







表1 数据集划分表

语料类别	积极	中性	消极
训练集	4820	2613	5222
验证集	1606	871	1740
测试集	1606	871	1741
(书籍、酒店住宿、手机的评论数据) 合计	8033	4355	8703

### 2.3 分词

本文使用结巴库中的 `lcut` 方法对中文句子进行切分, `jieba` 库作为 Python 自带的第三方中文分词库, 功能非常强大, 分词的准确率与速度均高于同类的分词软件。

### 2.4 训练情感词典

通过 Google 提出的 Word2Vec 模型来训练情感词典, 将文本的内容转换为向量来表征<sup>[5]</sup>。利用向量的夹角公式来计算词与词的相似度, 词向量一般是实数值形式的, 表征的范围更大、效率更高。通过 Wikipedia 获取大量的语料进行训练, 通过分词、去停用词、利用 Word2Vec 模型进行特征提取等最终将不同性质的词语归为一类并用高维的实型向量表示<sup>[6]</sup>。

### 2.5 创建当前语料库情感词典

创建单词到索引的映射以及单词到词向量的映射, 若单词的索引数小于 10, 则设置为 0, 若词语的索引为 0, 则词向量也为 0, 若词语的索引不为 0, 则词向量的分量为该词语在情感词典中的出现次数。接着分别获取训练集及其标签、验证集及其标签、测试集及其标签构造的词向量。词汇的维度假设为 100 维, 那么训练集、验证集及测试集的数据维度均为提取的词汇大小 $\times$ 100, 标签数据对应的维度均为词汇大小 $\times$ 3。

### 2.6 构建 LSTM 分类模型

本文通过 Keras 框架构建了一个 LSTM 神经网络模型, 其中模型的基本结构为: 输入层 `InputLayer` 为输入的序列编码成的向量维度, 有向量维度大小的神经元, 隐藏层 `HiddenLayer` 的大小可以自由设置, 这里设置为 64, 输出层 `Softmax` 的大小为情感分类的类别, 这里进行的是三分类, 所以输出层有 3 个神经元, 分别代表积极类、中性类与消极类<sup>[7]</sup>。部分结构代码设置如下:

```
np.concatenate((np.ones(len(pos), dtype=int), np.zeros(len(neu), dtype=int), -1*np.ones(len(neg), dtype=int)))/
*pos 类类别标签为 1, neu 类类别标签为 0, neg 类类别标签为 -1 */
```

```
model.add(LSTM(HIDDEN_LAYER_SIZE, dropout=
```

```
0.2, recurrent_dropout=0.2))
```

```
model.add(Dense(3, activation='softmax'))/
```

\*LSTM 情感分类模型搭建 \*/

构造 LSTM 模型之前, 先做如下预处理:

#### 1) 变长序列的处理

因为每一类别下的数据集中句子的长度都不相等, 所以需要将变长序列转换成定长序列来处理: 首先在所有句子中选取一个最大长度的句子, 这里设定序列最大长度 `maxlen` 为 100, 不足这个长度的序列补 0, 然后在 `embedding` 层中过滤掉指定字符, 通过该层能将序列映射到一个固定维度的空间中, 代码如下:

```
model.add(Embedding(output_dim=vocab_dim,
input_dim=n_symbols,
input_length=input_length))
```

#### 2) 数据归一化

数据归一化的目的是将原本不同量纲的数据经过标准化处理后, 让不同的数据处于同一数量级, 这样各数据指标之间会具有更好的可比性。这里使用 Z-score 标准化方法: 用样本的数据减去均值再除以标准差, 如下所示, 这里仅列举出训练集数据标准化代码:

```
x_train=np.array(x_train, dtype=np.float) #将数据类型转换为 float
```

```
x_train-=np.mean(x_train, axis=1).reshape(16870,1) # zero-center #求出每一行的均值
```

```
x_train /=np.std(x_train, axis=1) # normalize
```

#数据未归一化和归一化后的测试准确率比较如表 2 所示。

表2 归一化前后准确率对比表

	未归一化处理	Z-score 归一化
准确率	0.9187	0.8835

所以, 在模型构造时, 数据不经过归一化处理模型的效果更好。

#### 3) 权重初始化

LSTM 作为神经网络的一种模型, 训练过程中也要及时的对权重进行更新, 在更新前, 需要给每个权重一个初始值。这里有 3 种初始化权重的方式:

第一种方式是把索引为 0 的词语权重初始化为 0, 索引为 1 的词语开始每个词语按照其对应的词向量大小进行初始化。

第二种方式是 Xavier 初始化, 具体方式如下:

```
Weights=np.random.randn(node_in,node_out)/
np.sqrt(node_in)
```

