DEEP LEARNING INSTITUTE

# 深度学习基础知识

## 第 2 部分：如何训练神经网络

# 课程议题

# 议题 - 第 2 部分

- 回顾
- 一个简单的模型
- 从神经元到网络
- 激活函数
- 过拟合
- 从神经元到分类

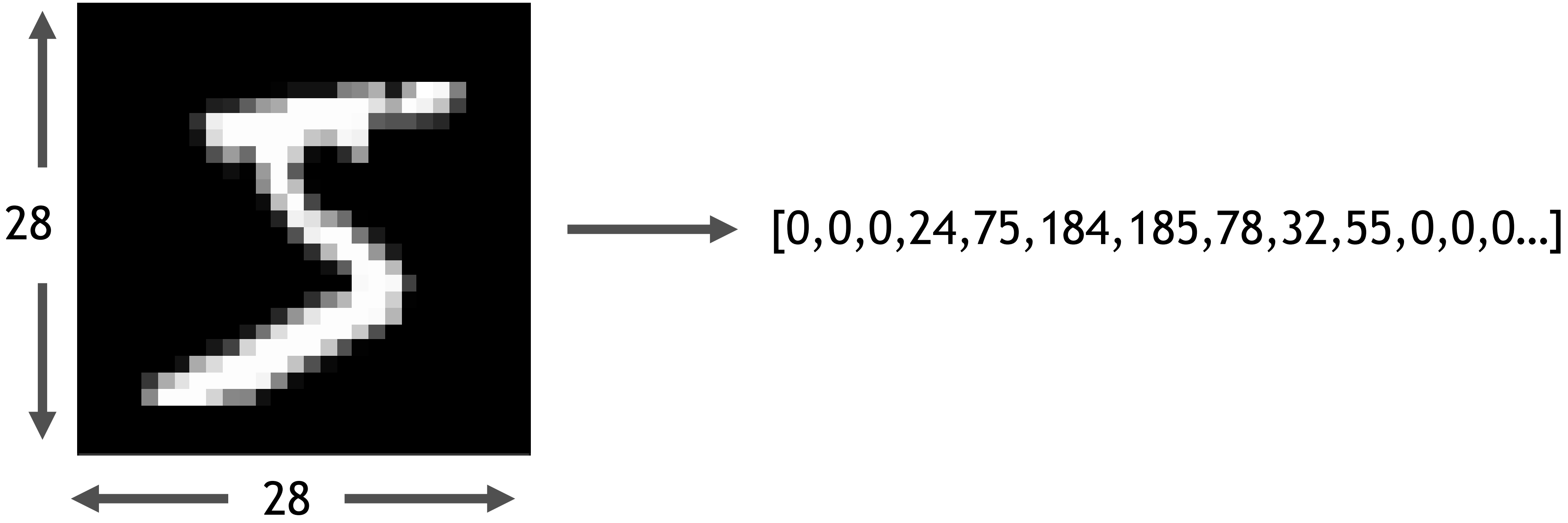
# 练习回顾

## 刚刚发生了什么?

- 加载数据并对数据进行了可视化
- 对数据进行了编辑（重构并进行归一化以供分类）
- 创建了模型
- 编译了模型
- 使用数据训练了模型

# 数据准备

## 以数组形式输入



# 数据准备

## 目标转换成类别

0 → [1,0,0,0,0,0,0,0,0,0]

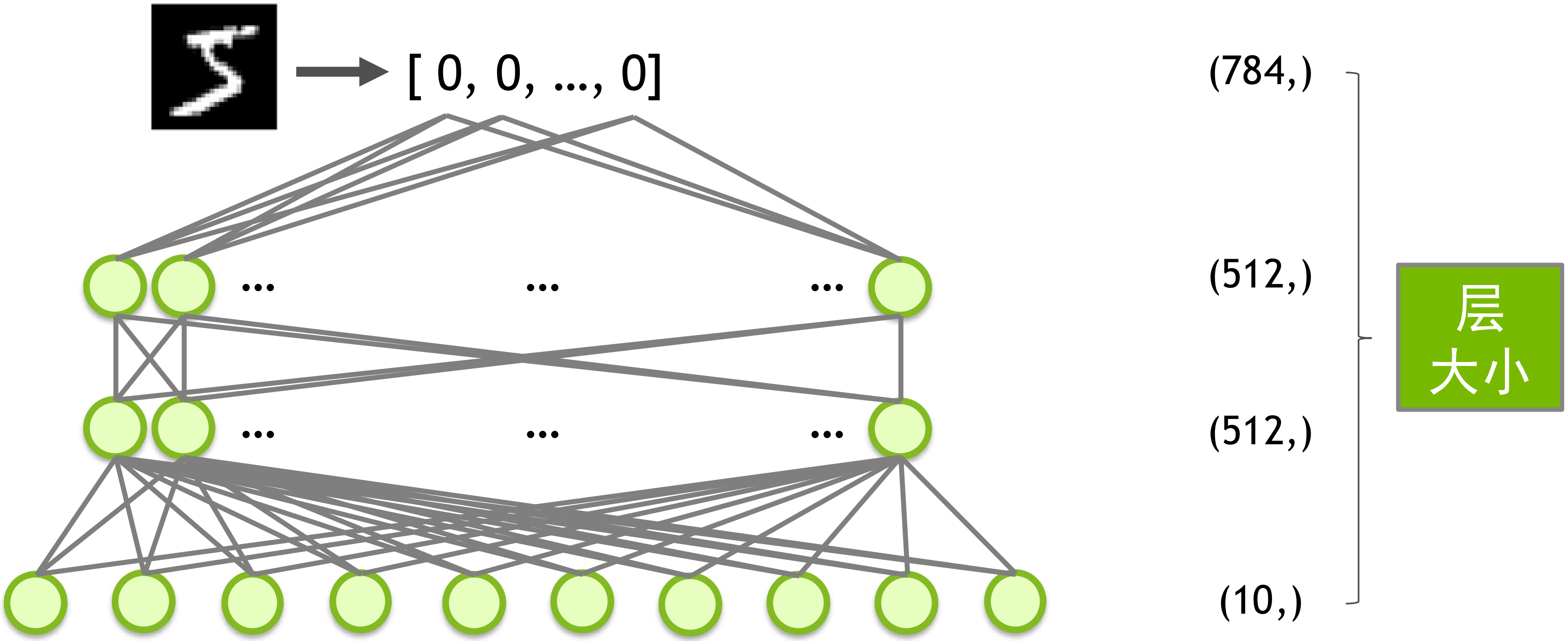
1 → [0,1,0,0,0,0,0,0,0,0]

2 → [0,0,1,0,0,0,0,0,0,0]

3 → [0,0,0,1,0,0,0,0,0,0]

⋮

# 未训练的模型

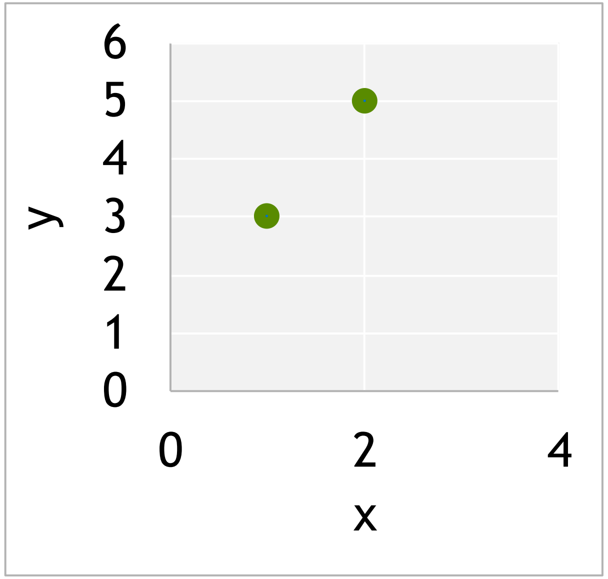


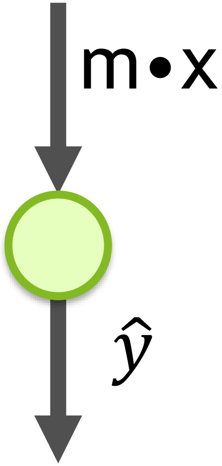
# 更简单的模型

# 更简单的模型

$$y = mx + b$$

| x | y |
|---|---|
| 1 | 3 |
| 2 | 5 |



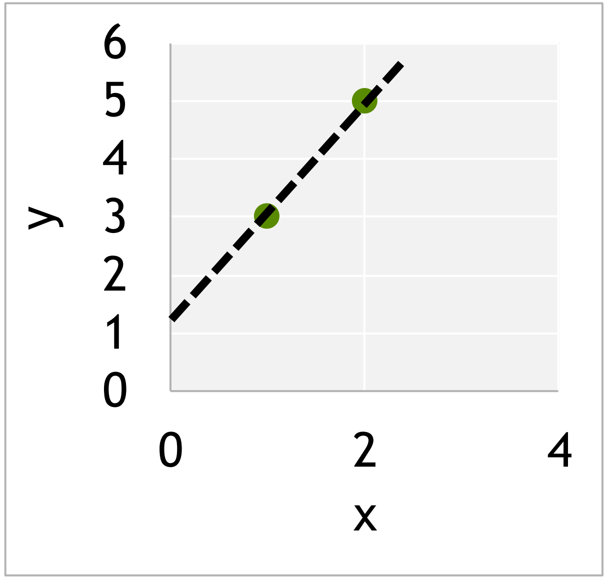

$m = ?$

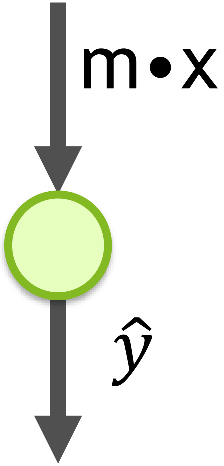
$b = ?$

# 更简单的模型

$$y = mx + b$$

| x | y |
|---|---|
| 1 | 3 |
| 2 | 5 |



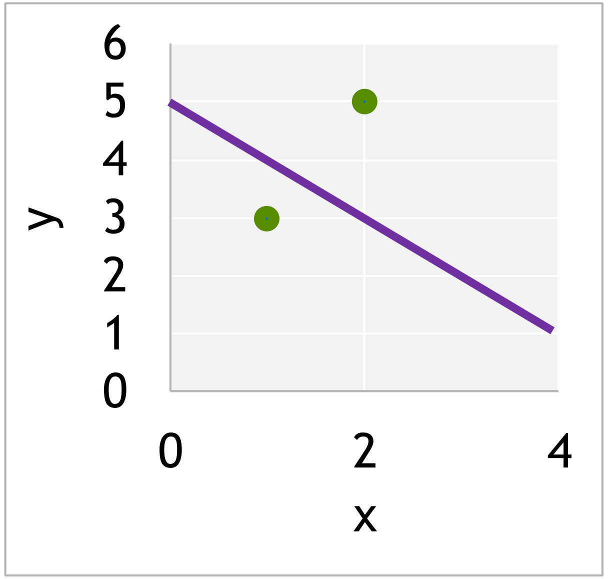

$m = ?$

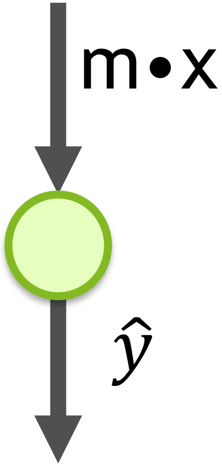
$\mathrm{b} = ?$

# 更简单的模型

$$y = mx + b$$

| x | y | $\hat{y}$ |
|---|---|---|
| 1 | 3 | 4 |
| 2 | 5 | 3 |




从随机值开始

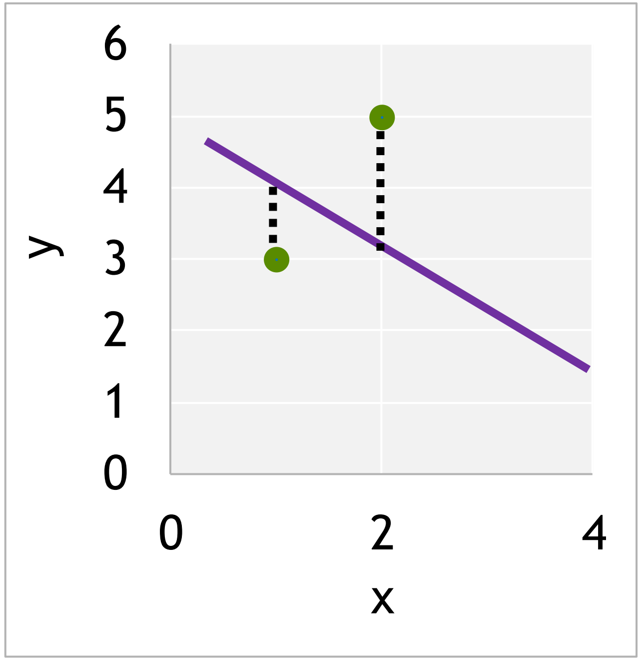
$$m = -1$$

$$b = 5$$

# 更简单的模型

$$y = mx + b$$

| x | y | $\hat{y}$ | $err^2$ |
|---|---|---|---|
| 1 | 3 | 4 | 1 |
| 2 | 5 | 3 | 4 |
| | | MSE = | 2.5 |
| | | RMSE = | 1.6 |




$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

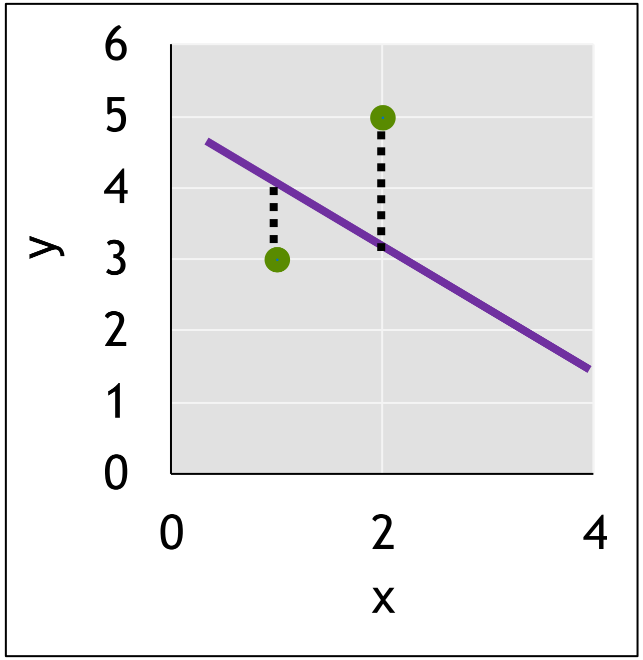
# 更简单的模型

$$y = mx + b$$

| x | y | $\hat{y}$ | $err^2$ |
|---|---|---|---|
| 1 | 3 | 4 | 1 |
| 2 | 5 | 3 | 4 |

