



DEEAO LEARNING  
迪 奥 教 育

# 迪奥数据分析实训营

## 梯度下降算法

# 课程学习目标



理解什么是梯度下降算法



理解梯度下降算法的相关问题



动手实现简单的梯度下降算法



使用梯度下降算法解决问题

# 目录

**01** 前情回顾

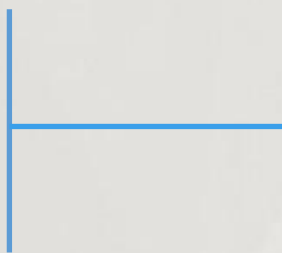
**02** 什么是梯度下降

**03** 实现简单的梯度下降

**04** 梯度下降的潜在问题

**05** 使用梯度下降算法

**06** 课后作业



01



# 前情回顾

# L11主要内容



线性回归



Ridge岭回归



Lasso套索回归



弹性网络

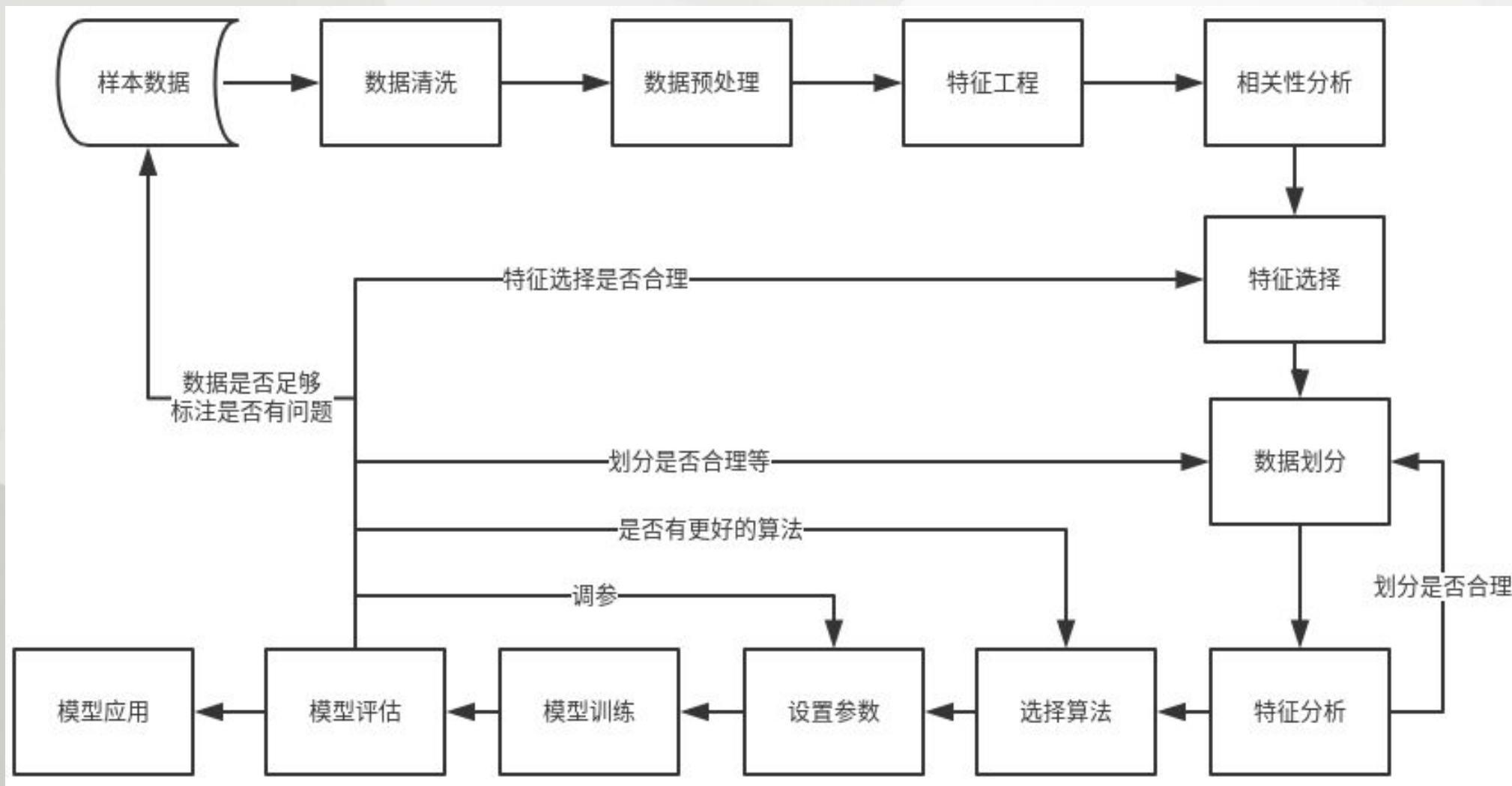


SVM回归



建立和优化波士顿房价预测模型

# 机器学习的工作过程





# AI薪资水平

企业	薪酬	岗位
腾讯	60w+和北京户口	机器学习基础研究
