人工智能与深度学习

2-1 只有一个隐藏层的神经网络 I

（一个训练样本）

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. 回顾逻辑回归模型

2. 前向传播

3. 后向传播

4. 批量梯度下降法

背景介绍

1. 只用一个训练样本$(\boldsymbol{x}, y)$，展示基本概念　$(y \in \{0, 1\})$

2. 神经网络模型是特征向量$\boldsymbol{x}$ 的函数，对应的模型参数为$\boldsymbol{\theta}$

3. 利用（批量）梯度下降法，得到模型参数$\boldsymbol{\theta}$ 的估计

4. 核心步骤是

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

- $\boldsymbol{\theta}^{(t)}$：当前模型参数
- 代价函数及其对参数的导数是什么呢？

偏导数的维度

1. 考虑 $\boldsymbol{f}(\boldsymbol{x})$

   - $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^{\mathrm{T}}$：一个 k 维列向量
   - $\boldsymbol{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$：一个 m 维列向量

2. 在本课程中，我们定义

$$\frac{\partial \boldsymbol{f}^{\mathrm{T}}}{\partial \boldsymbol{x}} = \left(\frac{\partial f_1}{\partial \boldsymbol{x}}, \ldots, \frac{\partial f_k}{\partial \boldsymbol{x}} \right) = \begin{pmatrix} \partial f_1(\boldsymbol{x})/\partial x_1 & \ldots & \partial f_k(\boldsymbol{x})/\partial x_1 \\ \partial f_1(\boldsymbol{x})/\partial x_2 & \ldots & \partial f_k(\boldsymbol{x})/\partial x_2 \\ \vdots & \ddots & \vdots \\ \partial f_1(\boldsymbol{x})/\partial x_m & \ldots & \partial f_k(\boldsymbol{x})/\partial x_m \end{pmatrix} \in \mathbb{R}^{m \times k}$$

   - 我们将在求偏导的分子或者分母中，使用转置符号，以清晰表达结果的维度
   - 使梯度下降法中的参数更新过程是严谨的（$k = 1$）：$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha \cdot \partial \mathcal{J} / \partial \boldsymbol{\theta} \left(\boldsymbol{\theta}^{(t)} \right)$

偏导数的维度

1．在一些高等数学课程中，偏导数的定义，可以用来简化对函数的线性近似（$\boldsymbol{\epsilon} \approx 0$）

$$\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{\epsilon}) = \begin{pmatrix} f_1(\boldsymbol{x}+\boldsymbol{\epsilon}) \\ f_2(\boldsymbol{x}+\boldsymbol{\epsilon}) \\ \vdots \\ f_k(\boldsymbol{x}+\boldsymbol{\epsilon}) \end{pmatrix} \approx \begin{pmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_k(\boldsymbol{x}) \end{pmatrix} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \cdot \boldsymbol{\epsilon}$$

- $\partial \boldsymbol{f}/\partial \boldsymbol{x}$：上一页我们定义梯度的转置

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{pmatrix} \partial f_1(\boldsymbol{x})/\partial x_1 & \dots & \partial f_1(\boldsymbol{x})/\partial x_m \\ \partial f_2(\boldsymbol{x})/\partial x_1 & \dots & \partial f_2(\boldsymbol{x})/\partial x_m \\ \vdots & \ddots & \vdots \\ \partial f_k(\boldsymbol{x})/\partial x_1 & \dots & \partial f_k(\boldsymbol{x})/\partial x_m \end{pmatrix} \in \mathbb{R}^{k \times m}$$

偏导数的维度

1．为了本课程讨论的方便，我们不采用某些高等数学课程中偏导数的形式

例子

1. 例 1：考虑 $f(\boldsymbol{x})$

   - $\boldsymbol{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$：列向量
   - $f(\boldsymbol{x})$：一维函数

2. 那么，我们有

$$\frac{\partial f}{\partial \boldsymbol{x}} = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_m} \right)^{\mathrm{T}}$$

