# 人工智能与深度学习

# 2-2 只有一个隐藏层的神经网络 II

（多个训练样本）

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）

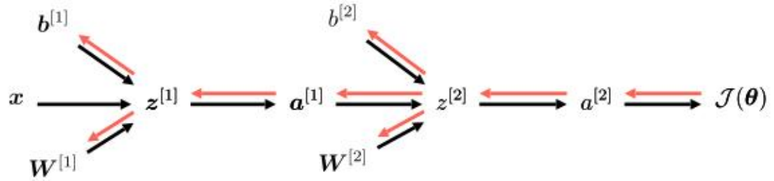
# 内容摘要

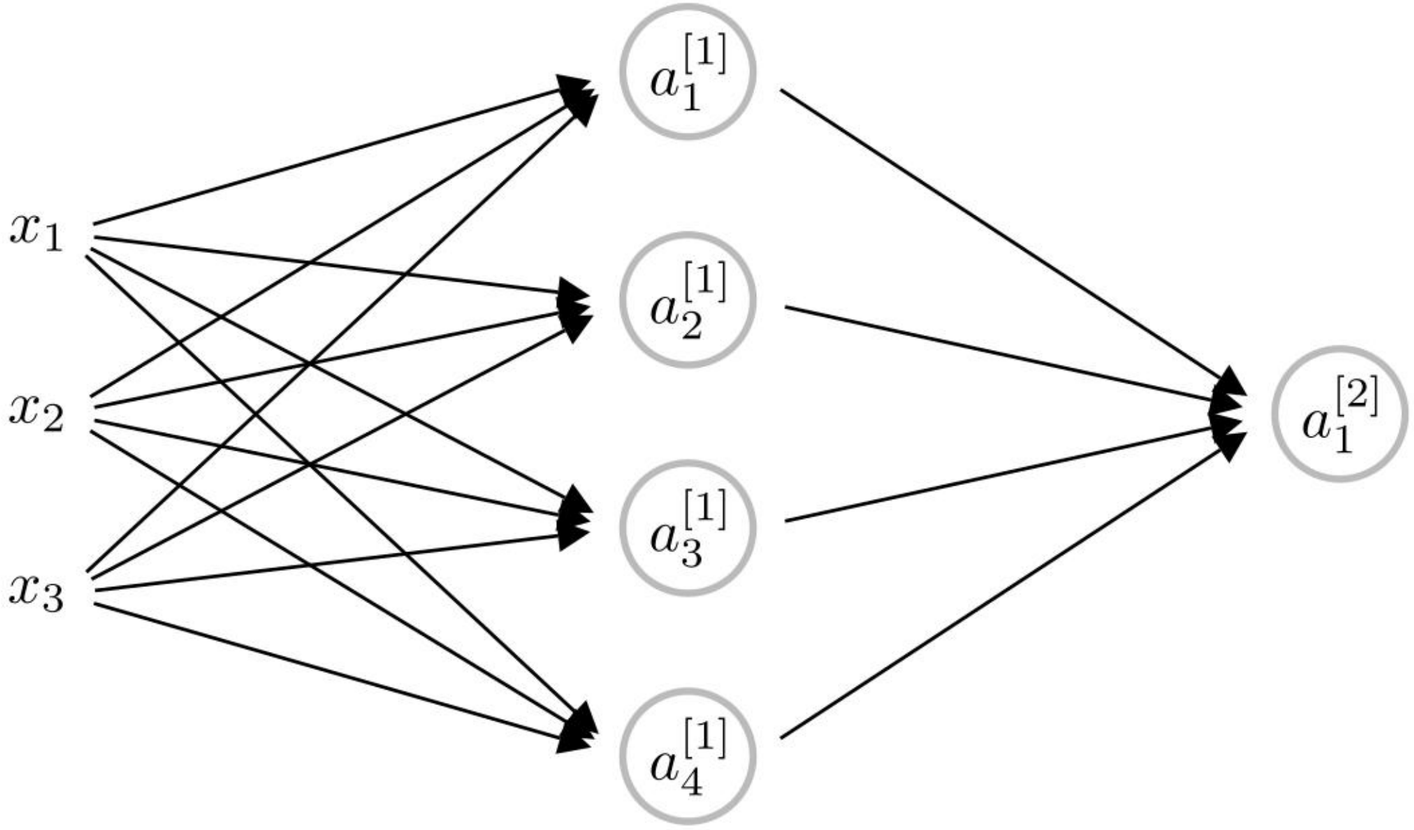
1. 前向传播

2. 后向传播

3. 批量梯度下降法

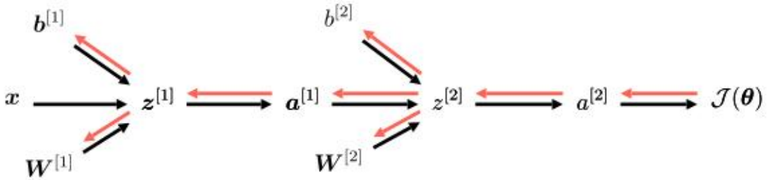
# 回顾




输入层 $(d^{[0]} = 3)$　　第一隐藏层 $(d^{[1]} = 4)$　　输出层 $(d^{[2]} = 1)$

# 回顾

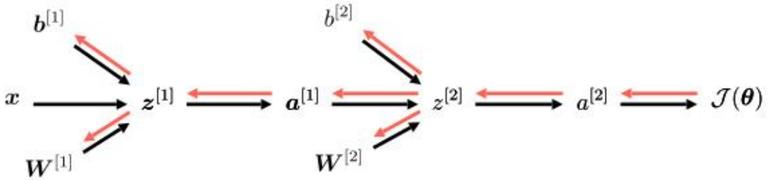



1．回顾

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

2．我们考虑一个 0-1 二分类问题

# 回顾




1．如果我们只有一个训练样本$(\boldsymbol{x}, y)$，前向传播的计算如下

$$\boldsymbol{z}^{[1]} = \boldsymbol{b}^{[1]} + \boldsymbol{W}^{[1]} \cdot \boldsymbol{x},$$

$$\boldsymbol{a}^{[1]} = \sigma(\boldsymbol{z}^{[1]}),$$

$$z^{[2]} = b^{[2]} + \boldsymbol{W}^{[2]} \cdot \boldsymbol{a}^{[1]},$$

$$a^{[2]} = \sigma(z^{[2]})$$

- Python 中的广播机制被用于计算 $\boldsymbol{z}^{[1]}$ 和 $\boldsymbol{a}^{[1]}$

2．以上出现的是列向量

3．代价函数为

$$\mathcal{J} = \mathcal{L} = -\left\{y \log a^{[2]} + (1-y) \log\left(1 - a^{[2]}\right)\right\}$$

