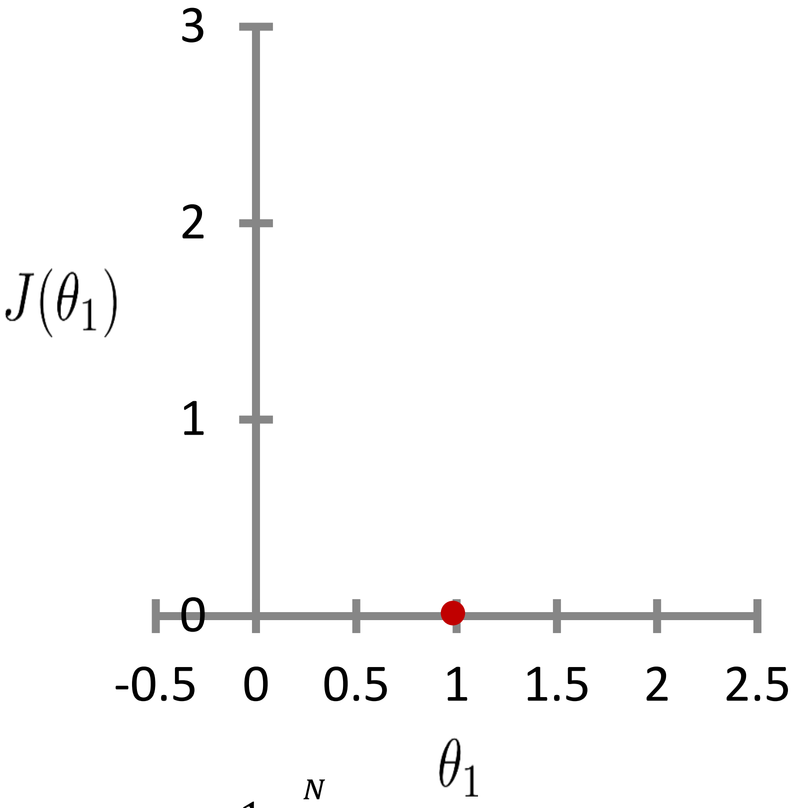
(for fixed θ_1, this is a function of x)



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

$J(\theta_1)$

(function of the parameter θ_1)

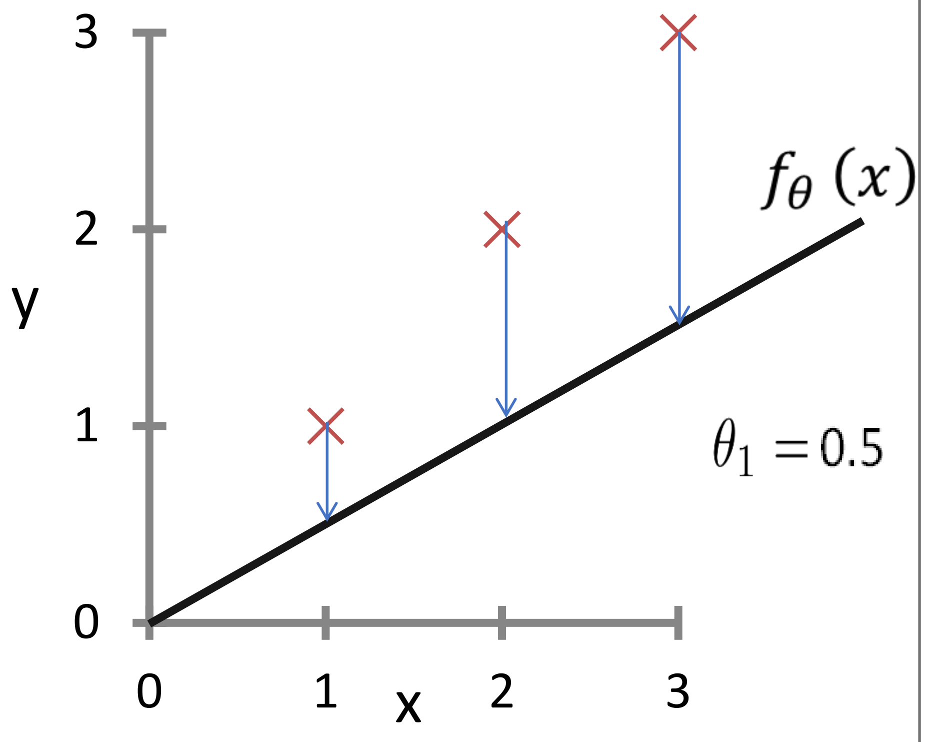


$$J(\theta_1) = \frac{1}{2N}\sum^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归 Linear regression with one variable

$f_\theta(x)$

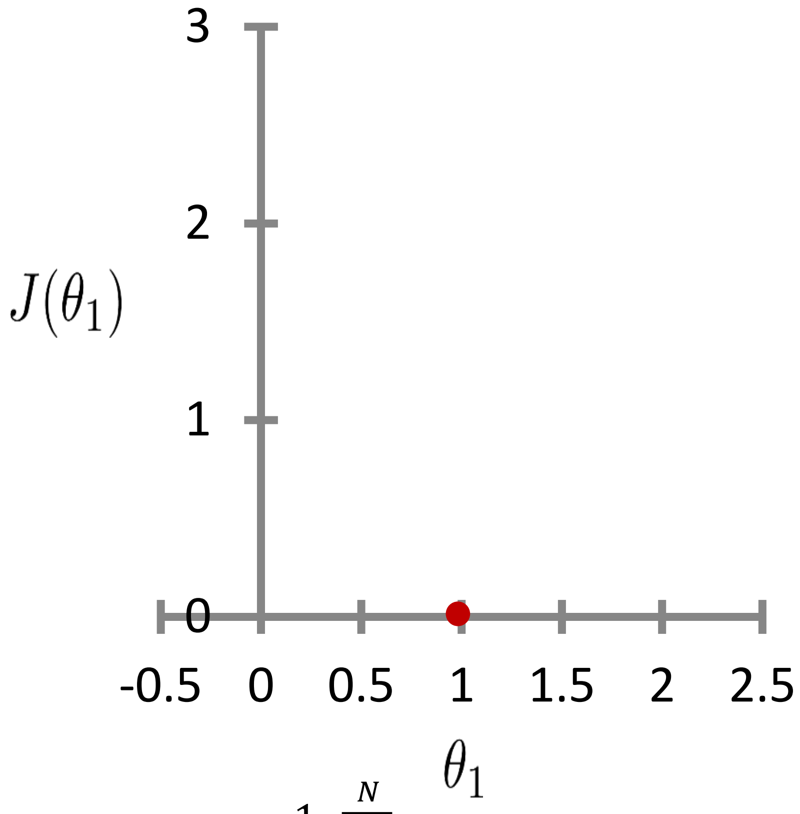
(for fixed θ_1, this is a function of x)



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

$J(\theta_1)$

(function of the parameter θ_1)

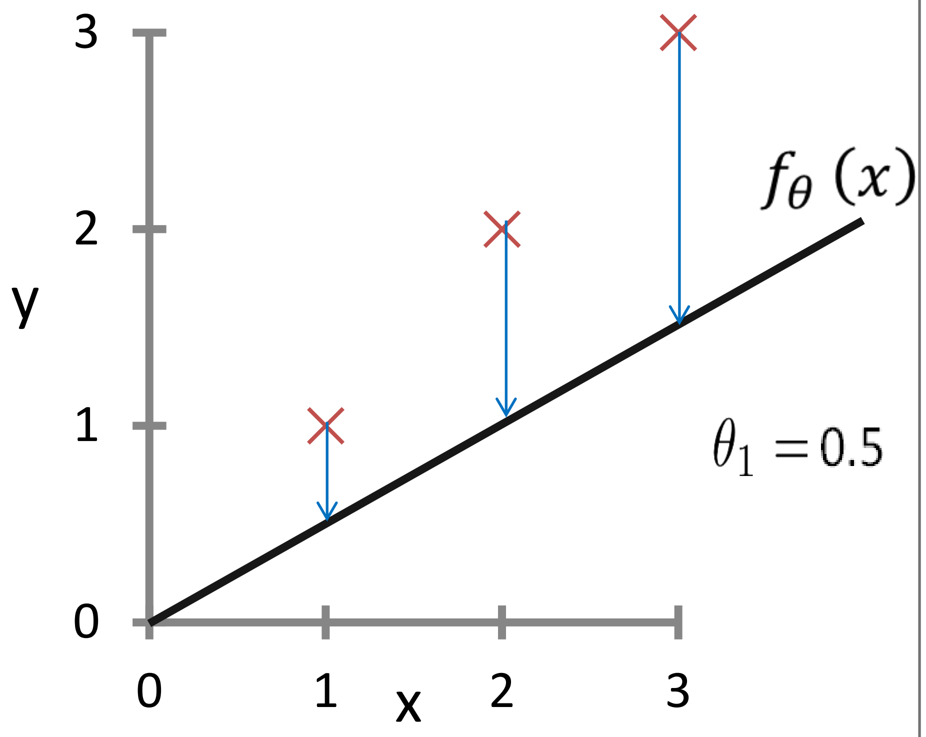


$$J(\theta_1) = \frac{1}{2N}\sum_{i=1}^{N}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

单变量线性回归 Linear regression with one variable

$f_\theta(x)$

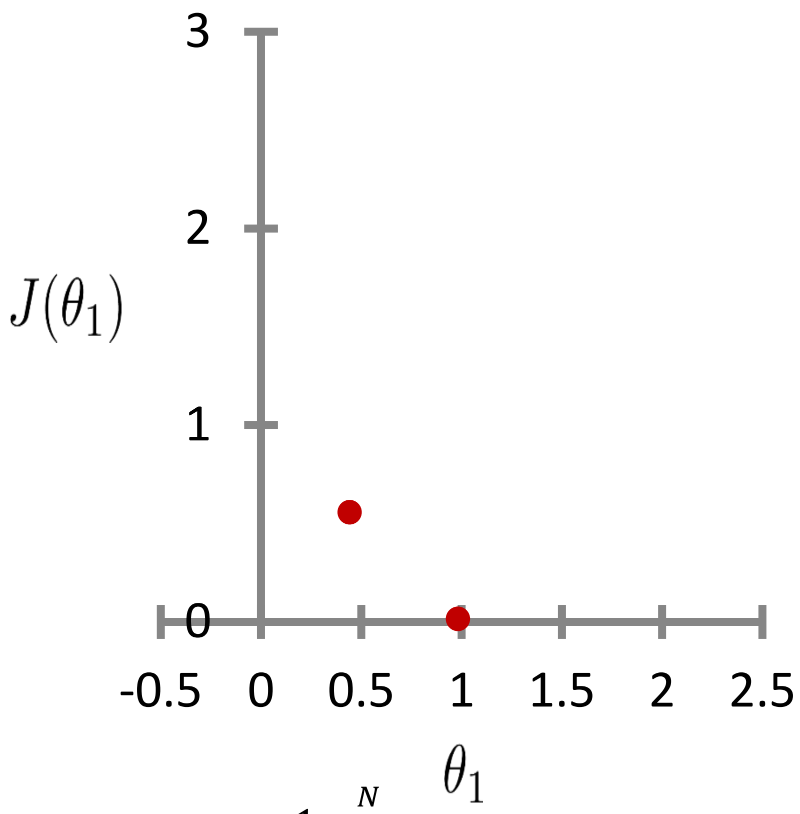
(for fixed θ_1, this is a function of x)



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

$J(\theta_1)$

(function of the parameter θ_1)

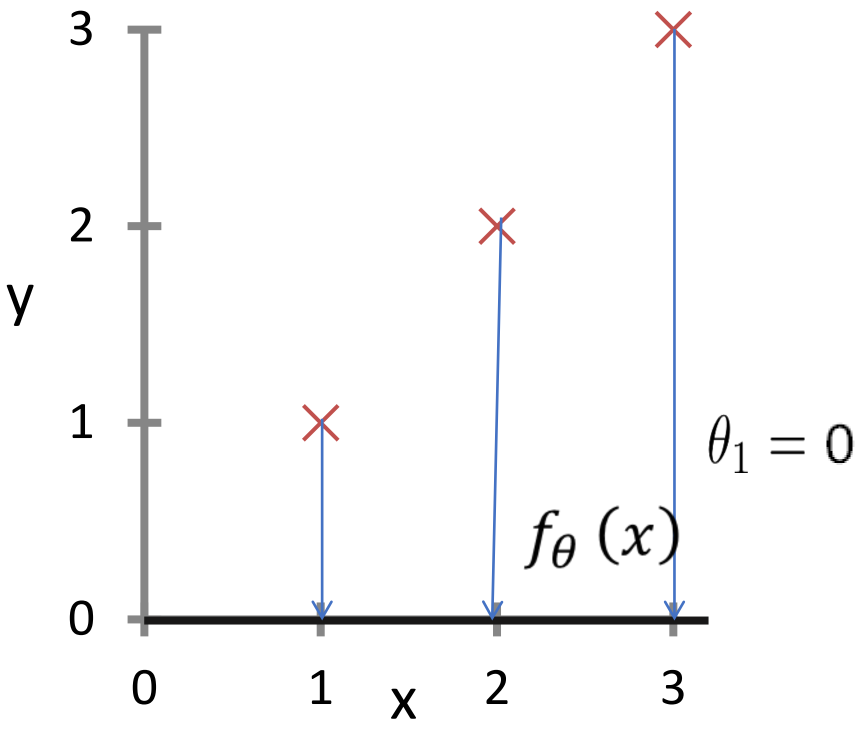


$$J(\theta_1) = \frac{1}{2N}\sum^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归 Linear regression with one variable

