



人工智能与深度学习

2-7 Softmax 回归

王中雷

经济学院、王亚南经济研究院、邹至庄经济研究院

厦门大学

(第一版)

内容摘要

1. 模型

2. 前向传播

3. 后向传播

4. 例子

介绍

1. 二分类问题

- $y \in \{0, 1\}$
- 模型通常用来估计条件概率 $P(y = 1 \mid \mathbf{x})$
- 损失函数：交叉熵

2. 实际问题中，有时需要处理两个以上的类别，记为 $K(> 2)$: $y \in \{0, 1, \dots, K - 1\}$

- 考虑独热编码 (one-hot representation)
- $\mathbf{y} = (0, \dots, 1, \dots, 0)^T \in \mathbb{R}^{K \times 1}$
- 如果 $y = k$, $\mathbf{y}$ 只有第 $(k + 1)$ 位置的元素为 1, 其他位置均为 0

模型

1. 对于特征向量 $\boldsymbol{x}$,

- 我们考虑一个有 K 个输出的神经网络模型
- 每一个输出, 对应于特定类别的“得分”(例如, 条件概率)

2. 与二分类模型主要的不同之处在于, 输出层神经元的个数

- 对于二分类问题, 输出层只有一个神经元
- 对于一般的分类问题, 输出层往往包含 $d^{[L]} = K$ 个神经元

Softmax 函数

1. 对于一个 d 维向量 $\mathbf{z} = (z_1, \dots, z_d)^T \in \mathbb{R}^{d \times 1}$, softmax 函数为

$$\text{Softmax}(\mathbf{z}) = \begin{pmatrix} \exp(z_1) / \sum_{i=1}^d \exp(z_i) \\ \exp(z_2) / \sum_{i=1}^d \exp(z_i) \\ \vdots \\ \exp(z_d) / \sum_{i=1}^d \exp(z_i) \end{pmatrix}$$

2. 备注

- 由于指数函数, Softmax 函数放大了自变量 $\{z_1, \dots, z_d\}$ 之间的差异
- 平移不变性: 对于任意 $c \in \mathbb{R}$, 我们有 $\text{Softmax}(\mathbf{z} + c) = \text{Softmax}(\mathbf{z})$
- 估计结果取决于相对大小关系

Softmax 函数

1. 记 $\mathbf{a} = \text{Softmax}(\mathbf{z}) \in \mathbb{R}^{d \times 1}$

- $\mathbf{a} = (a_1, \dots, a_d)^T$

2. 那么，我们有（请大家自行验证）

$$\frac{\partial \mathbf{a}^T}{\partial \mathbf{z}} = \left(\frac{\partial a_m}{\partial z_k} \right) = (a_m \cdot (\delta_{mk} - a_k))$$