腾讯	80w+（深圳）	图像识别算法研究
百度	30-35k月薪	图像识别算法研究
微软	50-55w	机器学习基础研究
谷歌	50-55w	人工智能研究员
美团	32k月薪+北京户口（北斗计划）	机器学习基础研究
滴滴	50w+	研究员
滴滴	25k（新锐计划）	算法工程师
今日头条	30-35k+住房补助	AI Lab研究员
网易	45w	人工智能研究员
华为	50w（25k月薪+奖金+补助）	算法研究员
大疆	35k	算法研究员
Face++	35-40k	Researcher
Face++	50w+	Researcher
商汤	35-40w	Researcher



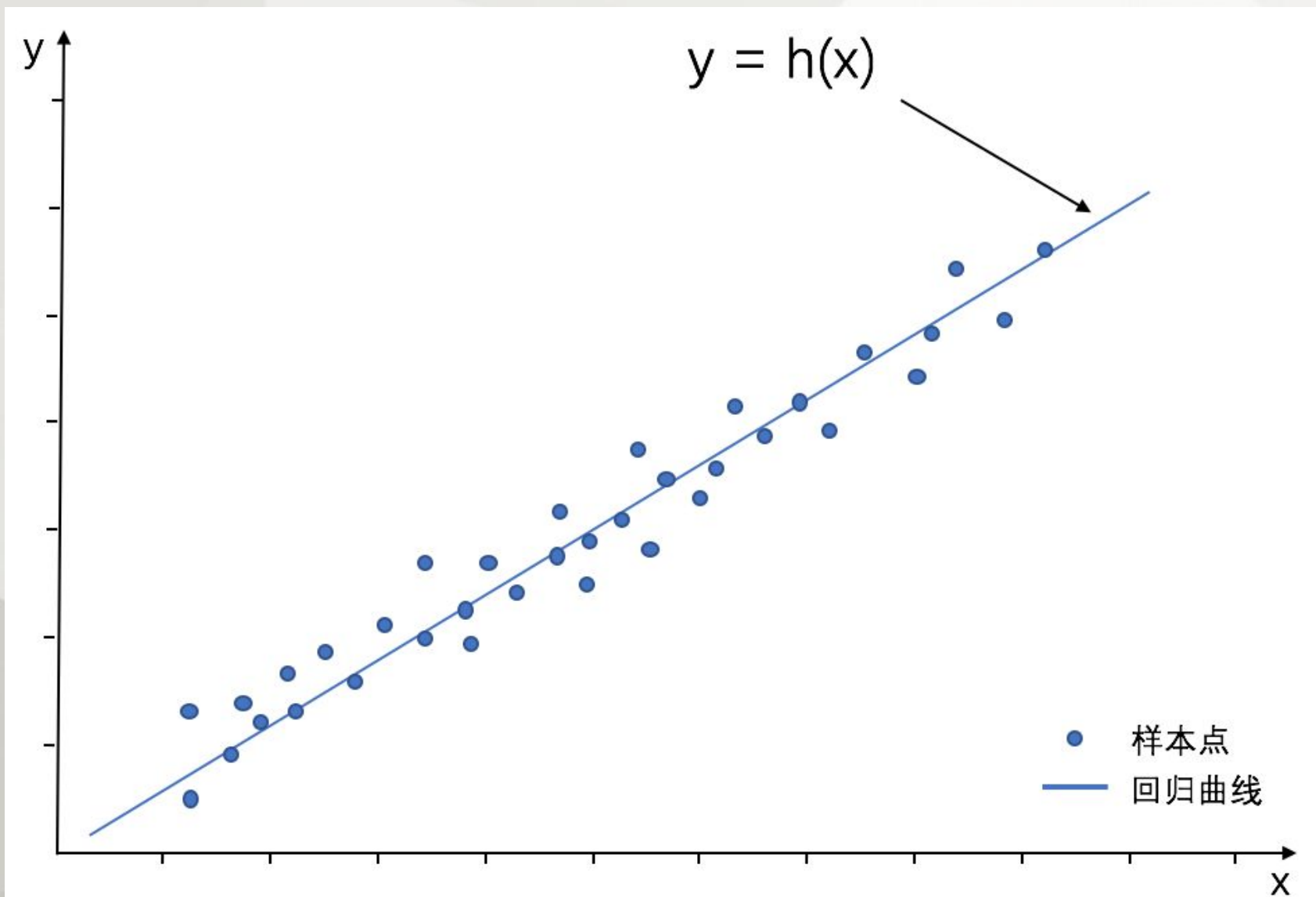
02



什么是梯度下降



# 线性回归模型



# 线性回归模型的数学表示

$$h_{\theta}(x) = \theta^T X = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