$f_\theta(x)$

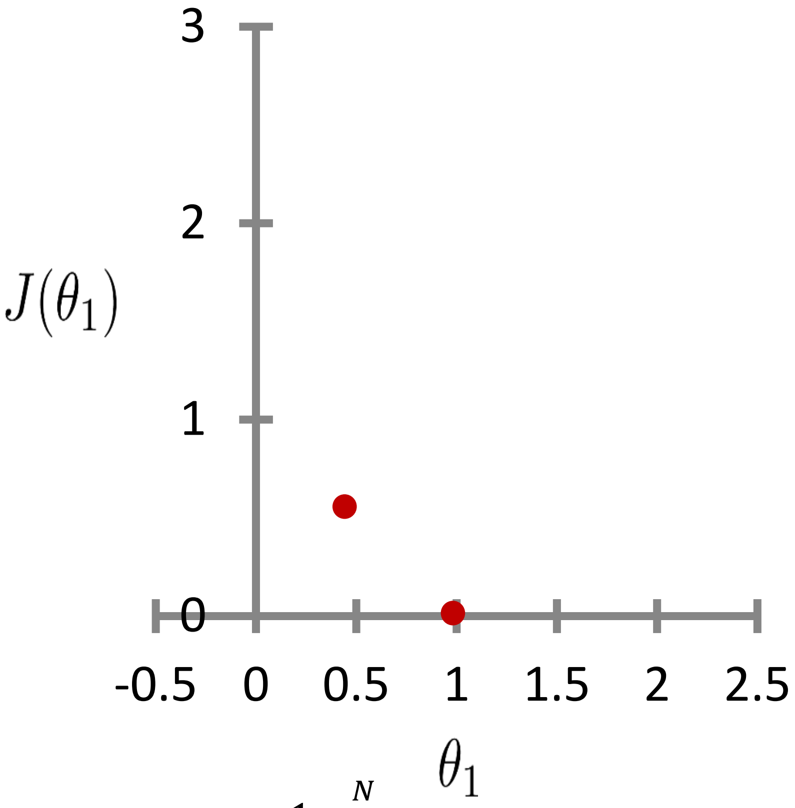
(for fixed θ_1 , this is a function of x)



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

$J(\theta_1)$

(function of the parameter θ_1)

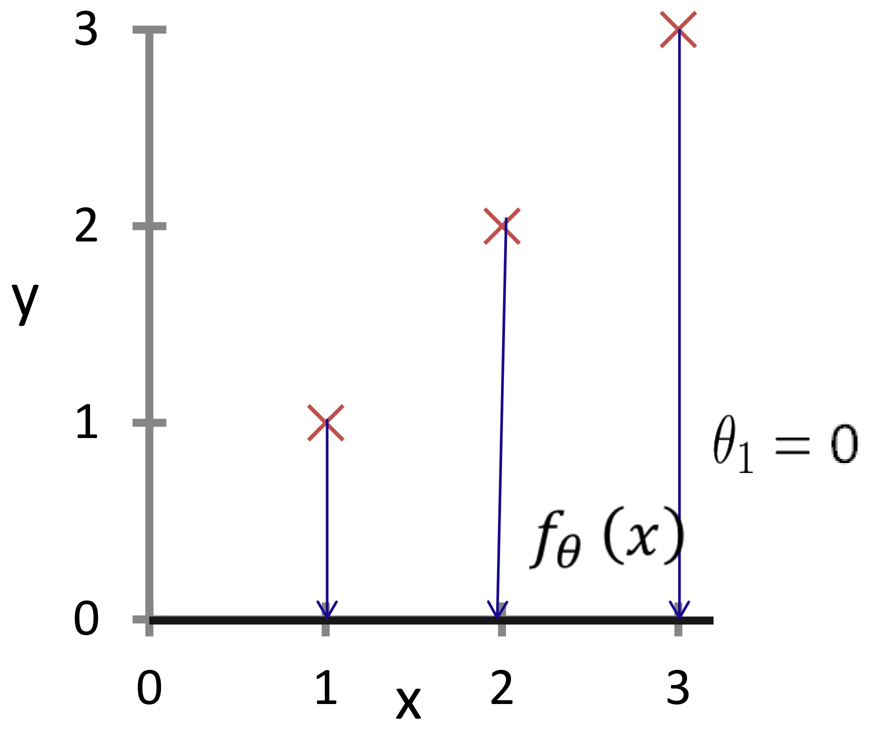


$$J(\theta_1) = \frac{1}{2N}\sum^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归 Linear regression with one variable

$f_\theta(x)$

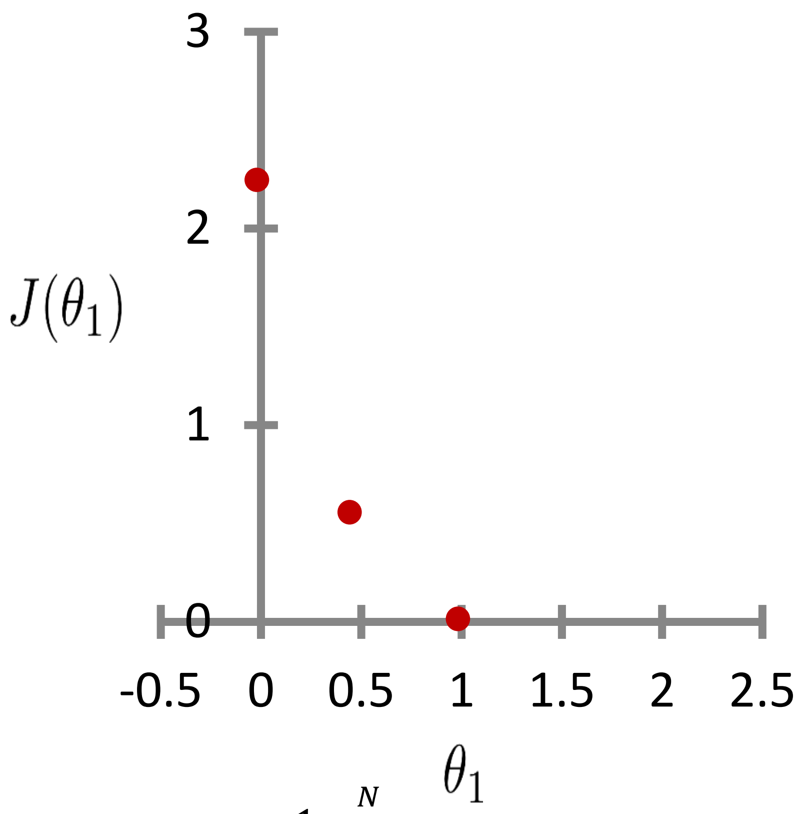
(for fixed θ_1, this is a function of x)



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

$J(\theta_1)$

(function of the parameter θ_1)

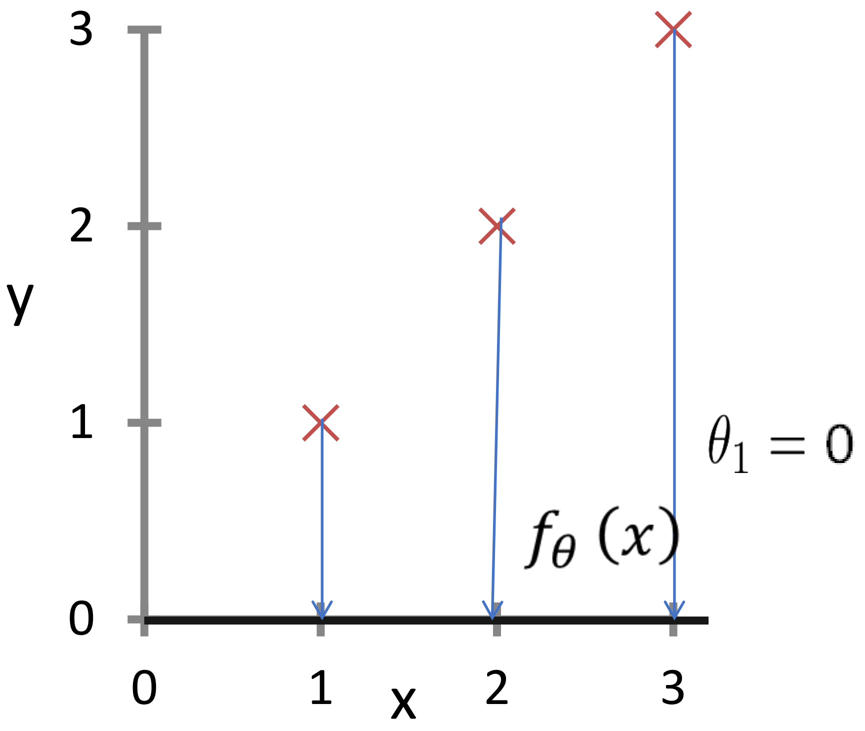


$$J(\theta_1) = \frac{1}{2N}\sum^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归 Linear regression with one variable

$f_\theta\,(x)$

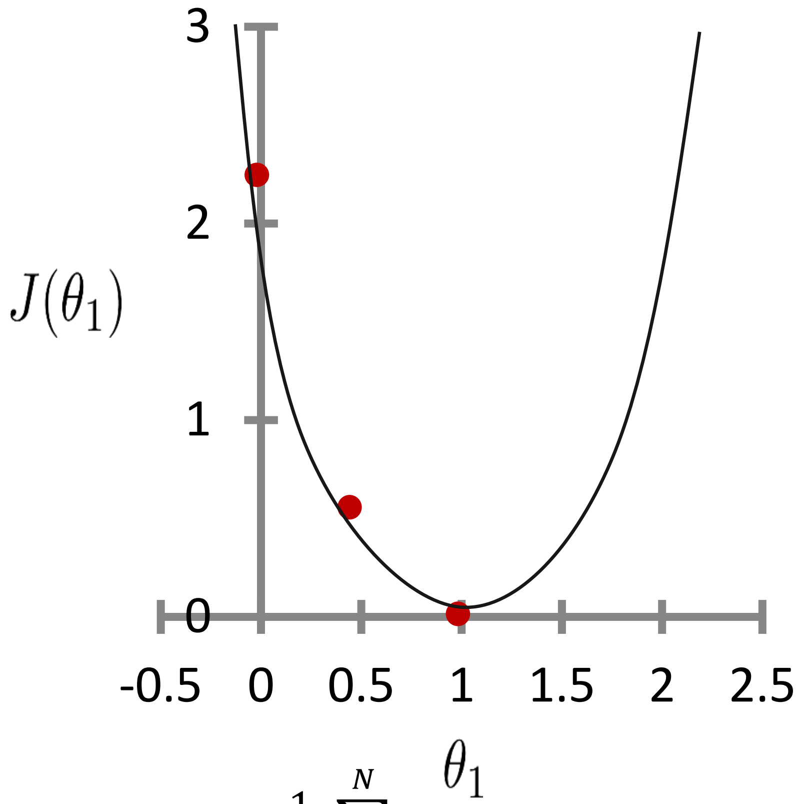
(for fixed θ_1, this is a function of x)