其中 `node_in` 为前一层神经元节点, `node_out` 为后一层神经元节点。

第三种初始化的方式是 He 初始化, 它是在 Xavier 初始化的基础上, 将分母中的方差节点除以 2 即可<sup>[8]</sup>。实验发现, 当使用 He 初始化时, 测试集数据的准确率最高。

## 2.7 LSTM 模型参数选择

当前训练的数据量达上万条, 所以需要将数据集分成多个小块, 实验发现当训练轮数 `epoch` 达到 30 轮之前 `val loss` 不断下降, `test loss` 不断下降; 当轮数超过 30 轮时, `val loss` 与 `test loss` 趋于稳定, 所以 `epoch=30` 为最优训练轮数。

影响 LSTM 模型准确率的因素非常多, 其中就包含各种不同性质的超参数, 比如损失函数、优化器种类、学习率的大小、激活函数的选取等, 这里选取影响 LSTM 模型的主要因素进行分析, 分别如下所示。

### 1) 损失函数 Loss

通过损失函数的连续变化, 使预测数据的真实值和实际值的误差不断减少, 这样就可以对给定的模型进行预测了, 对于多分类来说, 常见的损失函数可以为 `categorical_crossentropy` 或 `mean square error`。

当选用的损失函数为 `mean square error` 时, 测试集的总体准确率最高。

### 2) 学习率 Lr

根据经验, 学习率是用来控制模型的学习进度, 一般设置在 0.001~10 之间时样本训练的效果最好<sup>[9]</sup>。当学习率设置为 0.001, 0.01, 0.1, 1, 10 时, 不同学习率对应的准确率如图 4 所示。

从图 4 可以看出, 当学习率设置为 0.001 时, 模型的训练效果最好。

### 3) 优化器 Optimizer

因为 LSTM 模型可以设定的参数很多, 参数空间比较复杂, 所以优化问题比较困难, 为了寻求最优参数, 在模型编译时选择了不同的优化器进行优化, 其中包括 SGD、Momentum、AdaGrad、Adam 等<sup>[9]</sup>, 其中 SGD 又称随机梯度下降法, 函数会沿着当前位置呈“之”字形往下移动, 对参数更新时使用的学习率相同,

Momentum 借用了物理上“动量”的概念, 就像小球一样在地面上滚动, 它在面对小而连续的梯度时, 学习的更快, AdaGrad 优化器能在学习的过程中不断减少学习率, 并随着学习的进行可以适当的调整, Adam 优化器是将 Momentum 与 AdaGrad 方法融合到一起创建的, 它既可以按照类似小球在碗中滚动的物理规则进行移动, 也可以动态地调整更新过程中的学习率<sup>[9]</sup>。除此之外, 还有 Adamax、Nadam、RMSProp 等优化器。不同优化器准确率如图 5 所示。

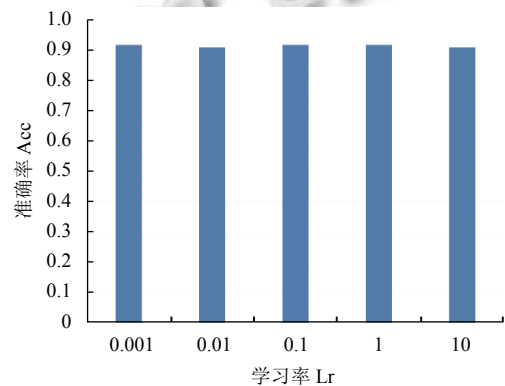


图 4 不同学习率对应的准确率图

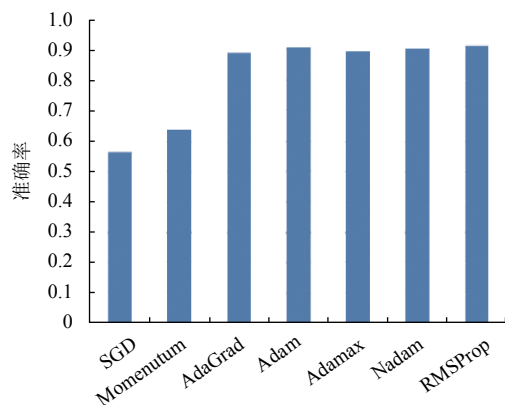


图 5 不同优化器对应的准确率图

可以看出, 当优化器选择 RMSProp 时, 模型训练效果最好。

### 4) 激活函数

激活函数可以将神经网络线性的输入转化到一定范围的非线性值, 从而使神经网络表达能力更加强大, 网络理论上可以逼近任意函数。在本文的多分类情感分析模型中, 主要选择了 3 种激活函数: `tanh`、`relu`、`Softmax`<sup>[10]</sup>, 在训练轮数、学习率、损失函数、优化器

等参数确定的情况下,通过实验可以得出不同激活函数作用下测试集总体的准确率.如表3所示.