## 代价函数

计算建立的模型对真实数据的误差，叫建模误差（Modeling Error）。误差越低，模型对数据拟合度越高。例如给出：

m：训练集的样本个数

n：训练集的特征个数（通常每行数据为一个 $x(0)=1$ 与n个 $x(i)$  (i from 1 to n)构成，所以一般都会将x最左侧加一列“1”，变成n+1个特征)

x：训练集（可含有任意多个特征，二维矩阵，行数m，列数n+1，即 $x_0=1$ 与原训练集结合）

y：训练集对应的正确答案（m维向量，也就是长度为m的一维数组）

h(x)：我们确定的模型对应的函数（返回m维向量）

theta：h的初始参数（常为随机生成。n+1维向量）

得代价函数J(theta)：

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

# 共性问题

线性回归的损失函数:  $J(\theta) = \frac{1}{2m} \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right)^2$

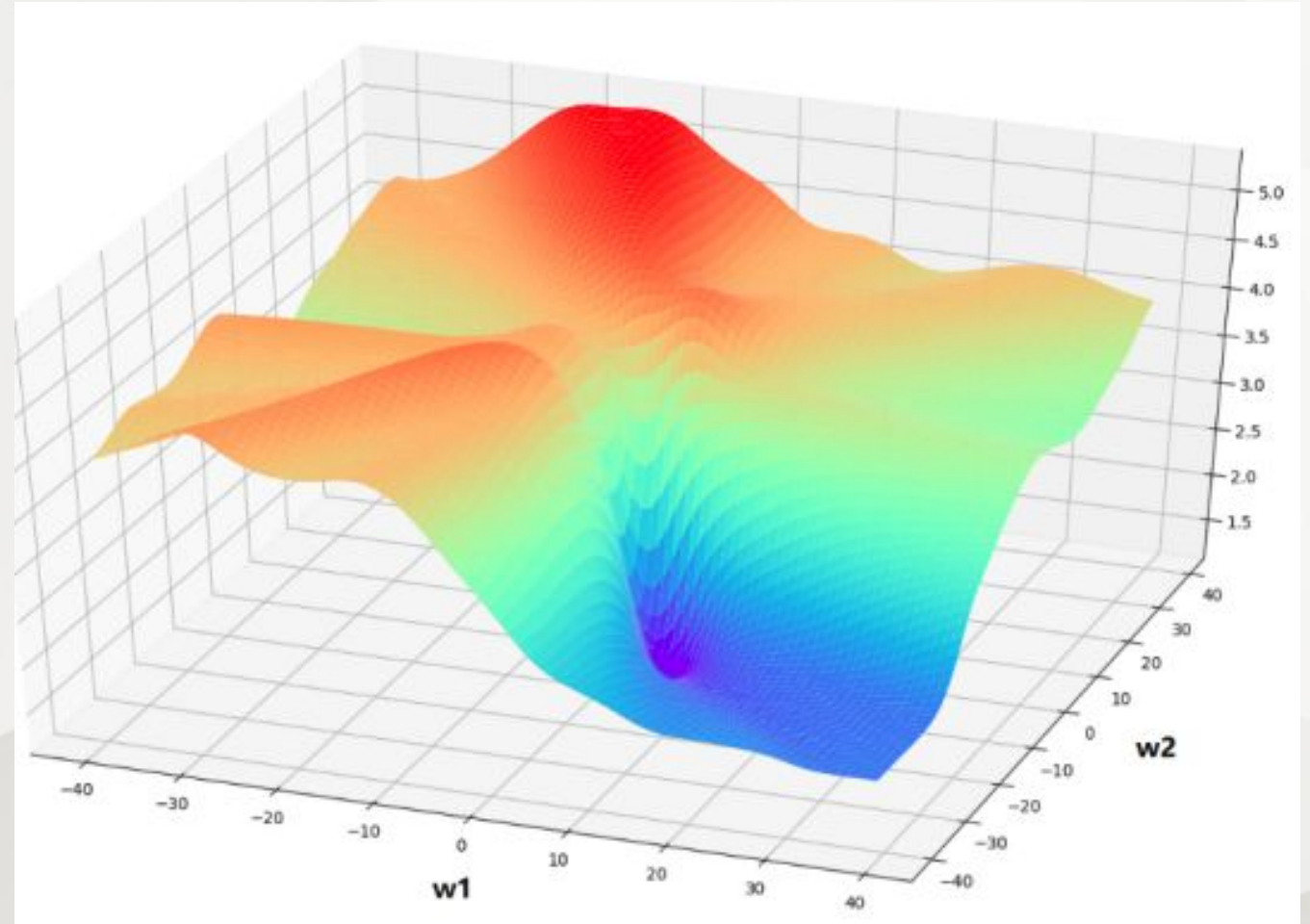
岭回归的损失函数:  $J(\theta) = \frac{1}{2m} \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^n \theta_j^2$