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

$J(\theta_1)$

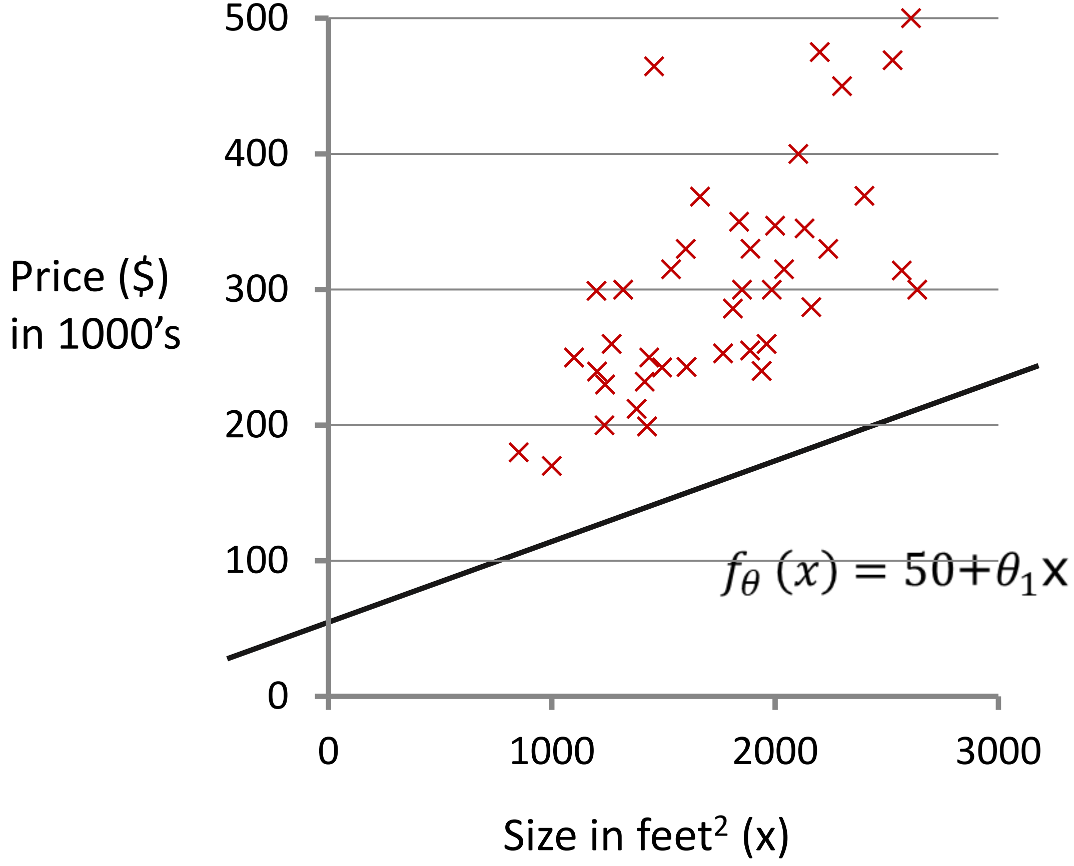
(function of the parameter θ_1)



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

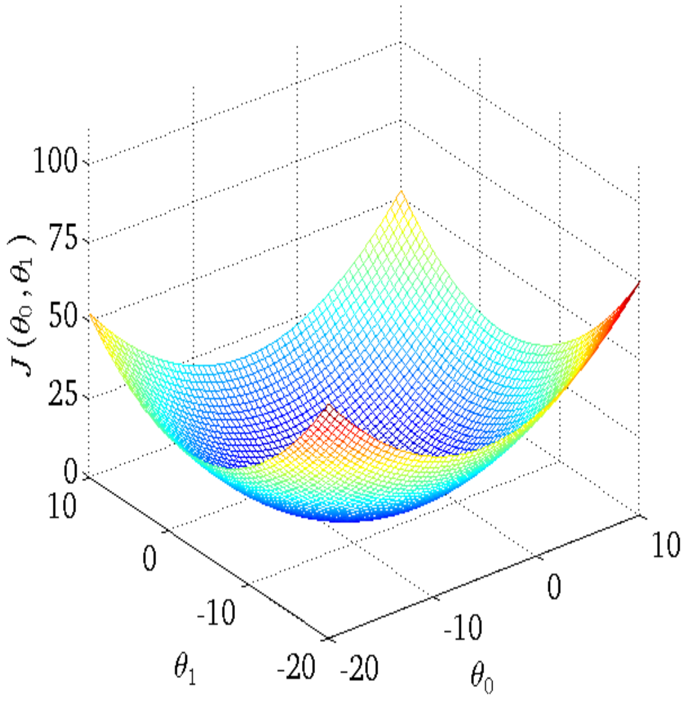
单变量线性回归 Linear regression with one variable

$f_\theta(x)$



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

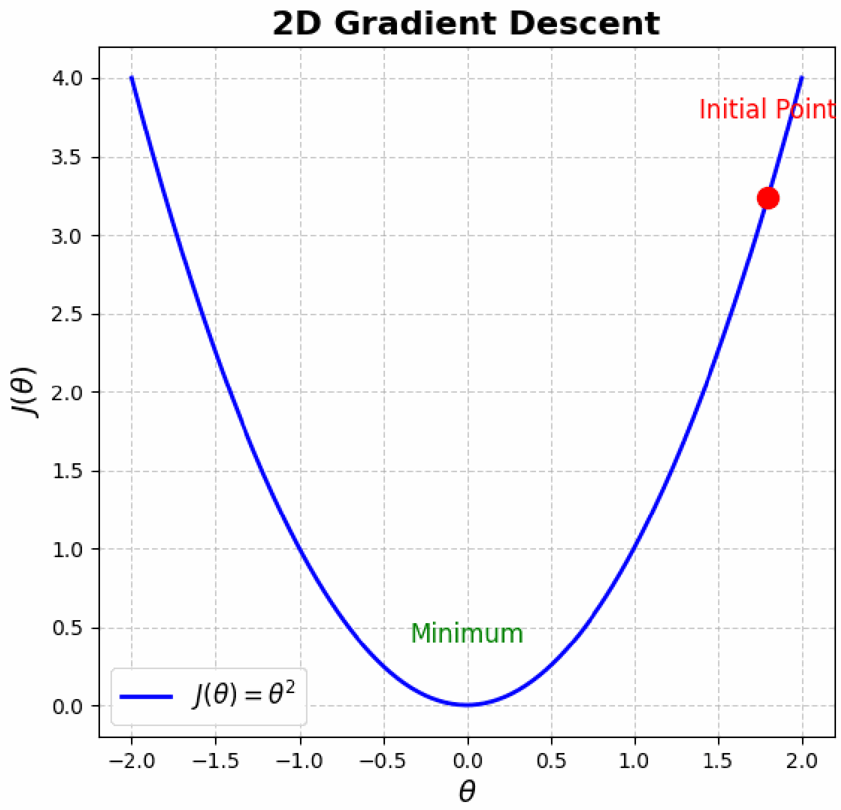
$J(\theta_0, \theta_1)$

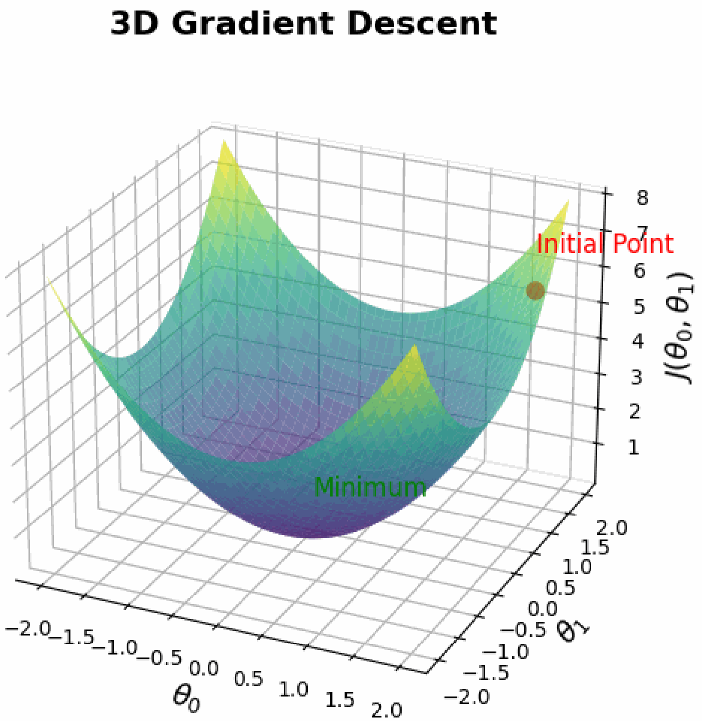


$$J(\theta_0, \theta_1) = \frac{1}{2N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})^2$$

单变量线性回归 Linear regression with one variable

$f_\theta(x)$ $J(\theta_0, \theta_1)$

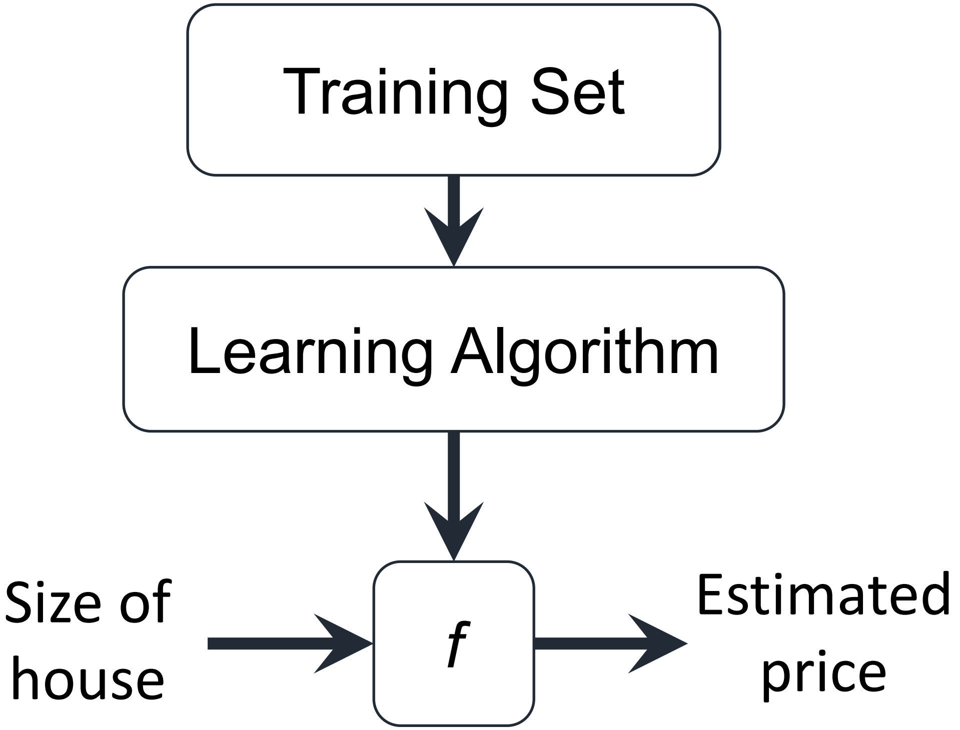




单变量线性回归 Linear regression with one variable

$f_\theta(x)$ $J(\theta_0, \theta_1)$

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:

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

单变量线性回归 Linear regression with one variable

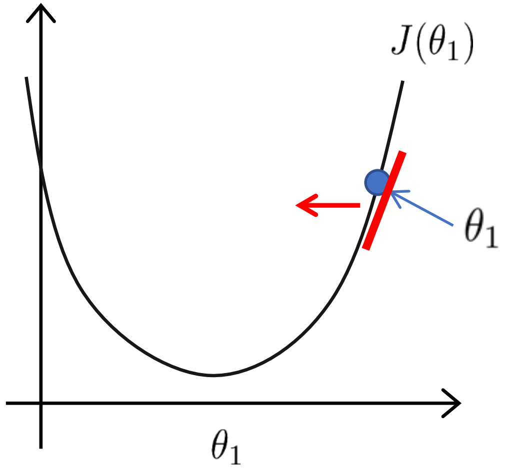
repeat until convergence {

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

}

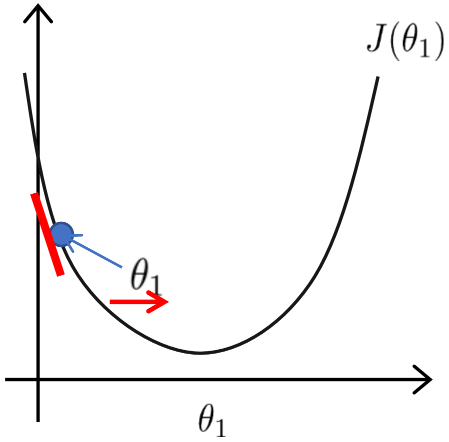
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} \underline{J(\theta_0, \theta_1)}$$