表3 不同激活函数作用下的准确率表

激活函数	tanh	relu	Softmax
准确率	0.9228	0.9206	0.9187

从表3可以看出,当激活函数选择 tanh 时测试集的准确率最高.

当然,为了防止过拟合,程序中使用了 Dropout 方法,该方法是在训练过程中随机选出一定比例隐藏层的神经元,将其删除,被删除的神经元不再进行信号的传递,输出的神经元要乘上训练时的删除比例然后再输出.经过多次实验证明,dropout=0.2时,效果最好.

## 2.8 实验环境

本次实验的硬件环境为:CPU为 i7-6700HQ @2.60 GHz,内存为 8 GB,显卡为 GTX965M;软件环境为:Eclipse 作为开发工具,使用 anaconda 自带的 Python 解释器作为 PyDev 的 Interpreter,使用 keras 框架来实现 LSTM 神经网络模型,使用 Python 的工具包 gensim 来训练 Word2Vec 模型,使用 matplotlib 库来绘制本实验中相关的图形.

## 2.9 实验结果

经过反复实验测试,当数据没有归一化、使用 He 初始化权重、学习率为 0.001、损失函数为 Mean Square Error(简称 MSE)、优化器选择 RMSProp、激活函数选择 tanh、训练轮数达到 30 轮时,模型总体测试集的准确率最高,这些参数组合到一起时训练的效果最好.这里采用准确率召回率和 f1 值进行评估<sup>[11]</sup>.此时,模型训练的准确率同 SVM 多分类方法比较如表4所示.

表4 不同方法性能对照表

性能	Word2Vec+SVM 方法	Word2Vec+LSTM 方法
准确率	0.823	0.9228
召回率	0.86	0.9002
f1 值	0.8412	0.9114

其中,SVM 方法先通过 PCA 方法将高维的特征向量降低到 100 维,对于三个类别,先将其转换成两个类别的分类问题,然后采取投票的方式,确定最终的类别.算法大致如下:

1) 将正类中性类负类的测试集票数都设置为 0.

2) 构造正类-负类分类器如果当前测试集数据判定的类别为正类,则正类票数加 1,否则负类票数加 1.

3) 接着构造正类-中性类分类器、中性类-负类分类器,按类似方式对测试的数据集进行投票.

4) 最终正类、中性类和负类中票数最多的那一类就是当前测试数据所属的类别.

验证集与测试集各自的损失函数值随训练轮数变化的曲线图如图6所示.

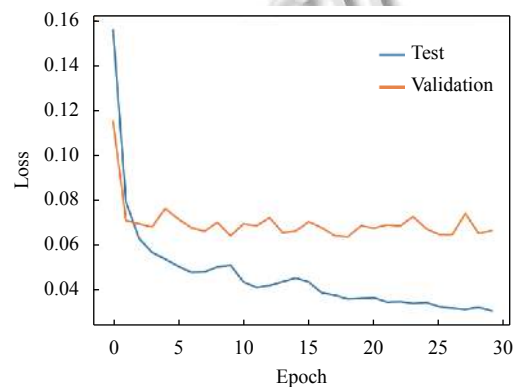


图6 验证集与测试集损失函数值随训练轮数变化图

随着训练轮数的增加,验证集与测试集的准确率不断接近,没有过拟合现象发生,验证集和测试集的准确率随着轮数的变化曲线图如图7所示.

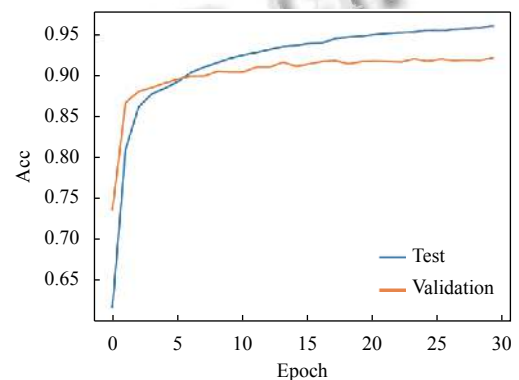


图7 验证集和测试集准确率随训练轮数变化图

将 model.compile() 函数中的 metrics 参数的值分别修改为 recall 和 f1,再分别定义出召回率 recall、精确率 precision 及 f1 三个函数,通过 30 个 epoch 的迭代计算,就可以得到验证集与测试集在每一轮中的召回率与 f1 值随训练轮数的变化结果.如图8和图9所示.