MSE = 2.5

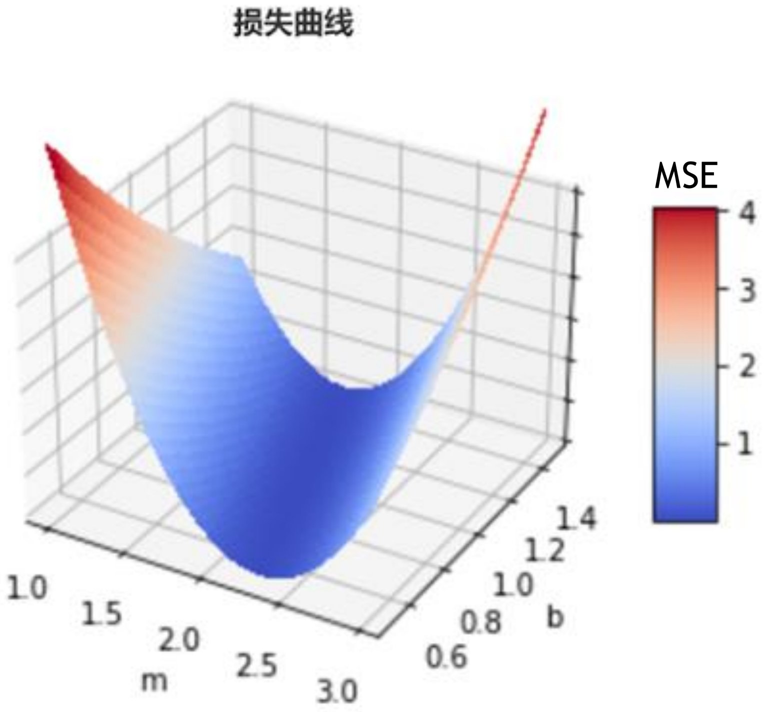
RMSE = 1.6



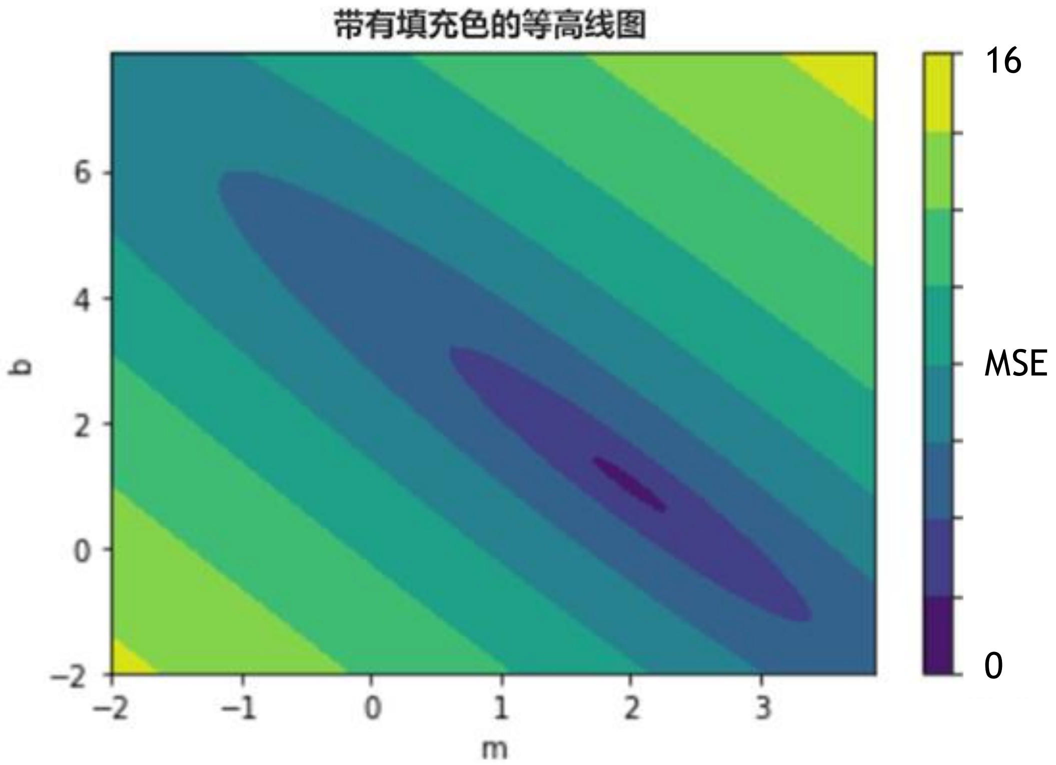

```python
data = [(1, 3), (2, 5)]
m = -1
b = 5


def get_rmse(data, m, b):
    """Calculates Mean Square Error"""
    n = len(data)
    squared_error = 0
    for x, y in data:
        # Find predicted y
        y_hat = m*x+b
        # Square difference between
        # prediction and true value
        squared_error += (
            y - y_hat)**2
    # Get average squared difference
    mse = squared_error / n
    # Square root for original units
    return mse ** .5
```

# 损失曲线

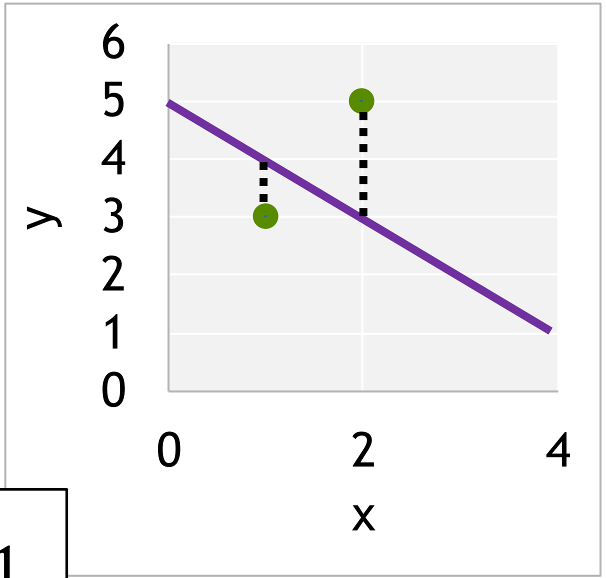


损失曲线
MSE
4
3
2
1
1.0
1.5
2.0
2.5
3.0
m
1.4
1.2
1.0
0.8
0.6
b

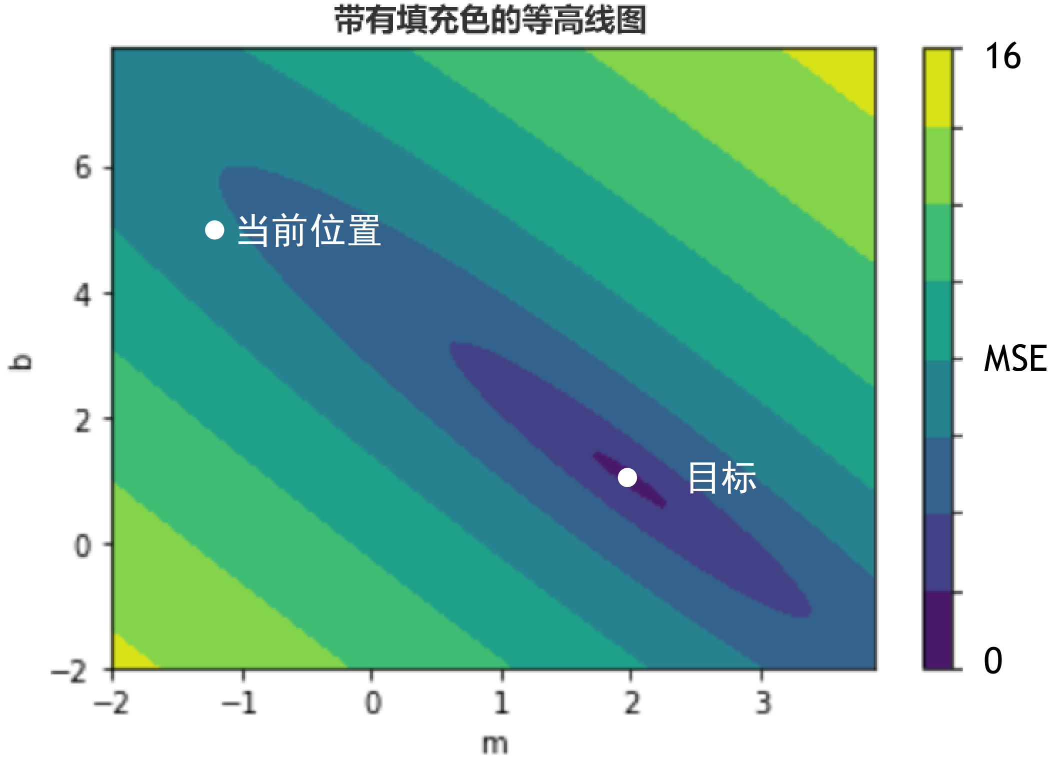
带有填充色的等高线图
16
MSE
0
6
4
2
0
−2
b
−2
−1
0
1
2
3
m