- $k = 1, \dots, d$: 行标
- $m = 1, \dots, d$: 列标
- δ_{mk} : 示性函数

$$\delta_{mk} = \begin{cases} 1, & m = k \\ 0, & m \neq k \end{cases}$$

符号

1. 回顾

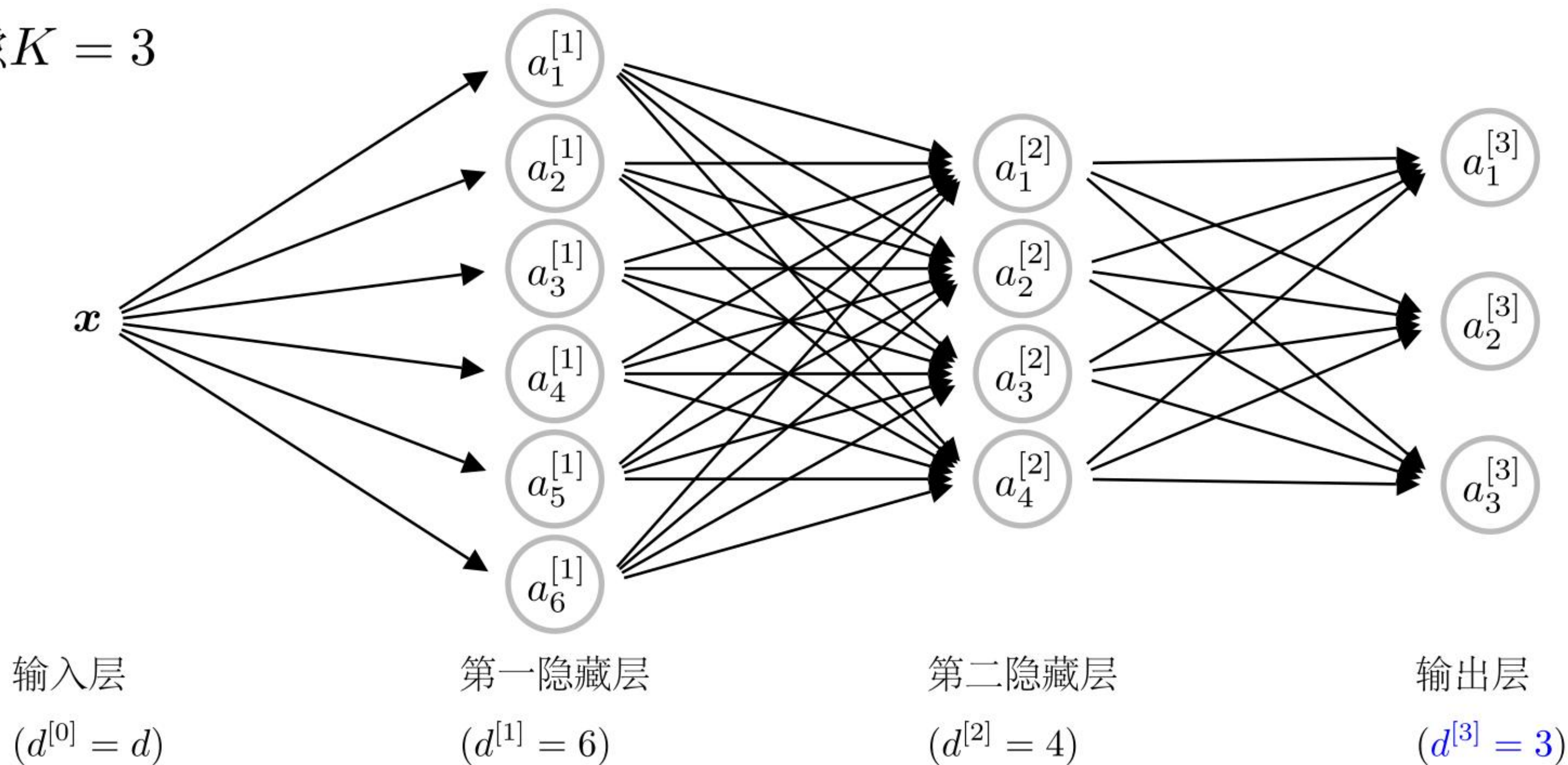
- L : 神经网络模型中的层数
- $d^{[l]}$: 第 l 层神经元个数 ($l = 0, \dots, L$)
- $\mathbf{a}^{[l]} = (a_1^{[l]}, \dots, a_{d^{[l]}}^{[l]})^T \in \mathbb{R}^{d^{[l]} \times 1}$
- $\mathbf{W}^{[l]} = (\mathbf{w}_1^{[l]}, \dots, \mathbf{w}_{d^{[l]}}^{[l]})^T \in \mathbb{R}^{d^{[l]} \times d^{[l-1]}}$
- $\mathbf{b}^{[l]} = (b_1^{[l]}, \dots, b_{d^{[l]}}^{[l]})^T \in \mathbb{R}^{d^{[l]} \times 1}$

2. 对于二分类问题, 我们有 $d^{[L]} = 1$

3. 对于一般分类问题, 我们有 $d^{[L]} = K$

符号

1. 考虑 $K = 3$



前向传播

1. 令 $\mathbf{A}^{[0]} = \mathbf{X} \in \mathbb{R}^{n \times d}$

2. 对于 $l = 1, \dots, L - 1$,

$$\mathbf{Z}^{[l]} = (\mathbf{b}^{[l]})^{\text{T}} + \mathbf{A}^{[l-1]} \cdot (\mathbf{W}^{[l]})^{\text{T}},$$

$$\mathbf{A}^{[l]} = \sigma^{[l]}(\mathbf{Z}^{[l]})$$

- $\sigma^{[l]}(z)$: 第 l 层对应的激活函数
- Python 中的广播机制, 用于计算激活值

前向传播

1. 对于最后一层（输出层），

$$\mathbf{W}^{[L]} \in \mathbb{R}^{K \times d^{[L-1]}}, \quad \mathbf{b} \in \mathbb{R}^{K \times 1}$$