# 前向传播

1. 假设我们有$n$ 个训练样本$\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$

2. 记$\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}} \in \mathbb{R}^{n \times d}$

3. 基于向量化的前向传播，我们有

$$\boldsymbol{Z}^{[1]} = (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{X} \cdot (\boldsymbol{W}^{[1]})^{\mathrm{T}} \in \mathbb{R}^{n \times d^{[1]}},$$

$$\boldsymbol{A}^{[1]} = \sigma(\boldsymbol{Z}^{[1]}) \in \mathbb{R}^{n \times d^{[1]}},$$

$$\boldsymbol{Z}^{[2]} = b^{[2]} + \boldsymbol{A}^{[1]} \cdot (\boldsymbol{W}^{[2]})^{\mathrm{T}} \in \mathbb{R}^{n \times d^{[2]}},$$

$$\boldsymbol{A}^{[2]} = \sigma(\boldsymbol{Z}^{[2]}) \in \mathbb{R}^{n \times d^{[2]}}$$

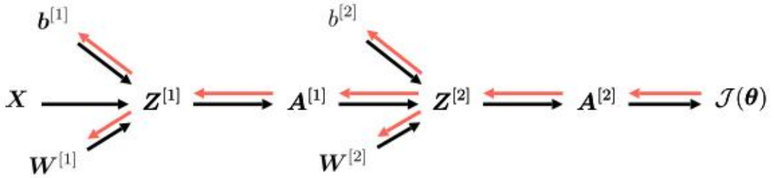
- Python 中的广播机制被用于以上计算

$$\boldsymbol{Z}^{[1]} = (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{pmatrix} \cdot (\boldsymbol{W}^{[1]})^{\mathrm{T}}$$

$$= \begin{pmatrix} (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{x}_1^{\mathrm{T}} \cdot (\boldsymbol{W}^{[1]})^{\mathrm{T}} \\ (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{x}_2^{\mathrm{T}} \cdot (\boldsymbol{W}^{[1]})^{\mathrm{T}} \\ \vdots \\ (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{x}_n^{\mathrm{T}} \cdot (\boldsymbol{W}^{[1]})^{\mathrm{T}} \end{pmatrix}$$

4. 实际上，我们将信息进行行拼接

# 前向传播




1. 假设我们有$n$ 个训练样本$\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$

2. 记$\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}} \in \mathbb{R}^{n \times d}$

3. 基于向量化的前向传播，我们有

$$\boldsymbol{Z}^{[1]} = (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{X} \cdot (\boldsymbol{W}^{[1]})^{\mathrm{T}} \in \mathbb{R}^{n \times d^{[1]}},$$

$$\boldsymbol{A}^{[1]} = \sigma(\boldsymbol{Z}^{[1]}) \in \mathbb{R}^{n \times d^{[1]}},$$

$$\boldsymbol{Z}^{[2]} = b^{[2]} + \boldsymbol{A}^{[1]} \cdot (\boldsymbol{W}^{[2]})^{\mathrm{T}} \in \mathbb{R}^{n \times d^{[2]}},$$

$$\boldsymbol{A}^{[2]} = \sigma(\boldsymbol{Z}^{[2]}) \in \mathbb{R}^{n \times d^{[2]}}$$

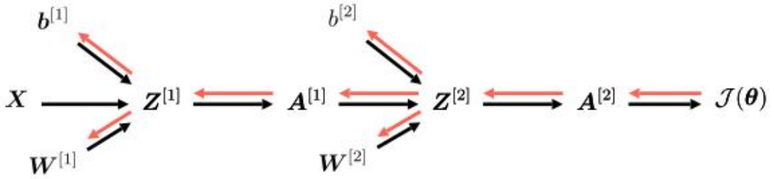
$$\boldsymbol{A}^{[1]} = \sigma(\boldsymbol{Z}^{[1]}) = (\sigma(Z_{ij}^{[1]}))$$

$$(\boldsymbol{Z}^{[1]} = (Z_{ij}^{[1]}); i = 1, \ldots, n; j = 1, \ldots, d^{[1]})$$