# 损失曲线

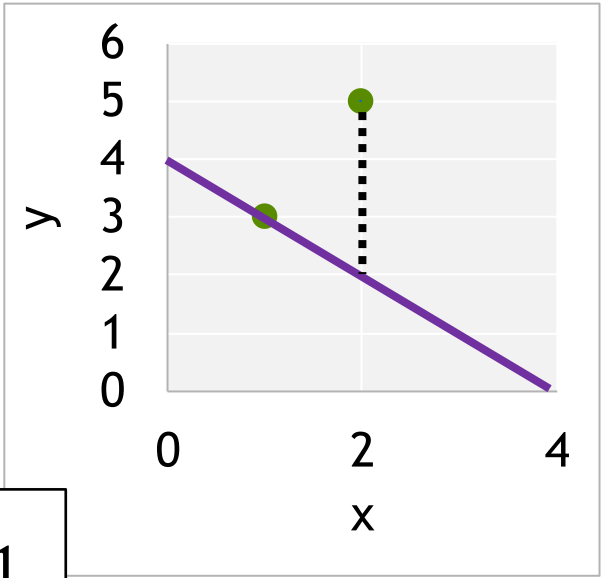



$$m = -1$$

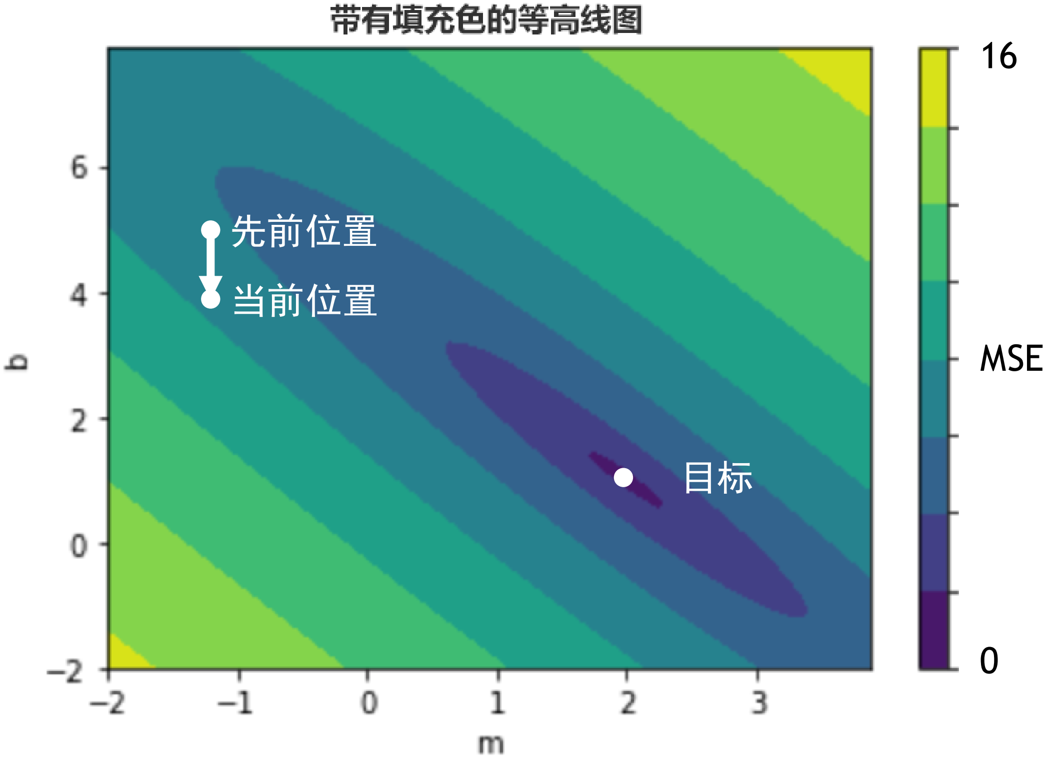
$$b = 5$$



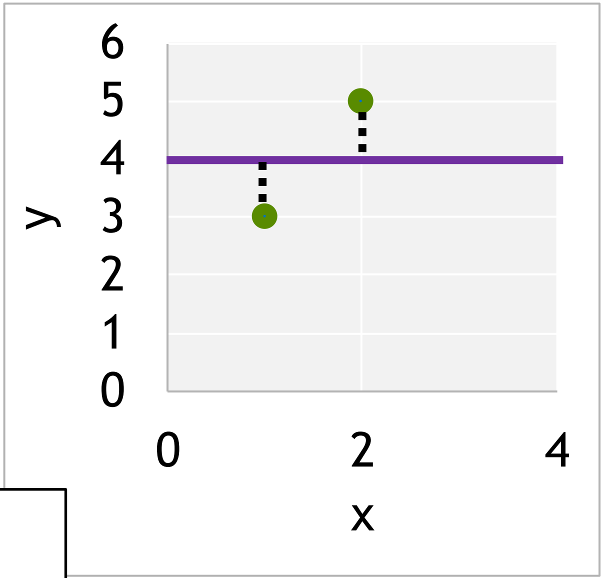
# 损失曲线




$$m = -1$$
$$b = 4$$

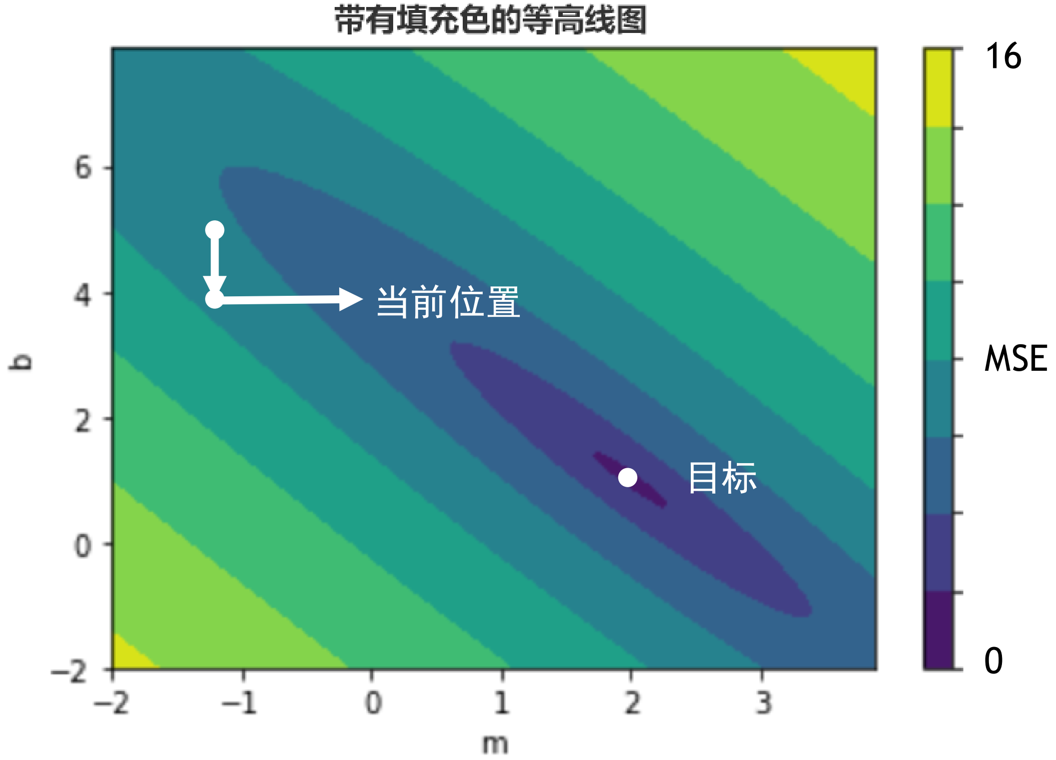


# 损失曲线



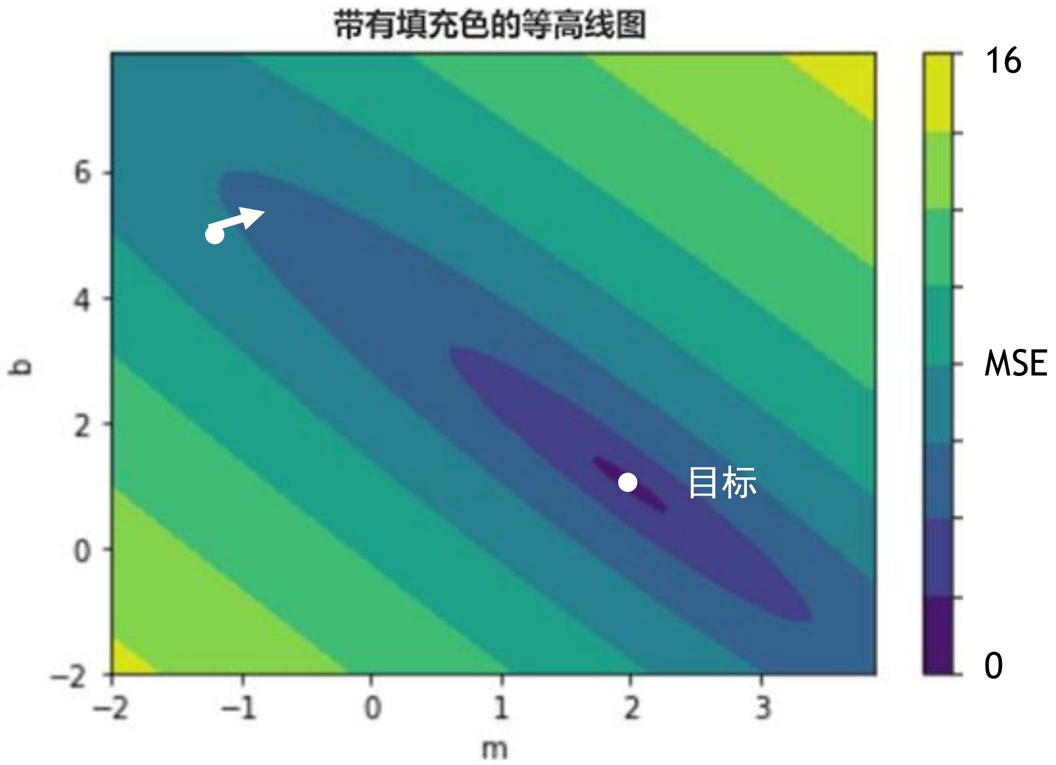

$$m = 0$$

$$b = 4$$



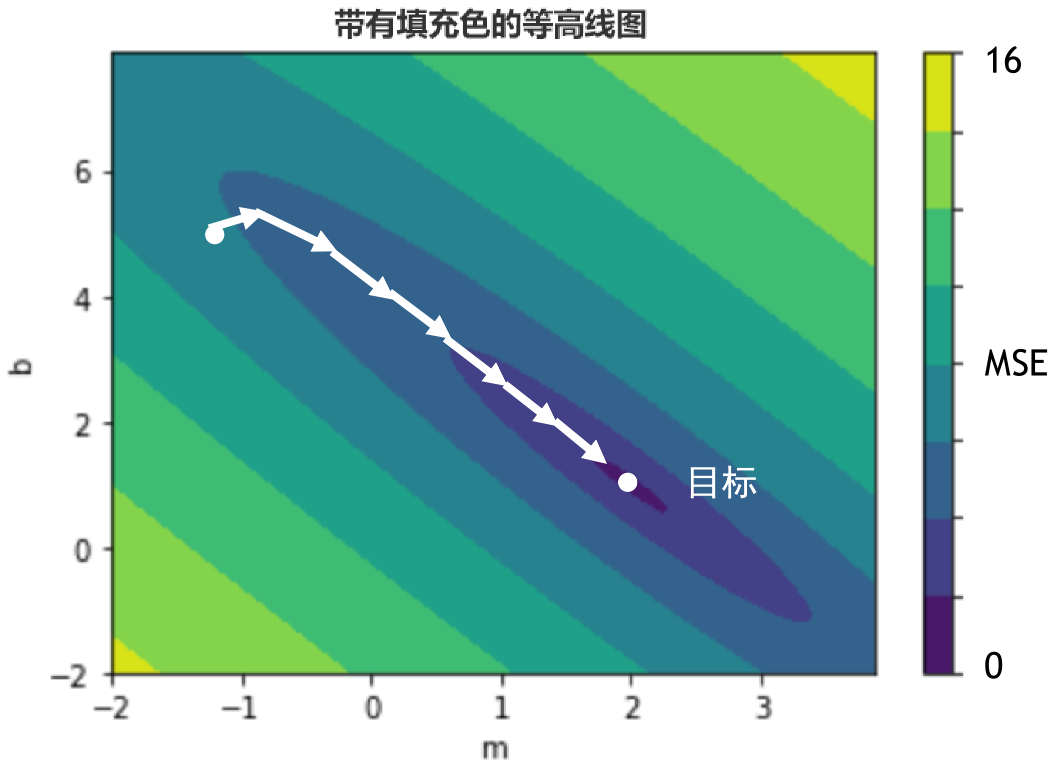
# 损失曲线

| | |
|---|---|
| 梯度 | 哪个方向损失减少最多 |
| λ：学习率 | 移动的距离 |
| 训练周期 | 使用完整数据集进行的一次模型更新 |
| 批量 | 完整数据集的样本 |
| 步 | 对权重参数的一次更新 |

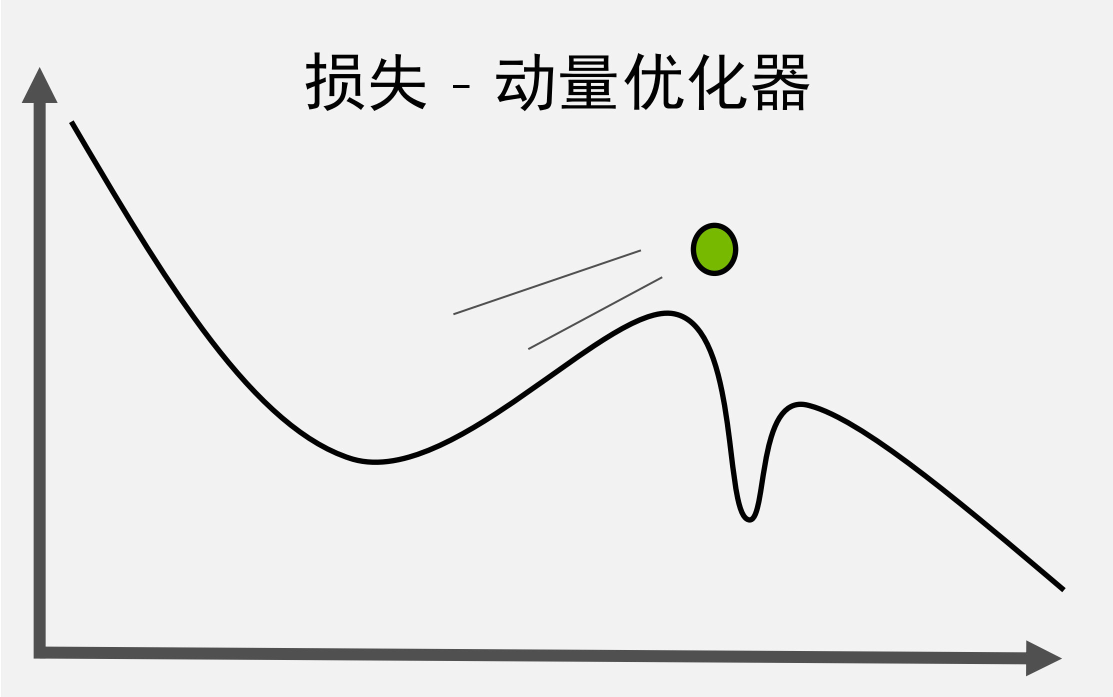


# 损失曲线

| | |
|---|---|
| 梯度 | 哪个方向损失减少最多 |
| λ：学习率 | 移动的距离 |
| 训练周期 | 使用完整数据集进行的一次模型更新 |
| 批量 | 完整数据集的样本 |
| 步 | 对权重参数的一次更新 |



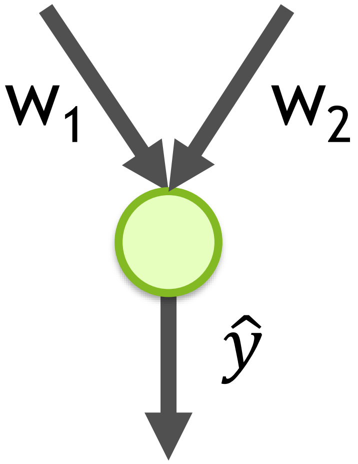
# 优化器




- Adam
- Adagrad
- RMSProp
- SGD

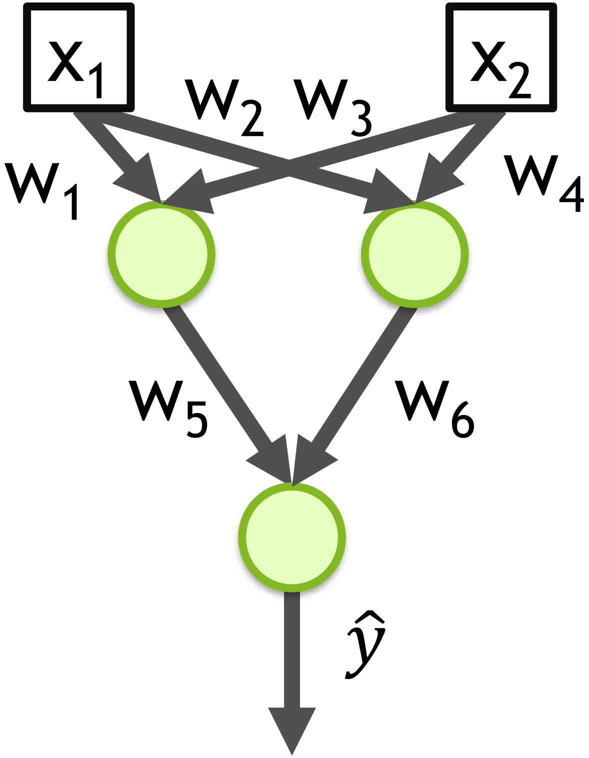
# 从神经元到网络

# 构建网络



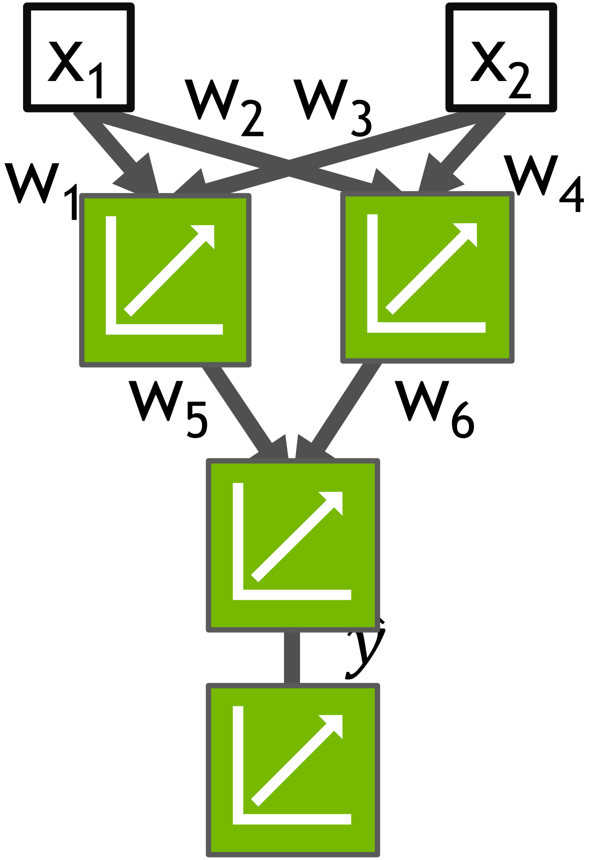

- 扩充到更多的输入

# 构建网络




- 扩充到更多的输入
- 能够串联神经元

# 构建网络




- 扩充到更多的输入
- 能够串联神经元
- 如果所有回归均为线性回归，输出也将为线性回归

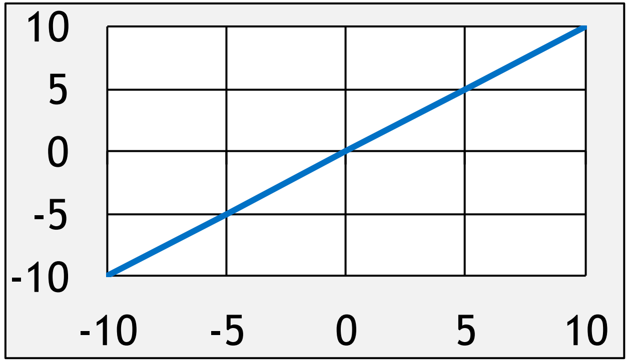
# 激活函数

# 激活函数

## Linear

$$\hat{y} = wx + b$$

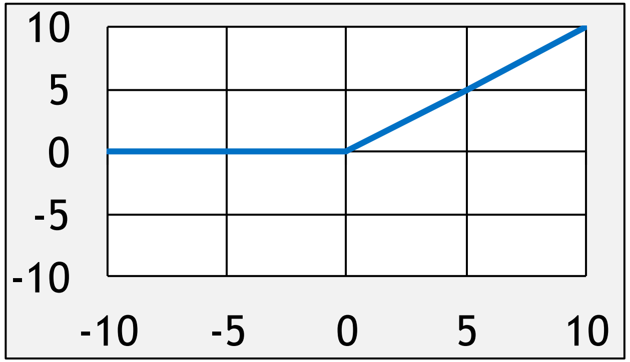
```
# Multiply each input
# with a weight (w) and
# add intercept (b)
y_hat = wx+b
```




