浙江水利水电学院

实验报告

（ 2019-2020 学年 2 学期）

|  |  |
| --- | --- |
| 课 程 名 称： | 神经网络与深度学习 |
| 班 级： | 软件工程S19-2 |
| 学 号： | 2019b31073 |
| 姓 名： | 徐文昊 |
| 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 4 月 18 日

实 验 名 称：使用神经网络识别手写体数字 指导教师：张鹏亮

实 验 日 期：2020.4.7/9 地 点：现南411

同组学生姓名：2019b31064\_陈威、2019b31068\_陈榆

|  |
| --- |
| 实验内容及要求：  **1.*实验内容：***  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~任务四，必做任务。主要为定义卷积神经网络模型，加载数据，定义训练过程。  第二阶段为任务五~任务六，扩展阶段。主要为模型训练、实验数据记录和分析。  详细内容参见《实验一指导书》  **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《实验一指导书》 |
| 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）  **实验项目工程结构：**    图 1-1 项目工程结构  其中各模块的说明如下：   |  |  | | --- | --- | | resources/ans/exp\_data | 文件夹，保存实验结果数据 | | resources/dataset | 文件夹，数据集 | | analysis.py | 实验结果数据分析 | | common.py | 任务1、2的实现 | | mnist.py | 实验相关的函数，给其他模块调用 | | module1.py – module6.py | 6种模型的训练 | | MyModule.py | 模型的定义 | | painter.py | 封装的plot绘图函数 |   **任务一代码设计：**   1. 载入数据集和测试集     图2-1 加载数据集流程图   1. 将标签向量转换为one-hot编码   设计：传入n\*1的标签向量y。预设一个n\*10的张量t。遍历每一个y[i]，将t中y[i]对应位置置为1。代码由徐文昊编写。流程如图2-2。    图2-2 转换one-hot编码流程图  **任务二代码设计：**  参考了电子教材3.5节展示FashionMNIST数据集的方式展示。因任务需要写出注解，所以下面由陈威给出具体实现的代码截图。见图2-3。    图2-3 展示数据集图片代码  **任务三代码设计：**  对于6种模型，将其抽象为两个类。分别表示单隐藏层模型与双隐藏层模型。通过传入的参数来实例化。具体实现由徐文昊编写代码。详情见类图2-4。    图2-4模型类图  其中，类Net2Layers为单隐藏层模型。参数说明如下*。*   |  |  |  | | --- | --- | --- | | hidden\_size | int | 隐藏层神经元个数 | | activation | str | 激活函数类型。  “sigmoid” or “relu”。 | | out\_activate | bool | 输出层是否使用激活函数。  True or False。 |   类Net3Layers为双隐藏层模型。参数说明如下*：*   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | hidden\_size | int[] | 数组，隐藏层神经元个数 |  |  |  | | activation | str | 激活函数类型。  “sigmoid” or “relu”。 |  |  |  | | out\_activate | bool | 输出层是否使用激活函数。  True or False。 |  |  |  |   **任务四代码设计：**   1. 根据不同的激活函数初始化权重及偏置   在设计神经网络模型类时，我们就考虑到了这个问题。我们将初始化权重及偏置逻辑封装在了模型的类中，通过传入不同的激活函数来初始化，隐藏了具体的实现细节，方便不同的组员调用。流程如图2-5。    图2-5 权重及偏置初始化流程图   1. 针对损失函数MSE转换标签形式   详见任务一代码设计第二小节。   1. 设计实验探寻最佳的超参数组合   定义不同的学习率、epoch、激活函数、输出层是否使用激活函数四个类型的参数数组，通过foreach循环穷举，定义模型并训练，保存实验数据以供分析。  任务流程详见图2-6。    图2-6 探索超参数流程图  **任务五代码设计：**  结合任务五与任务六。我组认为如果不统一记录数据的格式以及方式，会对后续的分析产生不利影响，很多封装的分析函数等将不能通用，也无法对数据进行很好的横向与纵向比较。所以经过我们一致讨论，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 |   代码由徐文昊实现，关键代码见图2-7。    图2-7 记录数据代码实现 |
| 实验数据记录或图片：（可附页）  **任务一：**  对于子任务1，采用pytorch的API，并未遇到特别的问题。  对于子任务2，通过断点调试，分析data\_loader中的成员变量，发现有data.dataset.data与data.dataset.targets两个成员变量分别保存数据集及对应标签，取得之后转换one-hot即可。由徐文昊给出了代码实现，由于任务很基础，并未遇到问题。结果见图3-1-1。Debug过程发现的成员变量见图3-1-2。    图3-1-1 任务一结果图    图3-1-2 data\_loader中存储的训练集和测试集形式图  **任务二：**  参考了电子教材3.5节展示FashionMNIST数据集的方式展示。具体的实现及注解已由陈威在任务二代码设计中给出。下面直接给出结果，见图3-2。    图3-2 任务二结果图  **任务三：**  具体的设计已在任务三代码设计中给出。在抽象成类的过程中，我们遇到的问题有：  （1）如何自定义一个模型类。  （2）模型类中如何实现自定义的隐藏层。  （3）如何在自定义的模型类中初始化权重和偏置。  