- Python 中的广播机制被用于以上计算

4. 实际上，我们将信息进行行拼接

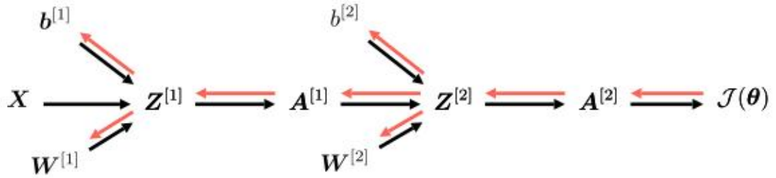
# 前向传播




1. 广播机制能够使 Python 代码更加简洁、高效，并能够节省内存

- 代码简洁并易于阅读
- 避免了临时变量的存储，内存优势明显
- 通过 C/Fortran 语言的优化，大规模提升计算性能

# 前向传播




1. 对于右下角的例子，我们有$d^{[0]} = 3, d^{[1]} = 4, d^{[2]} = 1$

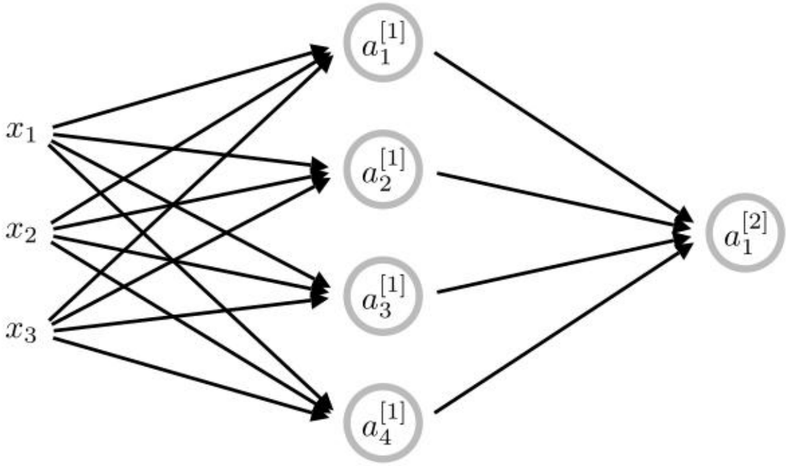
2. 因此，我们有

$$\boldsymbol{Z}^{[1]} = (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{X} \cdot (\boldsymbol{W}^{[1]})^{\mathrm{T}} \in \mathbb{R}^{n \times 4},$$

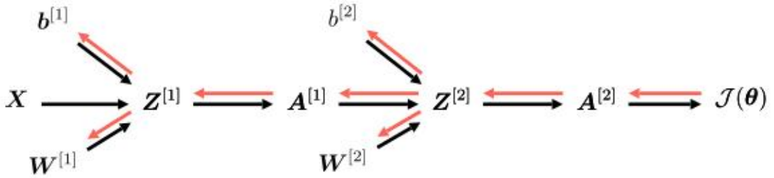
$$\boldsymbol{A}^{[1]} = \sigma(\boldsymbol{Z}^{[1]}) \in \mathbb{R}^{n \times 4},$$

$$\boldsymbol{Z}^{[2]} = b^{[2]} + \boldsymbol{A}^{[1]} \cdot (\boldsymbol{W}^{[2]})^{\mathrm{T}} \in \mathbb{R}^{n \times 1},$$

$$\boldsymbol{A}^{[2]} = \sigma(\boldsymbol{Z}^{[2]}) \in \mathbb{R}^{n \times 1}$$



# 前向传播



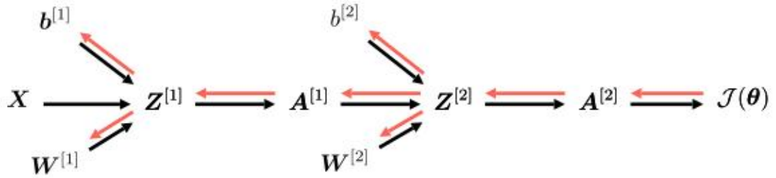

1．二分类问题的交叉熵代价函数为

$$\mathcal{J}=-\frac{1}{n}\sum_{i=1}^{n}\left\{y_i\log a_i^{[2]}+(1-y_i)\log\left(1-a_i^{[2]}\right)\right\}$$

- $a_i^{[2]}$：对$P(y_i=1\mid\boldsymbol{x}_i)$ 的估计值

2．损失函数的形式由具体问题决定

# 后向传播




1. 记$\mathrm{d}\cdot = \dfrac{\partial \mathcal{J}}{\partial \cdot}$

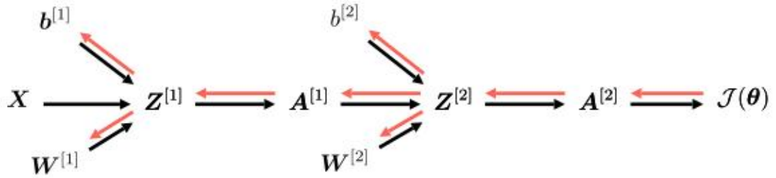
2. 回顾: 对于一个样本点$(\boldsymbol{x}, y)$，我们有

$$\mathrm{d}z^{[2]} = a^{[2]} - y, \quad \mathrm{d}b^{[2]} = \mathrm{d}z^{[2]}, \quad \mathrm{d}\boldsymbol{W}^{[2]} = \mathrm{d}z^{[2]} \cdot (\boldsymbol{a}^{[1]})^{\mathrm{T}},$$
$$\mathrm{d}\boldsymbol{z}^{[1]} = \mathrm{d}z^{[2]} \cdot \boldsymbol{D} \cdot (\boldsymbol{W}^{[2]})^{\mathrm{T}}, \quad \mathrm{d}\boldsymbol{b}^{[1]} = \mathrm{d}\boldsymbol{z}^{[1]}, \quad \mathrm{d}\boldsymbol{W}^{[1]} = \mathrm{d}\boldsymbol{z}^{[1]} \cdot \boldsymbol{x}^{\mathrm{T}}$$

   - $\boldsymbol{D} = \mathrm{diag}(\{a_j^{[1]}(1 - a_j^{[1]}) : j = 1, \ldots, d^{[1]}\})$

3. 我们有
   - $\mathrm{d}z^{[2]}$ 可通过 $\mathrm{d}a^{[2]}$ 得到，而后者通过代价函数得到
   - $\mathrm{d}\boldsymbol{z}^{[1]}$ 可通过 $\mathrm{d}\boldsymbol{a}^{[1]}$ 得到，而后者可利用计算图并基于 $\mathrm{d}z^{[2]}$ 直接得到