## ReLU

$$\hat{y} = \begin{cases} wx + b \; if \; wx + b > 0 \\ 0 \;\; otherwise \end{cases}$$

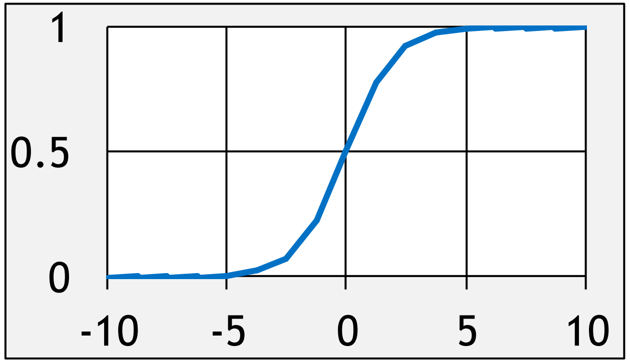
```
# Only return result
# if total is positive
linear = wx+b
y_hat = linear * (linear > 0)
```




## Sigmoid

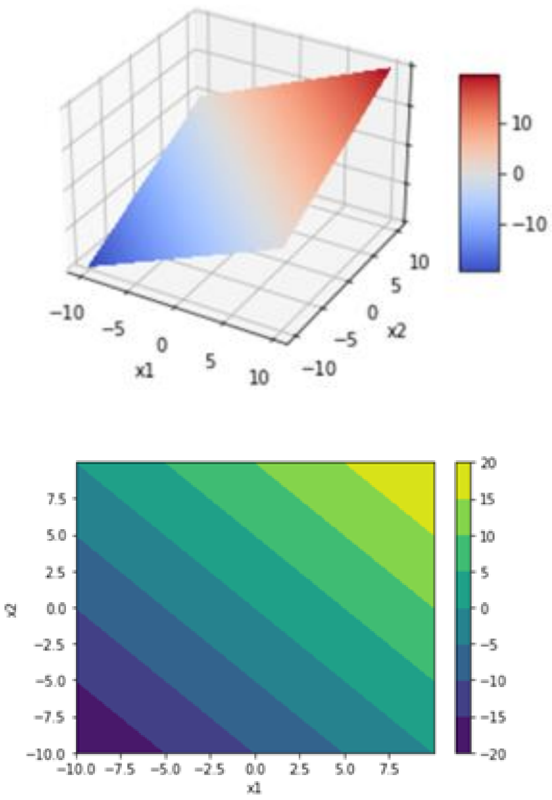
$$\hat{y} = \frac{1}{1 + e^{-(wx+b)}}$$

```
# Start with line
linear = wx + b
# Warp to - inf to 0
inf_to_zero = np.exp(-1 * linear)
# Squish to -1 to 1
y_hat = 1 / (1 + inf_to_zero)
```