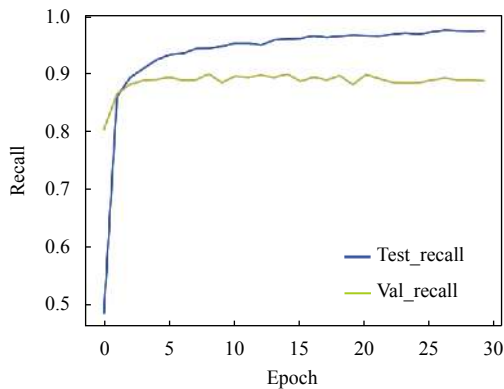


图8 验证集和测试集召回率随训练轮数变化图

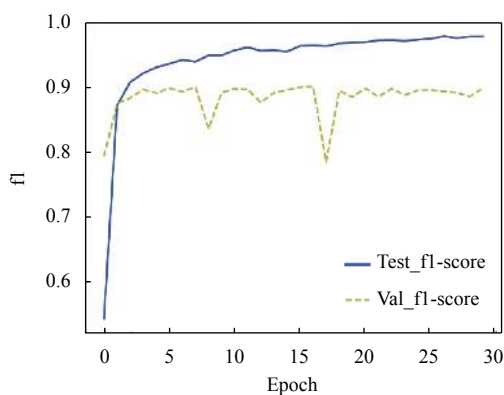


图9 验证集和测试集 f1 值随训练轮数变化图

### 3 总结

情感分类问题是一个在很多领域都很常见的问题,除本文用到的评论数据外,网上随处可见的一些新闻评论、微博评论、贴吧评论、qq 空间评论,还有歌曲评论、网络电影评论等,使用传统的方法,比如 SVM、KNN 来分类,不但准确率提升的幅度有限,而且需要人工提取特征向量,效率很低。

本文使用深度学习中的 LSTM 模型,同时结合 Word2Vec 工具创建出当前领域范围的情感词典,然后将语料中的词语用高维的词向量表示,接着用一个 Embedding 层将词向量嵌入到 LSTM 模型的隐层中,最后连上一个 Softmax 输出层,这样就建立了一个多

分类的情感分类器。影响神经网络模型精度的参数非常多,本文选取了学习率、损失函数、优化器、激活函数等为主要评价指标,分析了它们对当前模型准确率的影响,最终确定了一组最优的参数。实验证明,该方法能够在一定规模的数据集上有效地解决多类别情感分类问题!

### 参考文献

- 1 蓝雯飞,徐蔚,汪敦志,等.基于 LSTM-Attention 的中文新闻文本分类.中南民族大学学报(自然科学版),2018,37(3): 129-133.
- 2 赵勤鲁,蔡晓东,李波,等.基于 LSTM-Attention 神经网络的文本特征提取方法.现代电子技术,2018,41(8): 167-170. [doi: 10.16652/j.issn.1004-373x.2018.08.041]
- 3 石逸轩.基于深度学习的文本分类技术研究[硕士学位论文].北京:北京邮电大学,2018.
- 4 伊恩·古德费洛,约书亚·本吉奥,亚伦·库维尔.深度学习.赵申剑,黎彧君,符天凡,等译.北京:人民邮电出版社,2017. 187-189.
- 5 陈楠,陈进才,卢萍.基于深度学习的多元文本情感研究与分析.计算机科学与应用,2018,8(5): 669-686. [doi: 10.12677/csa.2018.85076]
- 6 马远浩,曾卫明,石玉虎,等.基于加权词向量和 LSTM-CNN 的微博文本分类研究.现代计算机,2018,(25): 18-22. [doi: 10.3969/j.issn.1007-1423.2018.25.004]
- 7 Greff K, Srivastava RK, Koutnik J, et al. LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(10): 2222-2232. [doi: 10.1109/TNNLS.2016.2582924]
- 8 李杰,李欢.基于深度学习的短文本评论产品特征提取及情感分类研究.情报理论与实践,2018,41(2): 143-148.
- 9 [日]斋藤康毅.深度学习入门:基于 Python 的理论与实现.陆宇杰,译.北京:人民邮电出版社,2018. 164-183.
- 10 王伟,孙玉霞,齐庆杰,等.基于 BiGRU-Attention 神经网络的文本情感分类模型.计算机应用研究. <https://www.cnki.net/KCMS/detail/51.1196.TP.20181011.1246.010.html>. [2018-10-15].
- 11 李妍坊,许歆艺,刘功申.面向情感倾向性识别的特征分析研究.计算机技术与发展,2014,24(9): 33-36.