Lasso回归的损失函数:  $J(\theta) = \frac{1}{2m} \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^n |\theta_j|$

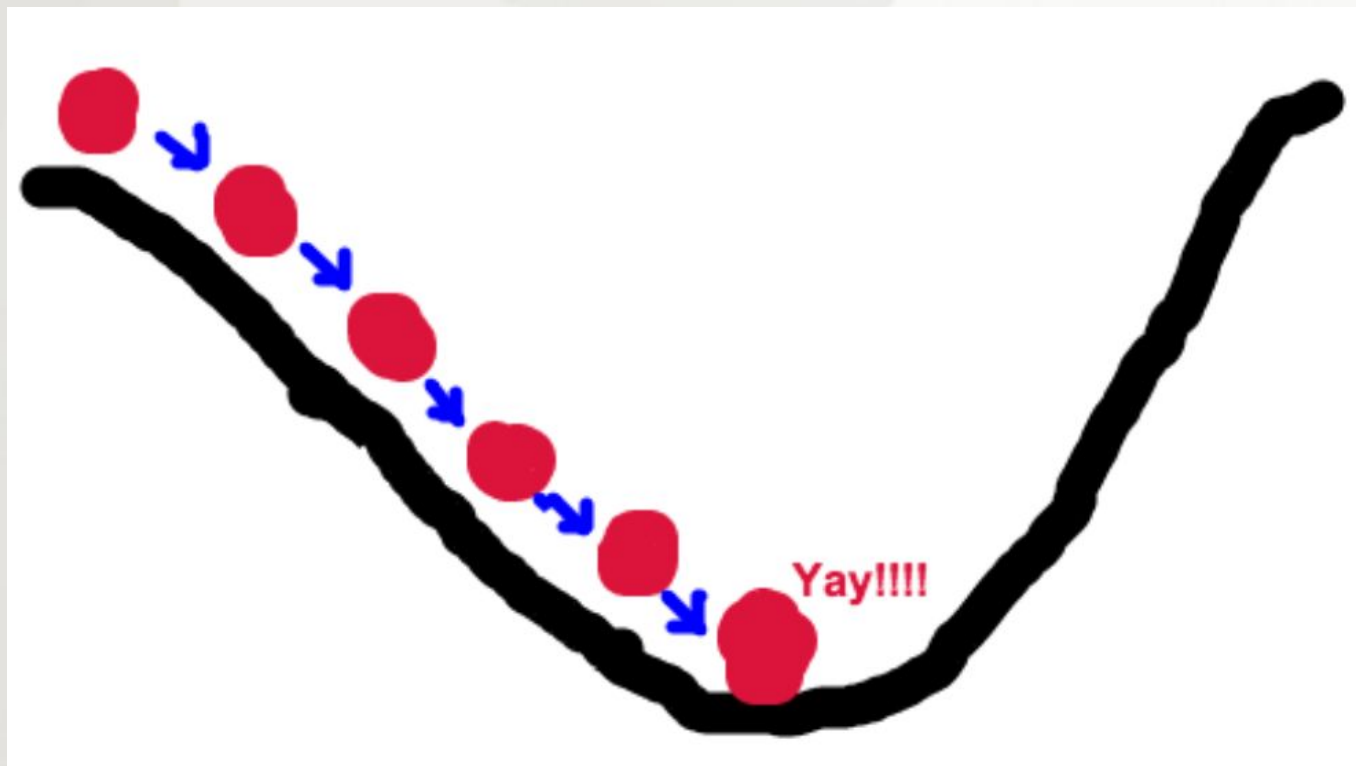
我们可以看到, 我们优化的目标是: **损失最小**。  
可是怎样才能损失最小呢?

# 下山问题

当你站在最高的山顶上，你要怎么才能下山？



# 简化的下山问题



可是问题来了：  
数学上怎么表达“向低的地方走”？

# 数学表达

The diagram illustrates the gradient descent update rule with the following components and callouts:

- current position**: A blue callout bubble pointing to  $\Theta^0$ .
- opposite direction**: A black callout bubble pointing to the minus sign.
- small step**: A green callout bubble pointing to the learning rate  $\alpha$ .
- direction of fastest increase**: A purple callout bubble pointing to the gradient  $\nabla J(\Theta)$ .
- next position**: A red callout bubble pointing to  $\Theta^1$ .