# 后向传播



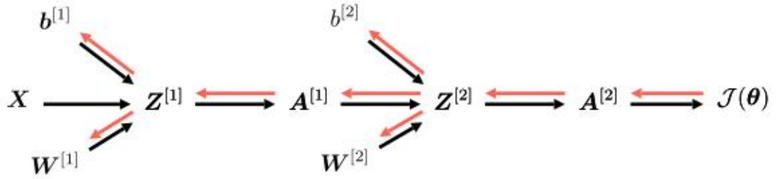

1. 一般地，如果我们有$n$ 个训练样本，则我们有

$$\mathrm{d}\boldsymbol{\theta} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial \mathcal{L}(y_i, a_i^{[2]})}{\partial \boldsymbol{\theta}}$$

   - $\mathcal{L}(y, a)$：基于观测标签 $y$ 及其估计值 $a$ 的损失函数
   - 针对于二分类问题，我们有 $\mathcal{L}(y, a) = -\{y \log a + (1-y)\log(1-a)\}$

2. 因此，为了得到$\mathrm{d}\boldsymbol{\theta}$，我们只需要对训练样本对应的偏导数取平均即可
3. 利用向量化

# 后向传播



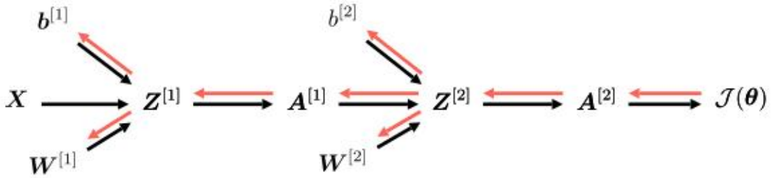

1．基于向量化，我们有

$$\mathrm{d}\boldsymbol{Z}^{[2]} = n^{-1}(\boldsymbol{A}^{[2]} - \boldsymbol{Y}), \quad \mathrm{d}b^{[2]} = (\mathrm{d}\boldsymbol{Z}^{[2]})^{\mathrm{T}} \cdot \mathbf{1}, \quad \mathrm{d}\boldsymbol{W}^{[2]} = (\mathrm{d}\boldsymbol{Z}^{[2]})^{\mathrm{T}} \cdot \boldsymbol{A}^{[1]},$$

$$\mathrm{d}\boldsymbol{Z}^{[1]} = (\mathrm{d}\boldsymbol{Z}^{[2]} \cdot \boldsymbol{W}^{[2]}) \circ \sigma'(\boldsymbol{Z}^{[1]}), \quad \mathrm{d}\boldsymbol{b}^{[1]} = (\mathrm{d}\boldsymbol{Z}^{[1]})^{\mathrm{T}} \cdot \mathbf{1}, \quad \mathrm{d}\boldsymbol{W}^{[1]} = (\mathrm{d}\boldsymbol{Z}^{[1]})^{\mathrm{T}} \cdot \boldsymbol{X}$$

- $\sigma'(z)$：对应于$\sigma(z)$ 的导数
  - 对于 sigmoid 函数$\sigma(z) = \{1 + \exp(-z)\}^{-1}$，我们有$\sigma'(z) = \sigma(z)\{1 - \sigma(z)\}$
- $\sigma'(\boldsymbol{Z}^{[1]}) = (\sigma'(z_{ij}^{[1]})) \in \mathbb{R}^{n \times d^{[1]}}$：利用广播机制，计算$\boldsymbol{Z}^{[1]}$ 每个元素对应的导数值
  - $\boldsymbol{Z}^{[1]} = (z_{ij}^{[1]}) \quad (i = 1, \ldots, n; j = 1, \ldots, d^{[1]})$
- $\circ$：Hadamard 乘积（矩阵对应位置元素相乘）

# 推导细节

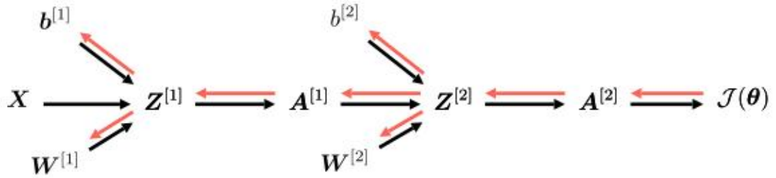



1. 我们首先展示，输出层相关导数的计算细节
2. 回顾：代价函数为

$$\mathcal{J} = -\frac{1}{n}\sum_{i=1}^{n}\left\{y_i \log a_i^{[2]} + (1-y_i)\log\left(1-a_i^{[2]}\right)\right\}$$

3. 为了得到 $\mathrm{d}\boldsymbol{A}^{[2]} = (\partial\mathcal{J}/\partial a_1^{[2]}, \ldots, \partial\mathcal{J}/\partial a_n^{[2]})^{\mathrm{T}} \in \mathbb{R}^{n\times 1}$，我们将 $\mathcal{J}$ 看成 $\boldsymbol{A}^{[2]}$ 的函数
   - $\boldsymbol{A}^{[2]} = (a_1^{[2]}, \ldots, a_n^{[2]})^{\mathrm{T}} \in \mathbb{R}^{n\times 1}$
   - $a_i^{[2]}$ 为对 $P(y_i = 1 \mid \boldsymbol{x}_i)$ 的估计量
4. 我们可得到 $\partial\mathcal{J}/\partial a_i^{[2]} = n^{-1}(a_i^{[2]} - y_i)/\{a_i^{[2]}(1-a_i^{[2]})\}$

# 推导细节




1. 因此，我们有

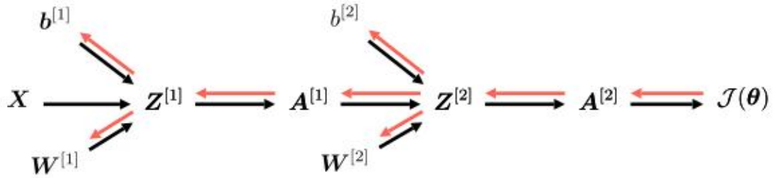
$$\begin{aligned}\mathrm{d}\boldsymbol{A}^{[2]} &= n^{-1}((a_1^{[2]}-y_1)/\{a_1^{[2]}(1-a_1^{[2]})\},\ldots,(a_n^{[2]}-y_n)/\{a_n^{[2]}(1-a_n^{[2]})\})^{\mathrm{T}} \\ &= n^{-1}\boldsymbol{D}^{-1}\cdot(\boldsymbol{A}^{[2]}-\boldsymbol{Y})\end{aligned}$$

