作业一:简单的文本分类器

姓名：梁辉

学号：10185501411

使用pytorch实现的的简单的英文文本分类网络。

环境要求

- python == 3.8
- torch ==1.10

其余见`requirements.txt`

使用方法

- 预处理数据执行`./preprocess.sh --config_path ./config/config_preprocess.json`
- 执行训练网络`python3 main.py --config_path ./config/config_mlp.json`或`python3 main.py --config_path ./config/config_rnn.json`

实现思路与过程

1、数据集处理

数据集json形式的20news，每一条json包含分好词的文本`text`和文本类型标签`label`

文章长度不一，文本类型一共有20个类型

预处理preprocess过程：

使用word2vec包将数据集中的单词转化为10维的向量，并将文本中的单词tokenizer化为索引。文章删去停用词后取出文章中150个词代表该篇文章并将文章中的词转化为向量，得到预训练的embedding layer。

之后将数据集划分为训练集和验证集。

将处理后的训练集和验证集数据，分别存储在`data`目录下。

`data`目录中`trainTensor0-4.pt`是标签为0-4的五分类的训练集数据，`trainLabels0-4.pt`是标签0-4的五分类训练集的标签数据,`trainTensor5-9`是标签5-9的训练数据集，其余的文件类似，`all_trainTensor.pt`是全部20分类的训练数据集，`all_trianLabels.pt`是全部20分类的训练集标签数据集。

2、模型实现

MLP(2 layer)

Linner + ReLu + Dropout + Linear +softmax

使用参数`config_mlp.json`

模型保存在`data\2layer_mlp.pkl`文件

RNN

使用nn.LSTM模型

模型保存在`data\lstm_rnn.pkl`文件

3、部分参数说明

config_preprocess.json中参数

```
{
  "data_path": "./data/20news.json",
  "stop_path": "./data/停用词.txt", #停用词目录
  "max_seqence_length": 150, #每篇文章取的最大单词数
  "embedding_len": 10, #每个词转变为词向量的长度
  "validation_split": 0.8, #训练集和测试集的占比
  "output_path": "./data",
  "use_5split": true #将数据集改为五分类任务或者全部20分类任务
}
```

config_mlp.json中的参数

```
{
  "output_path": "save/mlp_1",
  "gpu": false,
  "optimizer": "adagrad",
  "lr": 0.005, #学习率
  "lr_decay": 0,
  "num_epochs": 100,
  "batch_size": 128,
  "num_labels": 5, #标签数量，为几分类问题
  "embedding_size": 10, #词向量长度
  "type": "mlp", #网络类型
  "config_type": 0,#五分类的标签集，0代表0-4标签，1代表5-9标签，2代表10-14标签，3代表15-19标签
  "mlp": {
    "max_length": 1500,
    "dropout": 0.5,
    "hidden_size": 1500,
    "loss": "cross_entropy"
  }
}
```