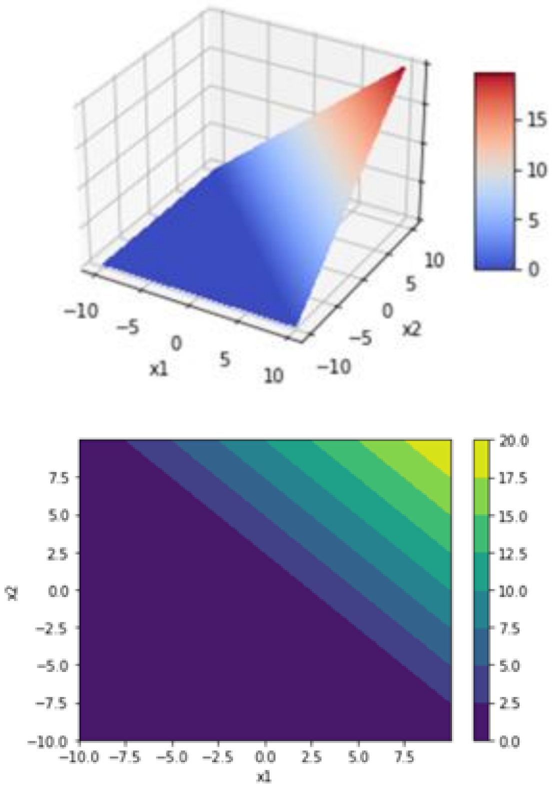


# 激活函数

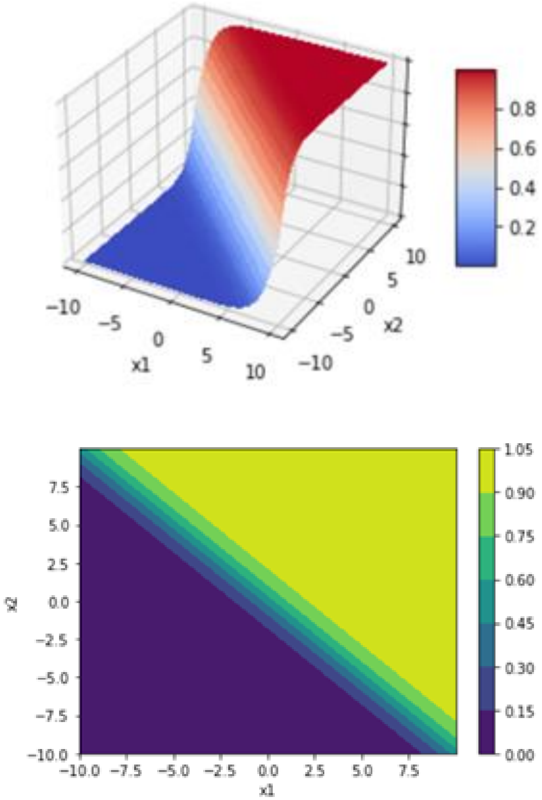
Linear



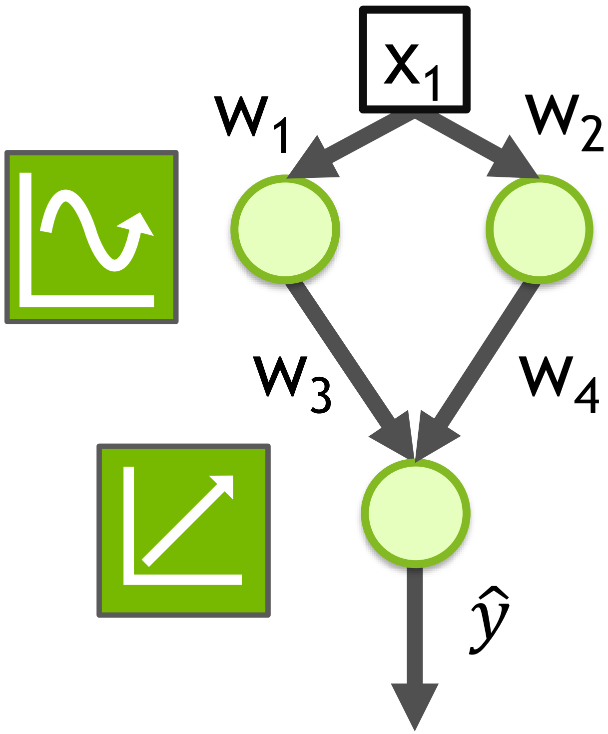

ReLU

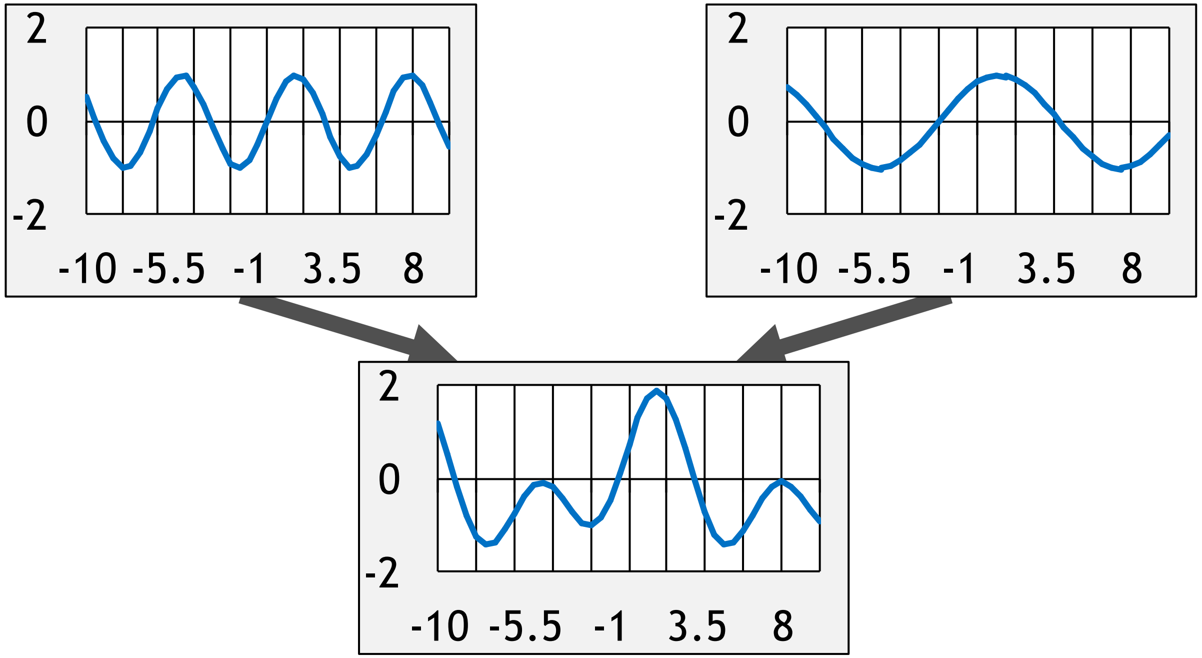



Sigmoid



# 激活函数



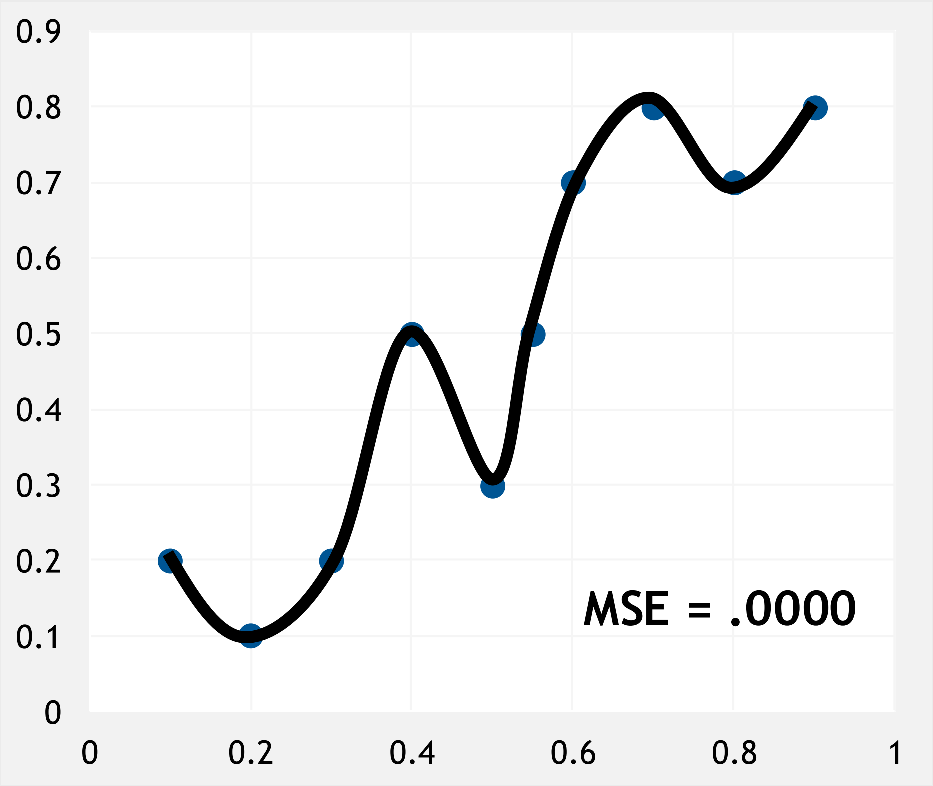
# 过拟合

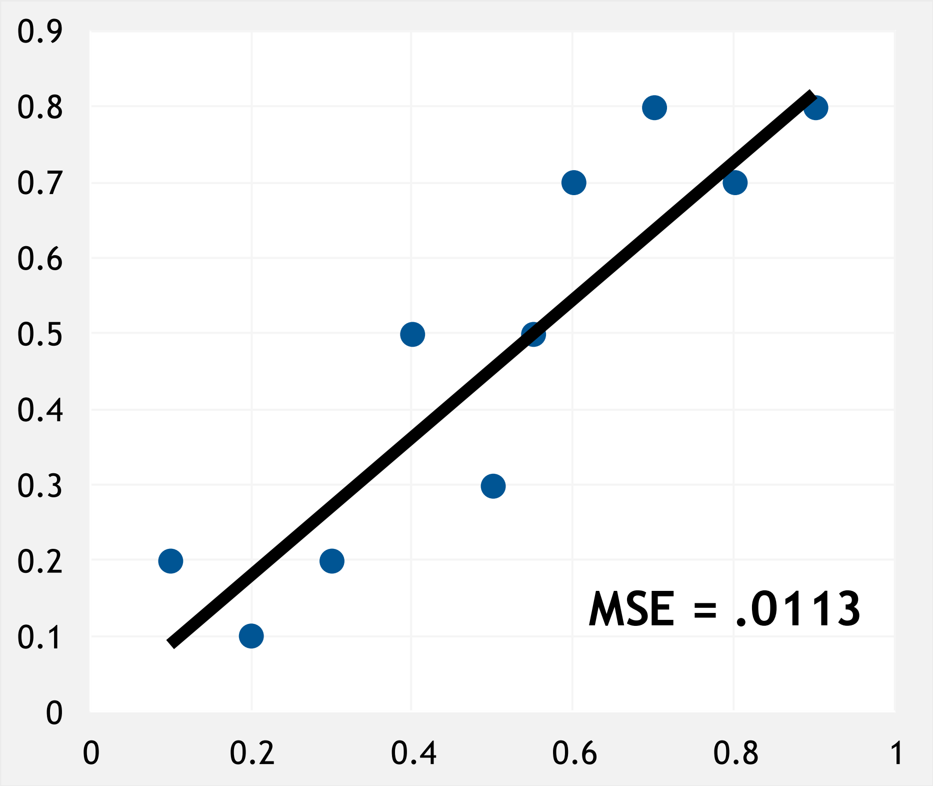
# 过拟合

## 为何不构建一个超大的神经网络呢？

# 过拟合

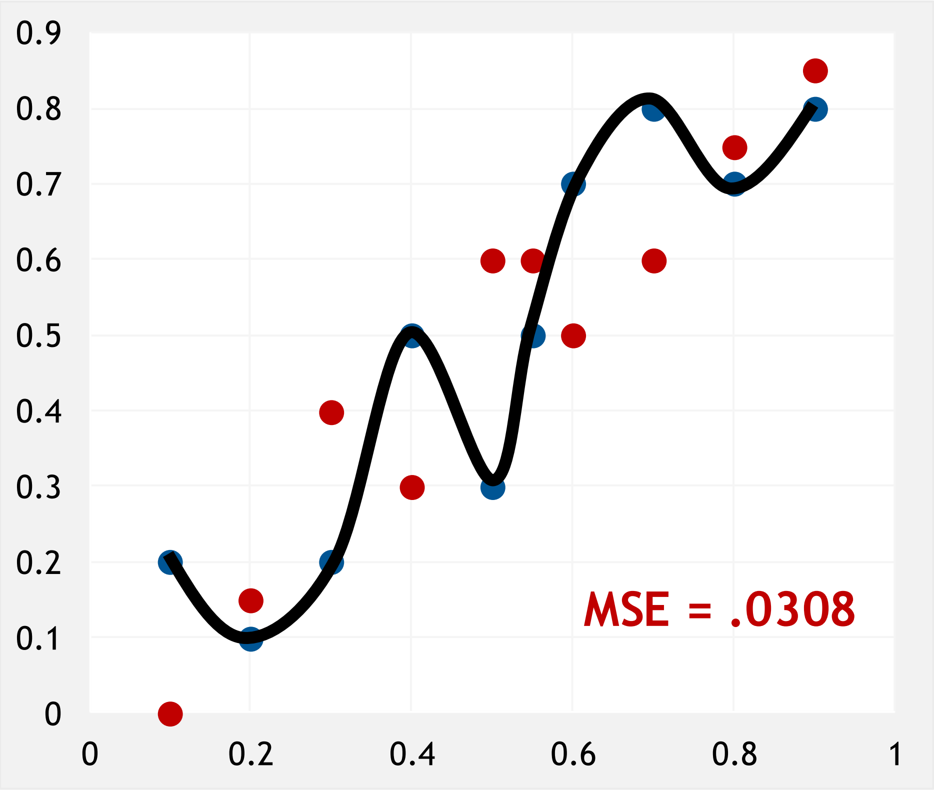
## 哪条趋势线更好？


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
MSE = .0000

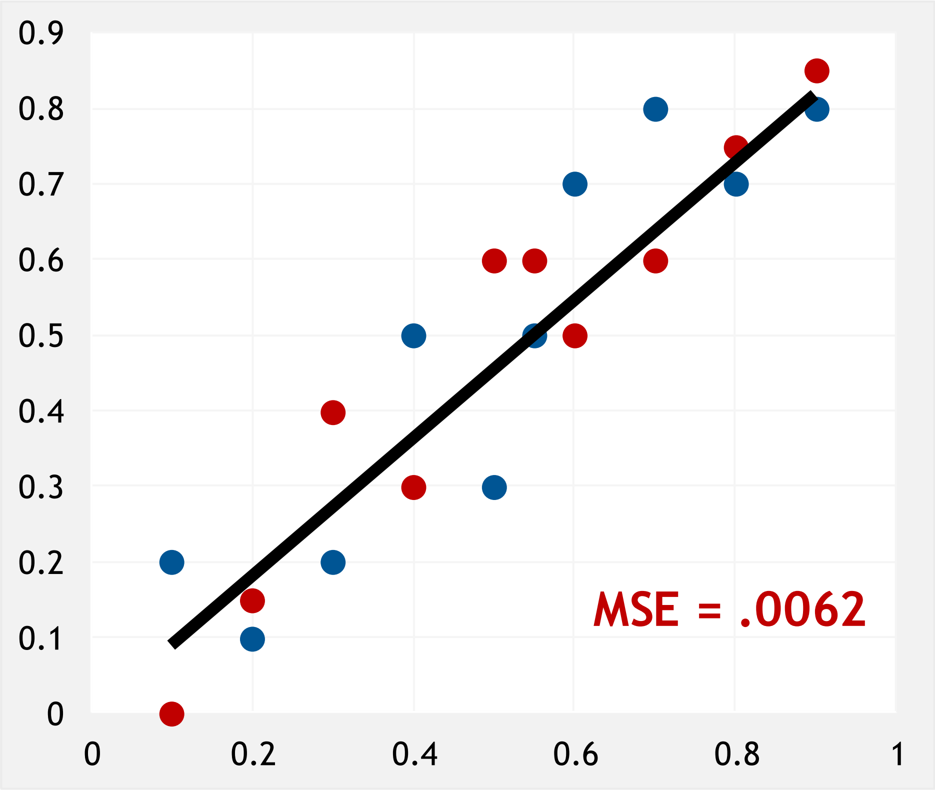

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
MSE = .0113

# 过拟合

## 哪条趋势线更好？



0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
MSE = .0308

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
MSE = .0062

# 训练数据和验证数据对比

## 避免记忆数据

### 训练数据

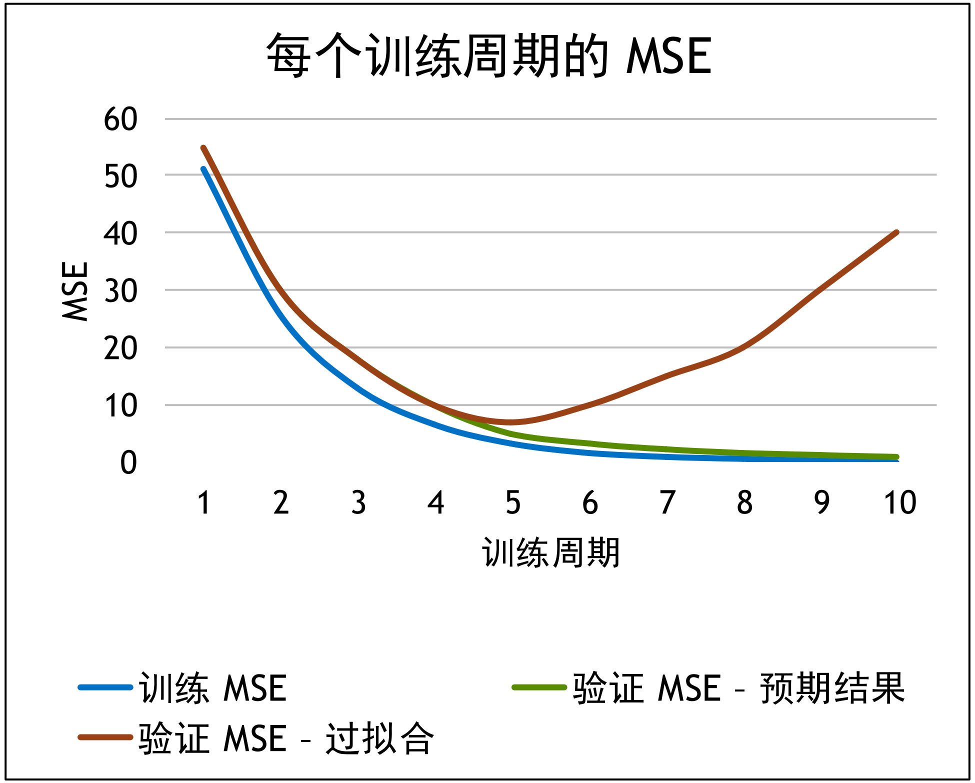
