

作业 2

2022 年 3 月 17 日

本次作业通过多层感知机实现非线性分类任务，帮助同学们理解机器学习的基本要素。作业内容为多层感知机的实现及模型优化求解方法，并在数据集上进行训练和测试。具体任务分为理论部分、编程部分以及作业报告。其中理论部分包含第1, 2题，所有同学均需完成，答案附在作业报告中；编程部分包含第3、4题，作业报告为第5题，已确认自选课题的同学需完成第6题。

1. 单选题（15 分）
2. 计算题（15 分）
3. 完成非线性分类器的程序代码（30 分）
4. 训练/测试/可视化（30 分）
5. 撰写作业报告（10 分）
6. 汇报自选课题进度（70 分）*

理论部分

1 单选题（15分）

1.1 以下关于神经网络的训练说法正确的是

- (A) 神经网络的权值时可以全部初始化为0。
- (B) 当网络的训练结果没有达到预计的效果，提高学习率一定可以提高网络的学习能力。
- (C) 加入L2正则化，会导致训练中的损失函数变大，进而恶化网络的训练效果。
- (D) 在标准的随机梯度下降法中，权值衰减正则化可以由 L_2 正则化进行替代。

1.2 以下哪种方法不能防止训练的过拟合

- (A) 进行正则化，例如采用权值衰减。
- (B) 对输入的数据进行数据增强。
- (C) 提高训练的轮数，增强网络训练的效果。

1.3 Tanh激活函数如下：

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1)$$

以下关于编程实现Tanh的说法正确的是：

- (A) 某些情况下分母的值将为0，需要在分母上加上应该极小值防止除零。
- (B) 某些情况下指数项会溢出，导致计算结果出错。
- (C) 某些情况下y的绝对值会大于1。

1.4 考虑使用复合的函数来作为激活函数，如下：

$$y = f(g(z)) + t(z) \quad (2)$$

在其误差反向传播过程如下，其中正确的是：

- (A) $\frac{\partial L}{\partial z} = \frac{\partial L}{\partial y} * (f'(z)g'(z) + t'(z))$
- (B) $\frac{\partial L}{\partial z} = \frac{\partial L}{\partial y} * (f'(y)g'(y) + t'(y))$
- (C) $\frac{\partial L}{\partial z} = \frac{\partial L}{\partial y} * (f'(g(z))g'(z) + t'(z))$
- (D) $\frac{\partial L}{\partial z} = \frac{\partial L}{\partial y} * (f'(y)g'(z) + t'(z))$

1.5 标量 $y = \mathbf{a}^T W \mathbf{x}$ ，其中 $\mathbf{a} \in R^{n \times 1}$, $W \in R^{n \times n}$, $\mathbf{x} \in R^{n \times 1}$ ，现求标量y对向量x的偏导数 $\frac{\partial y}{\partial \mathbf{x}}$ 为：

- (A) $\mathbf{a}^T W$
- (B) $W \mathbf{a}^T$
- (C) $W \mathbf{a}$
- (D) $W^T \mathbf{a}$

2 计算题（15 分）

2.1 设隐含层为 $\mathbf{z} = \mathbf{x}\mathbf{W}^T + \mathbf{b}$ ，其中 $\mathbf{x} \in R^{(1 \times m)}$ ， $\mathbf{z} \in R^{(1 \times n)}$ ， $\mathbf{W} \in R^{(n \times m)}$ ， $\mathbf{b} \in R^{(1 \times n)}$ 均为已知，其激活函数如下：

$$\mathbf{y} = \tanh(\mathbf{z}) = \frac{e^{\mathbf{z}} - e^{-\mathbf{z}}}{e^{\mathbf{z}} + e^{-\mathbf{z}}}$$

若训练过程中的目标函数为 L ，且已知 L 对 \mathbf{y} 的导数

$$\frac{\partial L}{\partial \mathbf{y}} = [\frac{\partial L}{\partial y_1}, \frac{\partial L}{\partial y_2}, \dots, \frac{\partial L}{\partial y_n}] \text{ 和 } \mathbf{y} = [y_1, y_2, \dots, y_n] \text{ 的值。}$$

