

# 《深度学习平台与应用》作业二

## 20251123

### 一、选择题

1. 在 GoogLeNet 的 Inception 模块中，使用  $1\times 1$  卷积作为「瓶颈层」的主要目的是什么？(B)
  - A.  $1\times 1$  卷积的主要作用是引入非线性，它并不会改变通道数，也几乎不影响计算量。
  - B. 在  $3\times 3$ 、 $5\times 5$  卷积之前加  $1\times 1$  卷积，可以显著减少通道数，从而降低后续  $3\times 3$ 、 $5\times 5$  卷积的计算量和参数量，并控制 Inception 模块输出通道数的增长。
  - C. 使用  $1\times 1$  卷积是为了彻底去掉池化操作，让整个网络只包含卷积层，从而更容易在 GPU 上并行。
  - D.  $1\times 1$  卷积只在最后的分类器部分使用，用来代替全连接层，不参与 Inception 模块内部的计算。
2. 关于卷积网络中的 Batch Normalization (BN)，以下哪一项描述是正确的？(C)
  - A. 训练和测试阶段都使用当前 mini-batch 的均值和方差，这样能够起到更强的正则化作用。
  - B. 训练阶段 BN 使用从整个训练集预先精确计算好的全局均值与方差，测试阶段则不做归一化。
  - C. 训练阶段 BN 使用当前 mini-batch 的均值和方差，并学习缩放参数  $\gamma$ 、偏移参数  $\beta$ ；测试阶段使用在训练过程中累计的均值和方差的移动平均来做归一化。
  - D. BN 只改变网络的前向传播公式，不会对梯度传播产生影响，所以不会缓解梯度消失/爆炸问题。
3. 下列哪种说法正确地比较了「一个  $7\times 7$  卷积」与「连续堆叠三层  $3\times 3$  卷积」的参数量与表达能力（忽略偏置）？(B)
  - A. 两者参数量完全相同，但三层  $3\times 3$  卷积更浅，非线性层更少。
  - B. 相比一个  $7\times 7$  卷积，三层  $3\times 3$  卷积具有相同的感受野，但参数

- 量更少，并且在中间多了两次非线性变换。
- C. 三层  $3 \times 3$  卷积的参数量比单个  $7 \times 7$  卷积多，因此只在小模型里使用。
- D. 单个  $7 \times 7$  卷积和三层  $3 \times 3$  卷积在感受野、参数量和非线性数量上都没有本质差别。
4. 在反向传播算法中，我们需要计算激活函数的导数。对于 Sigmoid 激活函数  $\sigma(x) = \frac{1}{1+e^{-x}}$ ，其导数  $\frac{d\sigma(x)}{dx}$  可以用  $\sigma(x)$  自身表示为：(B)
- A.  $1 - \sigma(x)$
- B.  $\sigma(x)(1 - \sigma(x))$
- C.  $\sigma(x)^2$
- D.  $e^{-x}$
5. 在 VGGNet 的设计中，主要特点是使用了多个堆叠的  $3 \times 3$  卷积层。请问堆叠 3 个步长为 1 的  $3 \times 3$  卷积层，其有效感受野 (Receptive Field) 等同于一个多大的卷积层？这样做相比于直接使用大卷积核有什么主要优势？(A)
- A. 等同于  $7 \times 7$  卷积层；优势是拥有更少的参数量和更多的非线性变换。
- B. 等同于  $9 \times 9$  卷积层；优势是计算速度更快。
- C. 等同于  $9 \times 9$  卷积层；优势是可以捕捉更细微的特征。
- D. 等同于  $7 \times 7$  卷积层；优势是能够保留更多的空间分辨率。
6. 关于激活函数的性质与选择，下列说法中错误的是：(D)
- A. Sigmoid 函数会将输出压缩到  $[0, 1]$  之间，且存在梯度消失的问题。
- B. Tanh 函数的输出是以 0 为中心的，这有助于下一层参数的更新。
- C. ReLU 函数在  $x > 0$  区域梯度不会消失，且计算效率非常高。
- D. 在训练深度神经网络时，应优先选择 Sigmoid 或 Tanh 函数，尽量避免使用 ReLU 及其变体。
7. 在神经网络参数初始化中，针对使用 ReLU 激活函数的深层网络，为了防止激活值在层间传递时逐渐消失或爆炸，课程介绍的最佳初始化方法是：(D)
- A. 将所有权重初始化为 0

- B. 使用标准差为 0.01 的高斯分布随机初始化
- C. Xavier 初始化
- D. Kaiming 初始化

二、计算题：

1. 给定 Sigmoid 激活函数  $\sigma(x) = \frac{1}{1 + e^{-x}}$ .