- $\boldsymbol{D}\in\mathbb{R}^{n\times n}$ 为对角阵，其第 $i$ 个对角元素为 $a_i^{[2]}(1-a_i^{[2]})$

2. 为了得到 $\mathrm{d}\boldsymbol{Z}^{[2]}=(\partial\mathcal{J}/\partial z_1^{[2]},\ldots,\partial\mathcal{J}/\partial z_n^{[2]})^{\mathrm{T}}\in\mathbb{R}^{n\times 1}$，我们利用链式法则，并将 $\mathcal{J}$ 看成 $\boldsymbol{A}^{[2]}$ 的函数，将 $\boldsymbol{A}^{[2]}$ 看成 $\boldsymbol{Z}^{[2]}$ 的函数

- $\boldsymbol{Z}^{[2]}=(z_1^{[2]},\ldots,z_n^{[2]})^{\mathrm{T}}\in\mathbb{R}^{n\times 1}$
- $z_i^{[2]}$ 是第 $i$ 个训练样本，在模型第二层线性变换的结果

# 推导细节




1. 由于 $\boldsymbol{A}^{[2]}$ 和 $\boldsymbol{Z}^{[2]}$ 都是列向量，我们有

$$\mathrm{d}\boldsymbol{Z}^{[2]} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{Z}^{[2]}} = \frac{\partial (\boldsymbol{A}^{[2]})^{\mathrm{T}}}{\partial \boldsymbol{Z}^{[2]}} \cdot \frac{\partial \mathcal{J}}{\partial \boldsymbol{A}^{[2]}} = \frac{\partial (\boldsymbol{A}^{[2]})^{\mathrm{T}}}{\partial \boldsymbol{Z}^{[2]}} \cdot n^{-1} \cdot \boldsymbol{D}^{-1} \cdot (\boldsymbol{A}^{[2]} - \boldsymbol{Y})$$

2. 注意到

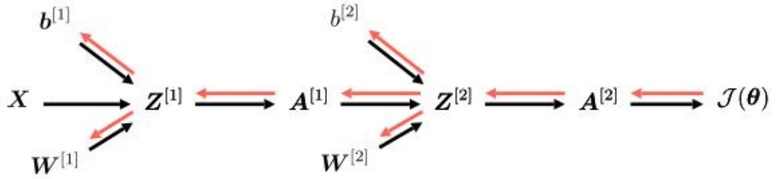
$$\frac{\partial (\boldsymbol{A}^{[2]})^{\mathrm{T}}}{\partial \boldsymbol{Z}^{[2]}} = \boldsymbol{D}$$

- 由于 $a_i^{[2]}$ 只是关于 $z_i^{[2]}$ 的函数, $\partial (\boldsymbol{A}^{[2]})^{\mathrm{T}} / \partial \boldsymbol{Z}^{[2]}$ 的结果为对角阵
- 由于 $a_i^{[2]} = \sigma(z_i^{[2]})$, 故 $\partial a_i^{[2]} / \partial z_i^{[2]} = a_i^{[2]}(1 - a_i^{[2]})$，其正是 $\boldsymbol{D}$ 的第 $i$ 个对角元素

3. 因此，我们有

$$\mathrm{d}\boldsymbol{Z}^{[2]} = n^{-1}(\boldsymbol{A}^{[2]} - \boldsymbol{Y})$$

# 推导细节

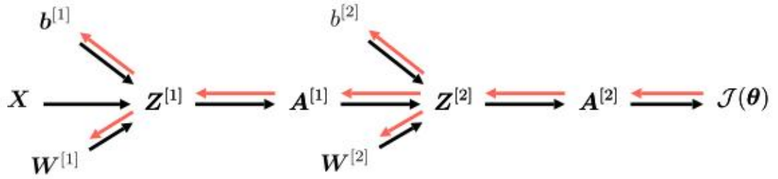



1. 为了得到 $\mathrm{d}b^{[2]} = \partial\mathcal{J}/\partial b^{[2]}$，我们利用链式法则，并将 $\mathcal{J}$ 看成 $\boldsymbol{Z}^{[2]}$ 的函数，将 $\boldsymbol{Z}^{[2]}$ 看成 $b^{[2]}$ 的函数
2. 因此，我们有

$$\mathrm{d}b^{[2]} = \frac{\partial\mathcal{J}}{\partial b^{[2]}} = \frac{\partial(\boldsymbol{Z}^{[2]})^{\mathrm{T}}}{\partial b^{[2]}} \cdot \frac{\partial\mathcal{J}}{\partial \boldsymbol{Z}^{[2]}} = \mathbf{1}^{\mathrm{T}} \cdot \mathrm{d}\boldsymbol{Z}^{[2]} = (\mathrm{d}\boldsymbol{Z}^{[2]})^{\mathrm{T}} \cdot \mathbf{1}$$

- $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}}$ 是一个元素均为 1 的长度为 $n$ 的列向量
- $\boldsymbol{Z}^{[2]}$ 每个元素都是 $b^{[2]}$ 的函数：$z_i^{[2]} = b^{[2]} + \boldsymbol{W}^{[2]} \cdot \boldsymbol{a}_i^{[1]}$
- 因此，$\partial z_i^{[2]}/\partial b^{[2]} = 1$ 以及 $\partial\boldsymbol{Z}^{[2]}/\partial b^{[2]} = (\partial z_1^{[2]}/\partial b^{[2]}, \ldots, \partial z_n^{[2]}/\partial b^{[2]})^{\mathrm{T}} = \mathbf{1}$

