反向传播习题

浙江大学

赵洲

题目

- 1.反向传播算法的主要目标是什么?

A. 初始化神经网络的权重

B. 计算神经网络的输出

C. 计算损失函数相对于每个参数的梯度

D. 选择合适的激活函数

解答

- **A. 初始化神经网络的权重**

- **含义**：指在训练开始之前，为神经网络的每一层、每个节点设定初始的权重（以及偏置），如随机初始化或使用特定分布进行初始化等。
- **是否属于反向传播的目标?**
  - **不是**。权重初始化在训练开始前完成，主要是为了使训练过程更有效地进行，并避免梯度消失或梯度爆炸等问题。
  - **反向传播（Backpropagation）**通常是在权重已被初始化之后进行的，用于根据损失函数对权重进行更新。

解答

- **B. 计算神经网络的输出**

- **含义**：也称前向传播（Forward Propagation），即将输入数据经过各层的加权、偏置和激活函数，最终得到神经网络的输出。
- **是否属于反向传播的目标?**
  - **不是。**
  - **前向传播**和**反向传播**是神经网络训练中相互配合的两个过程：
    - 前向传播：给定输入，计算网络输出和损失。
    - 反向传播：基于损失，对每个参数（权重、偏置）计算梯度，以便更新参数。
  - 前向传播是为了得到输出和损失，反向传播才是为了计算梯度。

解答

- **C. 计算损失函数相对于每个参数的梯度**

- **含义**：在训练中，我们需要知道如果改变某个权重或偏置，损失函数会如何变化，以此来更新网络参数，使损失下降。
- **是否属于反向传播的目标?**
  - **是**，这正是反向传播的核心任务。
  - 在前向传播得到损失之后，反向传播通过链式法则（Chain Rule）逐层向后计算偏导数，最终得到损失函数相对于每个参数（权重、偏置）的梯度。
  - 有了这些梯度，我们就可以采用梯度下降（或其变体，如 Adam、RMSProp 等）来更新网络参数。

解答

- **D. 选择合适的激活函数**

- **含义**：决定在神经网络的各层节点中使用哪种激活函数，如 Sigmoid、ReLU、Tanh、Leaky ReLU 等。
- **是否属于反向传播的目标?**
  - **不是。**激活函数的选择通常在网络结构设计阶段完成，属于模型设计的超参数选择。
  - 虽然激活函数在反向传播中需要计算其导数（如 ReLU 的梯度、Sigmoid 的梯度等），但“选择哪种激活函数”并不是反向传播所要做的事；反向传播只是在已经选定的激活函数基础上，通过链式法则计算其导数。

题目

- 2.在反向传播中，链式法则的作用是什么?

A. 优化损失函数

B. 分层计算各层的梯度

C. 初始化网络参数

D. 选择合适的学习率

解答

- **A. 优化损失函数**
- **含义**：指使用某种优化算法（如梯度下降、Adam 等）来最小化损失函数。
- **与链式法则的关系**：
  - 优化过程需要计算参数的梯度，但“优化”本身指的是整个训练过程或具体的优化算法。
  - **链式法则（Chain Rule）**只是其中的一部分，用于计算梯度，并不是直接去“优化”损失函数。
  - 因此，“优化损失函数”不是链式法则的直接作用，而是利用链式法则计算出的梯度来完成的更高层次目标。

解答

- **B. 分层计算各层的梯度**
- **含义**：在神经网络中，数据和误差都是层与层之间传递的；我们需要逐层计算损失对各个参数的偏导数（梯度）。
- **链式法则的作用**：正确
  - **核心**：在前向传播中，我们从输入开始，一层一层计算输出，最终得到损失。
  - 在反向传播中，就需要根据链式法则，从输出层的梯度开始，逐层向前（向输入方向）传递梯度信息。
  - 由于网络结构往往是由多层非线性组合而成，链式法则可以让我们“分段”或“分层”地计算导数，一层接一层地将梯度传递下去。
  - **因此**，“分层计算各层的梯度”就是链式法则在反向传播中的直接应用。

解答

- **C. 初始化网络参数**
- **含义**：在训练开始之前，对网络的权重、偏置进行一些初始化方法（如随机分布、He 初始化、Xavier 初始化等）。
- **链式法则的作用**：
  - 与初始化几乎无关。初始化是训练前的设置，链式法则是训练（尤其是反向传播）阶段计算梯度时使用的数学工具。
  - 因此，初始化网络参数并不是链式法则的作用。