3. $\partial f / \partial \boldsymbol{x}$ 与 $\boldsymbol{x}$ 具有相同维度

例子

1. 例 2：考虑 $f(\boldsymbol{x})$

   - $\boldsymbol{x} = (x_1, \ldots, x_m)$: 行向量
   - $f(\boldsymbol{x})$: 一维函数

2. 那么，我们有

$$\frac{\partial f}{\partial \boldsymbol{x}} = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_m} \right)$$

3. $\partial f / \partial \boldsymbol{x}$ 与 $\boldsymbol{x}$ 具有相同维度

例子

1. 考虑 $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{y} = \sum_{j=1}^{m} x_j y_j$

   - $\boldsymbol{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$
   - $\boldsymbol{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$

2. 那么，我们有

$$\frac{\partial f}{\partial \boldsymbol{x}} = \boldsymbol{y}, \qquad \frac{\partial f}{\partial \boldsymbol{y}} = \boldsymbol{x},$$

$$\frac{\partial f}{\partial \boldsymbol{x}^{\mathrm{T}}} = \boldsymbol{y}^{\mathrm{T}}, \qquad \frac{\partial f}{\partial \boldsymbol{y}^{\mathrm{T}}} = \boldsymbol{x}^{\mathrm{T}}$$

例子

1. 例 3：考虑 $f(\boldsymbol{X})$

   - $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}^{m \times k}$：一个$m \times k$ 维矩阵
   - $f(\boldsymbol{X})$：一维函数

2. 那么，我们有

$$\frac{\partial f}{\partial \boldsymbol{X}} = \left(\frac{\partial f}{\partial x_{ij}} \right)$$

3. $\partial f / \partial \boldsymbol{X}$ 与$\boldsymbol{X}$ 具有相同维度

例子

1．例 4（链式法则）：考虑

$$f(\boldsymbol{x}) = g \circ \boldsymbol{h}(\boldsymbol{x})$$

- $\boldsymbol{x}$：一个标量，或者列向量，或者行向量，或者矩阵
- $f(\boldsymbol{x})$: 一维函数
- $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x}))^{\mathrm{T}}$

2．那么，我们有

$$\frac{\partial f}{\partial \boldsymbol{x}} = \sum_{j=1}^{m} \frac{\partial h_j}{\partial \boldsymbol{x}} \cdot \frac{\partial g}{\partial h_j}$$

例子

1. 如果 $\boldsymbol{x}$ 是标量或者列向量，我们还有

$$\frac{\partial f}{\partial \boldsymbol{x}} = \sum_{j=1}^{m} \frac{\partial h_j}{\partial \boldsymbol{x}} \cdot \frac{\partial g}{\partial h_j} = \left(\frac{\partial h_1}{\partial \boldsymbol{x}}, \ldots, \frac{\partial h_m}{\partial \boldsymbol{x}} \right) \cdot \begin{pmatrix} \frac{\partial g}{\partial h_1} \\ \vdots \\ \frac{\partial g}{\partial h_m} \end{pmatrix} = \frac{\partial \boldsymbol{h}^{\mathrm{T}}}{\partial \boldsymbol{x}} \cdot \frac{\partial g}{\partial \boldsymbol{h}}$$

   - $\partial g / \partial \boldsymbol{h} = (\partial g / \partial h_1, \ldots, \partial g / \partial h_m)^{\mathrm{T}}$
   - $\partial \boldsymbol{h}^{\mathrm{T}} / \partial \boldsymbol{x} = (\partial h_1 / \partial \boldsymbol{x}, \ldots, \partial h_m / \partial \boldsymbol{x})$

2. 链式法则求导，可被向量化
3. 然而，核心步骤是得到 $\partial h_j / \partial \boldsymbol{x}$
4. 当 $\boldsymbol{x}$ 为矩阵时，向量化失效
5. 问题：当 $\boldsymbol{x}$ 是一个行向量时，我们可否进行向量化；如果可以，那结果又是什么呢？