2. 线性变换是一样的：

$$\mathbf{Z}^{[L]} = (\mathbf{b}^{[L]})^T + \mathbf{A}^{[L-1]} \cdot (\mathbf{W}^{[L]})^T$$

- $\mathbf{Z}^{[L]} \in \mathbb{R}^{n \times K}$

3. 激活函数为 Softmax 函数：

$$\mathbf{A}^{[L]} = \text{Softmax}(\mathbf{Z}^{[L]})$$

- $\mathbf{A}^{[L]} \in \mathbb{R}^{n \times K}$

- Softmax 函数作用于 $\mathbf{Z}^{[L]}$ 的每一行

- $\mathbf{A}^{[L]}$ 每一行元素之和为 1

前向传播

1. 输出层对应的 $\mathbf{A}^{[L]}$ 的维度

- $\mathbf{A}^{[L]} \in \mathbb{R}^{n \times 1}$: 二分类问题
- $\mathbf{A}^{[L]} \in \mathbb{R}^{n \times K}$: 一般的 softmax 回归问题 (K 分类问题)

2. Softmax 回归对应的代价函数为

$$\mathcal{J}(\boldsymbol{\theta}) = -n^{-1} \sum_{i=1}^n \sum_{k=1}^K y_{ik} \log a_{ik}^{[L]}$$

- $\boldsymbol{\theta}$: 模型参数
- $\mathbf{y}_i = (y_{i1}, \dots, y_{iK})^T$: 对应于第 i 个样本点的独热编码
- $a_{ik}^{[L]}$: 对第 i 个训练样本, 其标签为 $(k-1)$ 的条件概率 $P(y_i = k-1 \mid \mathbf{x}_i)$ 的估计

后向传播

1. 我们只需关注输出层的后向传播结果即可
2. $\mathbf{dA}^{[L]}$ 可通过代价函数得到
3. Softmax 回归问题的代价函数为

$$\mathcal{J}(\boldsymbol{\theta}) = -n^{-1} \sum_{i=1}^n \sum_{k=1}^K y_{ik} \log a_{ik}^{[L]}$$

4. 因此, $\mathbf{dA}^{[L]}$ 的位于 (i, k) 位置的元素为

$$\mathrm{d}a_{ik}^{[L]} = \frac{\partial \mathcal{J}}{\partial a_{ik}^{[L]}} = -n^{-1} \frac{y_{ik}}{a_{ik}^{[L]}}$$

5. $\mathbf{dA}^{[L]} = -n^{-1} \mathbf{Y} / \mathbf{A}^{[L]}$ (逐元素相除)

后向传播

1. 接下来，我们聚焦于 $d\mathbf{Z}^{[L]}$ 的计算
2. 对于 $i = 1, \dots, n$ 以及 $k = 1, \dots, K$, $z_{ik}^{[L]}$ 只被用于计算 $\mathbf{A}^{[L]}$ 的第 i 行
3. 对于 $h \neq i$, 我们有

$$\frac{\partial a_{hm}^{[L]}}{\partial z_{ik}^{[L]}} = 0 \quad (m = 1, \dots, K)$$

4. 对于 $i = 1, \dots, n; k = 1, \dots, K$, 我们有

$$dz_{ik}^{[L]} = \frac{\partial \mathcal{J}}{\partial z_{ik}^{[L]}} = \sum_{h=1}^n \sum_{m=1}^K \frac{\partial \mathcal{J}}{\partial a_{hm}^{[L]}} \frac{\partial a_{hm}^{[L]}}{\partial z_{ik}^{[L]}} = \sum_{m=1}^K \frac{\partial \mathcal{J}}{\partial a_{im}^{[L]}} \frac{\partial a_{im}^{[L]}}{\partial z_{ik}^{[L]}}$$

