浙江水利水电学院

实验报告

（ 2019-2020 学年 2 学期）

|  |  |
| --- | --- |
| 课 程 名 称： | 卷积神经网络应用 |
| 班 级： | 软件工程S19-1,2 |
| 学 号： |  |
| 姓 名： |  |
| 实验室（中心）名 称： | 软件工程实验室 |
| 教学单位： | 信息工程与艺术设计学院 |

2020年 5 月 10 日

实 验 名 称：卷积神经网络应用 指导教师：张鹏亮

实 验 日 期：2020.4.28/30 地 点：现南411

同组学生姓名：组员1学号\_姓名、组员2学号\_姓名、组员2学号\_姓名

|  |
| --- |
| 实验内容及要求：  本次实验为使用卷积神经网络识别手写体数字。  本次任务分为两阶段。  第一阶段为任务一~任务三，必做任务。主要为定义卷积神经网络模型，加载数据，训练和识别。  第二阶段为任务四，扩展阶段。本阶段选做。主要训练开放性思维和学习能力。  详细内容参见《实验二指导书》  **2.*实验要求：***  完成各阶段任务，完成各阶试验结果及其分析。  详细内容参见《实验二指导书》 |
| 主要仪器名称及型号：   1. 计算机：Windows，i5 3Ghz,4GB内存以上，或更高的配置 2. 开发工具：Python /Numpy/PyTorch 3. 开发环境：PyCharm/jupyter notebook |
| 实验过程：（可附页）  **任务一：**  对于模型对象，我组进行了封装。通过输入卷积核大小、卷积步长、池化核大小、池化步长来进行实例化卷积神经网络，并自动计算网络参数。包括各层的形状，参数个数，连接数等。因篇幅所限，此处仅给出构造函数的参数说明，如图1-1。具体的代码实现请见项目文件夹下LeNet5.py文件。    图1-1 卷积神经网络构造函数图  **任务二：**  流程图如下图：    图2-1 加载数据集流程图  **任务三：**   1. 实验数据记录表格   将记录实验数据的代码封装在训练方法中，结果记录到一个字典返回，通过key-value键值对的形式存储。方便后续取得分析以及数据持久化。在后续训练不同模型时调用。下面列出需要记录的数据key。   |  |  | | --- | --- | | learning\_rate | 学习率 | | num\_epochs | 迭代周期 | | batch\_size | batch大小 | | loss\_x | 训练集当前迭代次数 | | loss\_y | 训练集对应loss值 | | loss\_y\_test | 测试集loss值 | | accuracy\_train | 训练集精度 | | accuracy\_test | 测试集精度 |  1. 识别错误的图片及分析   **流程图：**    图3-1 寻找识别错误的图片流程图  **代码：**    图3-2 代码实现  任务四：  2019b31073\_徐文昊：子任务4.1  实验步骤：用不同的卷积核、池化核、步长等参数实例化模型并训练。对比同样参数下（学习率、batch等）模型的loss、误差等数据来分析。  代码：    图4-1 子任务4.1代码图  数据记录表格：  已统一封装成字典，key如下。   |  |  | | --- | --- | | learning\_rate | 学习率 | | num\_epochs | 迭代周期 | | batch\_size | batch大小 | | loss\_x | 训练集当前迭代次数 | | loss\_y | 训练集对应loss值 | | loss\_y\_test | 测试集loss值 | | accuracy\_train | 训练集精度 | | accuracy\_test | 测试集精度 |   2019b31064\_陈威：子任务4.2  实验步骤：寻找不同的模型，对每个模型分成两块，一块进行初始化，另一块则不进行初始化，然后得出相关数据。  代码截图：      实验数据格式：   |  |  | | --- | --- | | learning\_rate | 学习率 | | num\_epochs | 迭代周期 | | batch\_size | batch\_size | | loss\_x | 训练集当前迭代次数 | | loss\_y | 训练集对应loss值 | | loss\_y\_test | 测试集loss值 | | accuracy\_train | 训练集精度 | | accuracy\_test | 测试集精度 | | duration | 运行时间 |   组员3学号\_姓名：子任务4.3  。。。。。 |