2.1.1 请使用 \mathbf{y} 表示出 $\frac{\partial \mathbf{y}}{\partial \mathbf{z}}$

2.1.2 请使用 \mathbf{y} 和 $\frac{\partial L}{\partial \mathbf{y}}$ 表示 $\frac{\partial L}{\partial \mathbf{x}}$ ， $\frac{\partial L}{\partial \mathbf{W}}$ ， $\frac{\partial L}{\partial \mathbf{b}}$ 。

提示： $\frac{\partial L}{\partial \mathbf{x}}$ ， $\frac{\partial L}{\partial \mathbf{W}}$ ， $\frac{\partial L}{\partial \mathbf{b}}$ 与 \mathbf{x} ， \mathbf{W} ， \mathbf{b} 具有相同维度。

编程部分

编程部分包括第3，4题。

3 完成非线性分类器的程序代码（30 分）

使用多层感知机完成识别英文字符图像（非线性分类）任务，在本任务中，我们将使用MLP识别英文字符图像，即输入一张图像，模型输出识别结果。请注意图像样本包括大小写英文字母，识别时不区分大小写形式，即大写字母“A”和小写字母“a”都对应于一个类别，输出时转为大写字母“A”显示。

程序清单如下：

| 文件或目录 | 说明 | 注意事项 |
|----------------|-------------|------------|
| hw2.zip | 作业 2 程序压缩包 | 解压可以得到下列文件 |
| \data | 存放本次作业所用数据集 | 请勿修改 |
| \saved_models | 存放本次作业保存模型 | 请勿修改 |
| activations.py | 激活函数定义 | 已完成代码 |
| losses.py | 损失函数定义 | 需要完成代码 |
| network.py | 神经网络结构定义 | 需要完成代码 |
| recognition.py | 非线性分类程序 | 需要完成代码 |

需要完成的代码清单如下：每处需要完成的地方都有代码提示和步骤提示

activations.py文件中所有激活函数已经完成，感兴趣的同学可以阅读学习如何构造激活函数。

losses.py文件中待完成内容：

| 序号 | 行号 | 内容 | 说明 |
|--------|----|-----------|--|
| TODO 1 | 21 | 完成交叉熵损失函数 | 请按照step 1-3完成其前向函数， <u>注意数据需要转化为one_hot类型和防止log(0)</u> ；按照step 4完成后向函数 |

network.py文件中待完成内容：

| 序号 | 行号 | 内容 | 说明 |
|--------|-----|-------------|-------------------------|
| TODO 1 | 41 | 完成线性函数前向函数 | 注意要保持相关数据到ctx中 |
| TODO 2 | 58 | 完成线性函数后向函数 | 注意要计算多个参数的梯度 |
| TODO 3 | 82 | 完成线性层权值的初始化 | 注意初始化的处置和设置为可训练，可以参考作业1 |
| TODO 4 | 122 | 完成MLP模型 | 按照step 1-4 完成相关内容 |

recognition.py文件中待完成内容：

| 序号 | 行号 | 内容 | 说明 |
|--------|-----|-----------------|--|
| TODO 1 | 135 | 完成MLP模型和损失函数的定义 | 按照MLP()和损失函数的定义来使用 |
| TODO 2 | 173 | 计算loss和训练网络 | 按照step 1-6 来完成代码 |
| TODO 3 | 333 | 读取保存的模型 | 按照step 1-3 完成相关内容，可以参考test()中读取保存的模型代码 |

4 训练/测试/可视化（30分）

（1）在默认参数下，使用的是SGD优化器，同学们可以调整动量因子（momentum）来观察和记录其对训练、验证和测试性能的影响。这里给出一些参考值，比如选取动量因子为0.9和0.95。其中训练和测试的运行命令如下：

1) 训练模型，使用默认配置的命令如下：

增大hsize可以提高模型非线性表示能力，和权重衰减结合效果比较好（典型的如adam+64+2e-3+0.15），即使用sgd也有较好的效果，但是loss曲线不如adam平滑

```
python recognition.py --mode train
```

2) 测试模型，使用默认配置的命令如下：

```
python recognition.py --mode test
```

(2) 可以调整优化器为Adam优化器，可以修改其他参数如下，观察网络训练、验证和测试的性能。

hszie默认是32

```
python recognition.py --mode train --hsize 64 --lr 2e-3 --optim_type adam  
--momentum 0 --weight_decay 0.1
```

adam学习率不能太低，否则收敛太慢

adam好像也需要较大的hszie

注意：由于该分类的任务训练数据比较小，在增加隐含层层数时可能会导致性能下降，同时在该任务中Relu为较合适的激活函数，在选择其他激活函数时可能会有严重的性能下降。