≥ 0



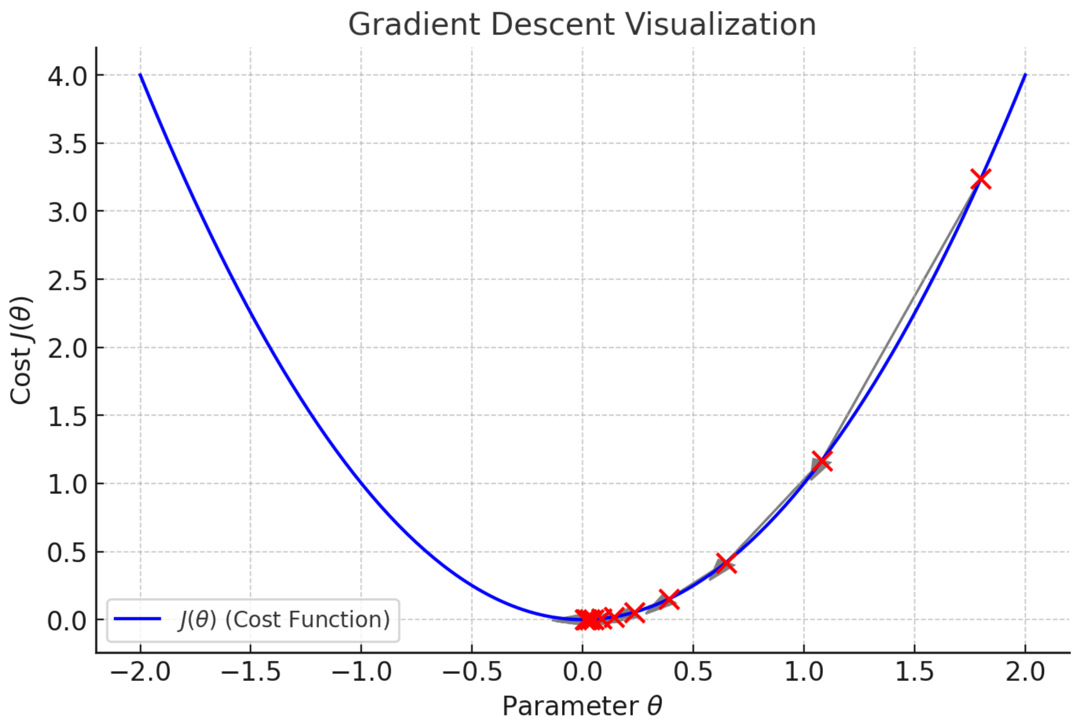
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} \underline{J(\theta_0, \theta_1)}$$

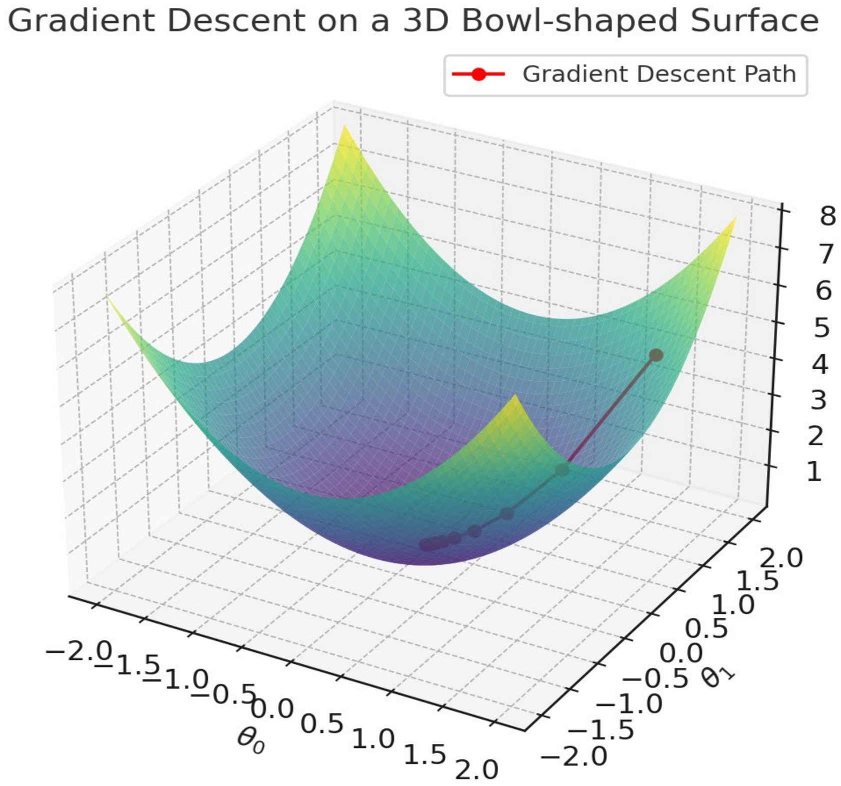
≤ 0



单变量线性回归 Linear regression with one variable

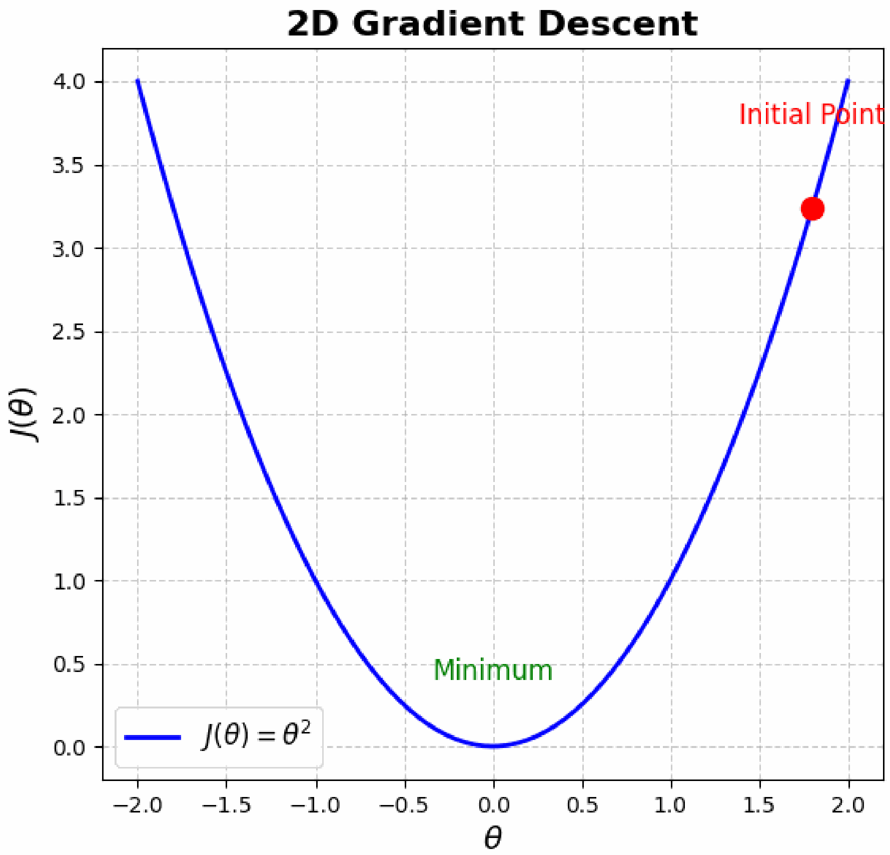
Gradient Descent

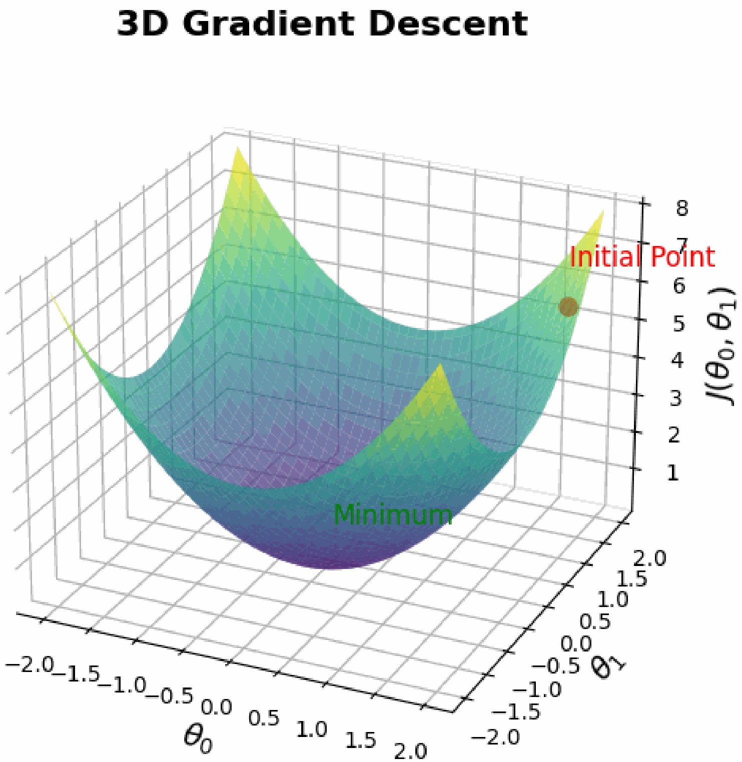




单变量线性回归 Linear regression with one variable

Gradient Descent





单变量线性回归 Linear regression with one variable

$$\text{repeat until convergence } \{$$

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

$$\}$$

单变量线性回归 Linear regression with one variable

$$\text{repeat until convergence } \{ \quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \}$$

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

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

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

单变量线性回归 Linear regression with one variable

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

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:

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

单变量线性回归 Linear regression with one variable

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

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

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

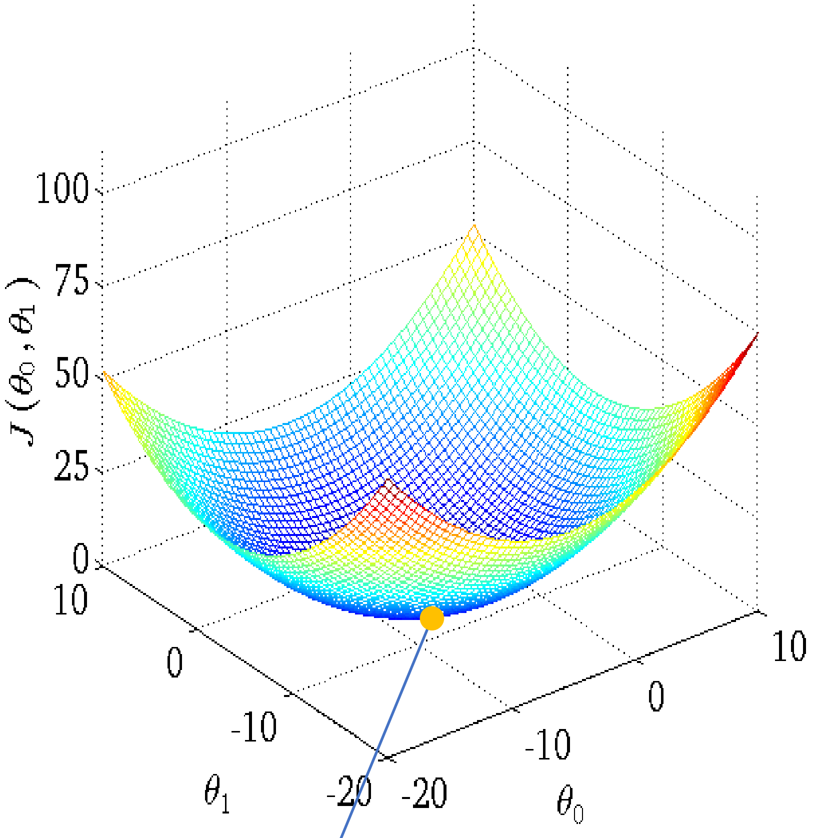
Repeat until convergece

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

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

update θ_0 and θ_1 simultaneously

凸函数 Bowled shape Convex Function

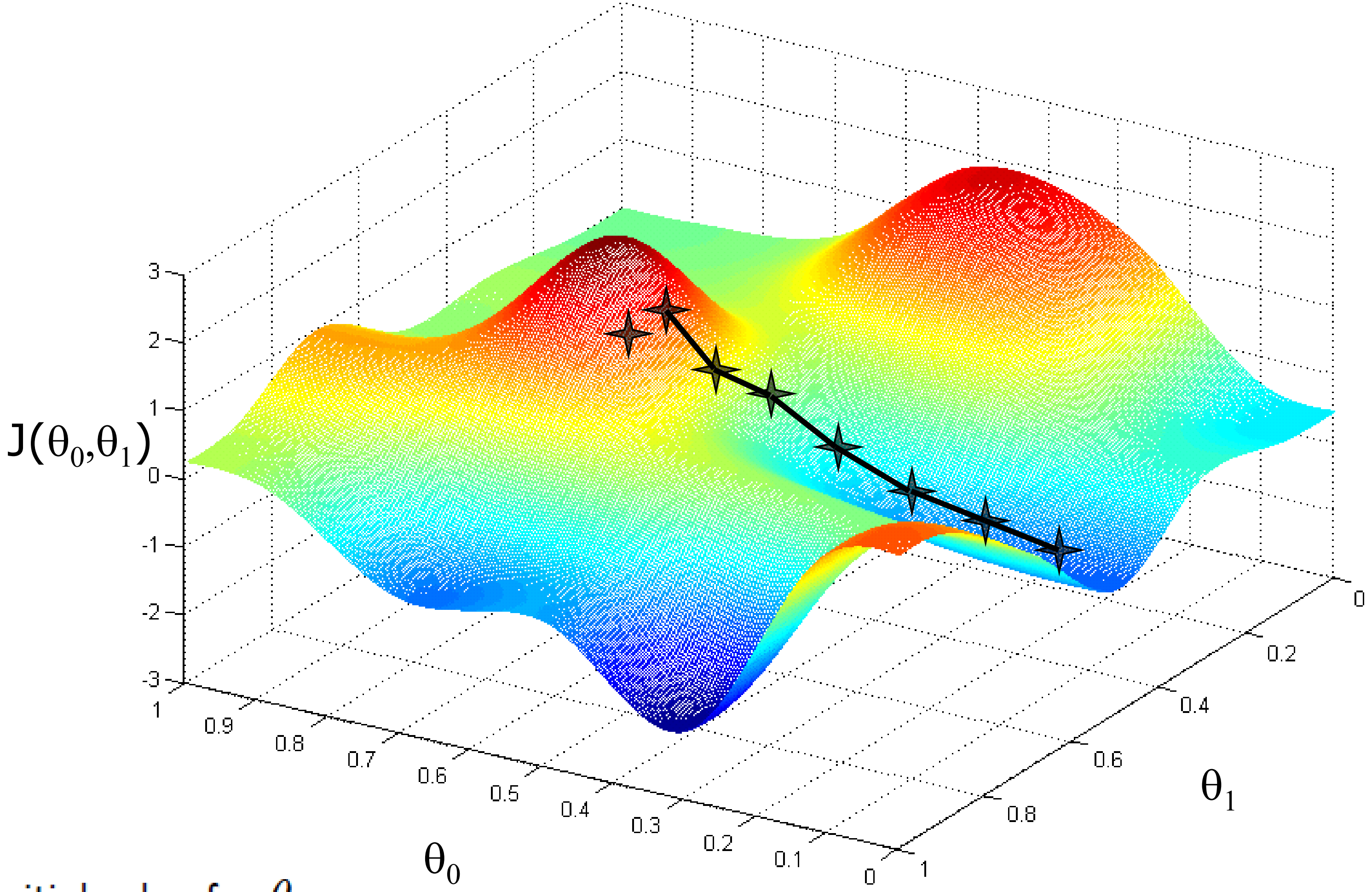


Unique Minimum

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

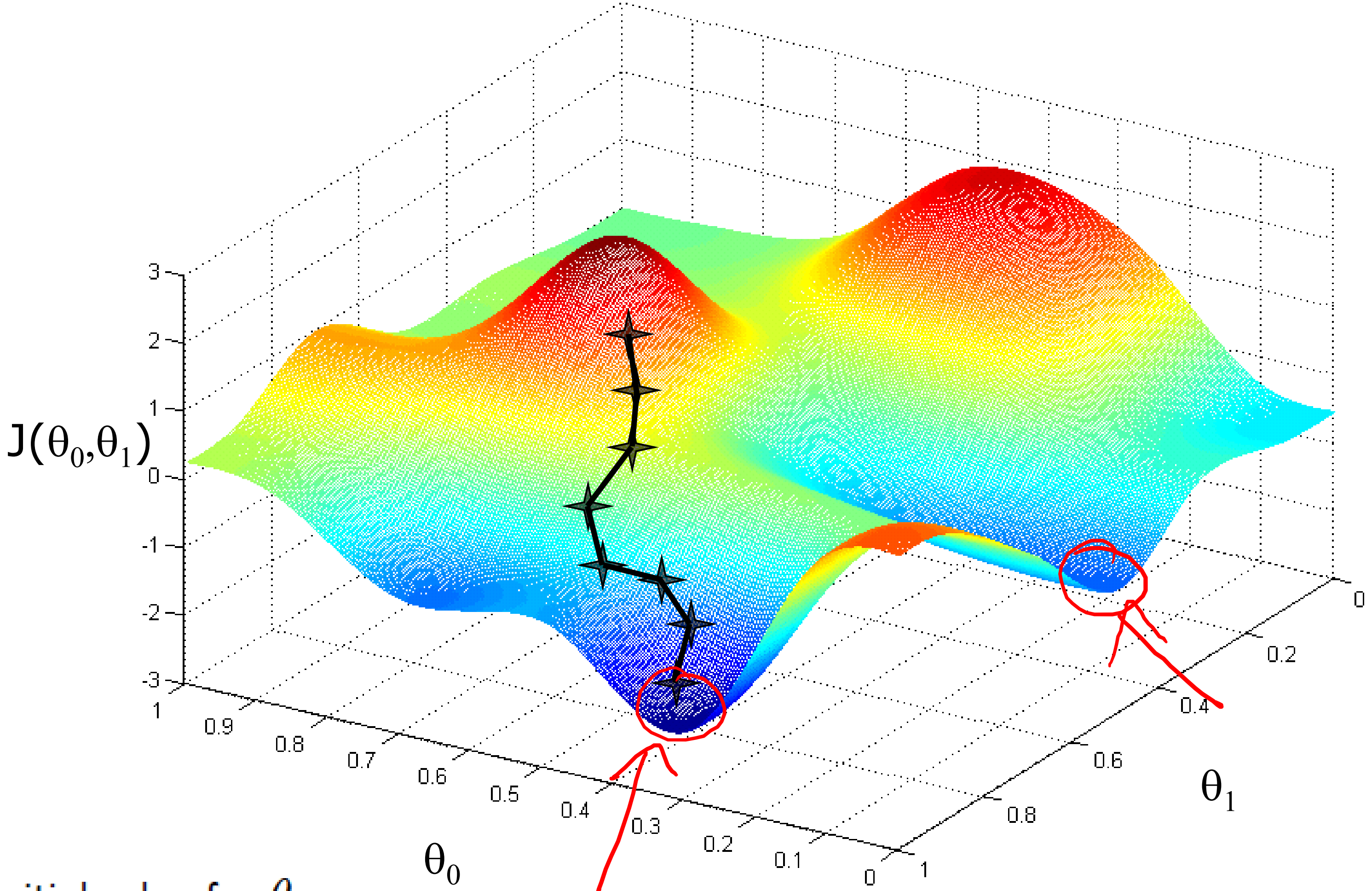
Different initial lead to the same optimum

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

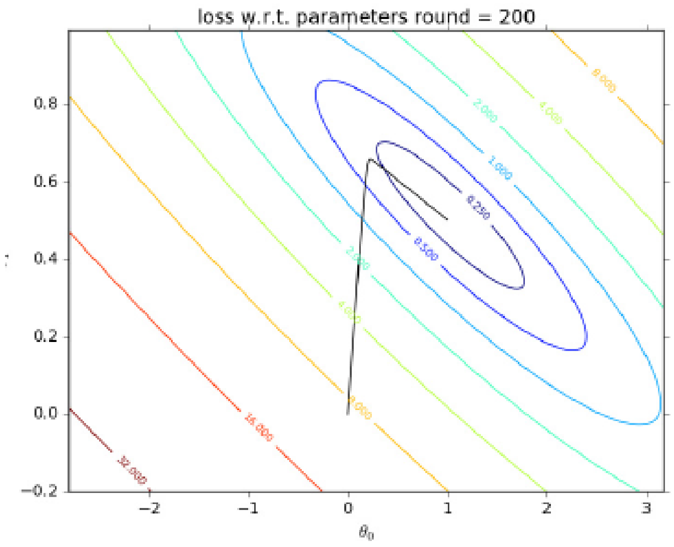
批量梯度下降 Batch Gradient descent

Each step of gradient descent uses all the training examples.

Update the parameters

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

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



随机梯度下降 Stochastic Gradient descent

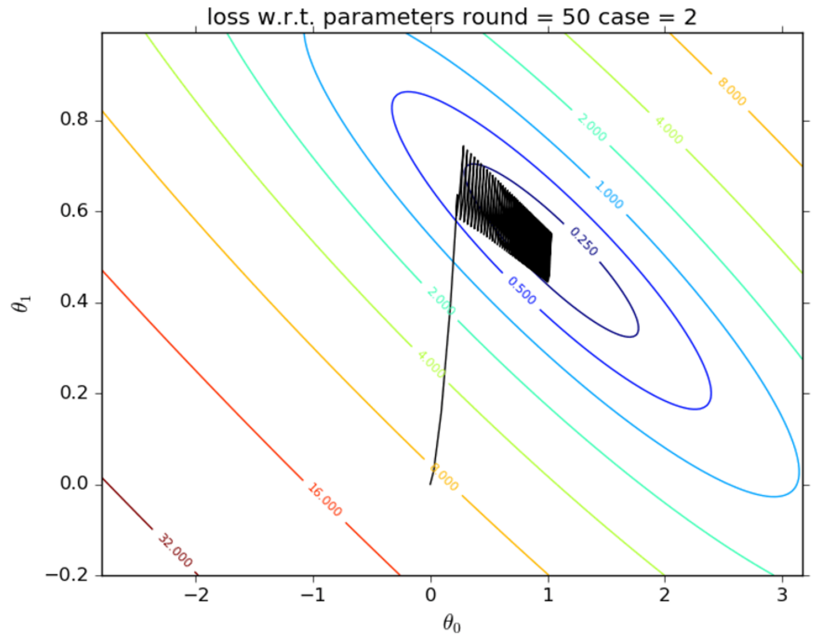
Each step of gradient descent uses single training example.

Iterate over the Dataset

Update the parameters

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

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



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

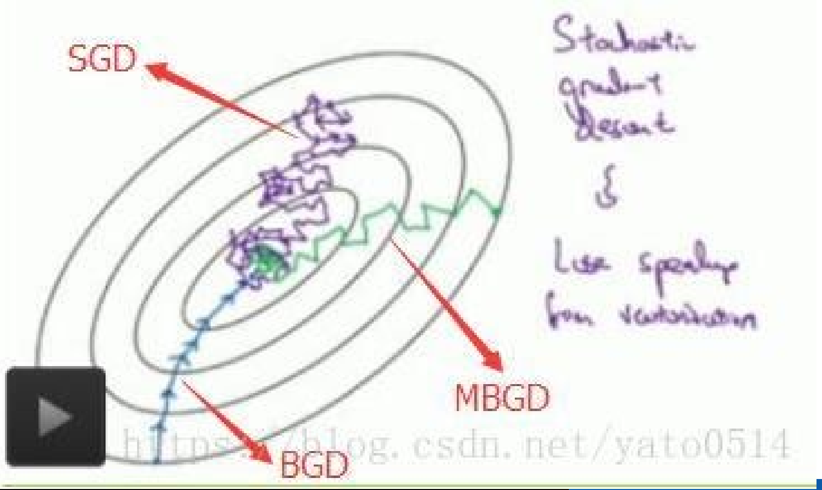
Split the whole dataset into k mini-batches.
Iterate over each mini-batch (once all mini-batches are processed, one epoch is complete)