- 模型学习所用的核心数据集

### 验证数据

- 新数据，用于验证模型是否已能真正作出理解（可进行泛化）

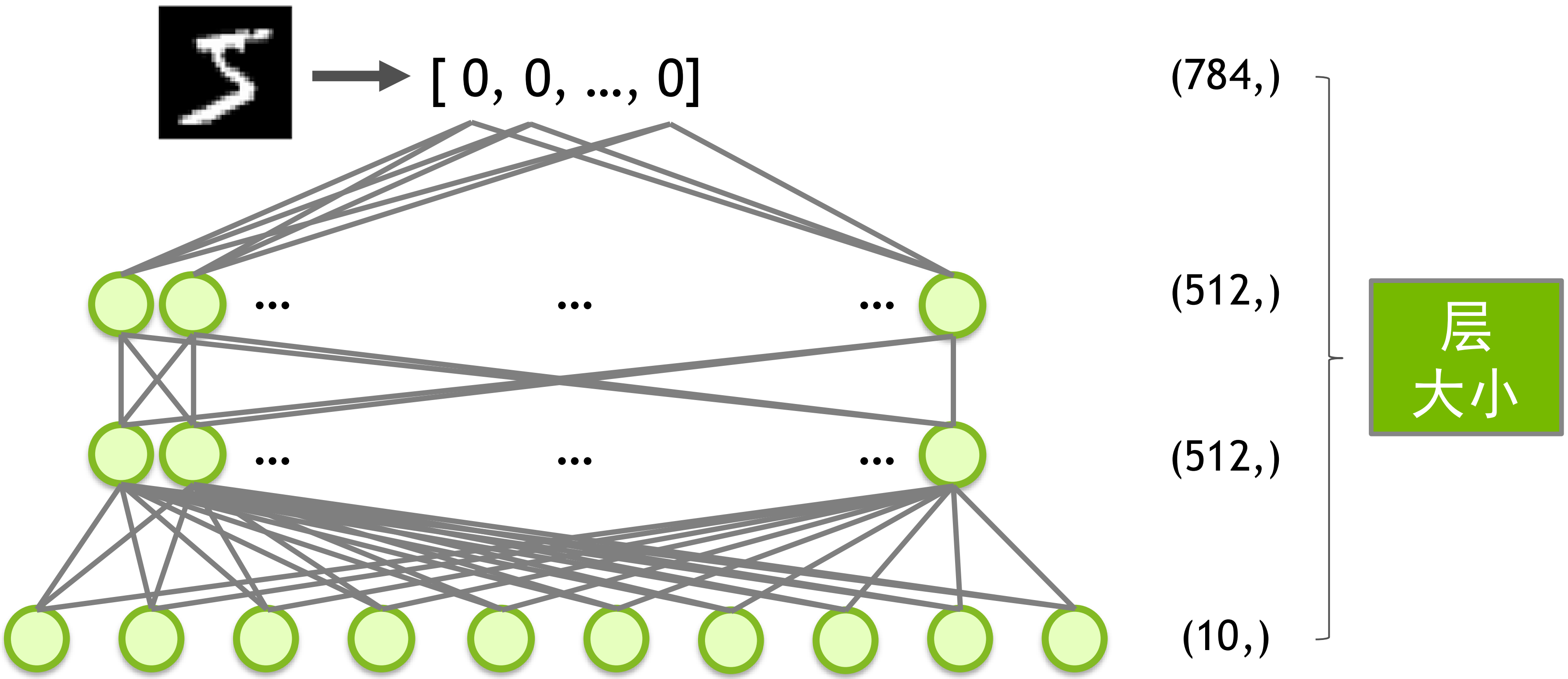
### 过拟合

- 模型在训练数据上表现出色，但对于验证数据表现不佳（表明模型只是在记忆数据）
- 理想情况下，模型在这两个数据集上表现出的准确性和损失应该相似

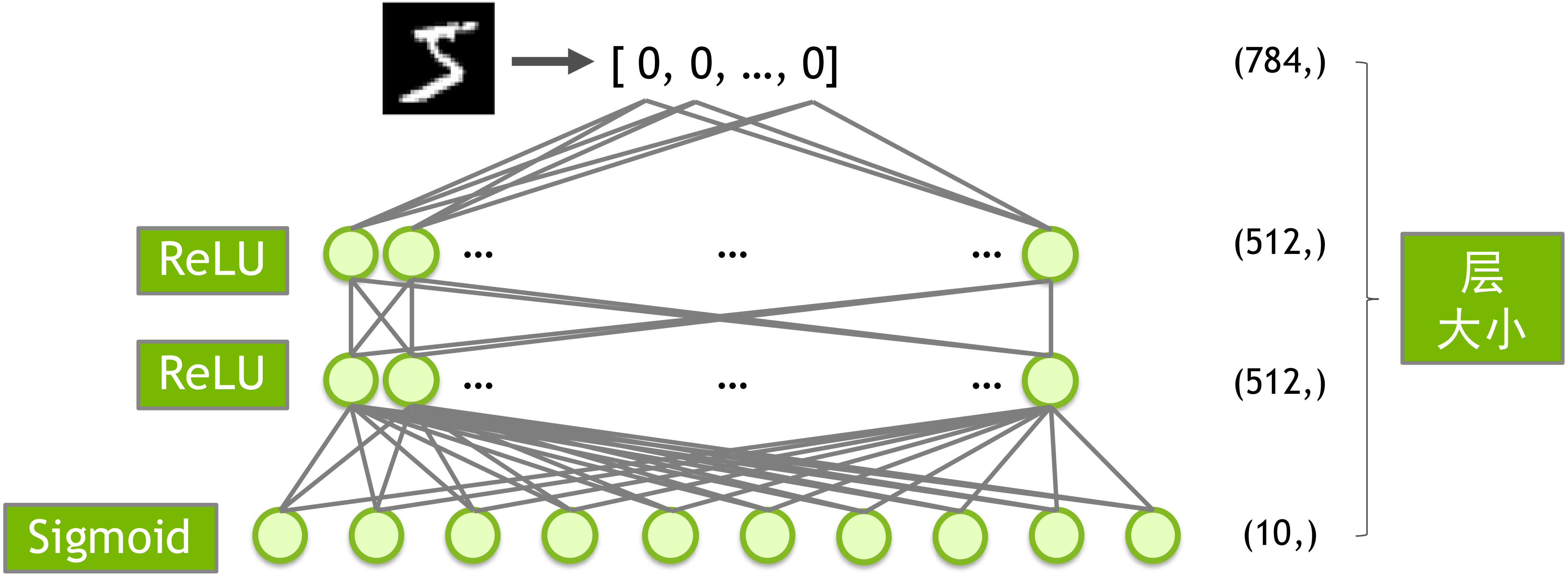


# 从回归到分类

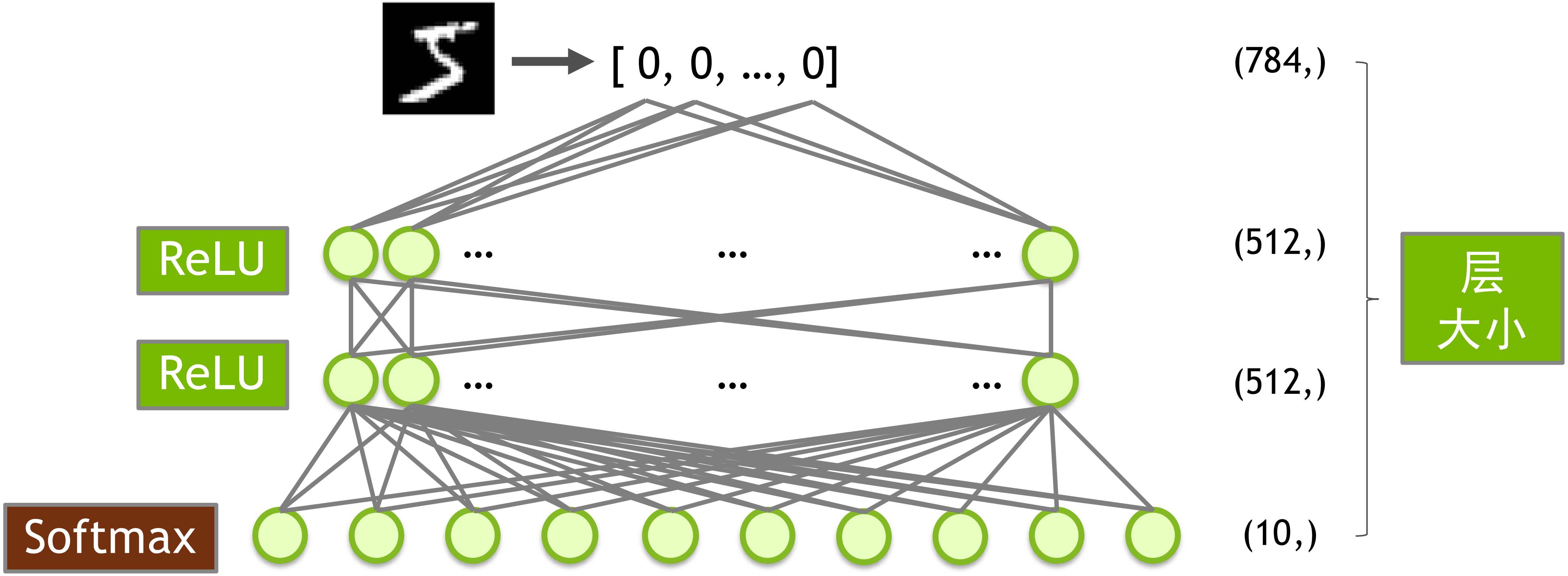
# MNIST 模型



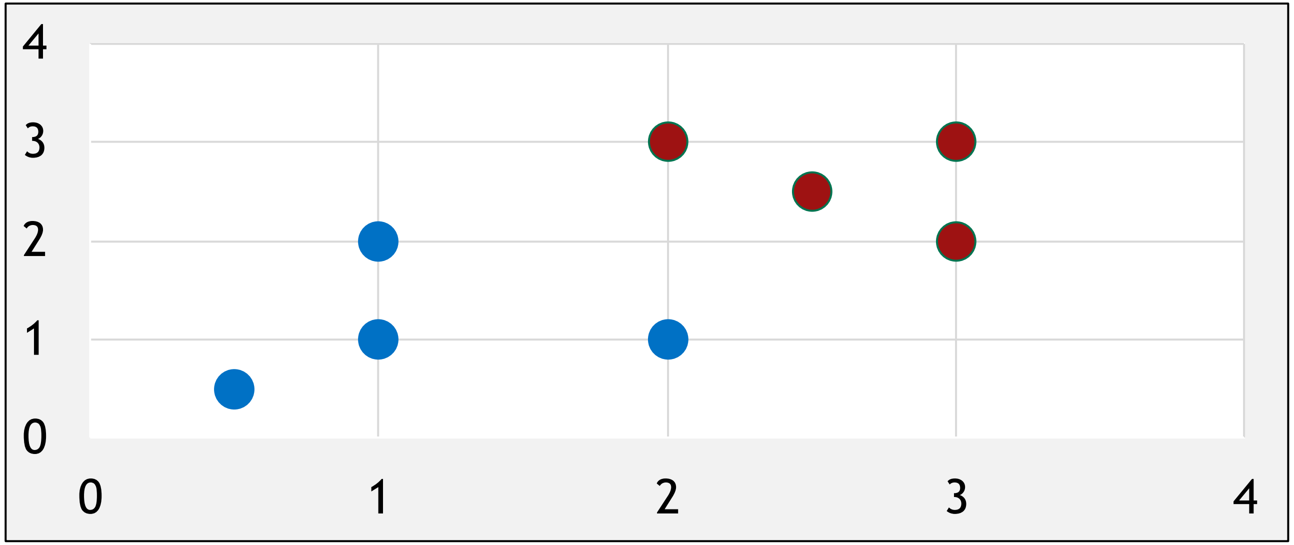
# MNIST 模型



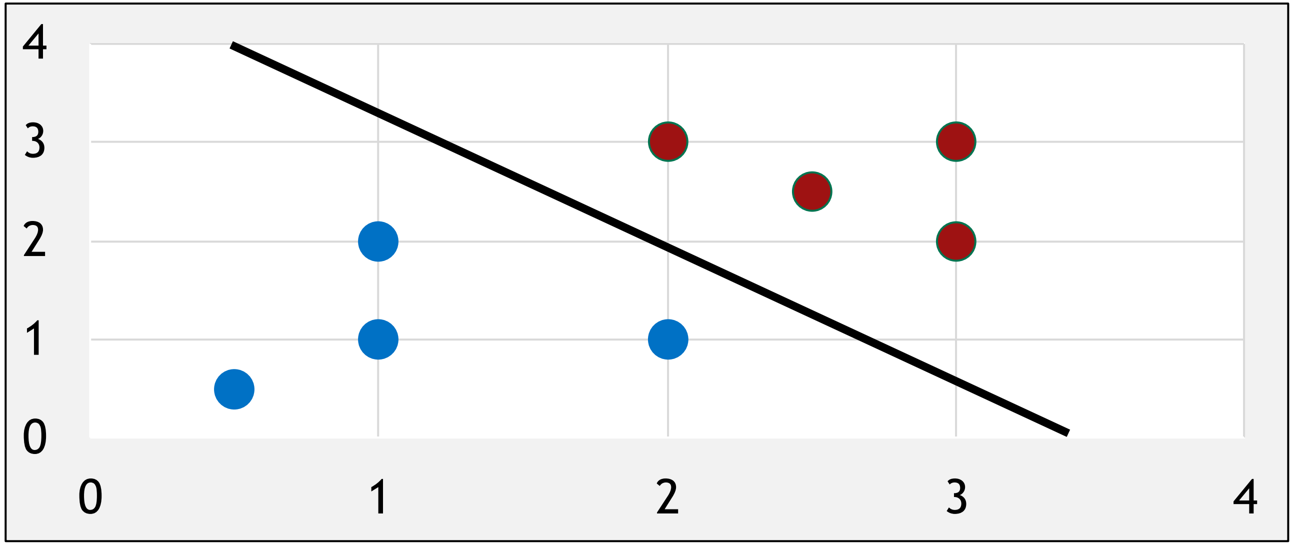
# MNIST 模型



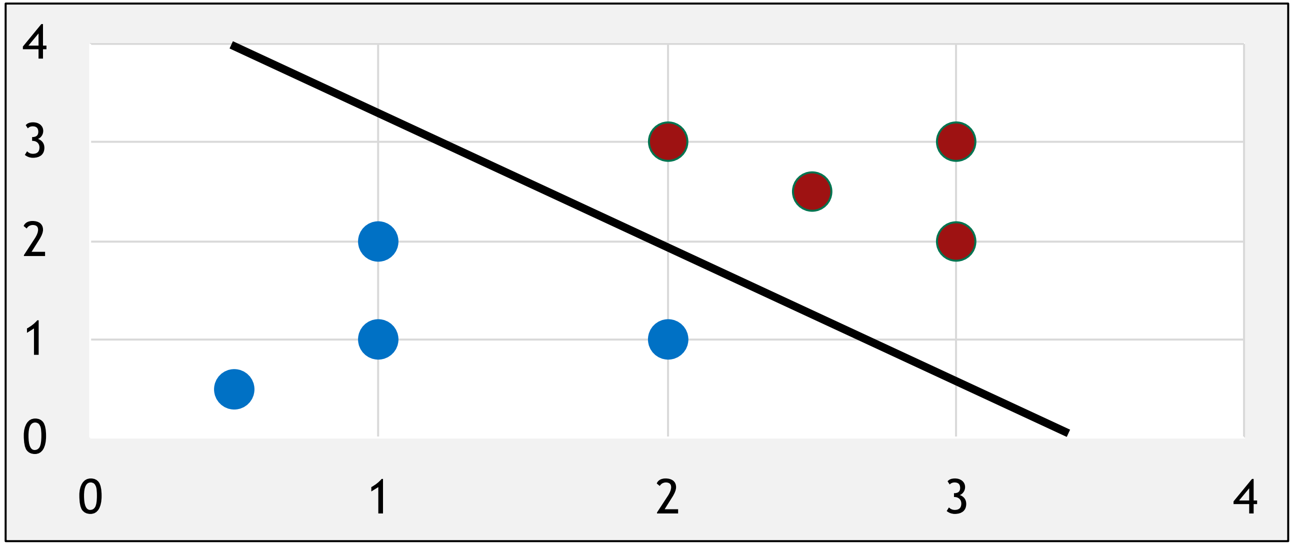
# 将 RMSE 用于概率 ？



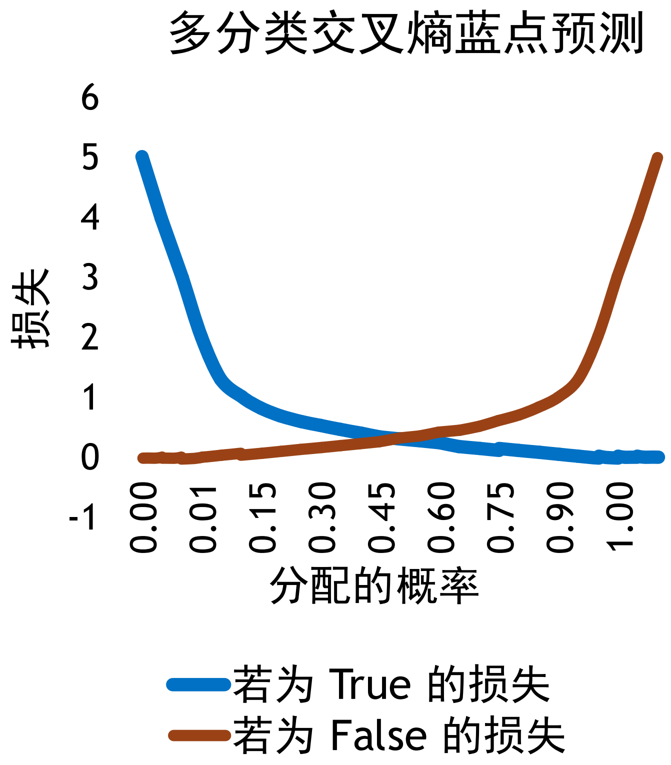
# 将 RMSE 用于概率？



# 多分类交叉熵

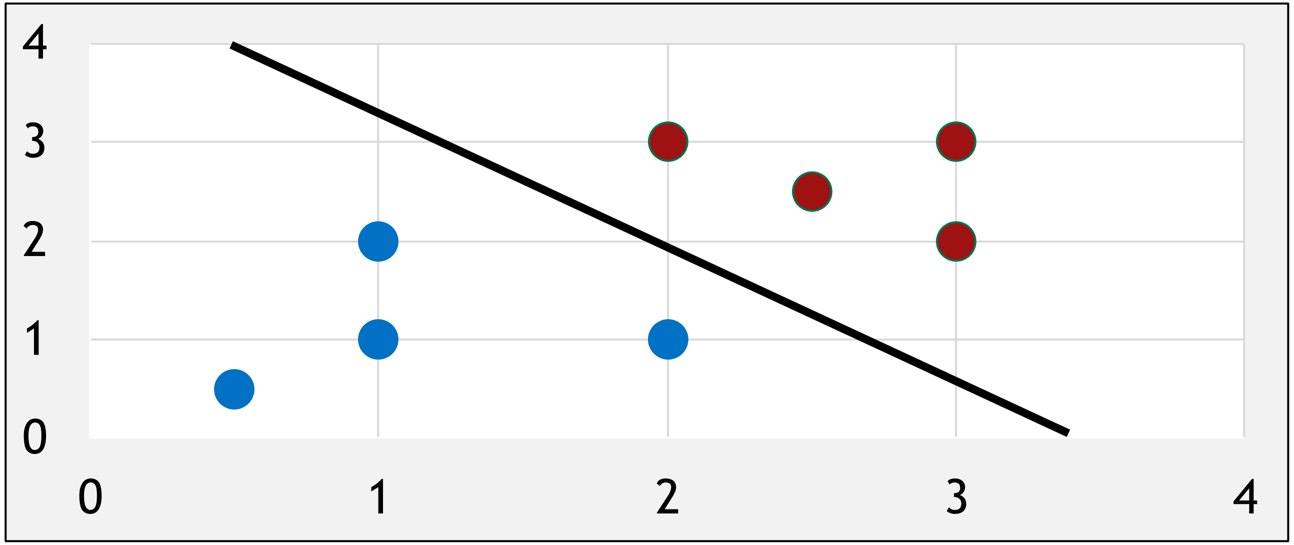


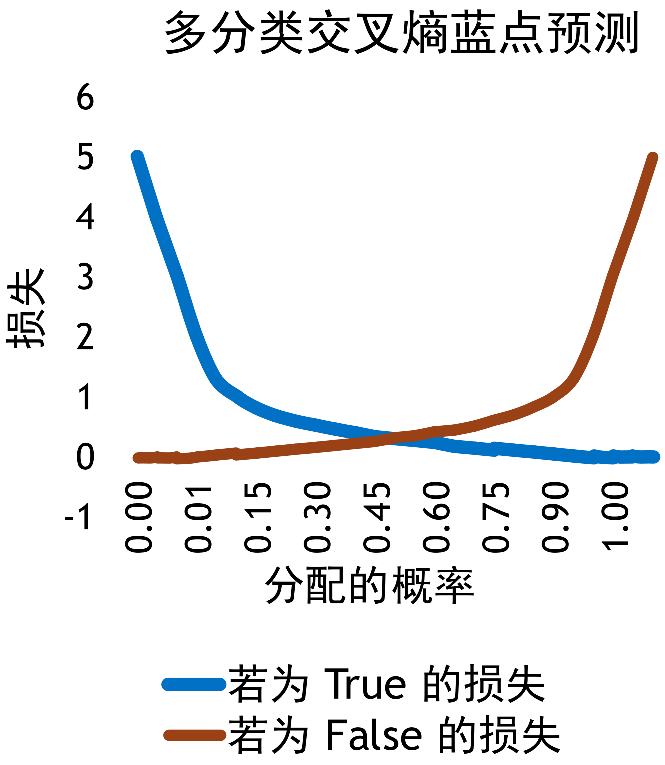
4
3
2
1
0
0
1
2
3
4

多分类交叉熵蓝点预测
6
5
4
3
2
1
0
-1
损失
0.00
0.01
0.15
0.30
0.45
0.60
0.75
0.90
1.00
分配的概率
若为 True 的损失
若为 False 的损失

# 多分类交叉熵



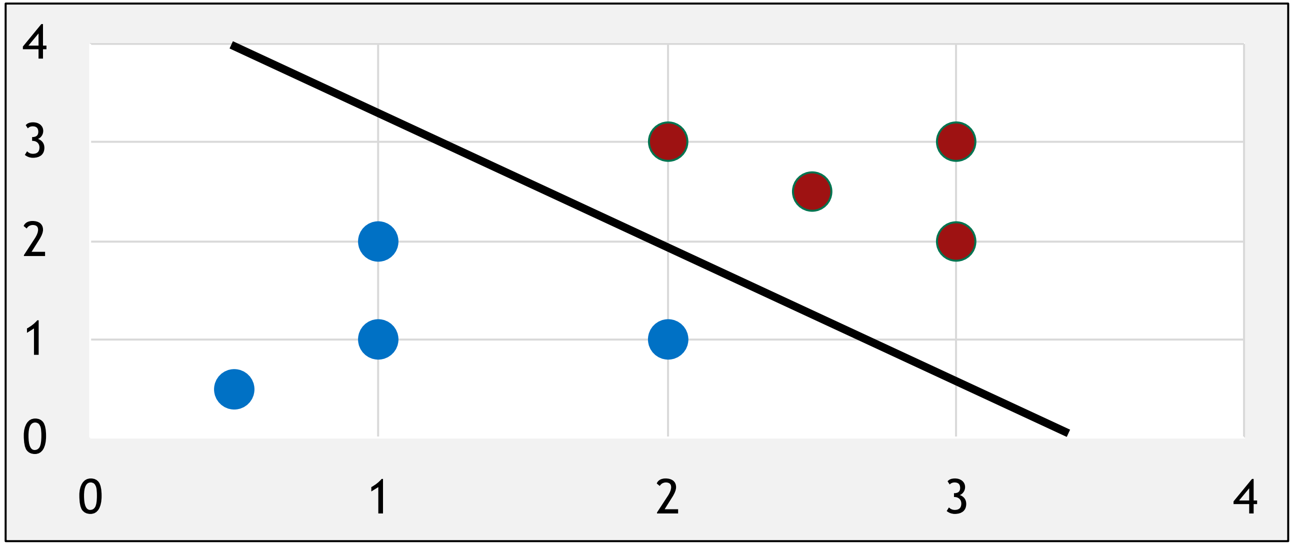