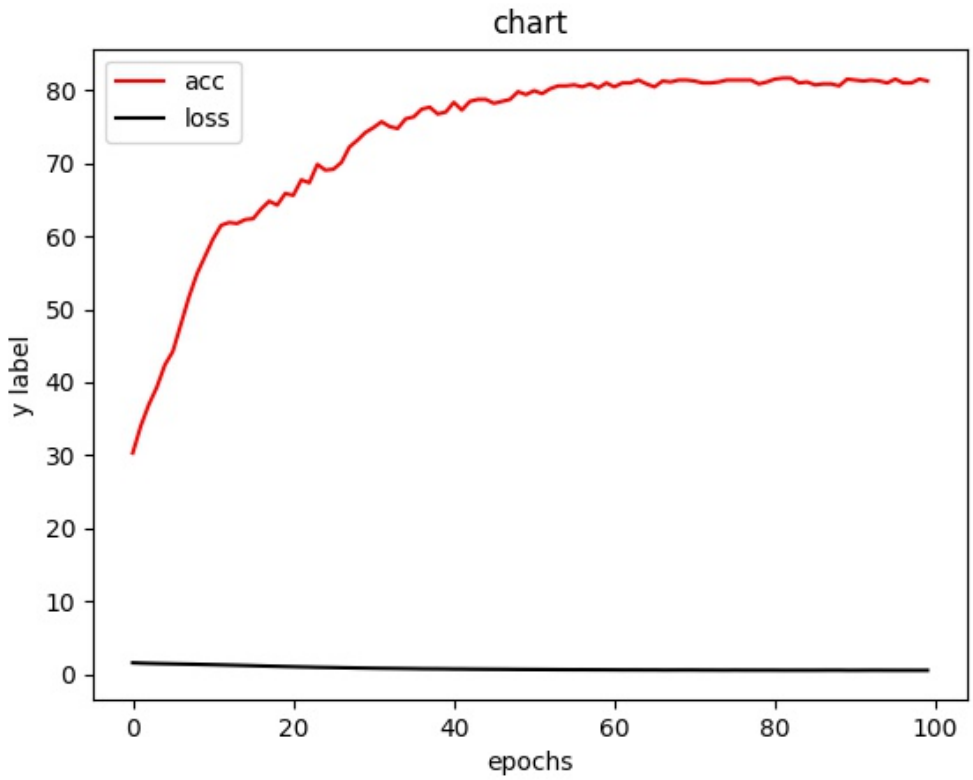
实验结果

- **MLP**

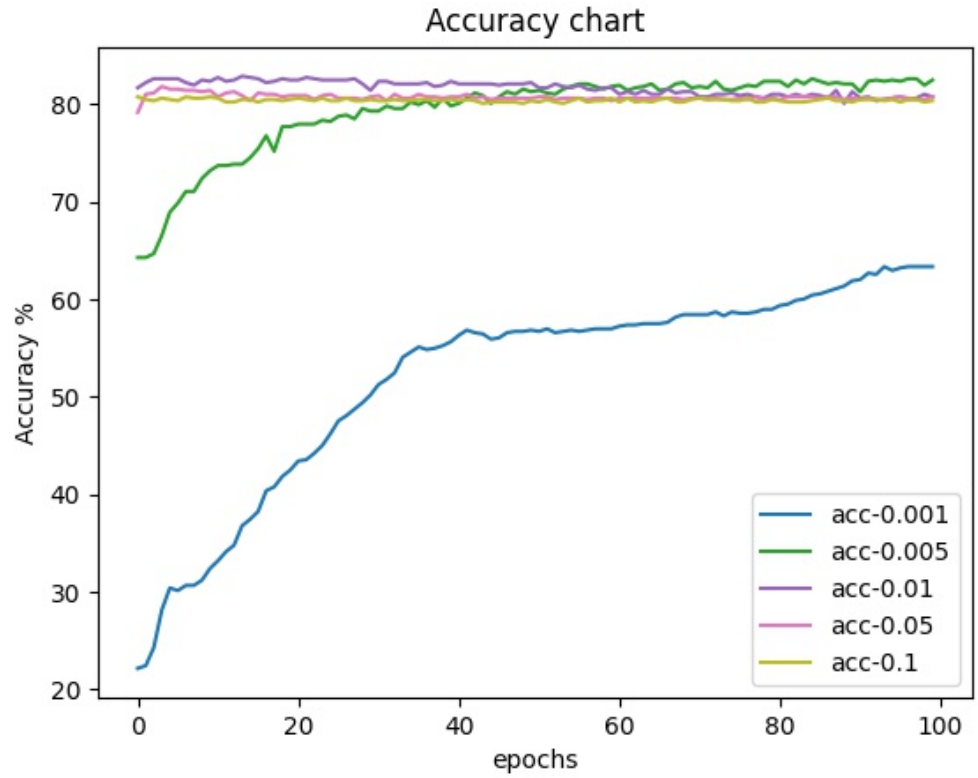
  采用五分类任务，总共训练100个epoch，取learning rate为0.005，得到的准确率为82.4%、83.2%、81.2%、81.0%和80.4%，

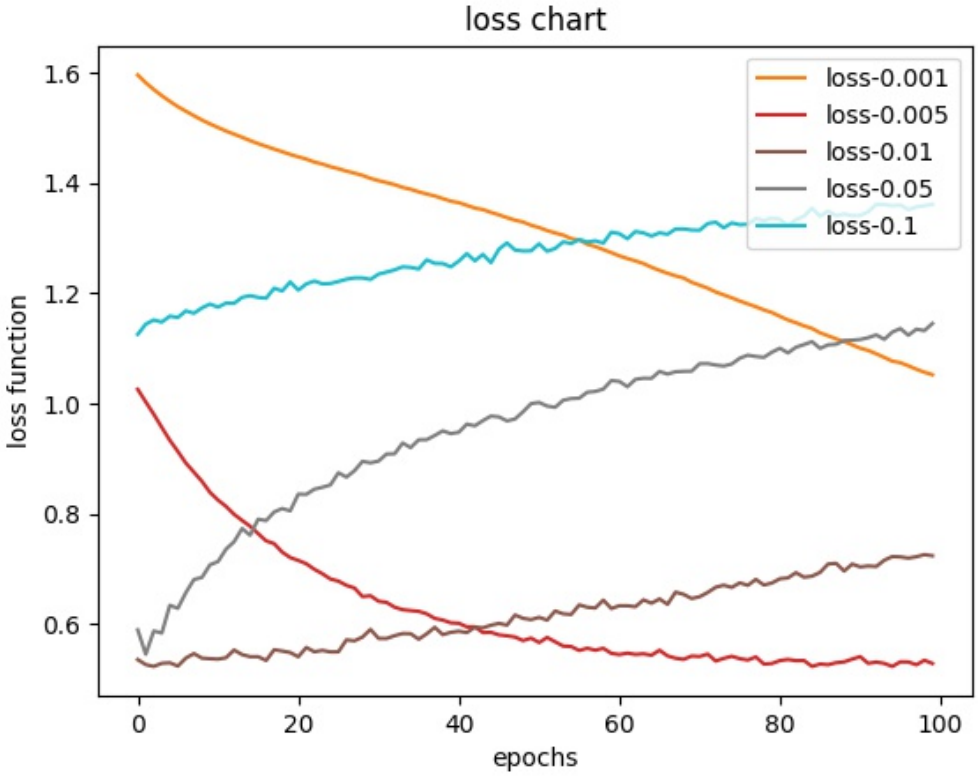
```
Test Error:
 Accuracy: 82.5%, Avg loss: 0.531348

Epoch 22
-------------------------------
loss: 0.119743  [    0/ 3011]
Test Error:
 Accuracy: 81.9%, Avg loss: 0.556799

Epoch 23
```