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

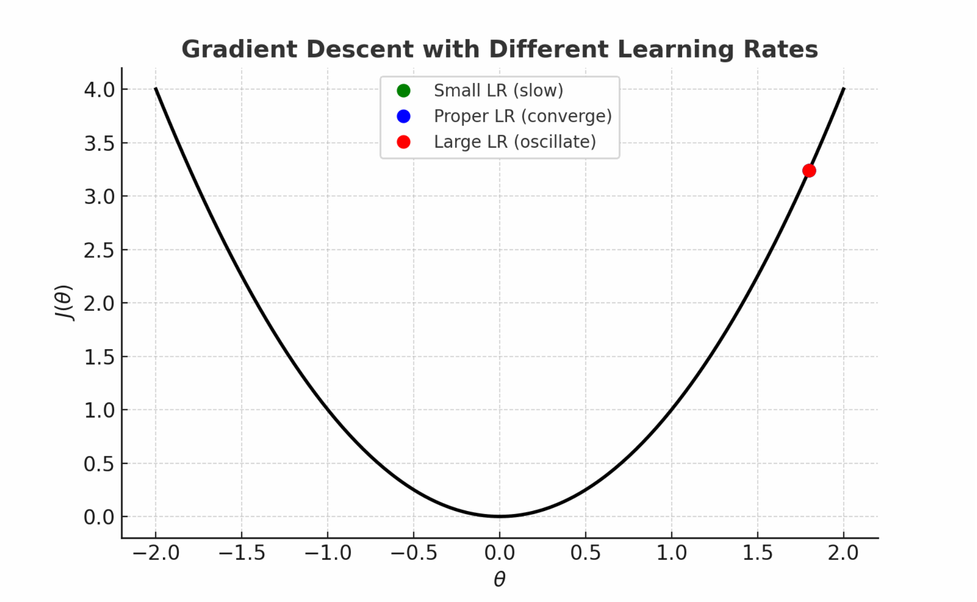
Update the parameters

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

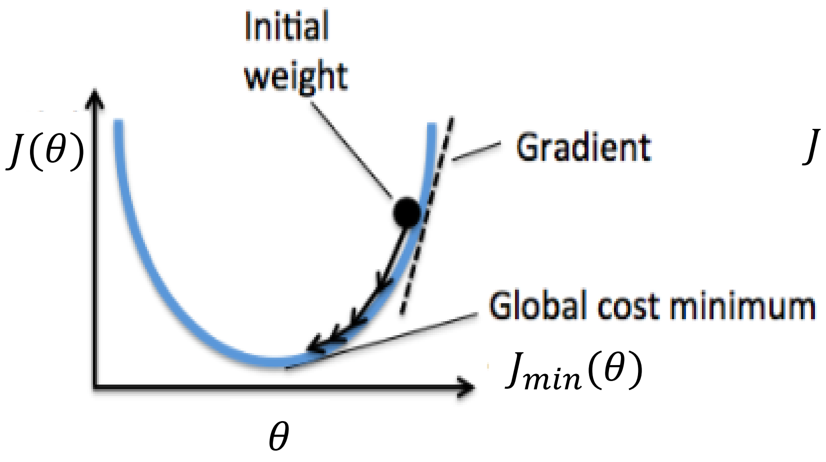
$$\theta_1 := \theta_1 - a\frac{1}{N_k}\sum_{i=1}^{N_k}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$



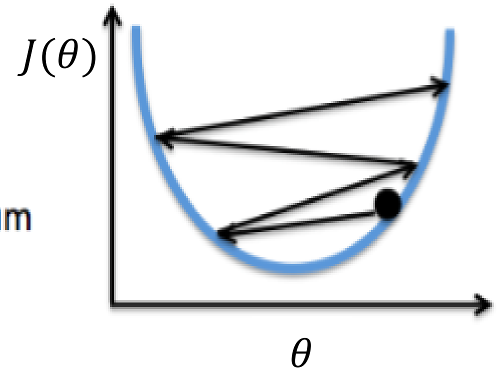
学习率选择 Choose learning rate



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

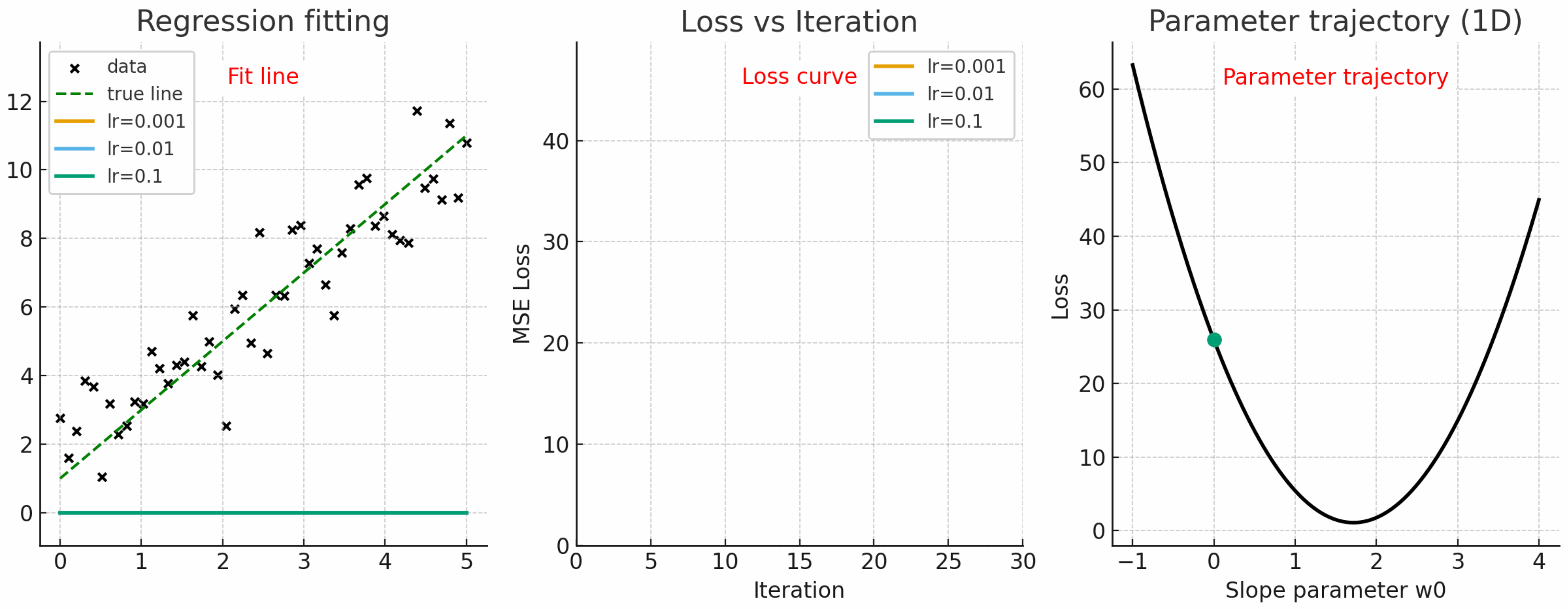


If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



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

学习率选择 Choose learning rate



多变量线性回归 Linear regression with multiple variable

Single features (variable)

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

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

Multiple features (variables)

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
…	…	…	…	…

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

$$= \theta_0 x_0 + \theta_1\, x_1 + \theta_2\, x_2 + \cdots + \theta_n\, x_n$$

$(x_0=1)$

单变量线性回归 Linear regression with one variable

Training Set

↓

Learning Algorithm

↓

Size of house → f → Estimated price

- 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:

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

Goal:

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

多变量线性回归 Linear regression with multiple variable

Hypothesis: $f_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$ (x_0=1)

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 $j = 0, \ldots, n$)

新的梯度 New Gradient Descent

Previously (n=1):

Repeat {

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

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

(simultaneously update θ_0, θ_1)

}

New algorithm (n≥1) ：

Repeat {

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

$j = 0, \ldots, n$

$(x_0=1)$

}

(simultaneously update θ_j)

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

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

$\cdots$

思　考

多变量线性回归相比单变量回归，可能会出现哪些问题？

思　考

多变量线性回归相比单变量回归，可能会出现哪些问题？

- Multicollinearity**多重共线性**

  When multiple predictors are highly correlated, the model becomes unstable.
  - Remove highly correlated variables
  - Regularization
  - Principal Component Analysis (PCA)
- Scaling Issues

  If features have significantly different scales, gradient descent algorithms can converge much more slowly
  - Standardization（标准化）
  - Normalization（归一化）

多变量线性回归 Linear regression with multiple variable

Single features (variable)

Size ($feet^2$)	Price ($1000)
2104	460
1416	232
1534	315
852	178
…	…

Multiple features (variables)

Size ($feet^2$)	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
…	…	…	…	…

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

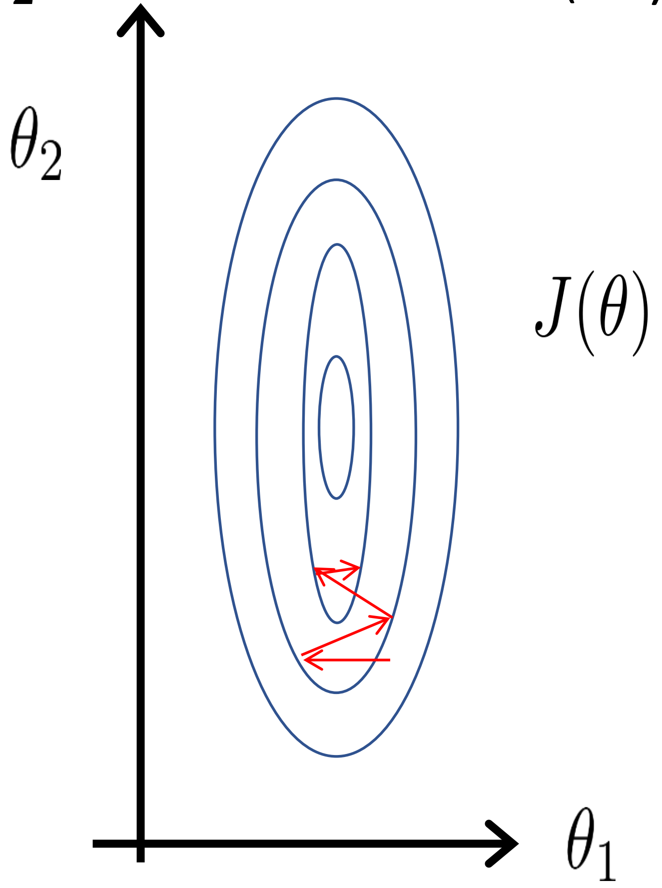
$$f_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

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 feet²)

x_2 = number of bedrooms (1-5)

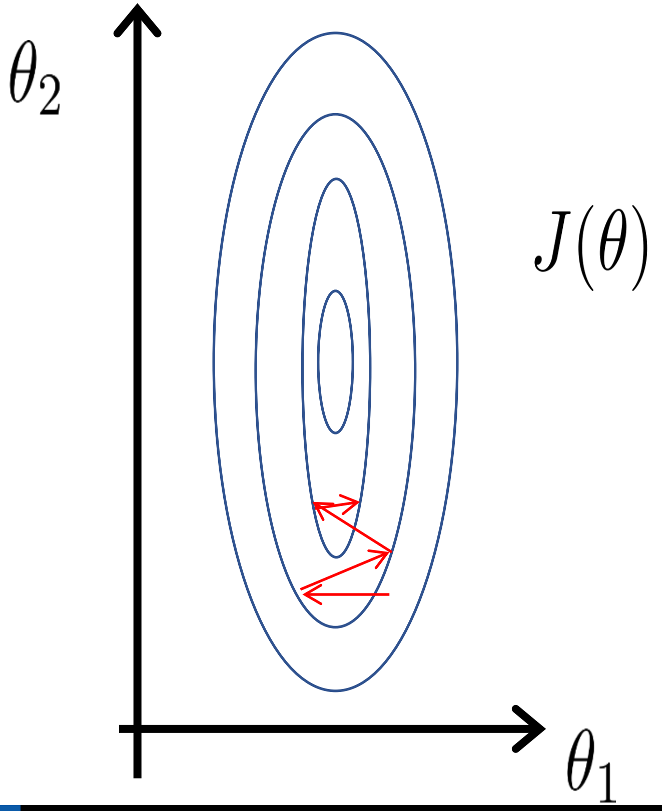


特征缩放 Feature Scaling

Idea: Make sure features are on a similar scale.

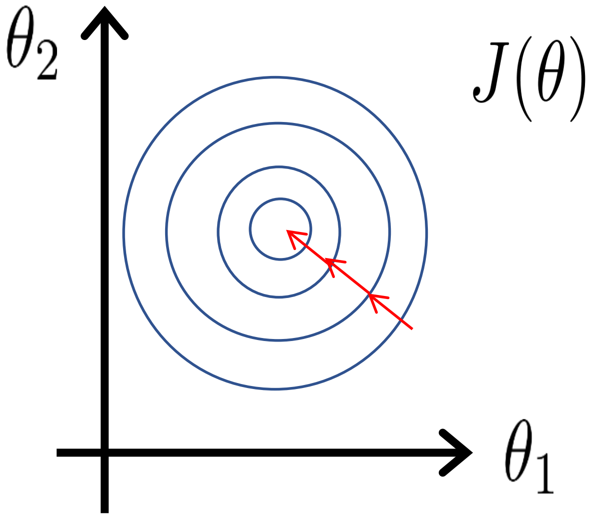
E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)



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

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



特征缩放 Feature Scaling

Get every feature into approximately a similar scale.

Normalization

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

Standardization

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

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

E.g. $x_1 = \frac{size-1000}{2000}$

$x_2 = \frac{\#bedrooms-2}{5}$

$$-0.5 \le x_1 \le 0.5, -0.5 \le x_2 \le 0.5$$

学习率 Learning rate

Previously (n=1):

Repeat {

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

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

(simultaneously update θ_0, θ_1)

}

New algorithm (n≥1) :

Repeat {

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

$j = 0, \ldots, n$

(x_0=1)

}

(simultaneously update θ_j)

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

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

$\ldots$

自适应学习率优化器 Adaptive learning rate optimizers

- **Momentum**

  accumulates past gradients to smooth the optimization process and accelerate convergence in consistent directions.