5. 为了得到 $d\mathbf{Z}^{[L]}$, 我们只需要得到它的所有元素 $\{dz_{ik}^{[L]} : i = 1, \dots, n; k = 1, \dots, K\}$ 即可

后向传播

1. 我们已经得到

$$\frac{\partial \mathcal{J}}{\partial a_{im}^{[L]}} = -n^{-1} \frac{y_{im}}{a_{im}^{[L]}}, \quad \frac{\partial a_{im}^{[L]}}{\partial z_{ik}^{[L]}} = a_{im}^{[L]} \cdot (\delta_{mk} - a_{ik}^{[L]})$$

2. 因此, 我们有

$$\begin{aligned} dz_{ik}^{[L]} &= \sum_{m=1}^K \frac{\partial \mathcal{J}}{\partial a_{im}^{[L]}} \frac{\partial a_{im}^{[L]}}{\partial z_{ik}^{[L]}} = -n^{-1} \sum_{m=1}^K \frac{y_{im}}{a_{im}^{[L]}} \cdot a_{im}^{[L]} \cdot (\delta_{mk} - a_{ik}^{[L]}) \\ &= -n^{-1} \sum_{m=1}^K y_{im} \cdot \delta_{mk} + n^{-1} a_{ik}^{[L]} \cdot \sum_{m=1}^K y_{im} = -n^{-1} (y_{ik} - a_{ik}^{[L]}) \end{aligned}$$

- 最后一个等式成立, 是因为 $\sum_{m=1}^K y_{im} = 1$

3. 因此, 我们有 $d\mathbf{Z}^{[L]} = -n^{-1}(\mathbf{Y} - \mathbf{A}^{[L]})$ (输出层的误差矩阵)

后向传播

1. 因此，对于输出层，我们有如下结果

$$d\mathbf{A}^{[L]} = -n^{-1}\mathbf{Y} / \mathbf{A}^{[L]} \quad (\text{逐元素相除})$$

$$d\mathbf{Z}^{[L]} = -n^{-1}(\mathbf{Y} - \mathbf{A}^{[L]}) \quad (\text{与二分类问题模型结论类似，但维度不同})$$

$$d\mathbf{W}^{[L]} = (d\mathbf{Z}^{[L]})^T \cdot \mathbf{A}^{[L-1]},$$

$$d\mathbf{b}^{[L]} = (d\mathbf{Z}^{[L]})^T \cdot \mathbf{1},$$

$$d\mathbf{A}^{[L-1]} = d\mathbf{Z}^{[L]} \cdot \mathbf{W}^{[L]}$$

- $\mathbf{1} \in \mathbb{R}^{n \times 1}$: 由元素 1 组成的维度为 n 的列向量

后向传播

1. Softmax 回归的隐藏层的计算模式，与二分类模型完全一致
2. 因此，二分类模型中得到的后向传播结果，可以直接使用
3. 即对于 $l = L - 1, \dots, 1$ ，假设 $\mathbf{dA}^{[l]}$ 已知，我们有