- (a) 请结合计算图的链式法则原理，推导 Sigmoid 函数关于输入  $x$  的导数  $\frac{d\sigma}{dx}$ ，并将其结果表示为  $\sigma(x)$  的函数形式。
- (b) 假设在一个神经网络的反向传播过程中，某神经元的输入  $x = 0$ 。若已知损失函数  $\mathcal{L}$  对该神经元输出  $\sigma$  的梯度为  $\frac{\partial \mathcal{L}}{\partial \sigma} = 4.0$ ，请计算损失函数对该神经元输入  $x$  的梯度  $\frac{\partial \mathcal{L}}{\partial x}$ 。

Sigmoid 函数为  $\sigma(x) = (1 + e^{-x})^{-1}$ 。

利用链式法则：

$$\begin{aligned}\frac{d\sigma}{dx} &= \frac{d}{dx} (1 + e^{-x})^{-1} \\&= -1 \cdot (1 + e^{-x})^{-2} \cdot \frac{d}{dx} (1 + e^{-x}) \\&= -(1 + e^{-x})^{-2} \cdot (e^{-x} \cdot -1) \\&= \frac{e^{-x}}{(1 + e^{-x})^2} \\&= \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} \\&= \frac{1}{1 + e^{-x}} \cdot \frac{(1 + e^{-x}) - 1}{1 + e^{-x}} \\&= \sigma(x) \cdot (1 - \sigma(x))\end{aligned}$$

计算梯度

当  $x = 0$  时，前向传播输出： $\sigma(0) = \frac{1}{1+e^0} = \frac{1}{2} = 0.5$ 。

$$\frac{d\sigma}{dx} = \sigma(0)(1 - \sigma(0)) = 0.5 \times 0.5 = 0.25。$$

根据链式法则，损失函数对输入的梯度：

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial \sigma} \cdot \frac{d\sigma}{dx} = 4.0 \times 0.25 = 1$$

2. 假设有一个输入图像，维度为  $32 \times 32 \times 3$ （宽  $\times$  高  $\times$  RGB通道）。我们对其应用一个卷积层，包含 10 个滤波器（Filters），每个滤波器大小为  $5 \times 5$ ，步长（Stride） $S = 1$ ，填充（Padding） $P = 2$ 。

请计算：（请列出计算过程）

- a) 该卷积层输出特征图的空间尺寸（宽  $\times$  高  $\times$  深度）。  
b) 该卷积层的总参数量（包括偏置项 bias）。

a) 输出特征图尺寸

公式：  $W_{out} = \lfloor \frac{W_{in} - K + 2P}{S} \rfloor + 1$

- 输入  $W_{in} = 32, H_{in} = 32$
- $K = 5, S = 1, P = 2$
- 计算：  $\lfloor \frac{32 - 5 + 2 \times 2}{1} \rfloor + 1 = \lfloor \frac{31}{1} \rfloor + 1 = 32$
- 输出通道数等于滤波器个数 = 10。
- 结果：  $32 \times 32 \times 10$

b) 总参数量

- 权重W：每个滤波器大小为  $5 \times 5 \times 3$  (输入通道数)。  
 $10 \times (5 \times 5 \times 3) = 10 \times 75 = 750$
- 偏置 (Bias)：每个滤波器 1 个偏置。  
 $10 \times 1 = 10$
- 总计：  $750 + 10 = 760$