- **Adagrad**(Adaptive Gradient)

  adapts the learning rate for each parameter based on the cumulative sum of past gradient squares, making it suitable for sparse data.

- **RMSProp** (Root Mean Square propagation)

  adjusts the learning rate dynamically by using an exponentially decaying average of past gradient squares, preventing the learning rate from shrinking too much, and is well-suited for non-stationary objectives.

- **Adam** (Adaptive Moment Estimation)

  combines the benefits of both Momentum and RMSprop by considering both the first moment (mean) and the second moment (variance) of the gradients, making it widely applicable in deep learning.

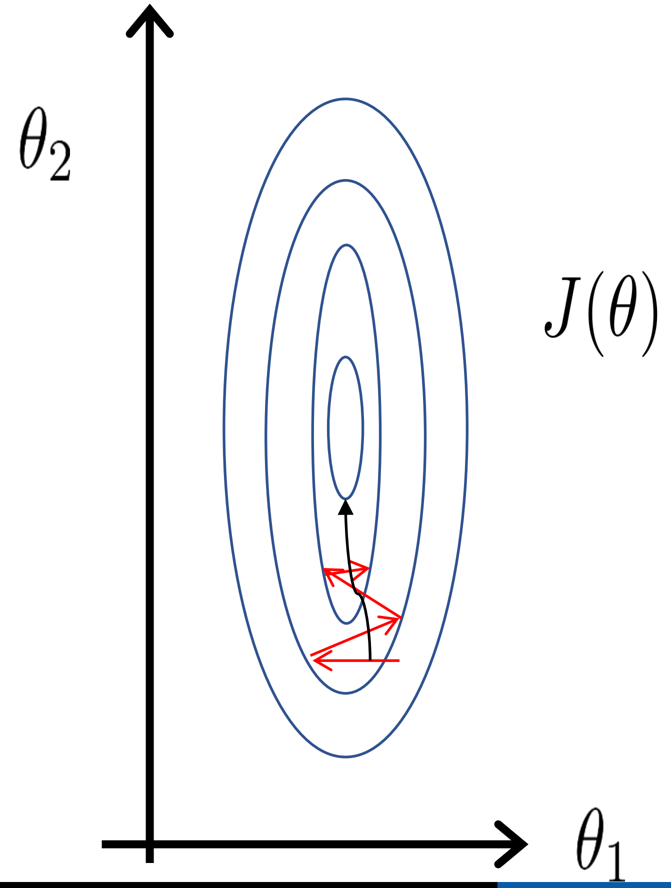
自适应学习率优化器 Adaptive learning rate optimizers

Adagrad

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

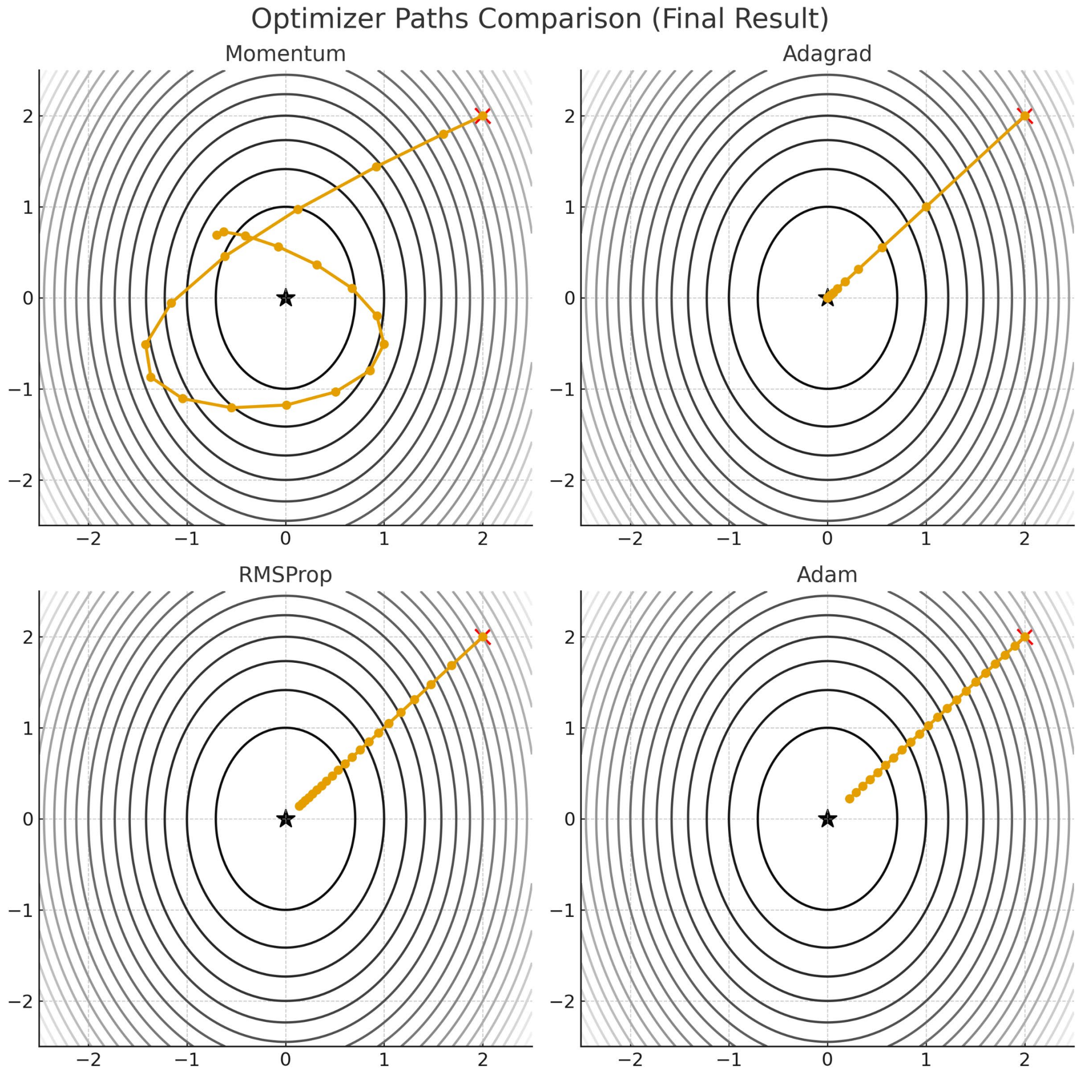
$$\theta^{(t+1)} := \theta^{(t)} - \frac{a}{\sqrt{\sum_{i=0}^{t}(g^{(i)})^2 + \varepsilon}} g^{(t)}$$

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



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

自适应学习率优化器 Adaptive learning rate optimizers



Momentum:路径有惯性，可能先震荡再稳定下来

Adagrad:步长逐渐缩小，后期收敛变慢

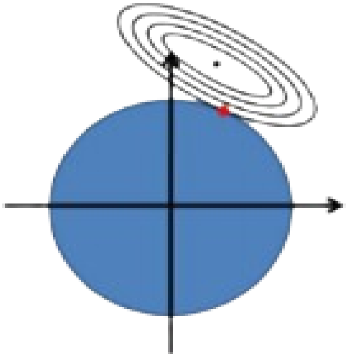
RMSProp:学习率保持稳定，路径平稳收敛

Adam:结合动量与自适应学习率，整体最快且最稳

正则化方法 Regularization

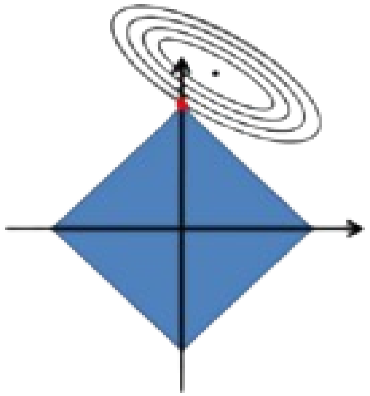
L2-Norm (Ridge):

$$\min_{\theta} \frac{1}{2N} \sum_{i=1}^{N} L(y^{(i)}, f_\theta(x^{(i)})) + \lambda \|\theta\|^2$$



L1-Norm (LASSO):

$$\min_{\theta} \frac{1}{2N} \sum_{i=1}^{N} L(y^{(i)}, f_\theta(x^{(i)})) + \lambda |\theta|$$



Elastic Net：

$$\min_{\theta} \frac{1}{2N} \sum_{i=1}^{N} L(y^{(i)}, f_\theta(x^{(i)})) + \lambda |\theta| \quad + \quad (1-\lambda) \ \|\theta\|^2$$

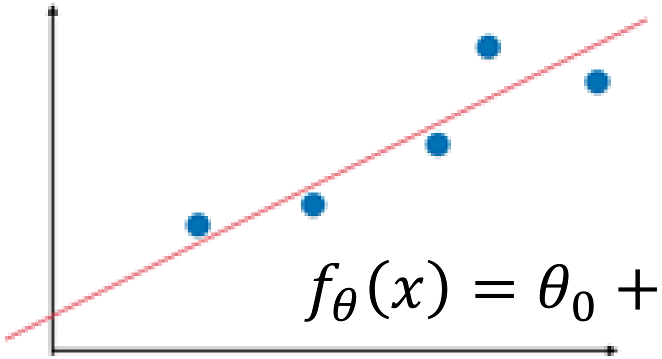
常见回归方法 Regression methods

- Linear Regression $$\min_{\theta} \frac{1}{2N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)})^2$$
- Ridge Regression $$\min_{\theta} \frac{1}{2N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\|\theta\|^2$$
- LASSO Regression $$\min_{\theta} \frac{1}{2N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)})^2 + \lambda|\theta|$$
- Elastic Net Regression $$\min_{\theta} \frac{1}{2N} \sum_{i=1}^{N} (f_\theta(x^{(i)}) - y^{(i)})^2 + +\lambda|\theta| + \ (1-\lambda)\ \|\theta\|^2$$

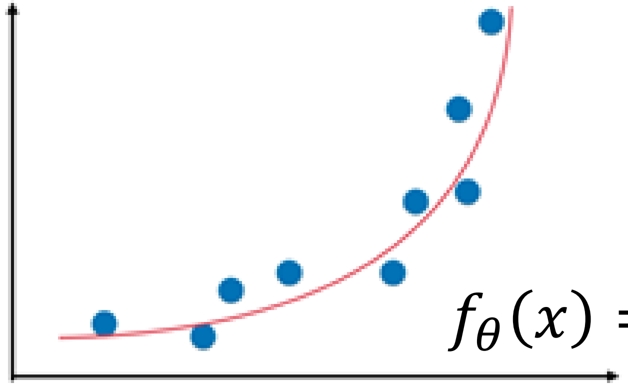
非线性回归方法 Nonlinear regression methods

- Polynomial Regression
- Decision Tree Regression
- Neural Networks
- Support Vector Regression, SVR
- Gradient Boosting Regression
- ….