$$\mathbf{dZ}^{[l]} = \mathbf{dA}^{[l]} \circ \sigma^{[l]'}(\mathbf{Z}^{[l]}),$$

$$\mathbf{dW}^{[l]} = (\mathbf{dZ}^{[l]})^T \cdot \mathbf{A}^{[l-1]},$$

$$\mathbf{db}^{[l]} = (\mathbf{dZ}^{[l]})^T \cdot \mathbf{1},$$

$$\mathbf{dA}^{[l-1]} = \mathbf{dZ}^{[l]} \cdot \mathbf{W}^{[l]}$$

- $\mathbf{1} \in \mathbb{R}^{n \times 1}$: 由元素 1 组成的维度为 n 的列向量

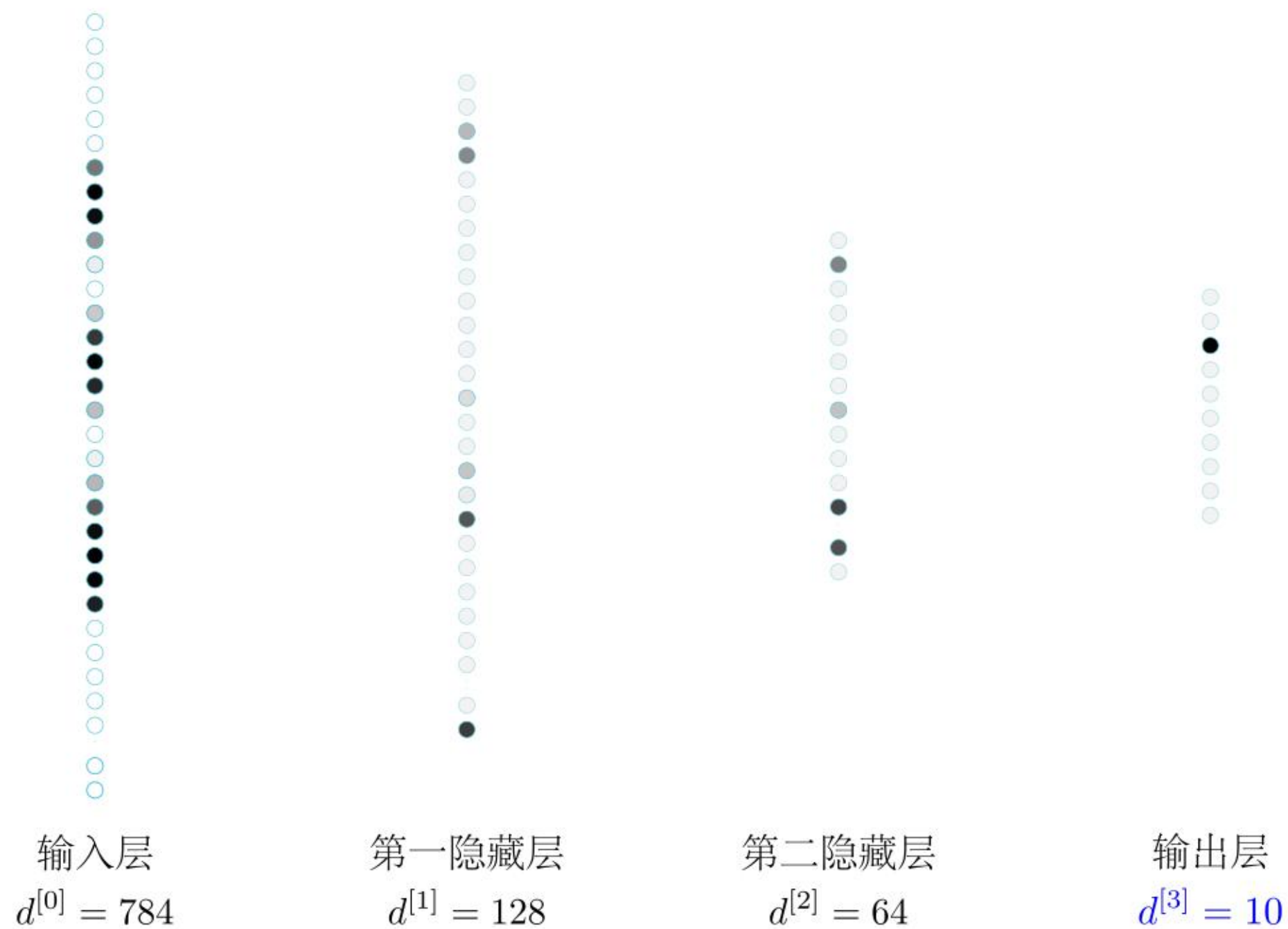
例子

测试图片



神经网络模型（两个隐藏层）

估计结果



课程总结

1. 理解 softmax 回归的四个方面

- 模型结果：输出层包含 K 个神经元，每个输出对应一个类别概率的估计
- 前向传播：隐藏层计算不变，输出层采用 Softmax 激活函数
- 代价函数：交叉熵
- 后向传播：隐藏层计算不变，输出层对应于样本平均后的预测误差， $d\mathbf{Z}^{[L]} = n^{-1}(\mathbf{A}^{[L]}$