$$Loss = -(t(x) \cdot \log(p(x) + \big(1 - t(x)\big) \cdot \log\big(1 - (px)\big))$$

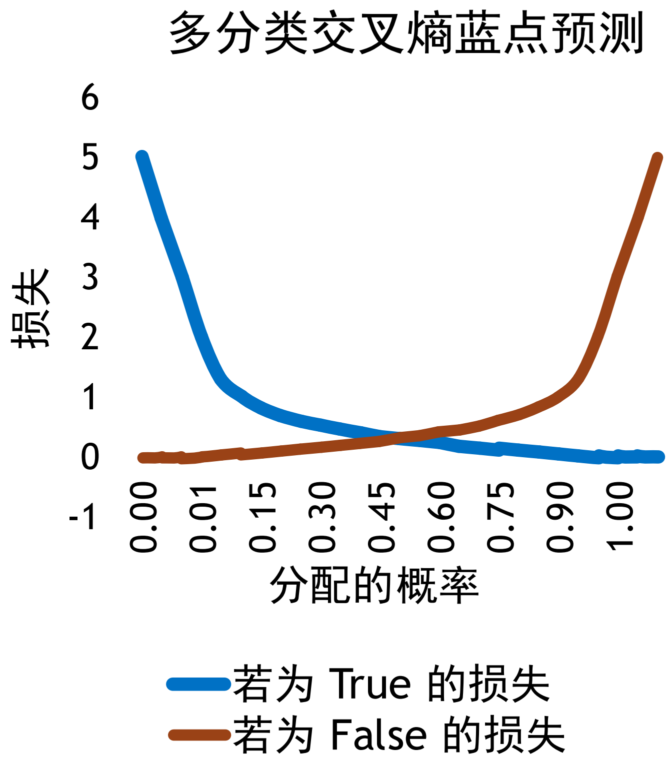
$$t(x) = target\ (0\ if\ False, 1\ if\ True)$$

$$p(x) = probability\ prediction\ of\ point\ x$$

# 多分类交叉熵



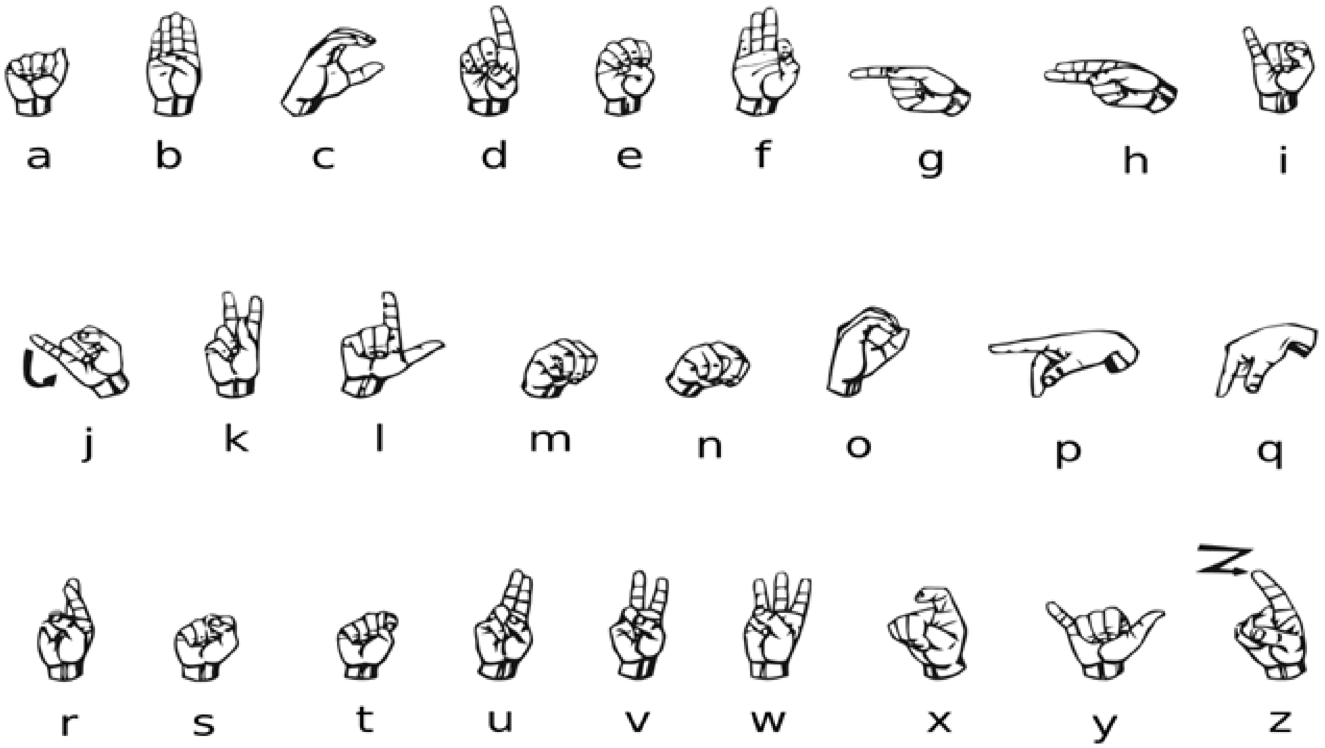

```
def cross_entropy(y_hat, y_actual):
    """Infinite error for misplaced confidence."""
    loss = log(y_hat) if y_actual else log(1-y_hat)
    return -1*loss
```



# 融会贯通

# 下一个练习

## 美国手语字母表



# 开始吧！

# 附录：梯度下降

帮助计算机欺骗微积分

# 从误差中学习

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2 = \frac{1}{n}\sum_{i=1}^{n}(y - (mx + b))^2$$

$$MSE = \frac{1}{2}((3 - (m(1) + b))^2 + (5 - (m(2) + b))^2)$$

$$\frac{\partial MSE}{\partial m} = 5m + 3b - 13 \qquad \frac{\partial MSE}{\partial b} = 3m + 2b - 8$$