The equation is:  $\Theta^1 = \Theta^0 - \alpha \nabla J(\Theta)$  evaluated at  $\Theta^0$



03



简单实现梯度下降



# 手动演算

问题：

对于 $\text{loss} = w * (w - 9)$ ，当前 $w_0 = 1$ ，学习率为0.1，找到 $\text{loss}$ 的最小值。

对 $\text{loss}$ 求微分： $2 * w - 9$

$$w_1 = w_0 - 0.1 * (2 * w_0 - 9) = 1.7, \quad \text{loss} = -12.41$$

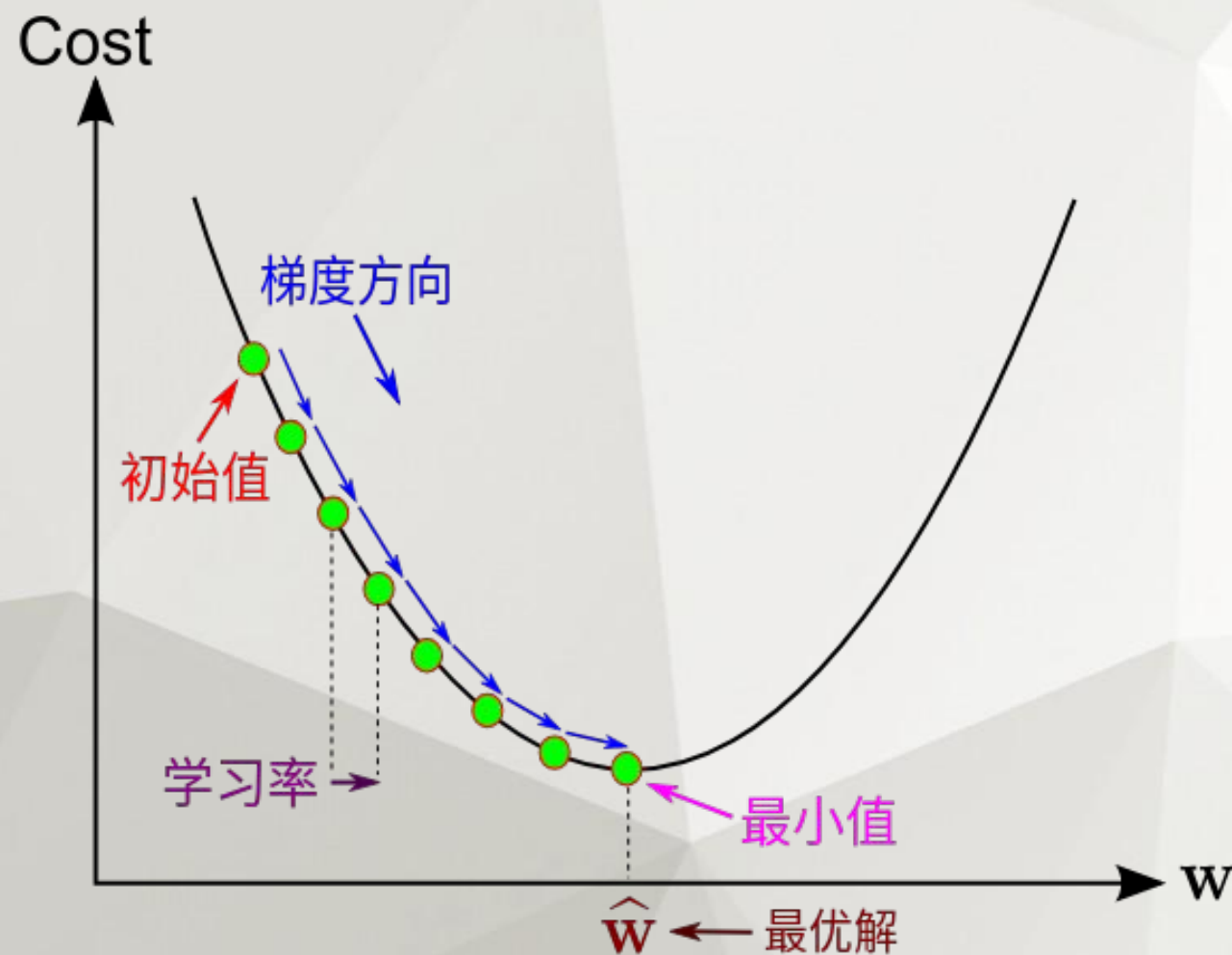
$$w_2 = w_1 - 0.1 * (2 * w_1 - 9) = 2.26, \quad \text{loss} = -15.23$$