(2) 可视化：在测试中会输出分类的可视化结果，其将高维分类的结果降维到二维空间，从而使其可视化。但是由于本次任务采用的训练数据较少，训练得到的模型分类性能较差，其可视化的结果也不是很理想。选取你认为结果比较好的可视化结果附在实验报告中。

(3) 使用训练好的模型预测图像类别：请选择你认为效果最好的一个模型，对新的图像进行识别（设置mode为predict）。可以使用我们提供的predict01.png和predict02.png，也可以自行收集更多图像进行测试。请把使用模型的情况、输入样本的预测结果写入作业报告。

使用训练好的模型预测输入样本的类别，使用命令例如：

```
python recognition.py --mode predict --im_path  
data/character_classification/new_images/predict02.png
```

当程序运行训练 (train) 和测试过程可视化显示相关图片时，在图片显示窗口工具栏有保存图片的按钮，可手动保存图片。手动关闭图片窗口后，程序可以继续运行至结束。可视化截图需要附在作业报告中。

5 撰写作业报告（10 分）

将hw2目录和作业报告打包为一个文件（例如 *.zip）提交到网络学堂。作业报告中包括选择题答案，计算题的解题步骤及答案、任务3、4运行结果及分析，本次作业遇到的问题及解决方法，对本次作业的意见及建议。推荐同学们使用随作业发布的LaTex模板HW2-template.zip完成作业报告。

6 自选课题进度汇报（70 分）*

请已确认自选课题的同学，完成简短的自选课题工作进度汇报，例如，文献阅读、或者研究方案设计、或者原型系统搭建及实验结果等内容。

关于作业迟交的说明：由于平时作业计入总评成绩，希望同学们能按时提交作业。若有特殊原因不能按时提交，请在提交截止时间之前给本次作业责任助教发 Email 说明情况并给出预计提交作业的时间。对于未能按时说明原因的迟交作业，将酌情扣分。

本次作业责任助教为章磊(Email: zhanglei21@mails.tsinghua.edu.cn)。

7 附录

我们通过argparse库进行参数设置，可以查看recognition.py中可以调节的参数（第405行到第435行），可调节参数如下表所示。

| | 参数 | 说明 |
|----------------------------|-----------------|---|
| 代码运行模式 | mode | train, test 或者 predict train 为模型训练; test 为模型测试; predict 为使用训练好的模型对新的样本进行识别 |
| 训练、验证和测试时将使用的参数 | im_dir | 存放训练/验证/测试图像的文件夹路径, 可以不用修改 |
| | train_file_path | 携带训练集图像名称和标签的 json 文件地址, 可以不用修改 |
| | val_file_path | 携带验证集图像名称和标签的 json 文件地址, 可以不用修改 |
| | test_file_path | 携带测试集图像名称和标签的 json 文件地址, 可以不用修改 |
| | batchsize | 训练、验证和测试时每一批数据中包含的数据个数 |
| | device | 训练、验证和测试时使用的设备, 可以为 cpu 或 cuda |
| mode 设为 train 时使用 | hsize | MLP 隐含层节点数, 多个隐含层以英文逗号分开 (不加空格) |
| | layer | MLP 层数, 应等于隐含层数量 + 1 (输出层) 若 layer=1 则 MLP 退化为线性层 |
| | act | 激活函数类型, 可以为 none, sigmoid, tanh, relu 中的一种 其中 none 表示无激活函数 |
| | norm_size | 图像归一化大小 |
| | epoch | 训练轮数 |
| | n_classes | 待分类类别数, 对于本任务 (英文字母识别) 是 26 |
| | valInterval | 进行验证的频率 |
| | lr | 学习率 |
| | optim_type | 优化器类型, 可以设为 SGD, adam, adagrad, rmsprop 或 adadelta |
| | momentum | SGD 优化器的动量, 只有 optim_type 设为 SGD 时使用 |
| | weight_decay | L2 正则化系数 |
| mode 设为 test 或 predict 时使用 | model_path | 训练好的模型保存的路径, 可以不用修改 |
| | im_path | 待识别图片路径, 我们提供两张图片 predict01.png 和 predict02.png, 存放在 data/character_classification/new_images/ 文件夹中 |