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -1$$

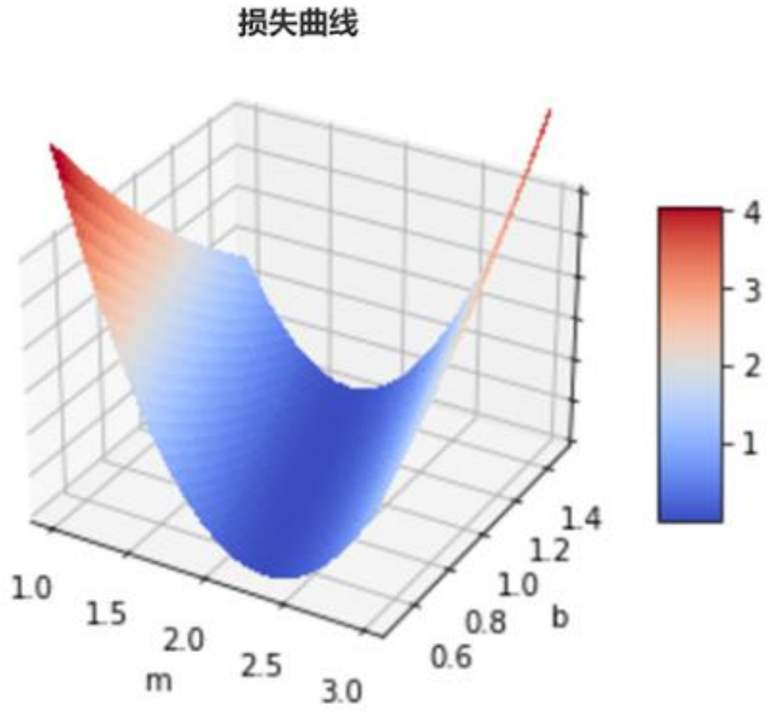
$m = -1$

$\mathrm{b} = 5$

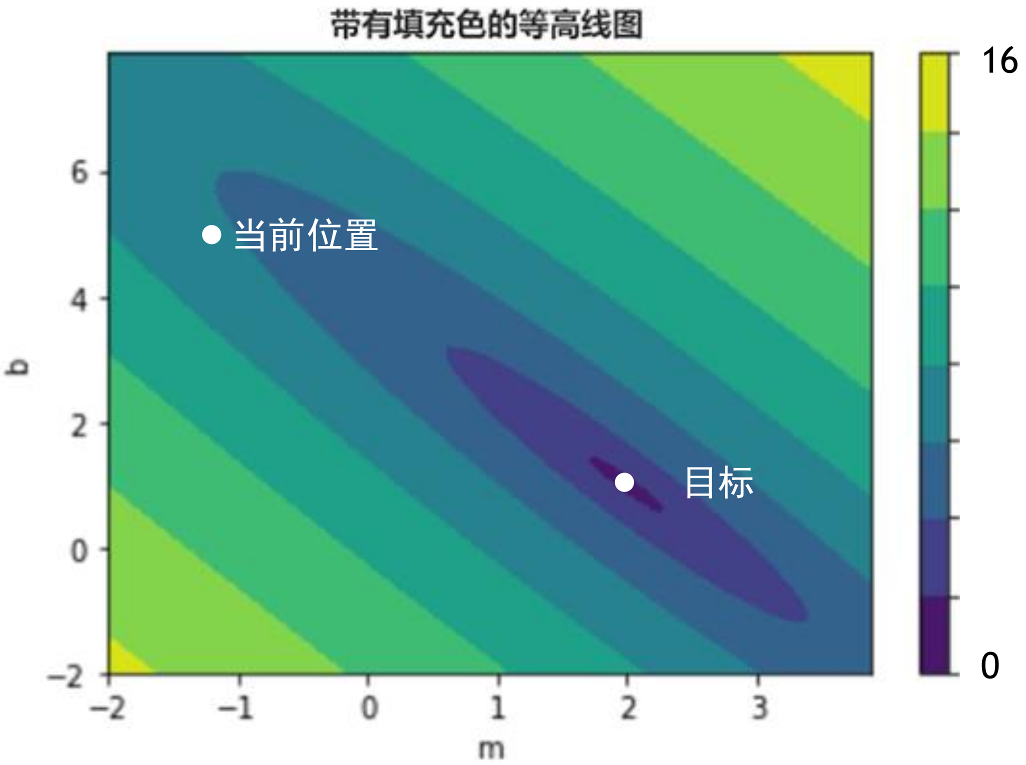


DEEP LEARNING INSTITUTE

# 损失曲线



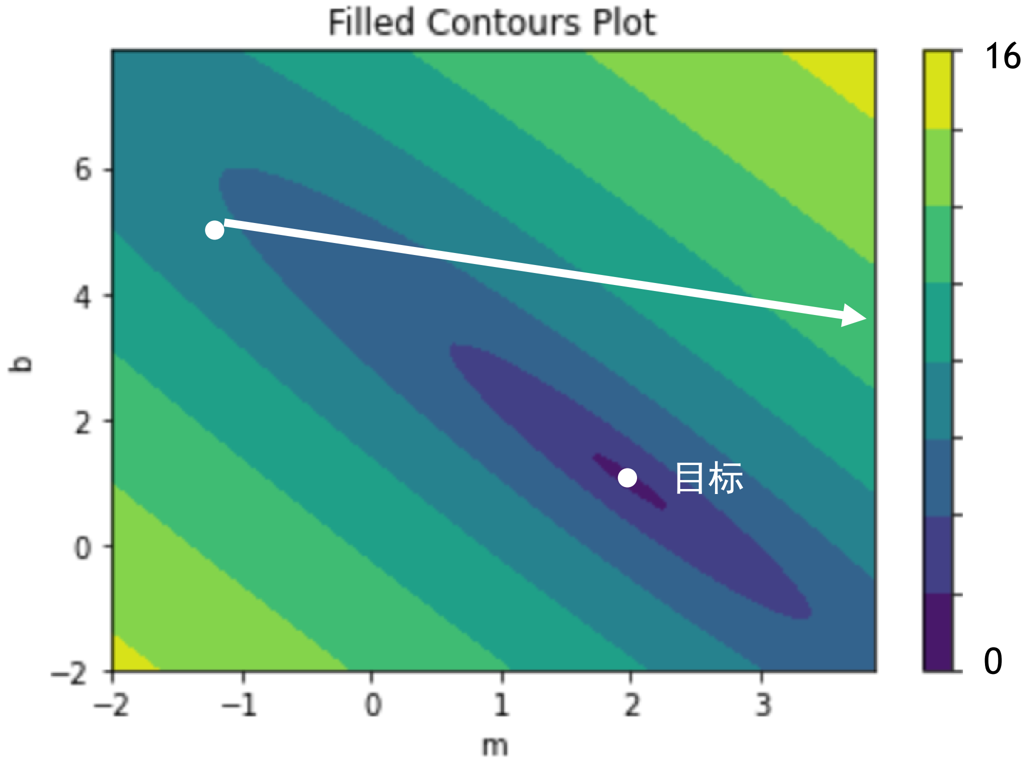
损失曲线
4
3
2
1
1.0
1.5
2.0
2.5
3.0
m
0.6
0.8
1.0
1.2
1.4
b

带有填充色的等高线图
当前位置
目标
16
0
6
4
2
0
−2
b
−2
−1
0
1
2
3
m

# 损失曲线

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -3$$

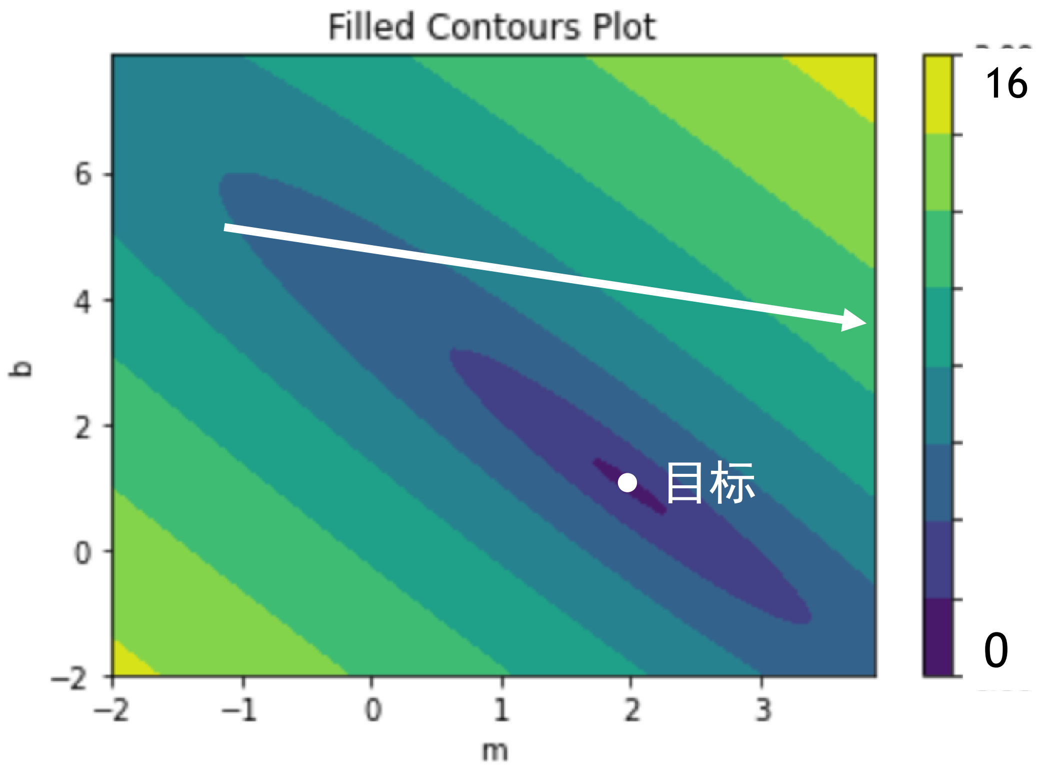


# 损失曲线

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -3$$

$$\mathrm{m} := \mathrm{m} - \lambda \frac{\partial MSE}{\partial m}$$

$$\mathrm{b} := b - \lambda \frac{\partial MSE}{\partial b}$$



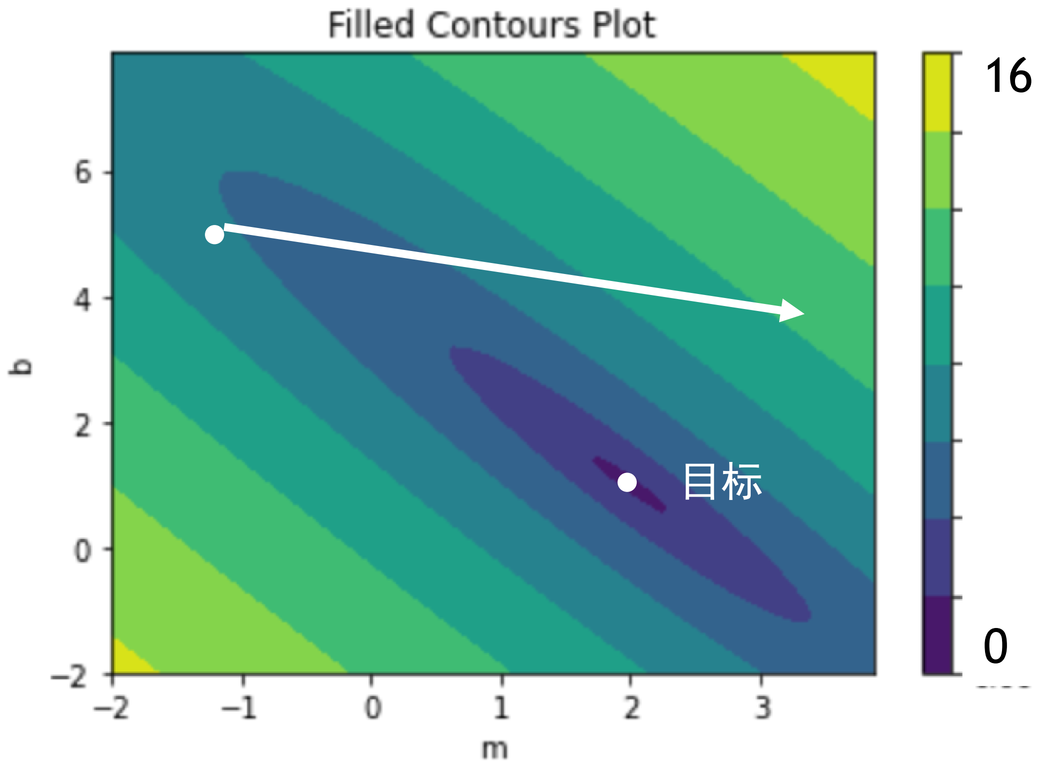
# 损失曲线

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -3$$

$$\mathrm{m} := \mathrm{m} - \lambda \frac{\partial MSE}{\partial m}$$

$$\lambda = .6$$

$$\mathrm{b} := b - \lambda \frac{\partial MSE}{\partial b}$$



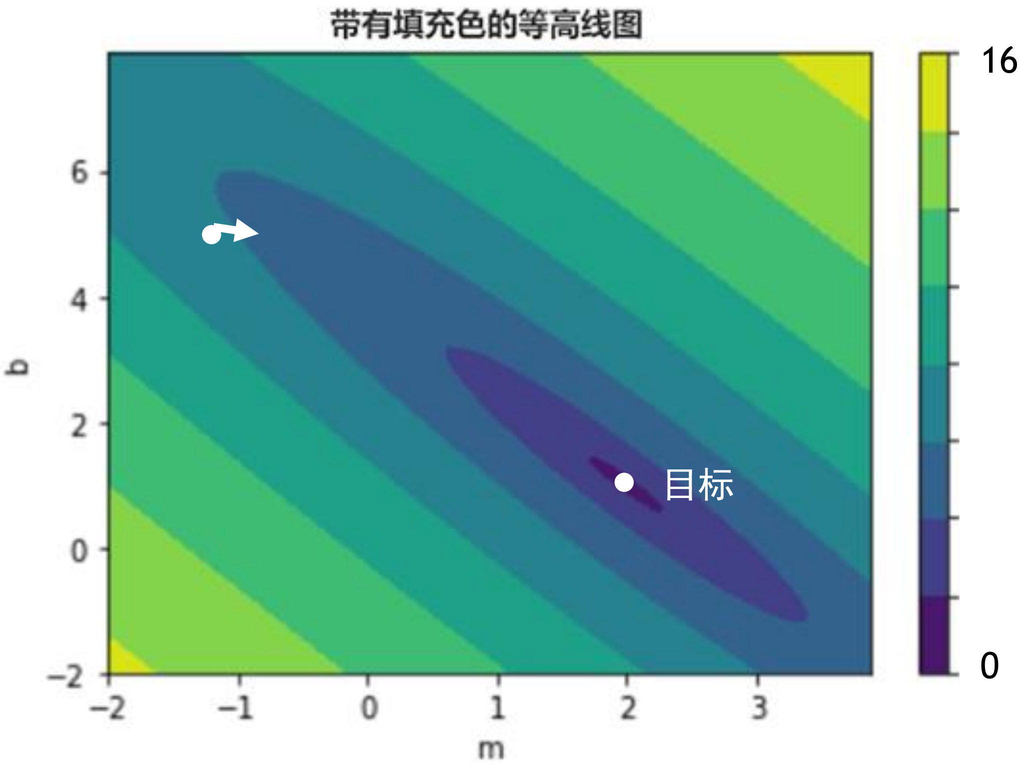
# 损失曲线

$$\frac{\partial MSE}{\partial m} = -7 \qquad \frac{\partial MSE}{\partial b} = -3$$

$$\text{m} := \text{m} - \lambda \frac{\partial MSE}{\partial m}$$

$$\lambda = .005$$

$$\text{b} := b - \lambda \frac{\partial MSE}{\partial b}$$

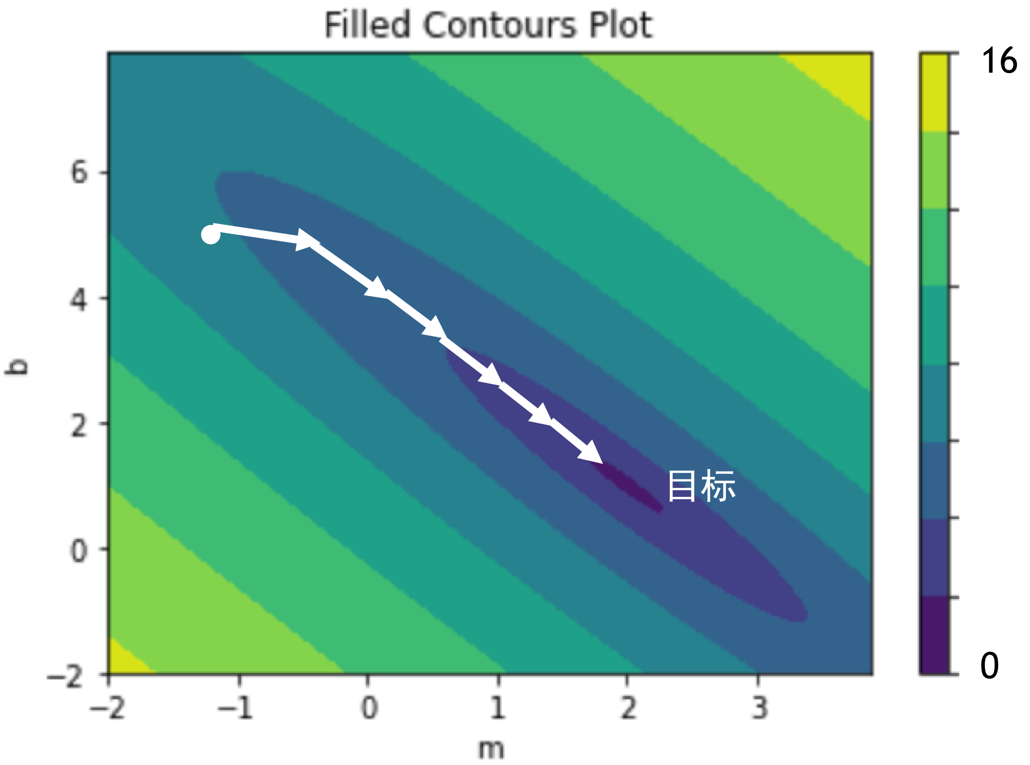


# 损失曲线

$$\lambda = .1$$

$$m := -1 + 7\,\lambda = -0.3$$

$$b := 5 + 3\,\lambda = 4.7$$



NVIDIA

DEEP
LEARNING
INSTITUTE

学习更多 DLI 课程，请访问 nvidia.cn/DLI