北 京 邮 电 大 学

本科毕业设计（论文）中期进展情况检查表

学院	计算机学院	专业	计算机科学与技术	班级	2014211307
学生姓名	王超	学号	2014211310	班内序号	07
指导教师姓名	刘晓鸿	所在单位	计算机学院	职称	副教授
设计（论文）题目	（中文）基于循环神经网络的影评情感分析系统的设计与实现				
	（英文）Design and implementation of the movie-review sentiment classification system based on RNN				

主要内容: (毕业设计（论文）进展情况，字数一般不少于 1000 字)

一、豆瓣短评爬虫

使用 python Requests 库对豆瓣电影进行爬虫，分析豆瓣网页请求，发现豆瓣分类网页中，后台发来 json response，前端从而进行解析显示。通过对该 url 发送请求来获取豆瓣电影的唯一标记 id 以及 url。

之后，根据电影 id 访问对应电影的短评首页，通过 BeautifulSoup 分析网页抓取短评所在元素以及该短评对应评分，从而根据评分分类 positive 和 negative 标签，将其按评分保存到各个 txt 中。

二、预处理

训练 word2vec 模型需要一个较好的提取语料特征的语料库，考虑到语料库的要求，使用了搜狗实验室提供的 2012 全网新闻数据(1.54G)。

由于该语料库为 XML 格式，并且其中文为 GBK 格式，而非通用的 UTF-8 格式，需要先进行转码及抓取指定元素内容的步骤，本步骤主要使用 iconv 及 grep 命令实现。

之后执行常规分词所要执行的预处理操作，首先过滤常见符号、数字、英文，之后使用 jieba 默认模型进行分词，之后使用该分词后语料库训练 Word2Vec 模型，并将其 vocabulary 及其对应关系模型保存为 numpy array，便于后续处理。得到的 Word2Vec 模型形状为 572297*300。

其次，对训练集进行预处理操作。对训练集过滤非中文字符，使用 jieba 模型进行分词，再根据网上开放的一份较大的停用词表进行去除停用词。

三、LSTM 模型对照实验

RNN 模型能够将信息持久化，能够处理序列数据。但对于长序列数据，会出现梯度下降、梯度爆炸等问题，导致无法得到优良的结果。而 LSTM 的提出解决了 RNN 在长期依赖方面的不足。本次实验主要设置了 LSTM 与传统 RNN 进行比照。

目前已完成任务

在设置循环神经网络模型参数前，为了确定神经网络步长，读取训练集每篇语料词长，以及频次，使用 matplotlib 可视化得知语料内容长度满足正态分布，绝大部分语料长度在 50 以内，因而确定步长为 50。超过 50 的则丢弃，不足则以 0 补齐。

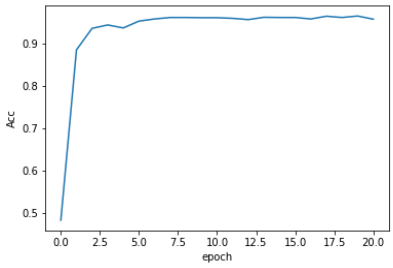
实验中设置的 LSTM 主要为 4 层，输入层、隐藏层、dropout 层和输出层。输入层，即 Embedding 层，设定输入参数句长为 50、向量维度（400）等。第二层，隐藏层，设定 LSTM 单元数为 64。第三层为 dropout 层，减少过拟合。第四层为输出层，即 softmax 层。

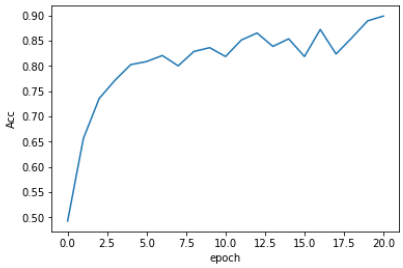
输入，通过 word2vec 模型提取特征矩阵 idx，将语料数值化，每条语料得到一个 50*400 的矩阵。

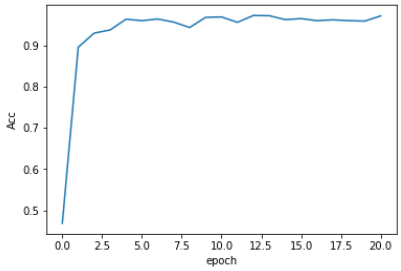
训练集经过 shuffle 打乱，防止模型不稳定，损失函数跳跃严重。

输入向量经过 embedding，馈送到 dynamic_rnn。之后主要采用 tensorflow 框架的 lstm 单元来完成循环神经网络模型的训练。

传统 RNN 模型和 GRU 模型参数设置与上面 LSTM 模型基本相同，将使用的 tensorflow 的 lstm 单元更换成了 rnn 单元，稍稍根据输出矩阵形状调整矩阵，主体与 LSTM 相同，仅设置了一层隐藏层。







四、结果分析

使用 matplotlib 可视化测试集的 accuracy 指标后可以看到，上左图是 LSTM 模型，右图是传统 RNN 模型，下左图是 GRU 模型。在设置了均衡训练集的情况下，三者训练时间相近，在经过 20 轮训练后已经趋于稳定。单层 RNN 的准确率在 81%，2 层的 RNN 准确率反而下降到了 76%，而单层 LSTM 的准确度有 96%，单层 GRU 的准确度高达 97%。可以看到，LSTM 和 GRU 相对传统 RNN 具有极大优势。根据文献得知，GRU 和 LSTM 性能在很多任务上不分伯仲。因此，在长序列文本的情感分析的问题上，将主要基于 LSTM 或 GRU 模型进行优化训练。

五、应用

搭建了一个根据输入评论内容可以返回情感分析结果的网页。前端提交输入的内容后，

<table>
<tr><td></td><td colspan="3">后端调用神经网络模型得出预测值，从而返回情感分析结果。技术上主要使用 python Flask Web 框架构建一个 wsgi 服务器，渲染引擎主要使用 Jinja2 后端渲染。不过由于模型较大，加载模型时间约 5 分钟，单次提交评论测试，响应时间略久，达到了近 10 秒之多。</td></tr>
<tr><td></td><td colspan="3">是否符合任务书要求进度</td></tr>
<tr><td rowspan="2">尚需完成的任务</td><td colspan="3">添加 ROC、F1 等评估标准对不平衡数据测试进行评估。
增加对照模型如 GRU、IndRNN，从而进行优化
完成毕业论文，并准备答辩</td></tr>
<tr><td colspan="3">能否按期完成设计（论文）</td></tr>
<tr><td rowspan="2">存在问题和解决办法</td><td>存在问题</td><td colspan="2">损失函数过拟合问题</td></tr>
<tr><td>拟采取的办法</td><td colspan="2">1. 调整模型参数
2. 增加训练数据
3. 添加 dropout 层</td></tr>
<tr><td colspan="2">指导教师签字</td><td></td><td>日期</td><td>年 月 日</td></tr>
<tr><td colspan="2">检查小组意见</td><td colspan="3">负责人签字： 年 月 日</td></tr>
</table>

注：可根据长度加页。