平均准确率为81.64%,用时2min10s，得到下图的准确率、损失曲线图

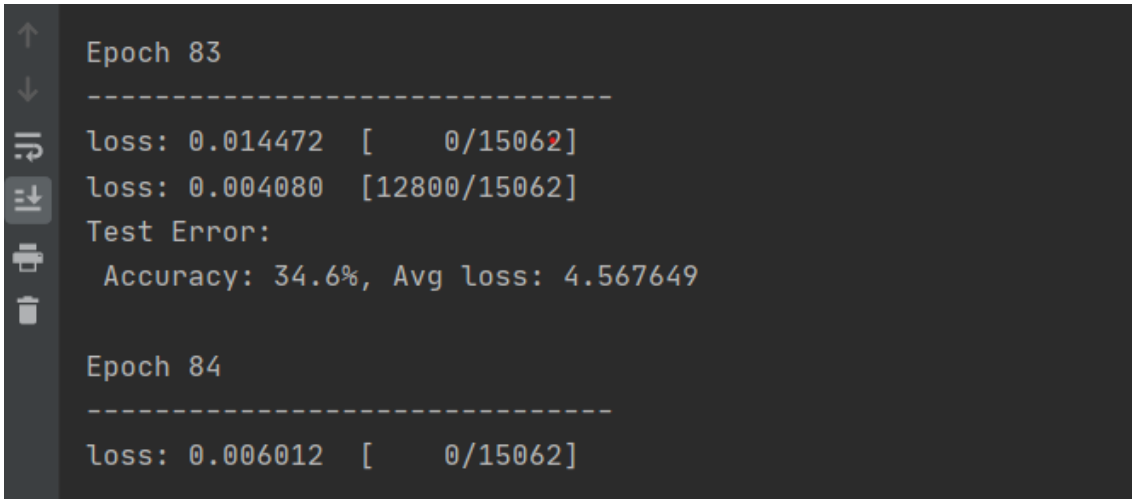


如果改变学习率画出多条曲线，我们可以选出最佳的学习率



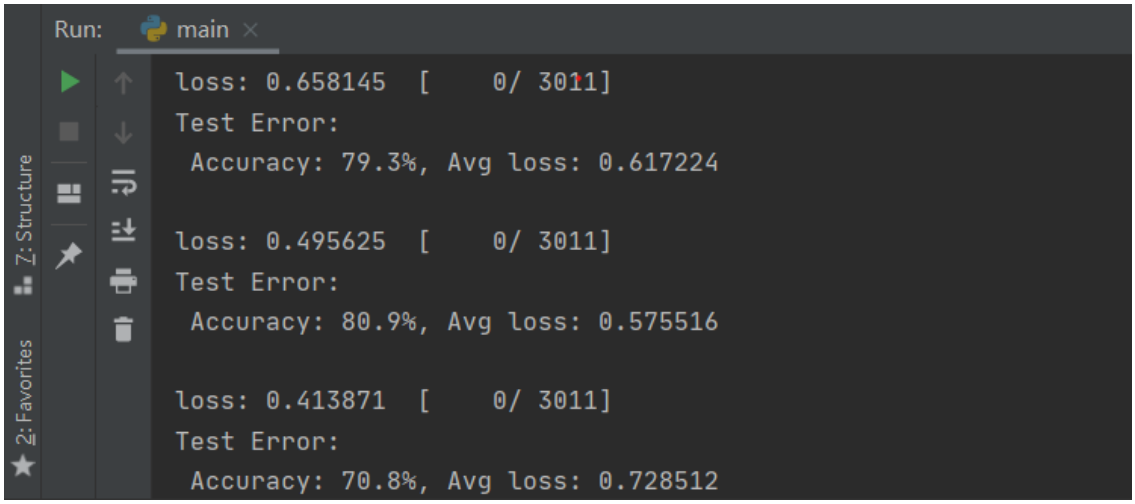


使用20个全部类别进行分类只有34%的准确率



- **RNN**

  使用RNN的LSTM网络进行训练，训练时间较长，但再五分类上的准确率只有80.9%，比采用MLP的五分类任务还要低好多（很迷），应该是我参数没找对，但参数太多确实不想再找了（摆烂）。



实验总结

这次实验采用两种不同的模型，分别进行了五分类任务和20分类任务，感受到了调参的痛苦和人工智能这门学科的玄学，而且正值考研复习阶段就没有花太多时间在调参，随便跑出一个模型结果就ok了。