目标

1．为了得到偏导数$\partial \mathcal{J}(\boldsymbol{\theta}^{(t)})/\partial \boldsymbol{\theta}$，我们需要引入

- 前向传播：基于当前模型参数，计算“激活值”以及代价函数值
- 后向传播：基于前向传播的计算结果，得到偏导数

2．我们用一个简单的例子，介绍前向和后向传播的计算过程

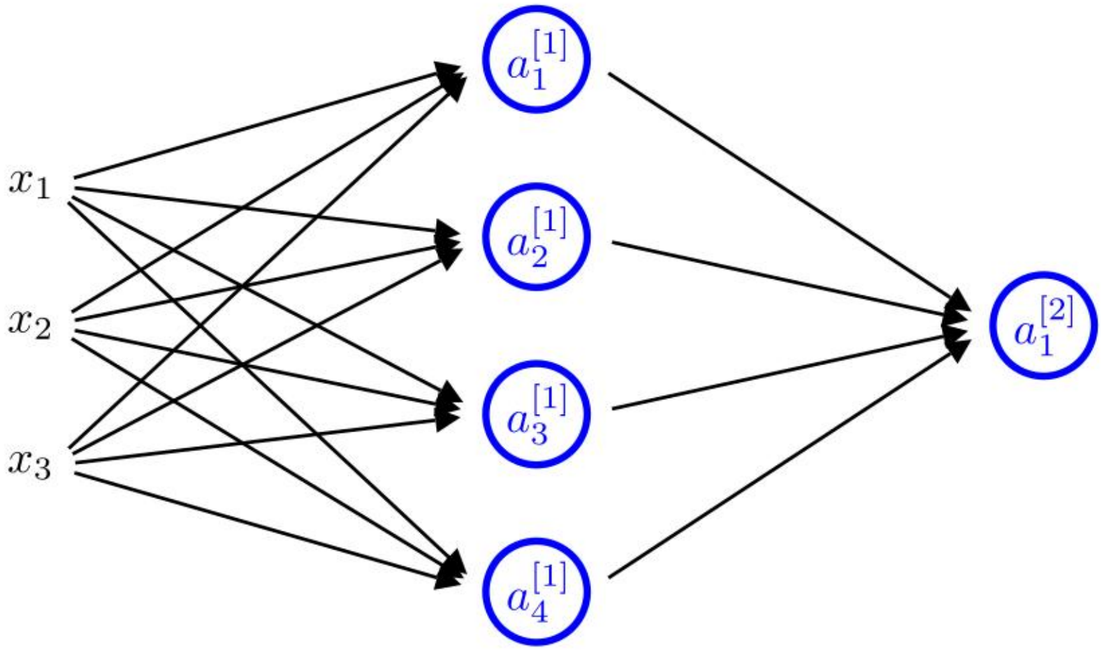
回顾逻辑回归模型

1．在处理实际问题时，这个模型过于简单

2．为什么不加入更多的“圆圈”呢?

神经网络的例子

1. 假设$\boldsymbol{x} = (x_1, x_2, x_3)^{\mathrm{T}}$



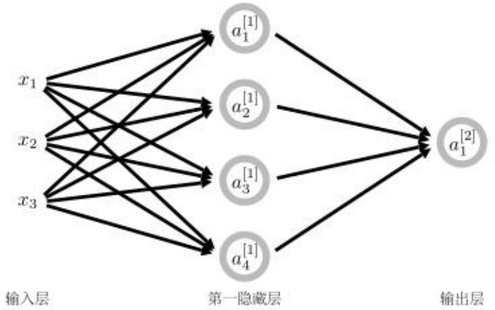
2. 每个圆圈包含如下两个运算

- 线性变换并对应两个模型参数（偏置以及权重）
- 激活变换 （非线性变换），其不对应模型参数

神经网络的例子

1．备注：

- 不同的“圆圈”对应不同模型参数，用来提取数据中的不同信息
- 我们在隐藏层中，“构建”了一组“新的”特征向量
- 对于输出层而言，隐藏层可被认为是它的特征向量
- 相比于逻辑回归模型，神经网络多了一个隐藏层，用来提取更多信息



