Assignment4 报告

一、 Requirement 1

本次实验我使用了 The 20 newsgroups text dataset 中全部训练数据并按 9:1 划分训练集和验证集。数据预处理和 assignment-2 一样，将标点符号替换为空格，以空格分割，将所有字母转成小写字母。使用了 fastNLP 的 Dataset 和 Vocabulary 来帮助预处理。数据集如下：

训练样本数：10183
验证样本数：1131
测试样本数：7532
词数：17249
类型数：20

1. 使用 LSTM 进行文本分类

采用和 assignment3 类似的 LSTM 进行文本分类，先对词进行 Embedding，然后对这个序列运行 LSTM，将 LSTM 的结果 $\text{batch sizes} \times \max \text{len} \times \text{hidden size}$ 的第二维求和得到 $\text{batch sizes} \times \text{hidden size}$，然后通过一个全连接层得到$\text{batch sizes} \times \text{class sizes}$的结果，与目标进行求交叉熵。

参数如下：
batch_size=10
n_epochs=20
embedding size=128
hidden size=256

验证集的正确率为 0.854111
测试集的正确率为 0.756506

2. 使用 CNNText 进行文本分类

这边采用和 fastNLP 模型中 CNNText 类似的模型，先对词进行 Embedding，然后对 embedding 的每一维进行卷积处理，对每个 kernel_size=[3,4,5]分别构造[25,25,25]个 Conv1d 卷积层。激励函数采用 ReLU 函数，进行卷积操作之后使用 max_pool1d 的取最大值操作，kernel_size 和卷积处理的 kernel_size 一样。然后对所有的输出进行拼接。由于测试中过拟合严重，再经过一层 dropout 层，最后经过全连接层得到预测值。与目标进行求交叉熵。

参数如下：
batch_size=10
n_epochs=20
embedding size=256
kernel_size=[3,4,5]
kernel_num=[25,25,25]
dropout=0.1

验证集的正确率为 0.83908
测试集的正确率为 0.708311

二、对 fastNLP 的想法

fastNLP 总体来说是一个非常优秀的框架，尤其是文档是中文的对中国人非常友好，并且在详细指南使用一个例子来进行讲解，在跟着指南完成这个任务后能够很快上手这个框架。除了作为一个框架外，各种模型源码写得非常好，具有大量注释和维度标注，具有易读性，很容易能够通过查看源码学习到各种神经网络结构，非常合适学生在这方面学习，可以说是一本带有代码的教科书。

但是在使用过程中有个问题,我在使用 fastNLP 时遇到问题，在没有 Callback 时使用 cuda 时内存不足报错没有提示内存错误而是提示其他的错误，我在自己编写训练过程后才发现这一问题，对第一次学习神经网络新手来说查出这个错误需要较长时间。

```
Traceback (most recent call last):
  File "E:/大三下文档/模式识别与机器学习/assignment-4/source.py", line 191, in <module>
    trainer.train()
  File "D:\anaconda\envs\tfenv\lib\site-packages\fastNLP\core\trainer.py", line 536, in train
    self.tester._format_eval_results(self.best_dev_perf), )
  File "D:\anaconda\envs\tfenv\lib\site-packages\fastNLP\core\tester.py", line 182, in _format_eval_results
    for metric_name, metric_result in results.items():
AttributeError: 'NoneType' object has no attribute 'items'
```

还有 test 过程中没有通过显示测试了多少数据，运行比较慢的机器测试比较大的测试集时要等待很久，建议向训练过程一样显示测试的进度条。