# 推导细节

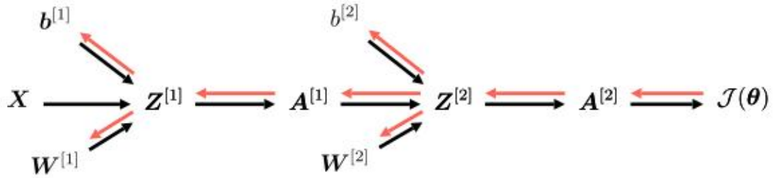



1. 为了得到 $\mathrm{d}\boldsymbol{W}^{[2]} = \partial\mathcal{J}/\partial\boldsymbol{W}^{[2]}$，我们利用链式法则，并将 $\mathcal{J}$ 看成 $\boldsymbol{Z}^{[2]}$ 的函数，将 $\boldsymbol{Z}^{[2]}$ 看成 $\boldsymbol{W}^{[2]}$ 的函数
2. 由于 $\boldsymbol{W}^{[2]}$ 是一个行向量，而不是列向量，我们有

$$\mathrm{d}\boldsymbol{W}^{[2]} = \sum_{i=1}^{n} \frac{\partial z_i^{[2]}}{\partial \boldsymbol{W}^{[2]}} \cdot \frac{\partial \mathcal{J}}{\partial z_i^{[2]}} = \left( \frac{\partial \mathcal{J}}{\partial z_1^{[2]}}, \cdots, \frac{\partial \mathcal{J}}{\partial z_n^{[2]}} \right) \cdot \begin{pmatrix} (\boldsymbol{a}_1^{[1]})^{\mathrm{T}} \\ \vdots \\ (\boldsymbol{a}_n^{[1]})^{\mathrm{T}} \end{pmatrix} = (\mathrm{d}\boldsymbol{Z}^{[2]})^{\mathrm{T}} \cdot \boldsymbol{A}^{[1]}$$

- $\boldsymbol{Z}^{[2]}$ 每个元素都是 $\boldsymbol{W}^{[2]}$ 的函数：$z_i^{[2]} = b^{[2]} + \boldsymbol{W}^{[2]} \cdot \boldsymbol{a}_i^{[1]}$
- 因此，$\partial z_i^{[2]}/\partial \boldsymbol{W}^{[2]} = (\boldsymbol{a}_i^{[1]})^{\mathrm{T}}$
- 利用向量化，我们得到最后一个等式

# 推导细节

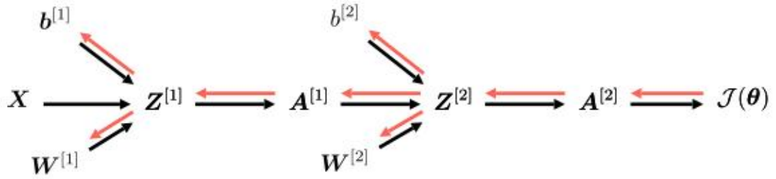



1. 由于 $(\boldsymbol{W}^{[2]})^{\mathrm{T}}$ 是列向量，得到 $\mathrm{d}\boldsymbol{W}^{[2]}$ 的第二种方法为

$$\mathrm{d}\boldsymbol{W}^{[2]} = \left(\frac{\partial \mathcal{J}}{\partial (\boldsymbol{W}^{[2]})^{\mathrm{T}}}\right)^{\mathrm{T}} = \left(\frac{\partial (\boldsymbol{Z}^{[2]})^{\mathrm{T}}}{\partial (\boldsymbol{W}^{[2]})^{\mathrm{T}}} \cdot \frac{\partial \mathcal{J}}{\partial \boldsymbol{Z}^{[2]}}\right)^{\mathrm{T}} = (\mathrm{d}\boldsymbol{Z}^{[2]})^{\mathrm{T}} \cdot \boldsymbol{A}^{[1]}$$

- $\boldsymbol{Z}^{[2]}$ 每个元素都是 $\boldsymbol{W}^{[2]}$ 的函数：$z_i^{[2]} = b^{[2]} + \boldsymbol{W}^{[2]} \cdot \boldsymbol{a}_i^{[1]}$
- 因此，$\partial z_i^{[2]} / \partial (\boldsymbol{W}^{[2]})^{\mathrm{T}} = \boldsymbol{a}_i^{[1]}$ 以及 $\partial (\boldsymbol{Z}^{[2]})^{\mathrm{T}} / \partial (\boldsymbol{W}^{[2]})^{\mathrm{T}} = (\boldsymbol{a}_1^{[1]}, \ldots, \boldsymbol{a}_n^{[1]}) = (\boldsymbol{A}^{[1]})^{\mathrm{T}}$

# 推导细节



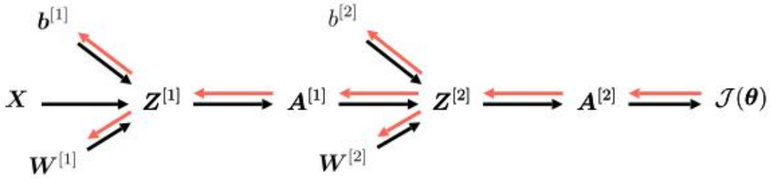

1. 为了得到 $\mathrm{d}\boldsymbol{A}^{[1]} = \partial \mathcal{J} / \partial \boldsymbol{A}^{[1]}$，我们利用链式法则，并将 $\mathcal{J}$ 看成 $\boldsymbol{Z}^{[2]}$ 的函数，将 $\boldsymbol{Z}^{[2]}$ 看成 $\boldsymbol{A}^{[1]}$ 的函数

2. 由于 $\boldsymbol{A}^{[1]} \in \mathbb{R}^{n \times d^{[1]}}$ 是一个矩阵，而不是列向量，我们有

$$\mathrm{d}\boldsymbol{A}^{[1]} = \sum_{i=1}^{n} \frac{\partial z_i^{[2]}}{\partial \boldsymbol{A}^{[1]}} \cdot \frac{\partial \mathcal{J}}{\partial z_i^{[2]}} = \sum_{i=1}^{n} \frac{\partial \mathcal{J}}{\partial z_i^{[2]}} \cdot \boldsymbol{e}_i \cdot \boldsymbol{W}^{[2]} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{Z}^{[2]}} \cdot \boldsymbol{W}^{[2]} = \mathrm{d}\boldsymbol{Z}^{[2]} \cdot \boldsymbol{W}^{[2]}$$