计算过程　　模型参数

$$z_1^{[1]} = b_1^{[1]} + \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{w}_1^{[1]}, \quad a_1^{[1]} = \sigma(z_1^{[1]}) \qquad b_1^{[1]} \quad \boldsymbol{w}_1^{[1]}$$

$$z_2^{[1]} = b_2^{[1]} + \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{w}_2^{[1]}, \quad a_2^{[1]} = \sigma(z_2^{[1]}) \qquad b_2^{[1]} \quad \boldsymbol{w}_2^{[1]}$$

$$z_3^{[1]} = b_3^{[1]} + \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{w}_3^{[1]}, \quad a_3^{[1]} = \sigma(z_3^{[1]}) \qquad b_3^{[1]} \quad \boldsymbol{w}_3^{[1]}$$

$$z_4^{[1]} = b_4^{[1]} + \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{w}_4^{[1]}, \quad a_4^{[1]} = \sigma(z_4^{[1]}) \qquad b_4^{[1]} \quad \boldsymbol{w}_4^{[1]}$$

$$z_1^{[2]} = b_1^{[2]} + (\boldsymbol{a}^{[1]})^{\mathrm{T}} \cdot \boldsymbol{w}_1^{[2]}, \quad a_1^{[2]} = \sigma(z_1^{[2]}) \qquad b_1^{[2]} \quad \boldsymbol{w}_1^{[2]}$$

向量化

1. 记

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

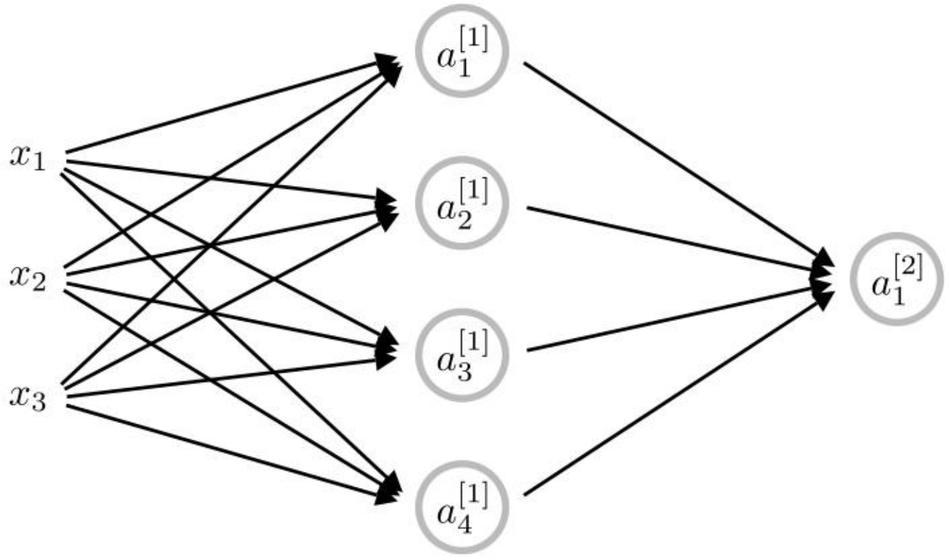
2. 模型参数

- $\{(\boldsymbol{b}^{[l]}, \boldsymbol{W}^{[l]}) : l = 1, \ldots, L\}$

向量化

1．对于之前考虑的例子，我们有

- $d^{[0]}=3, d^{[1]}=4, d^{[2]}=1$
- $\boldsymbol{W}^{[1]}=(\boldsymbol{w}_1^{[1]}, \boldsymbol{w}_2^{[1]}, \boldsymbol{w}_3^{[1]}, \boldsymbol{w}_4^{[1]})^{\mathrm{T}} \in \mathbb{R}^{4\times 3}, \boldsymbol{W}^{[2]}=(\boldsymbol{w}_1^{[2]})^{\mathrm{T}} \in \mathbb{R}^{1\times 4}$
- $\boldsymbol{b}^{[1]}=(b_1^{[1]}, b_2^{[1]}, b_3^{[1]}, b_4^{[1]})^{\mathrm{T}} \in \mathbb{R}^{4\times 1}, b^{[2]} \in \mathbb{R}$