$$w_3 = w_2 - 0.1 * (2 * w_2 - 9) = 2.71, \quad \text{loss} = -17.05$$

$$w_4 = 3.07, \quad \text{loss} = -18.21$$

$$w_5 = \dots\dots$$

# 手动演算



等等，对于单个特征的线性模型： $y=wx+b$ ，现在梯度下降计算出来 $w$ 了，可是这个 $b$ 呢？怎么不见了？

# 只有一个特征的线性模型

对于:  $y = a * x + b$

可以规范化为:  $y = a * x + b * x_0$ , 这里  $x_0 = 1$

在规范化一点:  $y = h(x) = w_1 * x_1 + w_0 * x_0$

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

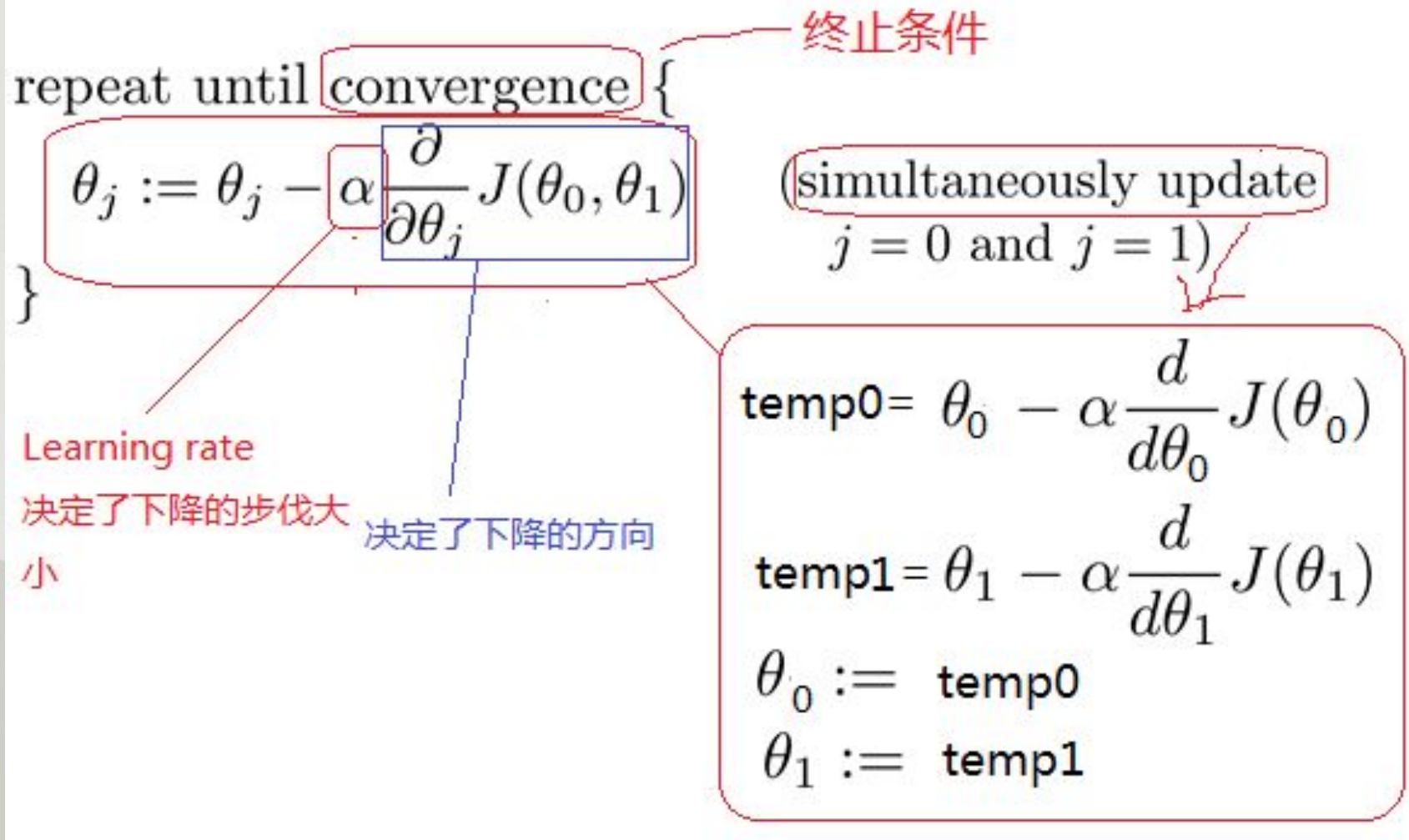
其中:

$x^{(i)}$  表示向量  $x$  中的第  $i$  个元素;

$y^{(i)}$  表示向量  $y$  中的第  $i$  个元素;

$h_{\theta}(x^{(i)})$  表示已知的假设函数;

# 梯度：偏微分



# 程序演算

问题: 对于点(2,1)和点(1,2), 拟合一个直线:  
 $y=a*x+b$

Step01: loss函数简化为:

$$\text{loss} = (2a+b-1)^2 + (a+b-2)^2$$

Step02: 对a和b分别求偏导数:

$$2*2(2a+b-1)+2(a+b-2) = 10a+6b-8$$

$$2(2a+b-1)+2(a+b-2) = 6a+4b-6$$

Step03: a和b初始为0, 学习率设置为0.1, 则:

$$a1 = 0 - 0.1*(10*0+6*0-8) = 0.8$$

$$b1 = 0 - 0.1*(6*0+4*0-6) = 0.6$$

# 程序演算

程序迭代100次, a和b的值大概如下:

(0.8, 0.60000000000000000001)

(0.43999999999999999999, 0.48)

(0.512, 0.62400000000000000001)

.....