解答

- **D. 选择合适的学习率**
- **含义**：在使用梯度下降或其变体时，需要设定学习率（或自适应学习率等超参数）。
- **链式法则的作用**：
  - 同样与学习率的选择没有直接关系。学习率是超参数，由训练过程的设计者或自动调参算法来决定。
  - 链式法则只是在已知网络结构和参数的基础上，计算梯度本身的数学工具，和如何选定学习率无关。

题目

- **3.哪种情况最有可能导致反向传播中的梯度消失问题?**

A. 使用ReLU激活函数

B. 使用Sigmoid激活函数

C. 使用大规模的训练数据

D. 使用批量归一化

解答

- **A. 使用 ReLU 激活函数**

- **ReLU (Rectified Linear Unit) 是一种常见且有效的激活函数，其定义为** ReLU(x)=max⁡(0,x)\text{ReLU}(x) = \max(0, x)ReLU(x)=max(0,x)
- **在正区间，** ReLU **的梯度为常数** 1**，不会像** Sigmoid **那样出现非常小的梯度，所以使用** ReLU **通常有助于缓解或减少梯度消失问题。**
- **但需要注意** ReLU **可能带来的另一个问题是“死亡 ReLU”（当输入始终小于** 0 **时，梯度为** 0**，节点不再更新），但这和“梯度消失”并不相同。**

解答

- **B. 使用 Sigmoid 激活函数正确**

- **Sigmoid 函数将输入映射到 (0,1)(0,1)(0,1) 的区间，函数形式是**

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- 在正向传播时，一旦输入过大或过小，输出会饱和到接近 1 或 0；在饱和区间，Sigmoid 的导数非常小（接近 0）。
- 在反向传播时，梯度在每一层会乘上非常小的导数，导致梯度**指数级衰减**，从而产生**梯度消失**问题。
- 因此，在深层网络中，Sigmoid 函数是引发梯度消失最典型的激活函数。

解答

- **C. 使用大规模的训练数据**

- 虽然数据规模会影响训练时间、泛化能力等，但与是否导致“梯度消失”并没有直接联系。
- 即使数据量很大，如果网络结构或激活函数等因素设计合理，也不一定会出现梯度消失问题。
- 大规模数据集更可能带来的是训练所需时间变长或需要更好的优化技巧、硬件加速等。

解答

- **D. 使用批量归一化（Batch Normalization）**
- **Batch Normalization (BN)** 通过对激活/输入数据进行标准化、平移和缩放，能有效稳定训练并加速收敛。
- 实际上，BN 通常能**缓解**梯度消失/梯度爆炸问题，而不是导致它。
- 因此，BN 往往被认为是帮助减轻梯度消失的一种技术。

题目

- 4.下面关于反向传播算法的说法，哪一个是错误的?

A. 反向传播可以应用于卷积神经网络

B. 反向传播只能用于全连接层

C. 反向传播依赖于损失函数的可微性

D. 反向传播通过梯度下降优化参数

解答

- **A. 反向传播可以应用于卷积神经网络**

- **正确性**：这是**正确**的说法。
  - 反向传播是一种通用的神经网络训练方法，不仅适用于全连接层，也可以应用于卷积神经网络、循环神经网络、注意力模型等。
  - 在卷积层中，同样会计算损失函数对滤波器（卷积核）参数的梯度，只不过卷积层的参数共享和局部连接使计算公式有所不同，但本质上仍然是通过链式法则来实现反向传播。

解答

- **B. 反向传播只能用于全连接层**
- **正确性**：这是**错误**的说法。
  - 反向传播**并不只**能用于全连接层，而是适用于各种可微的网络结构，包括卷积层、循环层、注意力机制等。
  - 只要网络计算图是可微的，就能通过反向传播来计算梯度并进行训练。

解答

- **C. 反向传播依赖于损失函数的可微性**

- **正确性**：这是**正确**的说法。
  - 反向传播的本质是运用链式法则来计算损失函数对网络参数的偏导数，所以损失函数（以及各层的运算）必须是**可微**的（或在绝大多数地方可微，比如 ReLU 在 0 处有不可导点，但通常我们取左/右导数来处理）。
  - 如果某些网络层或损失函数不可微，则无法直接通过标准反向传播来求解梯度，需要其他技巧（如次梯度、近似梯度、强化学习中策略梯度等）。