3. 考虑一个简单的 Inception 模块，输入特征图大小为  $28 \times 28 \times 256$ ，分支 1：  $1 \times 1$  卷积，输出通道数 128，分支 2：  $3 \times 3$  卷积，输出通道数 192，分支 3：  $5 \times 5$  卷积，输出通道数 96

a) 写出上述三个分支各自的卷积操作数，并给出该 Inception 模块总的卷积操作数（保留到百万级即可）。

b) 如果在  $3 \times 3$  分支和  $5 \times 5$  分支前各加一个  $1 \times 1$  卷积瓶颈层 来降低输入通道数：

- 分支 1： 仍为  $1 \times 1$  卷积，输出通道数 **128**（不变）
- 分支 2：
  - 先用  $1 \times 1$  卷积 将通道从 256 压缩到 64
  - 再用  $3 \times 3$  卷积 将通道从 64 升到 192

- 分支 3:
  - 先用  $1 \times 1$  卷积 将通道从 256 压缩到 32
  - 再用  $5 \times 5$  卷积 将通道从 32 升到 96

计算此时改进后模块的总卷积操作数。

计算改进后模块计算量相对于改进前下降的比例（保留到小数点后两位）。

a) 单层卷积操作数 (FLOPs) 近似公式:

$$H_{out} \times W_{out} \times C_{out} \times (K \times K \times C_{in})$$

分                      支    1                      (1x1):

$$28^2 \times 128 \times (1 \times 1 \times 256) \approx 784 \times 128 \times 256 \approx 25.7 \text{ M}$$

分                      支    2                      (3x3):

$$28^2 \times 192 \times (3 \times 3 \times 256) \approx 784 \times 192 \times 2304 \approx 346.8 \text{ M}$$

分                      支    3                      (5x5):

$$28^2 \times 96 \times (5 \times 5 \times 256) \approx 784 \times 96 \times 6400 \approx 481.7 \text{ M}$$

总计:  $25.7 + 346.8 + 481.7 = 854.2 \text{ M}$

b) 改进后模块计算量:

分支 1 (不变): 25.7 M

分支 2:

- 第一步 (256->64):  $28^2 \times 64 \times (1 \times 1 \times 256) \approx 12.8 \text{ M}$
- 第二步 (64->192):  $28^2 \times 192 \times (3 \times 3 \times 64) \approx 86.7 \text{ M}$
- 总共: 99.5 M

分支 3:

- 第一步 (256->32):  $28^2 \times 32 \times (1 \times 1 \times 256) \approx 6.4 \text{ M}$
- 第二步 (32->96):  $28^2 \times 96 \times (5 \times 5 \times 32) \approx 60.2 \text{ M}$
- 总共: 66.6 M

总共 (改进后):  $25.7 + 99.5 + 66.6 = 191.8 \text{ M}$

下降比例:  $\frac{854.2 - 191.8}{854.2} = \frac{662.4}{854.2} \approx 77.5\%$

4. 给定一个  $4 \times 4$  像素的图像区域 **I** 和一个  $2 \times 2$  的卷积核 **K**

(Kernel)。图像区域  $\mathbf{I}$ :  $\mathbf{I} = \begin{pmatrix} 10 & 20 & 30 & 40 \\ 50 & 60 & 70 & 80 \\ 90 & 100 & 110 & 120 \\ 130 & 140 & 150 & 160 \end{pmatrix}$ , 卷积核

$\mathbf{K}$ :  $\mathbf{K} = \begin{pmatrix} -1 & 0 \\ 1 & 2 \end{pmatrix}$ , 请计算:

a) padding=1, stride=1, 请计算卷积后的结果

b) padding=0, stride=2, 请计算卷积后的结果

a) :

$$\begin{pmatrix} 20 & 50 & 80 & 110 & 40 \\ 100 & 160 & 180 & 200 & 40 \\ 180 & 240 & 260 & 280 & 40 \\ 260 & 320 & 340 & 360 & 40 \\ 0 & -130 & -140 & -150 & -160 \end{pmatrix}$$

b):  $\begin{pmatrix} 160 & 200 \\ 320 & 360 \end{pmatrix}$

5. 前一层上的感受野和在输入图像上的感受野分别有什么不同? 卷积层, 池化层, 全连接层的感受野该如何计算?