- $\partial z_i^{[2]} / \partial \boldsymbol{A}^{[1]} \in \mathbb{R}^{n \times d^{[1]}}$ 与 $\boldsymbol{A}^{[1]}$ 具有相同维度
- $z_i^{[2]}$ 只是 $\boldsymbol{A}^{[1]}$ 第 $i$ 行的函数：$z_i^{[2]} = b^{[2]} + \boldsymbol{W}^{[2]} \cdot \boldsymbol{a}_i^{[1]}$
- 因此，$\partial z_i^{[2]} / \partial (\boldsymbol{a}_i^{[1]})^{\mathrm{T}} = \boldsymbol{W}^{[2]}$ 以及 $\partial z_i^{[2]} / \partial \boldsymbol{A}^{[1]} = \boldsymbol{e}_i \cdot \boldsymbol{W}^{[2]}$
- $\boldsymbol{e}_i = (0, \ldots, 1, \ldots, 0)^{\mathrm{T}}$ 是一个长度为 $n$ 的列向量；除了第 $i$ 个元素为 1 外，其他元素均为 0

# 推导细节



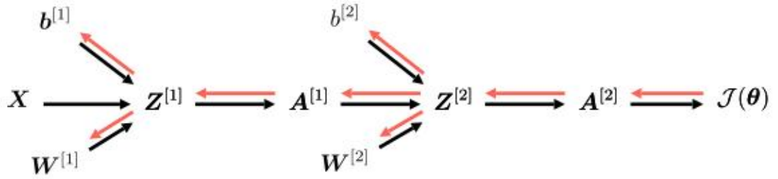

1. 为了得到 $\mathrm{d}\boldsymbol{Z}^{[1]} = \partial J / \partial \boldsymbol{Z}^{[1]}$，我们将 $J$ 看成 $\boldsymbol{A}^{[1]}$ 的函数，将 $\boldsymbol{A}^{[1]}$ 看成 $\boldsymbol{Z}^{[1]}$ 的函数

2. 由于 $\boldsymbol{A}^{[1]} \in \mathbb{R}^{n \times d^{[1]}}$，我们有

$$\mathrm{d}\boldsymbol{Z}^{[1]} = \sum_{i=1}^{n} \sum_{j=1}^{d^{[1]}} \frac{\partial a_{ij}^{[1]}}{\partial \boldsymbol{Z}^{[1]}} \cdot \frac{\partial J}{\partial a_{ij}^{[1]}} = \sum_{i=1}^{n} \sum_{j=1}^{d^{[1]}} \frac{\partial J}{\partial a_{ij}^{[1]}} \cdot \sigma'(z_{ij}^{[1]}) \cdot \boldsymbol{e}_i \cdot \boldsymbol{h}_j^{\mathrm{T}} = \mathrm{d}\boldsymbol{A}^{[1]} \circ \sigma'(\boldsymbol{Z}^{[1]})$$

- $\partial a_{ij}^{[1]} / \partial \boldsymbol{Z}^{[1]} \in \mathbb{R}^{n \times d^{[1]}}$ 与 $\boldsymbol{Z}^{[1]}$ 具有相同维度
- $a_{ij}^{[1]}$ 只是 $\boldsymbol{Z}^{[1]}$ 的第 $(i, j)$ 个元素的函数：$a_{ij}^{[1]} = \sigma(z_{ij}^{[1]})$
- 因此，$\partial a_{ij}^{[1]} / \partial z_{ij}^{[1]} = \sigma'(z_{ij}^{[1]})$ 以及 $\partial a_{ij}^{[1]} / \partial \boldsymbol{Z}^{[1]} = \sigma'(z_{ij}^{[1]}) \cdot \boldsymbol{e}_i \cdot \boldsymbol{h}_j^{\mathrm{T}}$
- $\boldsymbol{e}_i = (0, \ldots, 1, \ldots, 0)^{\mathrm{T}}$ 是一个长度为 $n$ 的列向量；除了第 $i$ 个元素为 1 外，其他元素均为 0
- $\boldsymbol{h}_j = (0, \ldots, 1, \ldots, 0)^{\mathrm{T}}$ 是一个长度为 $d^{[1]}$ 的列向量；除了第 $j$ 个元素为 1 外，其他元素均为 0

# 推导细节



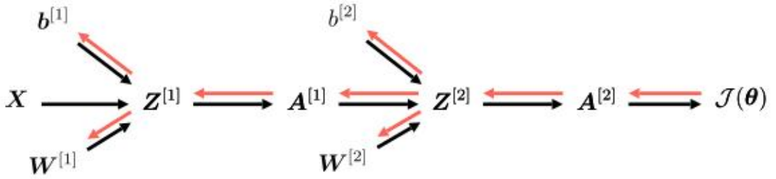

1. 为了得到 $\mathrm{d}\boldsymbol{b}^{[1]} = \partial\mathcal{J}/\partial\boldsymbol{b}^{[1]}$，我们将 $\mathcal{J}$ 看成 $\boldsymbol{Z}^{[1]} = (\boldsymbol{z}_1^{[1]}, \ldots, \boldsymbol{z}_n^{[1]})^{\mathrm{T}}$ 的函数，将 $\boldsymbol{Z}^{[1]}$ 看成 $\boldsymbol{b}^{[1]}$ 的函数