解答

- **D. 反向传播通过梯度下降优化参数**

- **正确性**：这是**大体正确**的说法。
  - 虽然最常见的做法是使用**梯度下降（Gradient Descent）**或其变体（如随机梯度下降 SGD、Adam、RMSProp 等）来更新参数，但是所有这些方法都需要用到反向传播算出的梯度。
  - 严格来说，反向传播是**计算梯度**的方法，随后我们再使用这些梯度配合梯度下降或者其他优化算法去更新参数。
  - 但通常人们也会简略地说“反向传播通过梯度下降来优化参数”，因为这两者常常绑定在一起使用。

题目

- 判断1.
- 反向传播算法不适用于深层神经网络

解答

- **错误**

  - 反向传播算法（Backpropagation）是训练神经网络的一种常用算法，它适用于任何深度的神经网络，包括深层神经网络（DNN）。反向传播通过计算误差并将其通过网络反向传播来更新网络权重。在现代深度学习中，反向传播是训练深层神经网络的核心算法。
  - 事实上，反向传播是深度神经网络训练中不可或缺的一部分，因此它适用于深层神经网络。

题目

- 判断2.
- 在使用反向传播算法时，学习率的选择不会影响训练过程的收敛速度。

解答

- 错误

- 学习率的选择对训练过程的收敛速度和最终效果有显著影响。过大可能导致不稳定，过小则可能导致收敛缓慢。

题目

- 判断3.
- 反向传播算法需要在每次迭代中都重新计算前向传播。

解答

- 正确

- 每次迭代中，反向传播需要依赖最新的前向传播结果，因此需要在每次迭代中先进行前向传播。

题目

题目

给定一个**单层感知器**（无隐藏层），输入维度 $d = 2$，输出维度 1。其参数和前向传播如下：

- **参数**：$\mathbf{w} = (w_1, w_2)$，偏置 b。
- **激活函数**：Sigmoid，定义

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

- **训练样本**：$(\mathbf{x}, y)$，其中
  $\mathbf{x} = (x_1, x_2)$，$y \in \{0, 1\}$。
- **网络输出**：

$$\hat{y} = \sigma(w_1 x_1 + w_2 x_2 + b).$$

- **损失函数**：使用**交叉熵损失**

$$L(\hat{y}, y) = -\Big[y \ln(\hat{y}) + (1 - y) \ln(1 - \hat{y})\Big].$$

请回答：

1. 写出前向传播得到的 $\hat{y}$（即给出其函数形式）。
2. 使用链式法则，分别计算损失函数对参数 w_1, w_2, b 的偏导数：

$$\frac{\partial L}{\partial w_1}, \quad \frac{\partial L}{\partial w_2}, \quad \frac{\partial L}{\partial b}.$$

解答

第 1 步：前向传播

网络的输出为

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{其中 } z = w_1 x_1 + w_2 x_2 + b.$$

这就是前向传播最终得到的预测值 $\hat{y}$。

解答

第 2 步：反向传播所需的中间量

1. **Sigmoid 函数的导数**

$$\sigma'(z) = \sigma(z)\big[1 - \sigma(z)\big].$$

对应到本题，就是

$$\frac{d\hat{y}}{dz} = \hat{y}\,(1 - \hat{y}).$$

2. **交叉熵损失函数对 $\hat{y}$ 的导数**

$$\frac{\partial L}{\partial \hat{y}} = -\Big[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\Big] = \frac{\hat{y} - y}{\hat{y}(1-\hat{y})}.$$

（可以记住一个常用的简化结果：对于二分类交叉熵，
$\frac{\partial L}{\partial \hat{y}} = \hat{y} - y$，
但那是针对输出层直接 $\hat{y} \in (0,1)$ 并使用 Sigmoid 结合的某些特殊实现。更一般的公式如上所示。）

解答

第 3 步：链式法则计算梯度

以 $\frac{\partial L}{\partial w_1}$ 为例，使用链式法则：

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1}.$$

1. $\frac{\partial L}{\partial \hat{y}} = \frac{\hat{y}-y}{\hat{y}(1-\hat{y})}$
2. $\frac{\partial \hat{y}}{\partial z} = \hat{y}(1-\hat{y})$
3. $\frac{\partial z}{\partial w_1} = x_1$（因为 $z = w_1 x_1 + w_2 x_2 + b$）

将它们连乘：

$$\frac{\partial L}{\partial w_1} = \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right) \cdot \left(\hat{y}(1-\hat{y})\right) \cdot x_1 = (\hat{y}-y)\, x_1.$$