解决方法：   1. 通过查阅pytorch官方文档，发现可以继承torch.nn.Module类。在\_\_init\_\_构造函数中申明各个层的定义，在forward中实现层之间的连接关系，实际上就是前向传播的过程。 2. 继续查阅官方文档，发现可以通过torch.nn.Sequential类来实现。它是一个连续的容器，torch.nn.Module将按照它们在构造函数中传递的顺序添加。同时，激活函数也可以通过该类定义。 3. 查看torch.nn.Module的源代码之后，发现参数是全部保存在私有成员变量\_parameters中的。结合老师给的代码，可以通过遍历字典named\_parameters方式，根据不同的激活函数，利用nn.init模块完成初始化权重和偏置。   部分实现：    图3-3 模型的代码实现图  **任务四：**  本次训练完全应用了pytorch的自动求导及反向传播特性。具体的设计以及代码已由前文给出。遇到的问题有：   1. 如何设置权重衰减。   解决方式：  （1）我组采用Adam优化器。查阅文档得知，Adam优化器的实例化方法中，有个参数是weight\_decay，即权重衰减的值。我们采用了推荐的1e-8。  **任务五：**  我组将数据的记录统一封装成了字典，具体格式已由前文给出。具体的代码实现由徐文昊编写。下面每个组员根据自己负责的模型给出部分实验数据的展示。  2019b31073\_徐文昊：  1.数据的代码的调试和解决：  在代码实现方面，因良好的编码水平，并未遇到需要调试解决的问题。  在实验需要记录的数据方面，由陈威提示，根据任务四寻找最佳的超参数epoch,需要记录测试集的loss，此key在结果字典中并未记录。通过补充训练方法在结果字典中添加了此key及对应的数据。  下面给出训练的60种参数组合之一的实验数据结果。  2.表格数据记录：   |  |  |  | | --- | --- | --- | | **key** | **value** | **备注** | | learning\_rate | 0.001 | 学习率 | | num\_epochs | 100 | epoch | | batch\_size | 5000 | batch\_size | | loss\_x | [0,1,2,…]，由下文可视化给出 | 迭代次数 | | loss\_y | […]，由下文可视化给出 | 对应loss | | loss\_y\_test | […] | 测试集对应loss | | accuracy\_train | 97.65241666666667 | 训练集精度 | | accuracy\_test | 97.9 | 测试集精度 | | duration | 598.3123314380646 | 训练时长(秒) |   3.数据可视化：    训练集上的损失函数变化趋势。x轴表示迭代次数，y轴表示对应的loss值。为更好的展示loss趋于更小时候的变化趋势，前几次迭代的部分较高的loss数据并未绘入图中。  2019b31064\_陈威：  1.数据的代码的调试和解决：  统一抽象成函数封装，调用即可。  2.表格数据记录：   |  |  |  | | --- | --- | --- | | **key** | **value** | **备注** | | learning\_rate | 0.001 | 学习率 | | num\_epochs | 100 | epoch | | batch\_size | 5000 | batch\_size | | loss\_x | [0,1,2,…1100] | 迭代次数 | | loss\_y | [0.3054652214050293…],具体参数根据下图 | 对应loss | | loss\_y\_test | […] | 测试集对应loss | | accuracy\_train | 96.92618333333333 | 训练集精度 | | accuracy\_test | 97.85 | 测试集精度 | | duration | 1114.10209441185 | 训练时长(秒) |   3.数据可视化：    x轴表示迭代次数，y轴表示对应的loss值。  根据图所示，随着迭代次数的增加，loss逐渐变小。在迭代次数较少向逐渐递增的趋势下，可以看到loss下降的趋势比较明显，随着迭代次数的增大，loss的趋势相对较缓。  2019b31068\_陈榆：  1.数据的代码的调试和解决：  调用写好的函数。  2.表格数据记录：   |  |  |  | | --- | --- | --- | | **key** | **value** | **备注** | | learning\_rate | 0.001 | 学习率 | | num\_epochs | 100 | epoch | | batch\_size | 5000 | batch\_size | | loss\_x | [0,1,2,…] | 迭代次数 | | loss\_y | […] | 对应loss | | loss\_y\_test | […] | 测试集对应loss | | accuracy\_train | 98.63368333333334 | 训练集精度 | | accuracy\_test | 98.08 | 测试集精度 | | duration | 860.1268711090088 | 训练时长(秒) |   3.数据可视化：    训练集上的损失函数变化趋势。x轴表示迭代次数，y轴表示对应的loss值。 |
| 实验结论：  （给出你对你的算法的评价，是否准确可靠，优缺点描述，改进描述。以及在实验过程中出现问题的原因分析，解决办法等）撰写实验报告时，请删除此段内容。请注意排版时注意格式。  针对各自所负责的模型，结合任务六以及任务三~任务五的要求和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。  对于任务六的子任务5~6，他们是综合性问题，请其中的一位组员进行分析阐述。  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。  组员1学号\_姓名（负责解释任务六的子任务5~6）：  分析解释  组员2学号\_姓名：  分析解释  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4月25日 |