(-0.9137542281992865, 2.8604514098404783)

(-0.9162708459042869, 2.864523382823859)

可以看到a值越来越接近-1, b值越来越接近3。

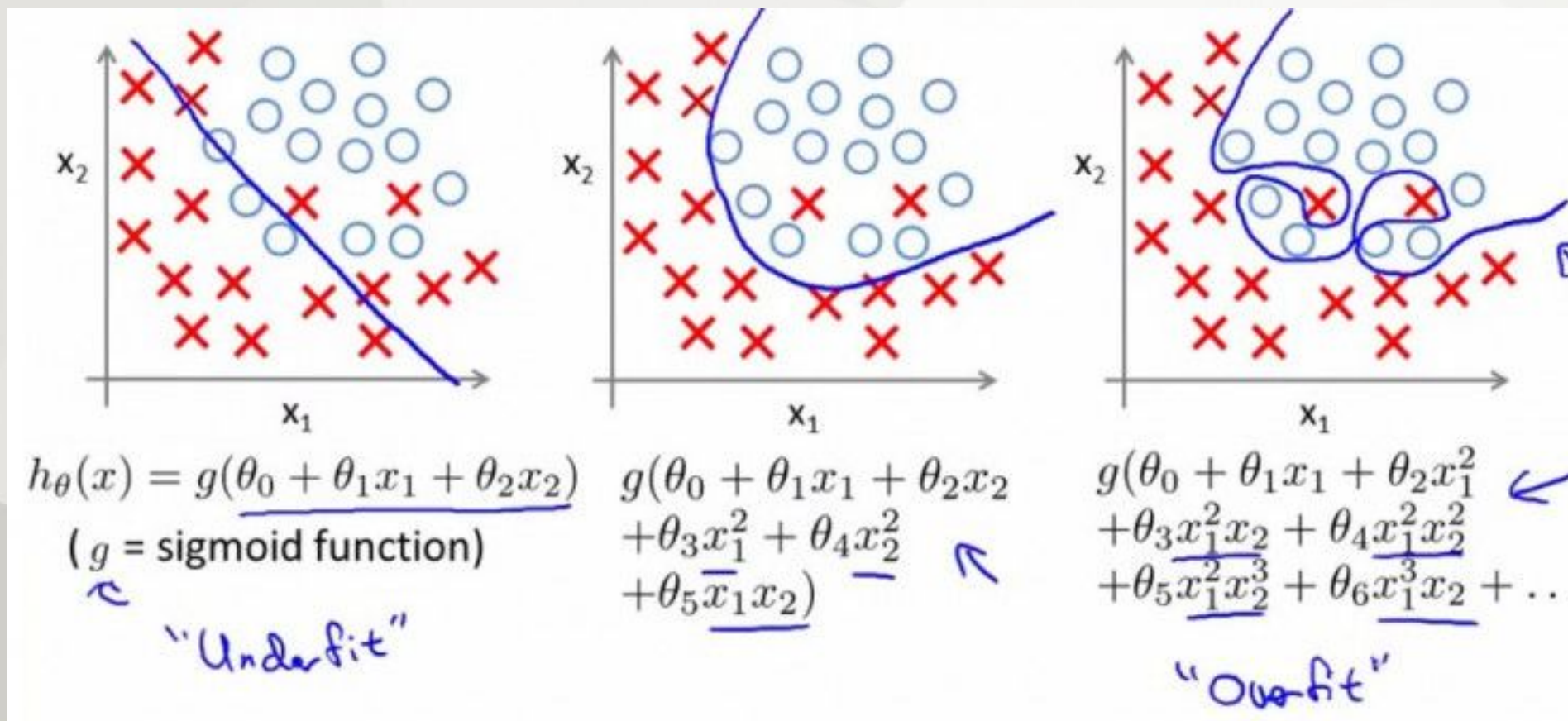
如果继续迭代到200次: (-0.996, 2.993)