同理，可得：

$$\frac{\partial L}{\partial w_2} = (\hat{y}-y)\, x_2.$$

以及对偏置 b 的偏导数：

$$\frac{\partial L}{\partial b} = (\hat{y}-y).$$

解答

因此，最终结果：

$$\frac{\partial L}{\partial w_1} = (\hat{y} - y)\, x_1, \quad \frac{\partial L}{\partial w_2} = (\hat{y} - y)\, x_2, \quad \frac{\partial L}{\partial b} = \hat{y} - y.$$

题目

网络结构

1. **输入层**：2 个输入 x_1, x_2。
2. **隐藏层**：只有 1 个神经元，使用 ReLU 激活函数

$$z = w_1 x_1 + w_2 x_2 + b_1, \quad h = \mathrm{ReLU}(z) = \max(0,\, z).$$

3. **输出层**：只有 1 个神经元，做线性输出

$$\hat{y} = w_3 \cdot h + b_2.$$

4. **损失函数**：对单个样本，使用均方误差(MSE)的形式

$$L = \frac{1}{2}(\hat{y} - y)^2.$$

已知具体数值

- **输入**：$x_1 = 1,\ x_2 = 2$，真实标签 $y = 2$。
- **参数初始值**：

$$w_1 = 1,\ w_2 = 1,\ b_1 = 0, \quad w_3 = 2,\ b_2 = 0.$$

请 **(1)** 做前向传播，算出 $\hat{y}$ 和 L 的数值；

(2) 使用反向传播，分别计算

$$\frac{\partial L}{\partial w_1},\ \frac{\partial L}{\partial w_2},\ \frac{\partial L}{\partial b_1},\ \frac{\partial L}{\partial w_3},\ \frac{\partial L}{\partial b_2}$$

解答

第 1 步：前向传播

1. **隐藏层线性变换**：

$$z = w_1 x_1 + w_2 x_2 + b_1 = 1 \cdot 1 + 1 \cdot 2 + 0 = 1 + 2 = 3.$$

2. **隐藏层激活 (ReLU)**：

$$h = \mathrm{ReLU}(z) = \max(0, 3) = 3.$$

3. **输出层**（线性）：

$$\hat{y} = w_3 \cdot h + b_2 = 2 \cdot 3 + 0 = 6.$$

4. **损失函数 (MSE)**：

$$L = \frac{1}{2}(\hat{y} - y)^2 = \frac{1}{2}(6 - 2)^2 = \frac{1}{2} \cdot 4^2 = \frac{1}{2} \cdot 16 = 8.$$

因此，**前向传播**结果：

$$z = 3, \quad h = 3, \quad \hat{y} = 6, \quad L = 8.$$

解答

第 2 步：反向传播

要计算各个参数的梯度，通常先写出链式法则的思路，并把需要的中间导数列清楚。

2.1. 对输出层参数 w_3, b_2 的梯度

1. **损失关于 $\hat{y}$ 的导数**

$$\frac{\partial L}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}}\left[\tfrac{1}{2}(\hat{y}-y)^2\right] = (\hat{y}-y) = 6-2 = 4.$$

（有时记作 $\delta = \hat{y}-y$）

2. $\hat{y} = w_3 \cdot h + b_2$
   - $\frac{\partial \hat{y}}{\partial w_3} = h = 3$
   - $\frac{\partial \hat{y}}{\partial b_2} = 1$

因此：

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_3} = 4 \cdot 3 = 12.$$

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b_2} = 4 \cdot 1 = 4.$$

解答

2.2. 对隐藏层参数 w_1, w_2, b_1 的梯度

这里需要把损失对 w_1 等参数的偏导，拆解为多步：

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial z} \cdot \frac{\partial z}{\partial w_1}.$$

1. **前两项（关于输出层）**

$$\frac{\partial L}{\partial \hat{y}} = 4, \quad \frac{\partial \hat{y}}{\partial h} = w_3 = 2.$$

2. $\frac{\partial h}{\partial z}$：**ReLU 的导数**

$$h = \mathrm{ReLU}(z) = \max(0, z). \quad \frac{\partial h}{\partial z} = \begin{cases} 1, & z > 0, \\ 0, & z \le 0. \end{cases}$$

由于 $z = 3$（大于0），所以这里 $\frac{\partial h}{\partial z} = 1$。

3. $\frac{\partial z}{\partial w_1}$

$$z = w_1 x_1 + w_2 x_2 + b_1 \quad \Longrightarrow \quad \frac{\partial z}{\partial w_1} = x_1 = 1.$$

