

提交时间：10月30日下午2:00前，以“姓名-学号”命名并提交到课程平台

1. [5分] 为什么在用反向传播算法进行参数学习时要采用随机参数初始化方式而不是直接令 $W=0$, $b=0$?

打破对称性：如果所有权重初始化为 0，神经网络的每一层神经元在正向传播时会计算出相同的输出，导致梯度更新时所有神经元学习相同的特征，无法形成多样化的表示。随机初始化可以打破这种对称性，使不同神经元学习不同的特征

避免梯度消失：当 $W=0$ 时，梯度更新可能为 0 或非常小（尤其在使用某些激活函数如 sigmoid 时），导致学习过程停滞或非常缓慢。随机初始化确保初始梯度有一定大小，有助于模型更快收敛

2. [15分] 如何解决神经网络的过拟合问题？请给出至少 3 种解决方案。

Dropout：在训练过程中，随机地将神经网络中的部分神经元丢弃，从而减少神经元之间的共适应性，迫使网络学习更鲁棒的特征

早停：监控验证集上的损失，当损失不再下降时提前停止训练，避免模型继续过拟合训练数据。

正则化：通过在损失函数中添加正则化项（如 L1 或 L2 正则化），对模型参数（如权重）施加惩罚，限制模型复杂度，从而防止模型过度拟合训练数据。

数据增强：通过对训练数据进行变换（如旋转、翻转、裁剪、添加噪声等），人为增加数据的多样性，使模型接触更多多样化的样本，从而提高泛化能力。

3. [10 分] 假定输入是一张 100×100 的 RGB 图像, 我们使用含有 100 个神经元的全连接层进行特征提取。请问, 该层含有多少个可学习的参数 (包括偏置) ?

输入维度: $100 \times 100 \times 3 = 30000$

权重参数: $30000 \times 100 = 3000000$

偏置个数: 即神经元个数, 100

总参数为: **3000100**

4. [10 分] 假定输入是一张 100×100 的 RGB 图像, 我们使用 100 个大小为 3×3 的卷积核进行特征提取。请问, 该卷积层含有多少个可学习的参数 (包括偏置) ?

每个卷积核: $3 \times 3 \times 3 = 27$

权重参数: $27 \times 100 = 2700$

偏置个数: 100

总参数为: **2800**

5. [10 分] 假定输入特征图维度是 $63 \times 63 \times 16$, 我们使用 32 个大小为 5×5 的卷积核进行特征提取, 步长为 2, 填充为 1。请问, 输出特征图的维度是多少?

输出通道数 = 卷积核数 = 32

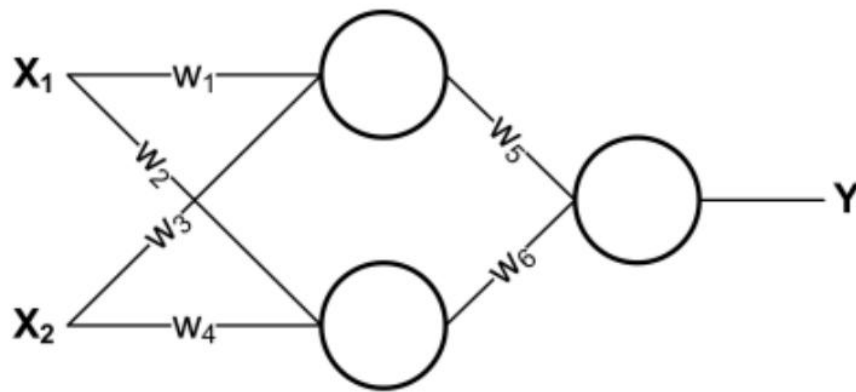
高度 = 宽度 = $(63 - 5 + 12) / 2 + 1 = 31$

输出维度为 **313132**

6. [10 分] 假定输入特征图维度是 $63 \times 63 \times 16$ ，我们使用 32 个大小为 5×5 的卷积核进行特征提取，步长为 1。请问，如果想要实现等宽卷积，填充应设置为多少？

代入公式 $63 = (63 - 5 + 2p) / 2 + 1$ ，解得填充 $P=32$

7. [40 分] 给定如下前馈神经网络， X_1, X_2 为网络输入， Y 为网络输出， w_1, w_2, w_3, w_4, w_5 ，以及 w_6 为神经元权重，所有神经元无偏置参数，无激活函数。