# 程序演算

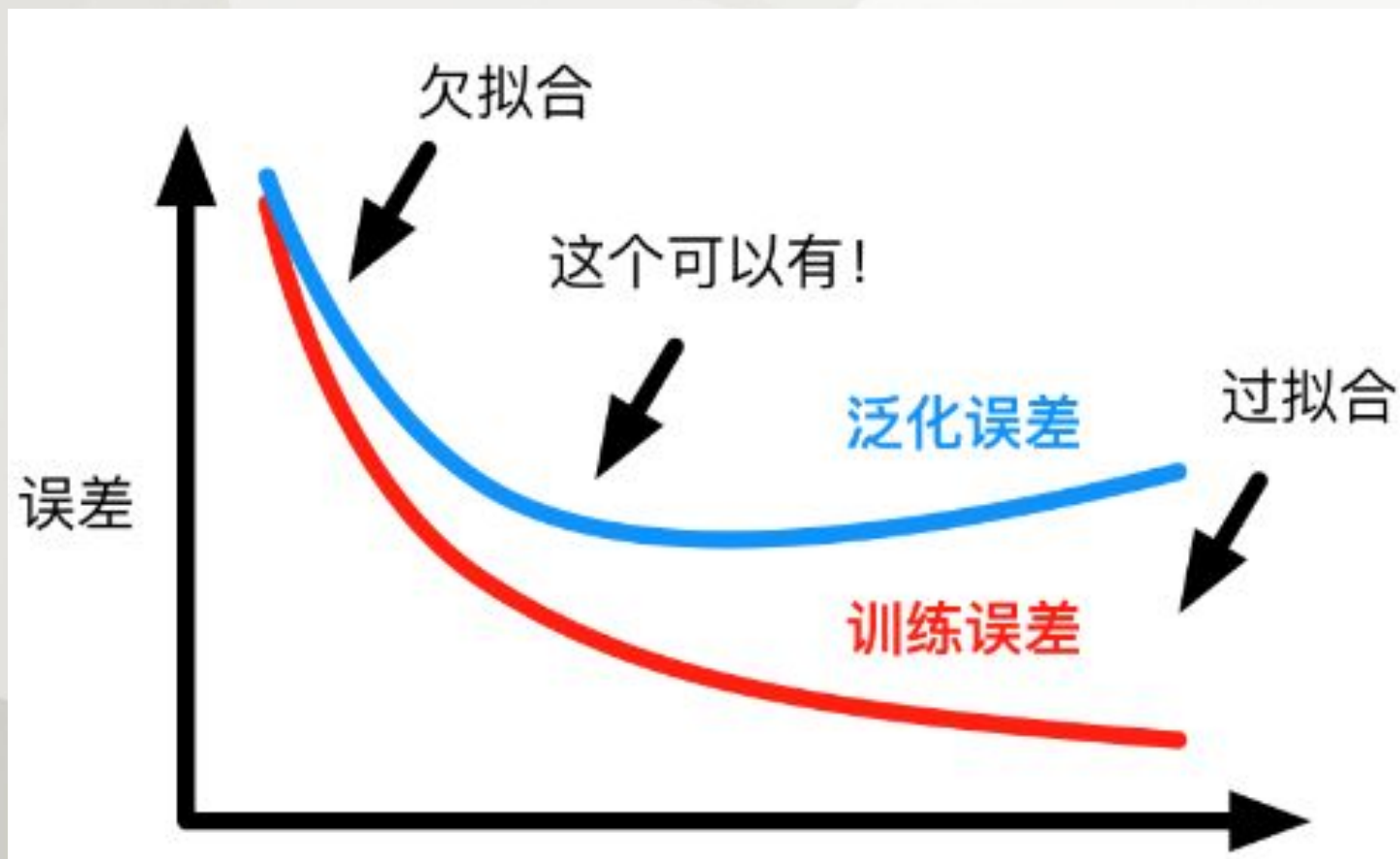
```
7 def gradient_a(a, b):  
8     return 10*a + 6*b - 8  
9  
10 def gradient_b(a, b):  
11     return 6*a + 4*b - 6  
12  
13 def update(a, b):  
14     return (a - 0.1 * gradient_a(a, b), b - 0.1 * gradient_b(a, b))  
15  
16 w = (0, 0)  
17 for i in range(100):  
18     w = update(w[0], w[1])  
19     print(w)
```



# 欠拟合、拟合与过拟合



# 欠拟合、拟合与过拟合










04



## 梯度下降的问题

# 需要解决的问题

-  迭代什么时候停止？
-  初始参数怎么选择？
-  学习率怎么选择？
-  如何避免局部最优值？
-  如何加快迭代速度？

# 重要概念



随机梯度下降



批量梯度下降



小批量梯度下降



动量



学习率优化策略



05



使用梯度下降



06



课后作业



# 重要概念

1. 给定三个点(1,1), (2,3), (3,4), 使用梯度下降算法拟合一个线性模型: $y=ax+b$ , 手动计算前5次迭代的参数值,  $a$ 与 $b$ 参数的初始值为0, 学习率为0.1。
2. 使用程序实现, 迭代100次时, 所得的参数值及其loss, 并绘制一条迭代次数与loss的曲线。
3. 学习率分别修改为0.01时, 迭代100次所得的参数值和loss。
4. 使用sklearn的梯度下降算法预测加利福尼亚州的房价。



DEEAO LEARNING  
迪 奥 教 育

感谢聆听