解答

综合相乘：

$$\frac{\partial L}{\partial w_1} = 4\,(\frac{\partial L}{\partial \hat{y}}) \times 2\,(\frac{\partial \hat{y}}{\partial h}) \times 1\,(\frac{\partial h}{\partial z}) \times 1\,(\frac{\partial z}{\partial w_1}) = 4 \times 2 \times 1 \times 1 = 8.$$

同理，

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial z} \cdot \frac{\partial z}{\partial w_2}.$$

- 依然 $\frac{\partial L}{\partial \hat{y}} = 4$
- $\frac{\partial \hat{y}}{\partial h} = w_3 = 2$
- $\frac{\partial h}{\partial z} = 1$ （因为 $z > 0$）
- $\frac{\partial z}{\partial w_2} = x_2 = 2$

$$\frac{\partial L}{\partial w_2} = 4 \times 2 \times 1 \times 2 = 16.$$

解答

而对偏置 b_1 的梯度：

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial z} \cdot \frac{\partial z}{\partial b_1}.$$

- 这里 $\frac{\partial z}{\partial b_1} = 1$，因为 $z = w_1 x_1 + w_2 x_2 + b_1$。
- 其它项和之前相同：4、2、1。

$$\frac{\partial L}{\partial b_1} = 4 \times 2 \times 1 \times 1 = 8.$$

解答

第 3 步：数值总结

1. **前向传播结果**

$$z = 3, \quad h = 3, \quad \hat{y} = 6, \quad L = 8.$$

2. **反向传播梯度**

$$\frac{\partial L}{\partial w_1} = 8, \quad \frac{\partial L}{\partial w_2} = 16, \quad \frac{\partial L}{\partial b_1} = 8, \quad \frac{\partial L}{\partial w_3} = 12, \quad \frac{\partial L}{\partial b_2} = 4.$$

题目

设有一个简单的单层神经网络，输出层不带非线性激活函数（等效于线性激活）。
该网络的输出计算方式如下：

$$z = w_1 x_1 + w_2 x_2 + b$$

损失函数 L 为均方误差（MSE），定义如下：

$$L = \frac{1}{2}(z - t)^2$$

其中，t 为目标值（标签），z 为网络输出。

给定以下数值：

- $x_1 = 1$
- $x_2 = 2$
- $w_1 = 2$
- $w_2 = 1$
- $b = 1$
- 目标值 $t = 10$

题目

请计算：

1. 前向传播所得的网络输出 z。
2. 由此产生的损失 L。
3. 该损失对各参数 w_1, w_2, b 的偏导数（即梯度）。

解答

1. 前向传播（Forward Pass）

根据公式

$$z = w_1 x_1 + w_2 x_2 + b$$

将题目给定的数值代入：

$$z = 2 \times 1 + 1 \times 2 + 1 = 2 + 2 + 1 = 5.$$

解答

2. 计算损失（Loss）

损失函数为

$$L = \frac{1}{2}(z - t)^2.$$

将 $z = 5$ 和 $t = 10$ 代入：

$$L = \frac{1}{2}(5 - 10)^2 = \frac{1}{2} \times (-5)^2 = \frac{1}{2} \times 25 = 12.5.$$

解答

3. 反向传播（Backward Pass）

由于这是单层且没有非线性激活，反向传播比较简单。我们先对输出层（即 z）求偏导，再逐层往回乘。

1. **损失对输出 z 的偏导数**

$$\frac{\partial L}{\partial z} = \frac{\partial}{\partial z}\left[\frac{1}{2}(z-t)^2\right] = (z-t).$$

将 $z = 5$ 和 $t = 10$ 代入：

$$\frac{\partial L}{\partial z} = 5 - 10 = -5.$$

解答

2. **损失对参数 w_1 的偏导数**

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial w_1} = (z - t) \cdot x_1.$$

将数值代入：

$$\frac{\partial L}{\partial w_1} = (-5) \times 1 = -5.$$

3. **损失对参数 w_2 的偏导数**

同理，

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial w_2} = (z - t) \cdot x_2 = (-5) \times 2 = -10.$$

4. **损失对偏置 b 的偏导数**

$$\frac{\partial z}{\partial b} = 1 \quad \Rightarrow \quad \frac{\partial L}{\partial b} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial b} = (-5) \times 1 = -5.$$

解答

最终结果

- 网络输出 z：5
- 损失 L：12.5
- 梯度：

$$\frac{\partial L}{\partial w_1} = -5, \quad \frac{\partial L}{\partial w_2} = -10, \quad \frac{\partial L}{\partial b} = -5.$$

End