前向传播

1．前向传播利用当前模型参数，计算“激活值”以及代价函数值

2．假设当前模型参数为：$\boldsymbol{b}^{[1]}, \boldsymbol{W}^{[1]}, b^{[2]}, \boldsymbol{W}^{[2]}$

3．（简化的）计算过程为

$$
\begin{aligned}
\boldsymbol{z}^{[1]} &= \boldsymbol{b}^{[1]} + \boldsymbol{W}^{[1]} \cdot \boldsymbol{x}, \\
\boldsymbol{a}^{[1]} &= \sigma(\boldsymbol{z}^{[1]}), \\
z^{[2]} &= b^{[2]} + \boldsymbol{W}^{[2]} \cdot \boldsymbol{a}^{[1]}, \\
a^{[2]} &= \sigma(z^{[2]})
\end{aligned}
$$

$$
\boldsymbol{z}^{[1]} = \begin{pmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{pmatrix} + \begin{pmatrix} (\boldsymbol{w}_1^{[1]})^{\mathrm{T}} \\ (\boldsymbol{w}_2^{[1]})^{\mathrm{T}} \\ (\boldsymbol{w}_3^{[1]})^{\mathrm{T}} \\ (\boldsymbol{w}_4^{[1]})^{\mathrm{T}} \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} b_1^{[1]} + (\boldsymbol{w}_1^{[1]})^{\mathrm{T}} \cdot \boldsymbol{x} \\ b_2^{[1]} + (\boldsymbol{w}_2^{[1]})^{\mathrm{T}} \cdot \boldsymbol{x} \\ b_3^{[1]} + (\boldsymbol{w}_3^{[1]})^{\mathrm{T}} \cdot \boldsymbol{x} \\ b_4^{[1]} + (\boldsymbol{w}_4^{[1]})^{\mathrm{T}} \cdot \boldsymbol{x} \end{pmatrix}
$$

前向传播

1．前向传播利用当前模型参数，计算“激活值”以及代价函数值

2．假设当前模型参数为：$\boldsymbol{b}^{[1]}, \boldsymbol{W}^{[1]}, b^{[2]}, \boldsymbol{W}^{[2]}$

3．（简化的）计算过程为

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

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

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

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

$$\boldsymbol{a}^{[1]} = \begin{pmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \\ \sigma(z_3^{[1]}) \\ \sigma(z_4^{[1]}) \end{pmatrix}$$

前向传播

1. 回顾：我们考虑二分类问题（$y \in \{0, 1\}$）

2. $a^{[2]}$ 是基于当前模型参数，对条件概率$P(y = 1 \mid \boldsymbol{x})$ 的估计

3. 考虑损失函数

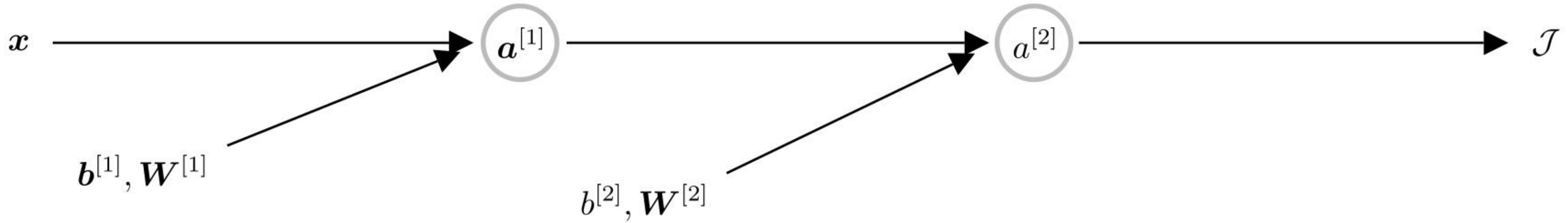
$$\mathcal{J} = \mathcal{L} = -\left\{y \log a^{[2]} + (1 - y) \log \left(1 - a^{[2]}\right)\right\}$$