前一层上的感受野等于当前层的卷积核大小。表示当前神经元看到了上一层特征图的多少个像素。

输入图像上的感受野表示当前神经元在原始输入图像上映射的区域大小。是累积的, 随着网络层数加深, 感受野会线性或指数级增长。

设  $R_L$  为第  $L$  层感受野,  $k_l$  为核大小,  $s_i$  为步长。(递推得到):

$$R_L = \sum_{l=1}^L \left( (k_l - 1) \prod_{i=1}^{l-1} s_i \right) + 1$$

卷积层: 使用上述公式累积,  $k$  为卷积核大小。

池化层: 计算方式同卷积层, 视为  $k \times k$  的卷积, 步长为  $s$ 。

全连接分两种情况, 前一层如果已经是全连接, 那么感受野不变, 如果前一层不是全连接层公式如上。

阅读参考: <https://distill.pub/2019/computing-receptive-fields/#return-from-solving-receptive-field-size>

## 6. 求出Alexnet网络中

- 每一层对输入图像的感受野（包含卷积层，池化层，全连接层）是多少？
- 计算每层的参数量和显存占用，找到网络中的参数瓶颈和显存占用瓶颈。

感受野：

CONV1: 11

MAX POOL1:  $11 + (3 - 1) \times (4 \times 1) = 19$

CONV2:  $19 + (5 - 1) \times (4 \times 2) = 51$

MAX POOL2:  $51 + (3 - 1) \times (8 \times 1) = 67$

CONV3:  $67 + (3 - 1) \times (8 \times 2) = 99$

CONV4:  $99 + (3 - 1) \times (16 \times 1) = 131$

CONV5:  $131 + (3 - 1) \times (16 \times 1) = 163$

MAX POOL3:  $163 + (3 - 1) \times (16 \times 1) = 195$

fc6:  $195 + (6 - 1) \times (16 \times 2) = 355$

fc7: 355

fc8: 355

参数：（分组版本）

CONV1:  $11 \times 11 \times 3 \times 96 + 96 = 34,944$

CONV2:  $5 \times 5 \times (96/2) \times 256 + 256 = 307,456$

CONV3:  $3 \times 3 \times 256 \times 384 + 384 = 885,120$

CONV4:  $3 \times 3 \times (384/2) \times 384 + 384 = 663,936$

CONV5:  $3 \times 3 \times (384/2) \times 256 + 256 = 442,624$

fc6:  $6 \times 6 \times 256 \times 4096 + 4096 = 37,752,832$

fc7:  $4096 \times 4096 + 4096 = 16,781,312$

fc8:  $4096 \times 1000 + 1000 = 4,097,000$

参数：（不分组）

CONV1:  $11 \times 11 \times 3 \times 96 + 96 = 34,944$

CONV2:  $5 \times 5 \times 96 \times 256 + 256 = 614,656$

CONV3:  $3 \times 3 \times 256 \times 384 + 384 = 885,120$

CONV4:  $3 \times 3 \times 384 \times 384 + 384 = 1,327,488$

$$\text{CONV5: } 3 \times 3 \times 384 \times 256 + 256 = 884,992$$

$$\text{FC6: } 6 \times 6 \times 256 \times 4096 + 4096 = 37,752,832$$

$$\text{FC7: } 4096 \times 4096 + 4096 = 16,781,312$$

$$\text{FC8: } 4096 \times 1000 + 1000 = 4,097,000$$

显存：（如果结果乘以字节数4也正确）

$$\text{input: } 227 \times 227 \times 3 = 154,587$$

$$\text{conv1: } 55 \times 55 \times 96 = 290,400$$

$$\text{pool1: } 27 \times 27 \times 96 = 69,984$$

$$\text{conv2: } 27 \times 27 \times 256 = 186,624$$

$$\text{pool2: } 13 \times 13 \times 256 = 43,264$$

$$\text{conv3: } 13 \times 13 \times 384 = 64,896$$

$$\text{conv4: } 13 \times 13 \times 384 = 64,896$$

$$\text{conv5: } 13 \times 13 \times 256 = 43,264$$

$$\text{pool5: } 6 \times 6 \times 256 = 9,216$$

$$\text{fc6: } 4,096$$

$$\text{fc7: } 4,096$$

$$\text{fc8: } 1,000$$