| 实验数据记录或图片：（可附页）  **任务一：**  下面验证网络结构，见图4-1。    图4-1 验证网络结构代码图    图4-2 网络结构及输出图  通过比较计算出的参数和实际网络的参数，对照网络的输出，可见网络初始化成功。  **任务二：**  运行结果：    图4-3 载入数据集结果图  经过测试，发现该函数的的下载路径是以python解释器执行调用该函数的.py文件的路径作为当前目录来参照的。同时可以使用绝对路径和相对路径。  **任务三：**  识别错误的图片如图3-2，图片上方第一个数字为标签，第二个数字为识别值。  根据课堂的内容知识，我们知道卷积神经网络通过卷积层与池化层的作用，对原图片进行了特征提取，且是局部的。而我们发现识别错误的图片，在识别值和标签的两个数中，有一些特征是很相似的。例如第一行的十张图片，我们圈出了其中一部分特征。比较有代表性的有第一行的第十章图片。它的标签是7，但是却可以提取出一个右上角的圆弧这种高维特征。这可能是导致卷积神经网络识别错误的原因。    图4-4 模型识别错误的图片  2019b31073\_徐文昊：子任务4.1  1.代码调试和解决  我组将训练过程封装，以供组员调用。为方便分析，统一数据的记录格式。此外，因为有实验一的基础，并未遇到需要调试的问题。  2.表格数据记录：  下面给出100batch, 20epoch, 1e-3学习率下的三种不同卷积核、池化核、步长的卷积神经网络的结果数据对比。此外在数据可视化中，给出三种卷积网络的loss对比。   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **卷积核** | **卷积步长** | **池化核** | **池化步长** | **训练精度** | **测试精度** | **训练时长** | | 5\*5 | 1 | 2\*2 | 2 | 98.62 | 98.88 | 694.82 | | 5\*5 | 1 | 2\*2 | 1 | 99.1875 | 99.04 | 1533.51 | | 3\*3 | 1 | 2\*2 | 1 | 99.28 | 98.71 | 1722.64 |   3.数据可视化：    图5-1 5\*5卷积核1卷积步长，2\*2池化核2池化步长的神经网络loss图    图5-2 5\*5卷积核1卷积步长，2\*2池化核1池化步长的神经网络loss图    图5-2 3\*3卷积核1卷积步长，2\*2池化核1池化步长的神经网络loss图  2019b31064\_陈威：子任务4.2  1.数据的选择：  根据任务的要求，我挑选了几个模型，分别观察它们的在设置了初始化参数和未设置初始化参数情况下所跑出的数据。  2.所选模型的数据：   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 学习率 | 迭代周期 | batch\_size | 训练集当前迭代次数 | 训练集对应loss值 | 测试集loss值 | 训练集精度 | 测试集精度 | 训练时间 | 是否初始化参数 | | 0.001 | 20 | 200 | [49……5980] | [0.57……0.00497] | [0.44……0.052] | 98.57 | 98.64 | 1451.71 | 否 | | 0.001 | 20 | 200 | [49……5980] | [0.57……0.00497] | [0.44……0.052] | 99.19 | 98.6 | 1488.345 | 是 | | 0.0001 | 20 | 200 | [49……5980] | [1.853……0.023] | [1.84……0.057] | 95.73 | 98.05 | 1426.16 | 否 | | 0.0001 | 20 | 200 | [49……5980] | [1.853……0.023] | [1.84……0.057] | 98.02 | 98.58 | 1419.04 | 是 | |
| 实验结论：  任务一~三，请和实验1进行对比分析，请从参数数量、训练时间、测试集上的准确率等各方面进行比较，说明卷积网络的优缺点。小组共同完成。  针对各自所负责的任务四的子任务，结合实验中出现的问题和现象，使用实验中记录的数据进行解释说明，并分析原因、提出解决办法。分析的过程可以使用图、表，结合文字进行解释说明。  对于任务四的子任务（4），一个开放性的问题，有兴趣的同学可以进行相关实验并解释。  每位组员在阐述自己的问题时，如需要重新设计实验或添加实验，也可在此部分添加相关内容并进行说明。  任务三：  下面给出学习率为1e-3，epoch为200，batch为256，激活函数都为relu的神经网络，与5\*5卷积核1卷积步长，2\*2池化核，2池化步长的LeNet5卷积神经网络的实验结果数据对比表格。（普通神经网络采用GPU训练，LeNet5用CPU）   |  |  |  |  | | --- | --- | --- | --- | | **神经网络类型** | **训练精度** | **测试精度** | **训练时长** | | LeNet5 | 98.62 | 98.88 | 694.82 | | 2-layer NN, 300隐藏层神经元 | 96.42 | 97.34 | 1075.64 | | 2-layer NN, 1000隐藏层神经元 | 97.65 | 97.9 | 598.31 | | 3-layer NN, 300+100隐藏层神经元 | 97.54 | 97.92 | 3680.98 | | 3-layer NN, 500+150隐藏层神经元 | 97.26 | 98.11 | 908.6 | | 3-layer NN, 500+300神经元 | 99.14 | 98.1 | 513.96 | | 2-layer NN, 800隐藏层神经元 | 98.63 | 98.08 | 860.13 |   得益与卷积神经网络的特征提取，卷积神经网络有着更小的参数值。