4. 问题：如果 $y \in \mathbb{R}$，那么 $a^{[2]}$ 和损失函数 $\mathcal{L}$ 的形式是什么？

5. 问题：如果标签 y 的取值只能非负呢？

前向传播

1．将计算过程可视化



前向传播

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

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

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

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

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

备注

1. 当计算第一层中的 $\boldsymbol{a}^{[1]}$ 时，$\boldsymbol{x}$ 为输入量
2. 当计算第二层中的 $a^{[2]}$ 时，$\boldsymbol{a}^{[1]}$ 可被视为输入量
3. 即当计算第 l 层的激活值时，前一层激活结果为对应的输入量
4. 这对于前向传播而言是普遍成立的，因此，前向传播可通过 for 循环编程实现

后向传播

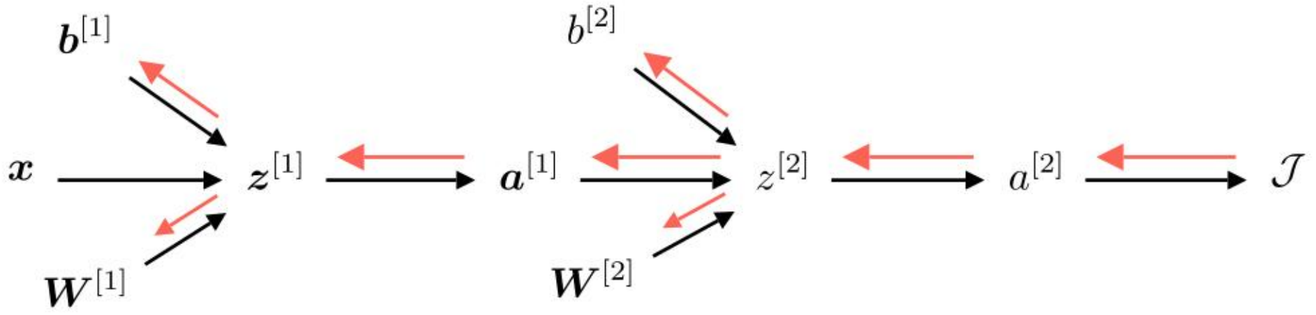
1．基于当前模型参数及前向传播计算结果，后向传播用于计算

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{b}^{[l]}}, \quad \frac{\partial \mathcal{J}}{\partial \boldsymbol{W}^{[l]}} \quad (l = 1, \ldots, L)$$

2．核心技术: 链式法则

- 记$f(\boldsymbol{x}) = g \circ h(\boldsymbol{x})$
- 我们有

$$\frac{\partial f}{\partial \boldsymbol{x}} = \frac{\partial h}{\partial \boldsymbol{x}} \cdot \frac{\partial g}{\partial h}$$

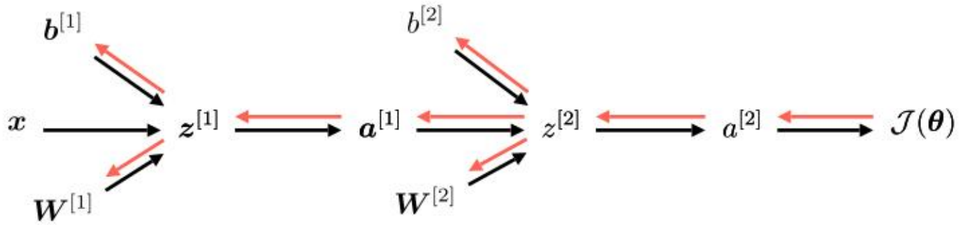


$$\frac{\partial \mathcal{J}}{\partial b^{[2]}} = a^{[2]} - y \qquad \frac{\partial \mathcal{J}}{\partial \boldsymbol{W}^{[2]}} = (a^{[2]} - y) \cdot (\boldsymbol{a}^{[1]})^{\mathrm{T}}$$