(1) 设计一新的无隐藏层的神经网络架构，实现和上述神经网络同样功能，请给出神经元权重。

无隐藏层的神经网络直接从输入 X_1, X_2 到输出 Y 其输出为：

$$Y = w_1 X_1 + w_2 X_2$$

原网络的输出 Y 是线性组合

第一个隐藏神经元输出： $h_1 = w_1 X_1 + w_3 X_2$

第二个隐藏神经元输出： $h_2 = w_2 X_1 + w_4 X_2$

输出 $Y = w_5 h_1 + w_6 h_2$

展开：

$$Y = w_5 (w_1 X_1 + w_3 X_2) + w_6 (w_2 X_1 + w_4 X_2)$$

$$Y = (w_5 w_1 + w_6 w_2) X_1 + (w_5 w_3 + w_6 w_4) X_2$$

所以 $W_1 = w_5 w_1 + w_6 w_2$

$$W_2 = w_5 w_3 + w_6 w_4$$

(2) 若上述神经网络的隐藏层采用 sigmoid 激活函数（即 Logistic 激活函数），输出层采用激活函数 $t(x)$, $t(x) = 1$ if $x > 0$, $t(x) = 0$, otherwise. 请给出 w_1, w_2, w_3, w_4, w_5 , 以及 w_6 的一种具体取值, 使得该神经网络实现 XOR 计算。注意: X_1, X_2 为二值输入, 所有神经元无偏置参数。

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

在无偏置条件下推导权重的一种方法是将问题转化为基于四个输入情形的不等式系统:

$$(0,0): w_5 \cdot 0.5 + w_6 \cdot 0.5 \leq 0 \rightarrow w_5 + w_6 \leq 0$$

$$(0,1): w_5 \cdot \text{sigmoid}(w_3) + w_6 \cdot \text{sigmoid}(w_4) > 0$$

$$(1,0): w_5 \cdot \text{sigmoid}(w_1) + w_6 \cdot \text{sigmoid}(w_2) > 0$$

$$(1,1): w_5 \cdot \text{sigmoid}(w_1 + w_3) + w_6 \cdot \text{sigmoid}(w_2 + w_4) \leq 0$$

由于 Sigmoid 函数的非线性特性, 该系统难以解析求解, 因此通常采用数值方法 (如梯度下降) 最小化一个损失函数, 该函数对违反上述条件的情形进行

惩罚（例如使用 hinge loss：对目标为 1 的样本，损失为 $\max(0, 1 - z)$ ；对目标为 0 的样本，损失为 $\max(0, 1 + z)$ ）。这一优化过程可在 PyTorch 等框架中实现：随机初始化权重，使用 Adam 等优化器训练数千轮，直至损失趋近于零。

通过该过程，可得到一组有效权重： **$w_1=3, w_2=1, w_3=3, w_4=1, w_5=4, w_6=-5$**

通过程序进行验证

```
26 # 配置权重，初始化w1到w6的值
27 w1 = 3 # 第一个隐藏层神经元的w1
28 w2 = 1 # 第一个隐藏层神经元的w2
29 w3 = 3 # 第二个隐藏层神经元的w3
30 w4 = 1 # 第二个隐藏层神经元的w4
31 w5 = 4 # 输出层的w5
32 w6 = -5 # 输出层的w6
33
34 # 测试四种输入情况 (X1, X2) 和对应的输出
35 inputs = [(0, 0), (0, 1), (1, 0), (1, 1)]
36 for X1, X2 in inputs:
37     output = xor_network(X1, X2, w1, w2, w3, w4, w5, w6)
38     print(f"输入 (X1={X1}, X2={X2}) 的输出: {output}")
39
```

题 输出 调试控制台 终端 端口

Microsoft Windows [版本 10.0.22631.5624]
© Microsoft Corporation。保留所有权利。

C:\Users\muke\Documents\A-作业 论文 复习\大三归档\IIIA 计算机视觉导论\实验\作业>python main.py

输入 (X1=0, X2=0) 的输出: 0
输入 (X1=0, X2=1) 的输出: 1
输入 (X1=1, X2=0) 的输出: 1
输入 (X1=1, X2=1) 的输出: 0

如表所示， z 仅在 XOR 应输出 1 的情况下为正，完全符合要求。

(X1, X2)	h1	h1 (sigmoid)	h2	h2 (sigmoid)	z	Y = t(z)	Expected XOR
(0, 0)	0	0.5000	0	0.5000	-0.5000	0	0
(0, 1)	3	0.9526	1	0.7311	0.1540	1	1
(1, 0)	3	0.9526	1	0.7311	0.1540	1	1
(1, 1)	6	0.9975	2	0.8808	-0.4140	0	0