多项式回归 Polynomial regression

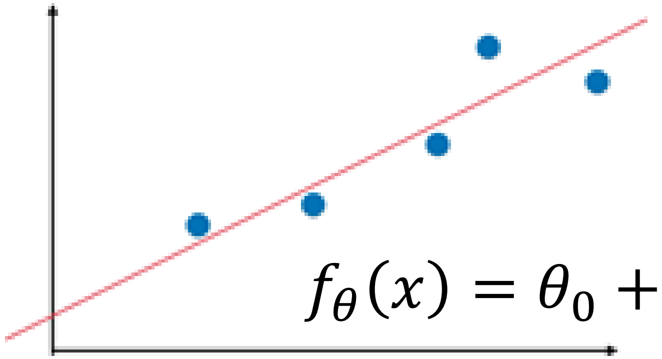


$$f_\theta(x) = \theta_0 + \theta_1 x_1$$

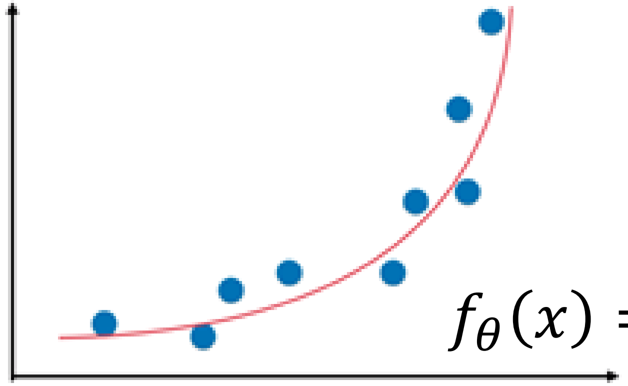


$$f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$$

多项式回归 Polynomial regression



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

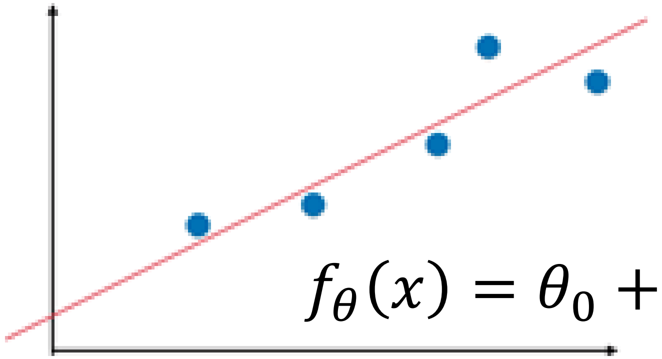


$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 \boxed{x_1^2}$$

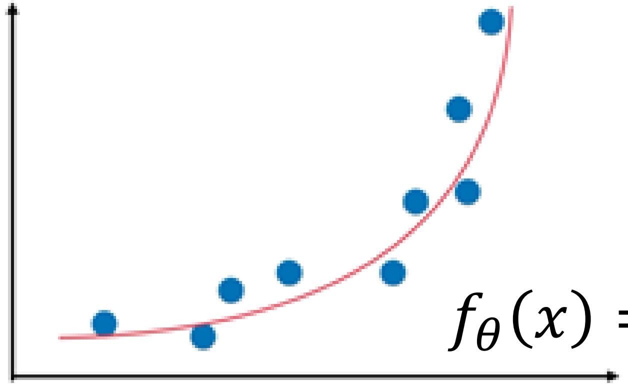
$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2\, \boxed{x_2}$$

Polynomial feature

多项式回归 Polynomial regression



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



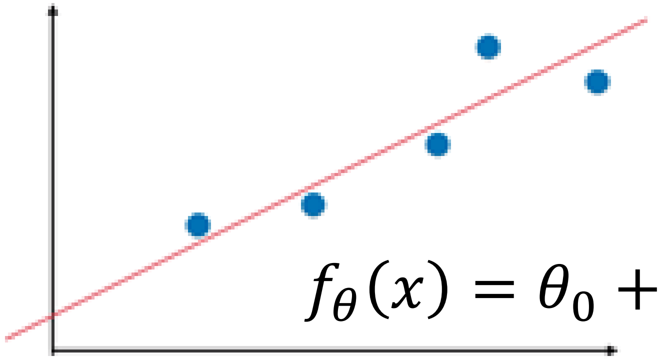
$$f_\theta(x) = \theta_0 + \theta_1\, x_1 + \theta_2 x_1^2$$

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

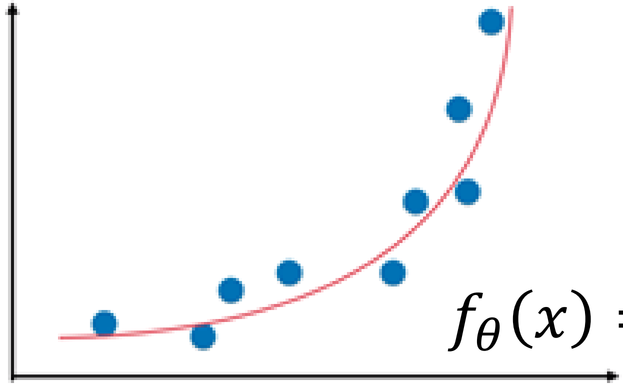
$$\Downarrow$$

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

多项式回归 Polynomial regression



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



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

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

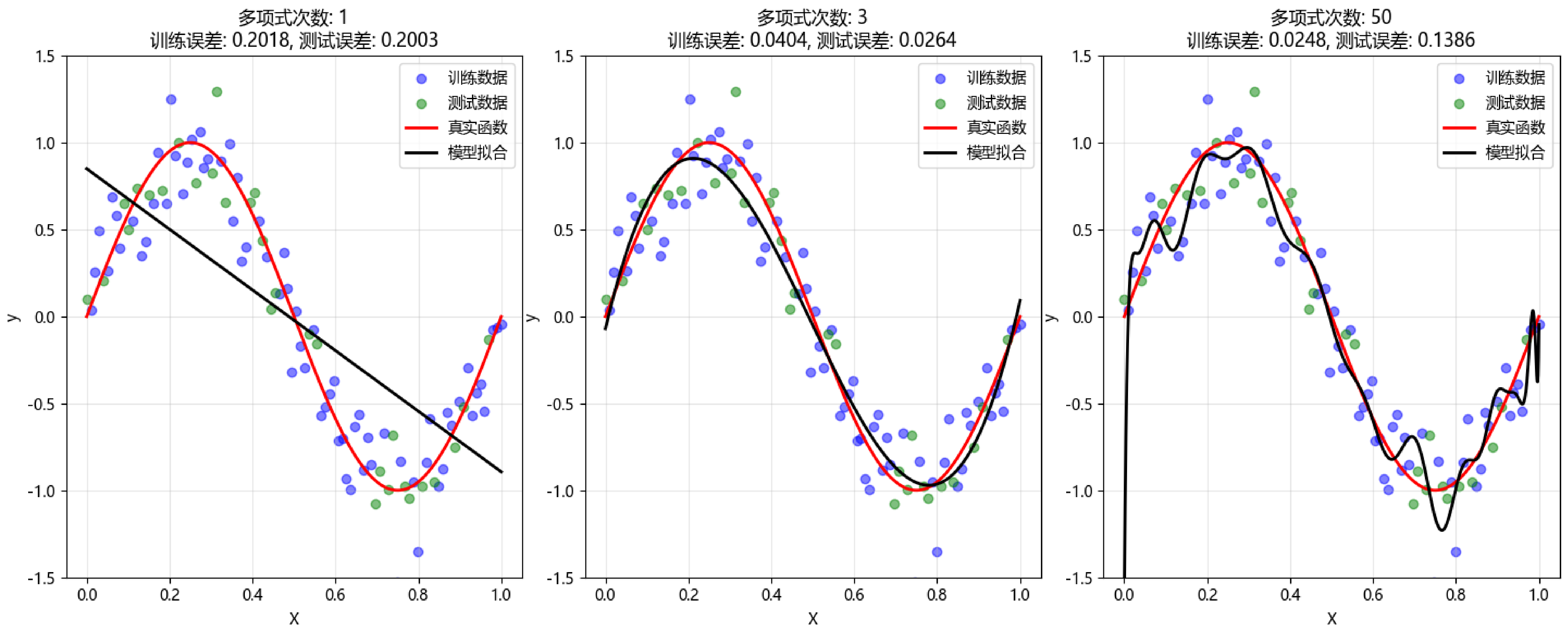
$$\Downarrow$$

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

✓ able to model all sorts of relationships

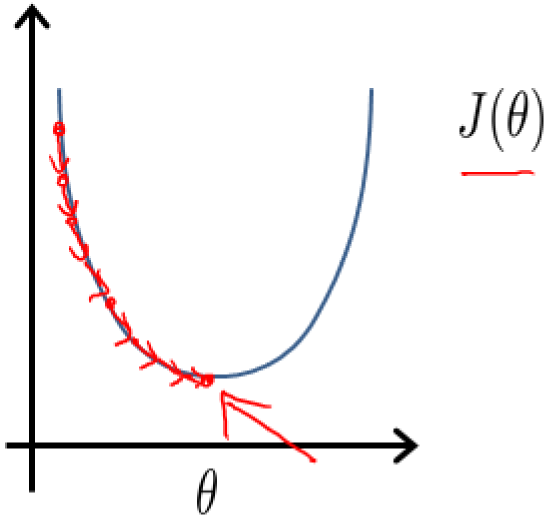
X easy to overfit

多项式回归 Polynomial regression



最小二乘法 Least square method

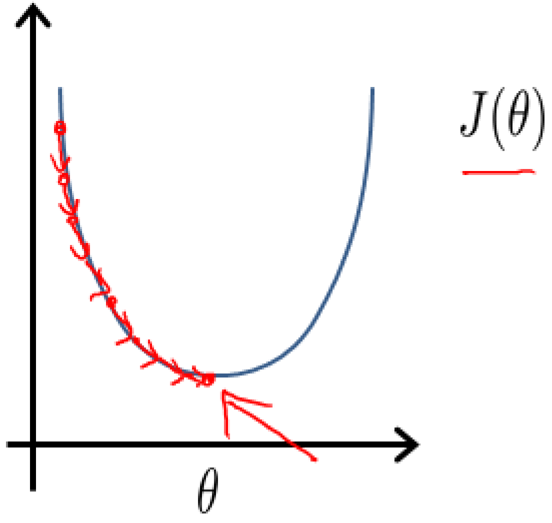
Gradient Descent



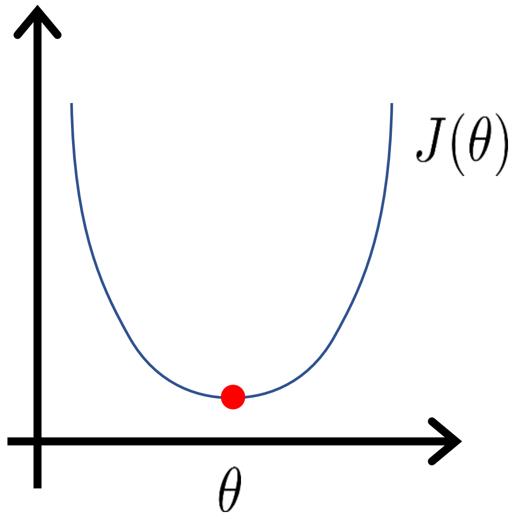
Other methods?

最小二乘法 Least square method

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

Examples:

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

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

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

最小二乘法 Least square method

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

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

- Gradient

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

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

$$=2X^T X\theta-2X^T y$$

- Solution

if $X^T X$ is invertible

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

最小二乘法 Least square method

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

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

- Gradient

$$\frac{\partial J(\theta)}{\partial\theta}=\frac{\partial}{\partial\theta}(\theta^TX^TX\theta-y^TX\theta-\theta^TX^Ty+y^Ty)$$

$$=\frac{\partial}{\partial\theta}(\theta^TX^TX\theta-2\theta^TX^Ty+y^Ty)$$

$$=2X^TX\theta-2X^Ty$$

3. 向量对向量的求导

假设 $\mathbf{a},\mathbf{b}\in\mathbb{R}^n$，则以下是几个常用的向量求导公式：

- $\frac{\partial(\mathbf{a}^T\mathbf{b})}{\partial\mathbf{a}}=\mathbf{b}$

4. 矩阵对向量的求导

假设 $\mathbf{X}\in\mathbb{R}^{n\times m}$，$\mathbf{a}\in\mathbb{R}^m$，则以下是常见的公式：

- $\frac{\partial(\mathbf{X}\mathbf{a})}{\partial\mathbf{a}}=\mathbf{X}^T$
- $\frac{\partial(\mathbf{a}^T\mathbf{X}\mathbf{a})}{\partial\mathbf{a}}=2\mathbf{X}\mathbf{a}$ （当 $\mathbf{X}$ 是对称矩阵时）

- Solution

if X^TX is invertible

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

最小二乘法 Least square method

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

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

$$\Rightarrow \quad (X^T X + \lambda \mathrm{I})\theta = X^T y$$

$$\Rightarrow \theta = (X^T X + \lambda \mathrm{I})^{-1} X^T y$$

Ridge Regression adds the regularization term, which ensures that even if $X^T X$ is not invertible, the model can still solve for the coefficients.

正规方程求解 Normal equation method

Examples:

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

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

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

最小二乘法 Least square method

Examples:

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

梯度下降 VS 最小二乘法

Gradient descent VS Least square method

- ■ Suitable for large-scale and high-dimensional data.
- ■ Flexible, can use adaptive optimizers.
- ■ Suitable for online or incremental learning.
- □ Iterative process can be slow. Requires tuning hyperparameters (learning rate, batch size, etc.).
- □ Improper learning rate may lead to non-convergence or slow convergence.

✓ Large-scale datasets.
✓ High-dimensional feature datasets

- ■ Direct computation, no need for iterations.
- ■ Yields the global optimal solution.
- ■ No need to tune hyperparameters.

learning.

- □ High computational complexity.
- □ Requires large memory.
- □ Not suitable for high-dimensional or large datasets.
- □ Sometimes cannot be directly calculated (if $X^T X$ is noninvertible).

✓ Small datasets.
✓ Datasets with few features.

回归评价标准 Regression Evaluation metrics

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

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

R-Squared（r2score）R方/决定系数

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