2. 由于 $\boldsymbol{b}^{[1]} \in \mathbb{R}^{d^{[1]}\times 1}$ 是列向量，我们有

$$\mathrm{d}\boldsymbol{b}^{[1]} = \sum_{i=1}^{n} \frac{\partial(\boldsymbol{z}_i^{[1]})^{\mathrm{T}}}{\partial\boldsymbol{b}^{[1]}} \cdot \frac{\partial\mathcal{J}}{\partial\boldsymbol{z}_i^{[1]}} = \sum_{i=1}^{n} \boldsymbol{I} \cdot \frac{\partial\mathcal{J}}{\partial\boldsymbol{z}_i^{[1]}} = \sum_{i=1}^{n} \frac{\partial\mathcal{J}}{\partial\boldsymbol{z}_i^{[1]}} = (\mathrm{d}\boldsymbol{Z}^{[1]})^{\mathrm{T}} \cdot \mathbf{1}$$

- $\boldsymbol{Z}^{[1]}$ 每一行都是 $\boldsymbol{b}^{[1]}$ 的函数：$(\boldsymbol{z}_i^{[1]})^{\mathrm{T}} = (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{x}_i^{\mathrm{T}} \cdot (\boldsymbol{W}^{[1]})^{\mathrm{T}}$
- 因此，$\partial(\boldsymbol{z}_i^{[1]})^{\mathrm{T}}/\partial\boldsymbol{b}^{[1]} = \boldsymbol{I}$
- $\boldsymbol{I}$：维度为 $d^{[1]} \times d^{[1]}$ 的单位阵

# 推导细节



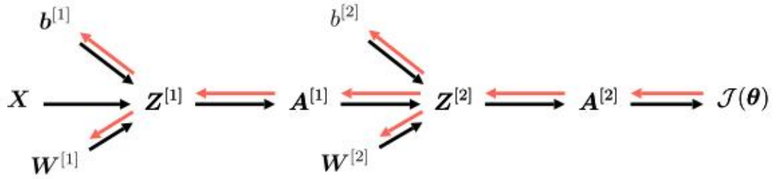

1. 为了得到 $\mathrm{d}\boldsymbol{W}^{[1]} = \partial \mathcal{J} / \partial \boldsymbol{W}^{[1]}$，我们将 $\mathcal{J}$ 看成 $\boldsymbol{Z}^{[1]} = (z_{ij}^{[1]})$ 的函数，将 $\boldsymbol{Z}^{[1]}$ 看成 $\boldsymbol{W}^{[1]}$ 的函数

2. 由于 $\boldsymbol{W}^{[1]} \in \mathbb{R}^{d^{[1]} \times d^{[0]}}$ 是一个矩阵，我们有

$$\mathrm{d}\boldsymbol{W}^{[1]} = \sum_{i=1}^{n} \sum_{j=1}^{d^{[1]}} \frac{\partial z_{ij}^{[1]}}{\partial \boldsymbol{W}^{[1]}} \cdot \frac{\partial \mathcal{J}}{\partial z_{ij}^{[1]}} = \sum_{i=1}^{n} \sum_{j=1}^{d^{[1]}} \frac{\partial \mathcal{J}}{\partial z_{ij}^{[1]}} \cdot \boldsymbol{h}_j \cdot \boldsymbol{x}_i^{\mathrm{T}} = \sum_{i=1}^{n} \frac{\partial \mathcal{J}}{\partial \boldsymbol{z}_i^{[1]}} \cdot \boldsymbol{x}_i^{\mathrm{T}} = (\mathrm{d}\boldsymbol{Z}^{[1]})^{\mathrm{T}} \cdot \boldsymbol{X}$$

- $z_{ij}^{[1]}$ 只是 $\boldsymbol{W}^{[1]} = (\boldsymbol{w}_1^{[1]}, \ldots, \boldsymbol{w}_{d^{[1]}}^{[1]})^{\mathrm{T}}$ 第 $j$ 行的函数：$z_{ij}^{[1]} = b_j^{[1]} + (\boldsymbol{w}_j^{[1]})^{\mathrm{T}} \cdot \boldsymbol{x}_i$
- 因此，$\partial z_{ij}^{[1]} / \partial \boldsymbol{W}^{[1]} = \boldsymbol{h}_j \cdot \boldsymbol{x}_i^{\mathrm{T}}$
- $\boldsymbol{h}_j = (0, \ldots, 1, \ldots, 0)^{\mathrm{T}}$ 是一个长度为 $d^{[1]}$ 的列向量；除了第 $j$ 个元素为 1 外，其他元素均为 0

# 备注




1. 简单起见，记 $\boldsymbol{A}^{[0]} = \boldsymbol{X}$

2. 前向传播和后向传播均可使用 for 循环实现

   - 前向传播：基于当前模型参数，$\boldsymbol{A}^{[l-1]}$ 可被视为计算 $\boldsymbol{Z}^{[l]}$ 和 $\boldsymbol{A}^{[l]}$ 的“输入”
   - 后向传播：$\mathrm{d}\boldsymbol{A}^{[l]}$ 是用来计算 $\mathrm{d}\boldsymbol{Z}^{[l]}$、$\mathrm{d}\boldsymbol{b}^{[l]}$、$\mathrm{d}\boldsymbol{W}^{[l]}$ 以及 $\mathrm{d}\boldsymbol{A}^{[l-1]}$ 的“输入”

# 批量梯度下降法

步骤1. 随机初始化$\boldsymbol{\theta}^{(0)}$

步骤2. 基于$\boldsymbol{\theta}^{(t)}$ 计算

$$\nabla \mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right)=\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(t)}\right)$$

步骤3. 更新参数

$$\boldsymbol{\theta}^{(t+1)}=\boldsymbol{\theta}^{(t)}-\alpha \cdot \nabla \mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right)$$

步骤4. 回到步骤 2 直至收敛，或达到最大迭代次数

# 课程总结

1．将单隐层神经网络由单样本计算推广到多样本的向量化训练

- 数据：将各训练样本的特征向量按行堆叠，构成设计矩阵 $\boldsymbol{X}$
- 前向传播：利用当前模型参数，计算代价函数值，借助广播机制提高计算效率
- 后向传播：基于链式法则和向量化，得到代价函数对参数的平均梯度值
- 梯度下降法