同时更加关注高维的特征，在大大减少训练时间的同时，精度也有着一定程度的突破。缺点在于，如上文分析识别错误图片提到的，卷积神经网络有些时候会被部分不那么清晰的特征给迷惑，做出错误的识别。对特征提取的依赖非常大。所以就对卷积层和池化层有着很高的要求。  2019b31073\_徐文昊：任务4.1  不同卷积层和池化层参数对识别结果的影响（其他参数均相同）：   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **卷积核** | **卷积步长** | **池化核** | **池化步长** | **训练精度** | **测试精度** | **训练时长** | | 5\*5 | 1 | 2\*2 | 2 | 98.62 | 98.88 | 694.82 | | 5\*5 | 1 | 2\*2 | 1 | 99.1875 | 99.04 | 1533.51 | | 3\*3 | 1 | 2\*2 | 1 | 99.28 | 98.71 | 1722.64 | | 4\*4 | 1 | 2\*2 | 1 | 97.37 | 98.56 | 412.35 | | 4\*4 | 2 | 2\*2 | 1 | 99.18 | 98.83 | 1243.68 |   不同卷积网络的输出大小及参数数量：   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **卷积层和池化层参数** | **卷积层1输出大小** | **卷积层1参数个数** | **池化层1输出大小** | **卷积层2输出大小** | **卷积层2参数个数** | **池化层2输出大小** | | 5-1-2-2 | 24\*24 | 156 | 12\*12 | 8\*8 | 416 | 4\*4 | | 5-1-2-1 | 24\*24 | 156 | 23\*23 | 19\*19 | 416 | 18\*18 | | 3-1-2-1 | 26\*26 | 60 | 25\*25 | 23\*23 | 160 | 22\*22 | | 4-1-2-1 | 25\*25 | 102 | 24\*24 | 21\*21 | 272 | 20\*20 | | 4-2-2-1 | 13\*13 | 102 | 12\*12 | 5\*5 | 272 | 4\*4 |   我们可以看到，在精度方面，五个模型的差别都不是很大。但是在训练时长方面，差异非常巨大。可见卷积神经网络非常依赖卷积层和池化层。好的卷积层和池化层，能够做出优秀的局部连接，进行高性能的参数共享，使识别更有效率。这在工程上具有巨大的意义。  2019b31064\_陈威：任务4.2  表格数据：   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 学习率 | 迭代周期 | batch\_size | 训练集当前迭代次数 | 训练集对应loss值 | 测试集loss值 | 训练集精度 | 测试集精度 | 训练时间 | 是否初始化参数 | | 0.001 | 20 | 200 | [49……5980] | [0.57……0.00497] | [0.44……0.052] | 98.57 | 98.64 | 1451.71 | 否 | | 0.001 | 20 | 200 | [49……5980] | [0.57……0.00497] | [0.44……0.052] | 99.19 | 98.6 | 1488.345 | 是 | | 0.0001 | 20 | 200 | [49……5980] | [1.853……0.023] | [1.84……0.057] | 95.73 | 98.05 | 1426.16 | 否 | | 0.0001 | 20 | 200 | [49……5980] | [1.853……0.023] | [1.84……0.057] | 98.02 | 98.58 | 1419.04 | 是 |   本次实验，我们主要采用的初始化参数方法是Kaiming初始化。  理由：首先，对于初始化权重的目的是为了防止深度神经网络的正向传播过程中出现激活函数的输出损失梯度出现爆炸或消失。其次，通过所列出的表格数据以及实验所得出的其他数据对比，可以看出运用了初始化参数方法的模型，训练集精度和测试集精度相对于未初始化的模型较高，随着学习率的越来越小，两者之间的对比就更为明显。  组员3学号\_姓名：  分析解释 |
| 教师评语：（成绩部分学生请勿填写和更改）  总体情况：  完成情况：优  实验报告：优  拓展能力：选择一项。  组员1： (请同学们填上学号\_姓名) 成绩：选择一项。  组员2： (请同学们填上学号\_姓名) 成绩：选择一项。  组员3： (请同学们填上学号\_姓名) 成绩：选择一项。  批阅教师：张鹏亮  2020年5月15日 |