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{b}^{[1]}} = (a^{[2]} - y) \cdot \boldsymbol{D} \cdot (\boldsymbol{W}^{[2]})^{\mathrm{T}} \qquad \frac{\partial \mathcal{J}}{\partial \boldsymbol{W}^{[1]}} = (a^{[2]} - y) \cdot \boldsymbol{D} \cdot (\boldsymbol{W}^{[2]})^{\mathrm{T}} \cdot \boldsymbol{x}^{\mathrm{T}}$$

1. $\boldsymbol{D} = \mathrm{diag}(\{a_j^{[1]}(1 - a_j^{[1]}) : j = 1, \ldots, d^{[1]}\})$
2. 我们只需缓存$\boldsymbol{a}^{[1]}$ 和$a^{[2]}$

备注



1. $\partial \mathcal{J} / \partial a^{[2]}$ 的形式，由代价函数 $\mathcal{J}$ 的形式决定

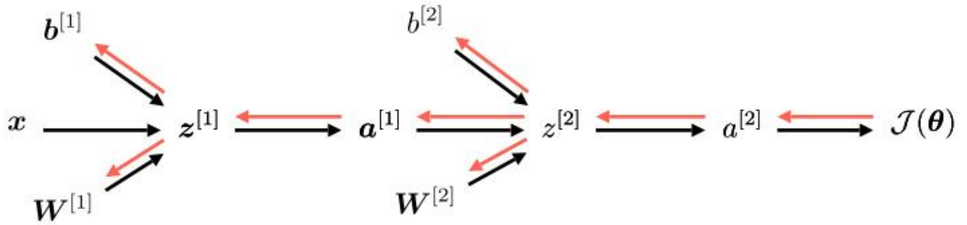
   - 我们已经针对二分类问题，讨论了后向传播的形式
   - 请针对回归问题中的代价函数 $\mathcal{J} = (y - a^{[2]})^2/2$，推导 $\partial \mathcal{J} / \partial a^{[2]}$ 的形式

2. 当我们得到 $\partial \mathcal{J} / \partial a^{[2]}$ 后，我们可根据链式法则得到

$$\frac{\partial \mathcal{J}}{\partial z^{[2]}} = \frac{\partial a^{[2]}}{\partial z^{[2]}} \cdot \frac{\partial \mathcal{J}}{\partial a^{[2]}}$$

   - $\partial a^{[2]} / \partial z^{[2]}$ 的结果，取决于得到 $a^{[2]}$ 的激活函数的形式
   - 对于二分类问题，$\partial \mathcal{J} / \partial z^{[2]} = a^{[2]} - y$

备注



1．当我们得到 $\partial \mathcal{J}/\partial z^{[2]}$，我们将很容易得到

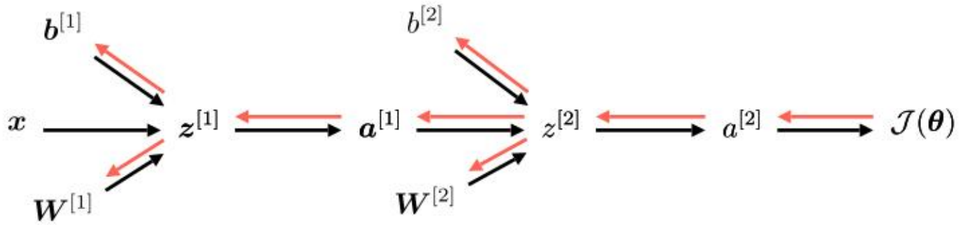
$$\frac{\partial \mathcal{J}}{\partial b^{[2]}} = \frac{\partial z^{[2]}}{\partial b^{[2]}} \cdot \frac{\partial \mathcal{J}}{\partial z^{[2]}} = \frac{\partial \mathcal{J}}{\partial z^{[2]}},$$

