|  |  |  |
| --- | --- | --- |
| 第三章 深度学习应用实验 | | |
| 姓名：吴文韬 | 学号：20373914 | 班级：212111 |
| **注意：课程实验代码和实验报告需独立完成，严禁抄袭!** 实验3.1基于VGG19实现图像分类(40’’) **1.什么是卷积层？卷积层相比全连接层有哪些优势？**(10’’)  **卷积层是卷积神经网络中的核心组成部分。主要功能就是对输入数据进行卷积运算，以提取输入数据的特征，原理是通过一个或多个卷积核（也称为滤波器）在输入数据上滑动并进行卷积运算，从而提取出输入数据的局部特征。从本质上讲，其实卷积层就是一个稀疏的全连接。**  **理论上卷积层和全连接层可互换，下面是卷积层的相对优势**   1. **输入更灵活，可以接受任意分辨率的图像；** 2. **减少计算量，只需要做一次前向就可以得到一张热力图，然后可以根据stride和polling等参数反算出原图大致的位置。**   **2.池化层的作用是什么？常见的池化方式有哪些？**(10’’)  **池化层是模仿人的视觉系统对数据进行降维，用更高层次的特征表示图像。**  **实施池化的目的：(1) 降低信息冗余；(2) 提升模型的尺度不变性、旋转不变性；(3) 防止过拟合等等**  **常见的池化方式：最大值池化，均值池化，随机池化，中值池化，组合池化等**  **3.Inception的网络设计有什么特点？VGG19的网络设计有什么特点？** (10’’)  **在 Inception 出现之前，大部分流行 CNN 仅仅是把卷积层堆叠得越来越多，使网络越来越深，以此希望能够得到更好的性能，比如AlexNet，Inception网络的出现则是CNN分类器发展史上的里程碑，它的目的是设计一种具有优良局部拓扑结构的网络，即对输入图像并行地执行多个卷积运算或池化操作，并将所有输出结果拼接为一个非常深的特征图。因为 1\*1、3\*3 或 5\*5 等不同的卷积运算与池化操作可以获得输入图像的不同信息，并行处理这些运算并结合所有结果将获得更好的图像表征。**  **VGG是一种经典的卷积神经网络架构，VGG的开发是为了增加此类CNN的深度，以提高模型性能。至于VGG-19的特征，它相比其他网络具有更深的网络结构（19层）。**  **4.请自己找几张测试图片，用模型预测图片的类别。请在报告中展示测试图片以及对应的预测结果的截图。**(10’’)  dog  预测结果：  CleanShot 2023-11-06 at 11.13.43  查询imagenet\_classes.txt得到207 class对应的序号为n02099601，再在imagenet\_synsets.txt内查找该序号对应的含义：  CleanShot 2023-11-06 at 11.14.43  注：golden retriever为“黄金猎犬”，识别成功  pig  预测结果：(虽然没通过check pool的校验)  CleanShot 2023-11-06 at 10.51.04  查询imagenet\_classes.txt得到341 class对应的序号为n02395406，再在imagenet\_synsets.txt内查找该序号对应的含义：  CleanShot 2023-11-06 at 10.54.16 实验3.3 非实时图像风格迁移(60’’) **1.图像风格迁移的训练过程和实验2.1MNIST手写数字分类任务的训练过程有何不同？**(20’’)  网络结构不同，MNIST手写数字分类使用的是简单CNN卷积神经网络，图像风格迁移训练则使用的是预训练的深度卷积神经网络（VGG-19）;  损失函数不同：MNIST使用的是交叉熵损失函数，用于衡量预测的数字标签和真实标签之间的差距。而图像风格迁移的损失函数通常包含两部分：内容损失和风格损失，分别用于衡量生成图像与内容图像在内容上的差距和生成图像与风格图像在风格上的差距  训练过程也不同：MNIST基于大量数字标注的手写图像进行，而图像风格迁移是基于单个内容图像和风格图像进行的  **2.内容损失与风格损失分别是如何计算的？尝试解释为何要这样计算。**(20’’)  **def content\_loss(content, target):**  **return torch.mean((content - target)\*\*2)**  **内容损失是用于保持生成图像与内容图像在内容上的一致性。它通常定义为生成图像和内容图像在某一层卷积层输出的特征图之间的欧氏距离。这样定义的原因是，深层的卷积层通常能够捕获图像的高级语义信息，比如物体的形状和位置等**  **def gram\_matrix(input):**  **a, b, c, d = input.size()**  **features = input.view(a \* b, c \* d)**  **G = torch.mm(features, features.t())**  **return G.div(a \* b \* c \* d)**  **def style\_loss(style, target):**  **return torch.mean((gram\_matrix(style) - gram\_matrix(target))\*\*2)**  **风格损失是用于使生成图像匹配风格图像的风格。它通常定义为生成图像和风格图像在某一层卷积层输出的特征图的格拉姆矩阵之间的欧氏距离。（格拉姆矩阵是特征图的各个通道之间的内积）**  **3.请问如何控制最终生成的风格迁移图像更接近目标内容图像还是目标风格图像？**(10’’)  **调整内容损失和风格损失的权重即可，如果希望生成的图像更接近内容图像，可以增大内容损失的权重，减小风格损失的权重。这样，优化过程会更加重视内容的保持，而相对忽视风格的匹配。反之优化过程会更加重视风格的匹配，而相对忽视内容的保持**  **4.尝试用本实验的方法生成自己独特的风格迁移图像。**(10’’)  output_0 | | |