零．文本分类问题的概述

一．文本获取

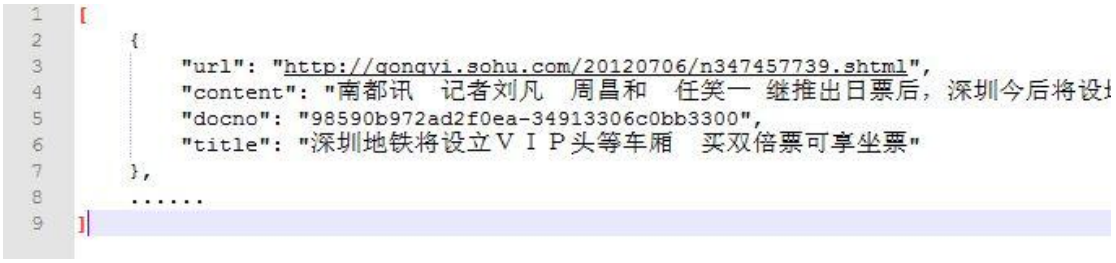
搜狗实验室的搜狐新闻数据

二．文本清洗

2.1 转换编码和格式

下载下来是 xml 格式，GBK 编码，转成 UTF8

然后转换成 json



2.2 类别数据

搜狗新闻的数据没有直接提供分类，而是得通过新闻来源网址的 url 来查其对应得分类，比如 http://gongyi.sohu.com 的 url 前缀对应的新闻类型就是“公益类”。对着他提供的对照表查，1410000+的总数据，成功标出来的有 510000+，标不出来的新闻基本都来自 http://roll.sohu.com，搜狐的滚动新闻

2.3 样本筛选

对成功标出的类剔除样本稀少的类，最终留下 11 个类，每个类抽 2000 个新闻，按 4：1 分成训练与测试

2.4 观察文本长度分布

大量的文本长度分布在 0-200 之间，算是短文本居多

2.5 文本截取

将标题与内容拼合，截取每篇文本的前一百字

2.6 分词清洗

通过 jieba 分词去掉标点符号并分词

最终得到训练文本与训练标签分别 17600 行，测试文本与测试标签分别 4324 行，每行对应一条新闻

三．传统机器学习模型

在文本分类方面表现良好的传统机器学习模型最有代表性的是 SVM 和朴素贝叶斯

四．深度学习模型

4.1 自训练 word embedding

所有新闻中提取到 65604 个词组，使用 keras 的 Tokenizer 将单词转换为索引序列，再使用 pad_sequences 对长度不足的新闻进行填充，同时将标签转换为 onehot 向量

1. CNN

在 embedding 层后加入 dropout 层，（来自知乎，一般用 CNN 时的 dropout 层加在 embedding 后，往往不在卷积层后）

模型的参数根据 A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification 论文给出的一些参数建议来设置，重点有以下参数建议：

* 词嵌入维度： 300 维， 主要针对预训练
* Filter 大小：7 的 filter 最优，不同数据集上最优组合不一致，相差不多
* Filter 的个数：推荐 100 ~ 600 个， 最好靠近 600
* 激活函数：Iden, ReLU， tanh 比 sigmoid,cube 要好， 默认 ReLU 好。
* Pooling 技术：推荐 1-max pooling，不要用 average，效果不好。
* 正则化：dropout rate 不要超过 0.5， l2 正则化效果不明确。

filter 大小和数量可以调一调， 其他调不调意义一般

对数损失，RMSprop 优化器 metrics：accuracy

在 2 个 epoch 上的 test acc 最佳

4.2 预训练 word embedding

五．模型性能综合总结

在最初的 Convolutional Neural Networks for Sentence Classification 论文中，CNN 是基于 Pretrained 的 Word2vec 的结果去做的， 其实效果还没有完全超越 SVM，但是经过尝试，可能由于短文本加数据集不够大，因而自训练效果反而不如预训练。

		Epoch=2			Epoch=4		
		Train-acc	Val-acc	Test-acc	Train-acc	Val-acc	Test-acc
cnn-pre	14s	0.85	0.88	0.82	0.93	0.88	0.82
cnn	45s/epoch	0.89	0.88	0.82	0.98	0.88	0.83
lstm-pre	50s-70s	0.83	0.87	0.83	0.88	0.88	0.84
lstm	110s/epoch	0.74	0.76	0.70-77	0.91	0.86	0.80
mlp	43s	0.98	0.90	0.86	0.99	0.90	0.85
Bys0.01/0.1/1	1s	0.99/8/5		0.85/5/5			
Bys_tf	1s	0.96		0.85			
svm	80s	0.99		0.84			
svm-w2v	150s	0.89		0.81			

Lstm epoch=6 0.96 0.88 0.83 随着 epoch 提升

Lstm-pre epoch=6 0.9 0.9 0.85,epoch=10 后 test acc 到达 0.86

深度学习模型中

Lstm 模型需要更多轮训练（约 6-10 才能达到稳定的效果），而 cnn 只需要 2 轮，但 LSTM 多轮训练后效果稍微强于 cnn，差异并不明显

相比于 lstm,Cnn 对是否预训练的反映并不大

使用预训练的词向量简单平均就能达到更快更优的效果

传统机器学习模型中

贝叶斯模型的表现异常好，时间极短，准确率高

使用预训练词向量的简单平均搭配 svm 效果不佳

Tfidf 特征在多层感知机上效果不错，很可能 tfidf 在 100 长度的中短新闻文本的特征体现能力已经比较强了，词向量可能在更短的文本或者说词义粒度更细致的文本分类中能够发挥更

大的作用
同时，经过测试发现 tf 特征与 tfidf 特征在贝叶斯模型上表现几乎无差别，这说明影响分类的关键词语对逆文档频率并不敏感，也就是说，语言中频率极高和极低的词基本较为中性，对新闻分类类别难以构成影响

通过对几个模型进行 bagging 投票法融合，得到最终准确率为 0.866