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{W}^{[2]}} = \frac{\partial z^{[2]}}{\partial \boldsymbol{W}^{[2]}} \cdot \frac{\partial \mathcal{J}}{\partial z^{[2]}} = (\boldsymbol{a}^{[1]})^{\mathrm{T}} \cdot \frac{\partial \mathcal{J}}{\partial z^{[2]}}$$

2．为了得到代价函数对于第二层模型参数的导数，我们需要得到 $\partial \mathcal{J}/\partial z^{[2]}$

3．相同的结论，适用于第一层模型参数的导数计算

备注



1. 当我们得到 $\partial \mathcal{J}/\partial \boldsymbol{a}^{[1]}$，我们将很容易得到

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{z}^{[1]}} = \frac{\partial (\boldsymbol{a}^{[1]})^{\mathrm{T}}}{\partial \boldsymbol{z}^{[1]}} \cdot \frac{\partial \mathcal{J}}{\partial \boldsymbol{a}^{[1]}} = \boldsymbol{D} \cdot \frac{\partial \mathcal{J}}{\partial \boldsymbol{a}^{[1]}},$$

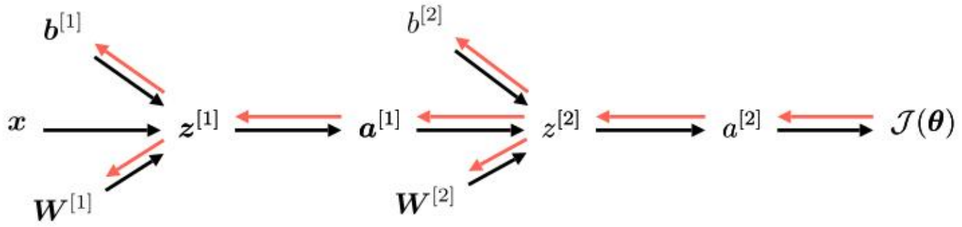
$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{b}^{[1]}} = \frac{\partial (\boldsymbol{z}^{[1]})^{\mathrm{T}}}{\partial \boldsymbol{b}^{[1]}} \cdot \frac{\partial \mathcal{J}}{\partial \boldsymbol{z}^{[1]}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{z}^{[1]}},$$

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{W}^{[1]}} = \sum_{k=1}^{d^{[1]}} \frac{\partial z_k^{[1]}}{\partial \boldsymbol{W}^{[1]}} \cdot \frac{\partial \mathcal{J}}{\partial z_k^{[1]}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{z}^{[1]}} \cdot \boldsymbol{x}^{\mathrm{T}}$$

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

2. 剩下只需讨论，如何从 $\partial \mathcal{J}/\partial z^{[2]}$ 出发，得到 $\partial \mathcal{J}/\partial \boldsymbol{a}^{[1]}$

备注



1．基于计算图，我们只需要讨论一步计算即可；即

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{a}^{[1]}} = \frac{\partial z^{[2]}}{\partial \boldsymbol{a}^{[1]}} \cdot \frac{\partial \mathcal{J}}{\partial z^{[2]}} = (\boldsymbol{W}^{[2]})^{\mathrm{T}} \cdot (a^{[2]} - y)$$

备注

1. 为了更新第 l 层的参数，我们只需要知道 $\partial \mathcal{J} / \partial \boldsymbol{a}^{[l]}$ 即可

2. 为了从 $\partial \mathcal{J} / \partial \boldsymbol{z}^{[l]}$ 得到 $\partial \mathcal{J} / \partial \boldsymbol{a}^{[l-1]}$，我们只需要考虑第 l 层的线性变换即可

3. 因此，后向传播也可通过 for 循环进行编程

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. 更新模型参数

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

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．将逻辑回归模型，拓展到神经网络模型

2．介绍了获得代价函数对模型参数梯度的方法

- 前向传播：基于当前模型参数，计算模型输出以及对应的代价函数值
- 后向传播：基于链式法则，得到代价函数对参数的梯度值

3．梯度下降法