计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(1):130−136 [doi: 10.15888/j.cnki.csa.007227]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

Word2Vec+LSTM 多类别情感分类算法优化①

邬明强[1], 邬佳明[2], 辛伟彬[1]

[1](广东东软学院 数字媒体与设计学院, 佛山 528200)
[2](开封大学 软件职业技术学院, 开封 475004)
通讯作者: 辛伟彬, E-mail: xinweibin@nuit.edu.cn

摘　要: 随着网民的数量不断增加, 用户上网产生的数据量也在成倍增多, 随处可见各种各样的评论数据, 所以构建一种高效的情感分类模型就非常有必要. 本文结合 Word2Vec 与 LSTM 神经网络构建了一种三分类的情感分类模型: 首先用 Word2Vec 词向量模型训练出情感词典, 然后利用情感词典为当前训练集数据构建出词向量, 之后用影响 LSTM 神经网络模型精度的主要参数来进行训练. 实验发现: 当数据不进行归一化, 使用 He 初始化权重, 学习率为 0.001, 损失函数选择均方误差, 使用 RMSProp 优化器, 同时用 tanh 函数作为激活函数时, 测试集的总体准确率达到了 92.28%. 与传统的 Word2Vec+SVM 方法相比, 准确率提高了大约 10%, 情感分类的效果有了明显的提升, 为 LSTM 模型的情感分类问题提供了新的思路.

关键词: Word2Vec; LSTM; 情感分类; 学习率; 损失函数; 激活函数

引用格式: 邬明强,邬佳明,辛伟彬.Word2Vec+LSTM 多类别情感分类算法优化.计算机系统应用,2020,29(1):130–136. http://www.c-s-a.org.cn/1003-3254/7227.html

Optimization of Word2Vec and LSTM Multi-Category Sentiment Classification Algorithm

WU Ming-Qiang[1], WU Jia-Ming[2], XIN Wei-Bin[1]

[1](Digital Media and Design Academy, Neusoft Institute Guangdong, Foshan 528200, China)
[2](Software Vocational and Technical College, Kaifeng University, Kaifeng 475004, China)

Abstract: With the increasing number of netizens, the users on the Internet has doubled, and a variety of comment data can be seen everywhere. So, it is very necessary to construct an efficient emotional classification model. This study combined Word2Vec with LSTM neural network to construct a three-class emotional classification model. Firstly, Word2Vec word vector model is used to train the emotion dictionary. Then, we construct word vectors for the current training set data by using emotional dictionary. Then, this study used the main parameters that affecting the accuracy of LSTM neural network model to train the model. The experiment found that when the data are not normalized, using the weight of He is initialized, the learning rate is 0.001, the loss function is mean square error, the RMSProp optimizer is used, the training rounds are 30, and the accuracy of traditional Word2Vec + SVM method improves by about 10%. The effect of affective classification promotes obviously, which provides a new way of thinking for LSTM model's sentiment classification.

Key words: Word2Vec; LSTM; sentiment classfication; learining rate; loss function; activation function

进入大数据时代、互联网+时代, 数据呈现指数级增长. 微博的社交关系, 淘宝的购物记录, 不同类型的新闻信息, 诸如娱乐、军事、体育等这些形形色色的数据包括了人们的各种行为活动的细节[1]. 据统计,

① 基金项目: 佛山市科技创新项目 (2017AG100132)
Foundation item: Science and Technology Innovation Project of Foshan (2017AG100132)
收稿时间: 2019-05-15; 修改时间: 2019-06-21, 2019-07-08; 采用时间: 2019-07-15; csa 在线出版时间: 2019-12-27

1998 年 3 月, 我国第一笔互联网网上交易成功, 标志着网上购物在中国的兴起; 2001 年底, 我国互联网用户数增长为 3370 万, 网上购物的交易额仅为 6 亿元; 2010 年中国网络购物交易规模就已经达到了 5000 亿. 线下没有的, 照样可以网购到, 而在琳琅满目的物品中挑选出合适物品的一个很重要途径的就是看评价. 因为其他顾客购买该物品后, 一般会对产品进行评论, 所以只要他或她挑选出了中意的商品, 再结合评论区的评论就可以很容易的对所购的商品进行更全面的了解.

同时, 随着智能手机的快速普及, 一些饮食类 APP, 如美团、饿了么应运而生, 住宿类 APP 如去哪儿网、携程网等人们已司空见惯, 面对众多的餐馆、鱼龙混杂的酒店、宾馆, 用户如何根据自己的需求选择一个性价比最优的呢? 显然最好的方式是查看大量已有的评价数据. 本文正是在此背景下从众多的数据集中选取了一定数量的其中 3 类数据集, 借助计算机信息技术的优势对已有的文本进行处理, 进而构造一个可靠的情感分类系统, 通过情感分类算法对大量的情感分类数据进行训练构造出相应的模型, 一旦有新的评价数据到来, 只需要导入该系统就可以快速地完成情感的分析分类.

1 Word2Vec+LSTM 情感分类方法

1.1 情感分类方法

1.1.1 情感分类方法介绍

常见的情感分类方法有五大类: 基于词典的方法、基于机器学习的方法、词典与机器学习相结合的方法、基于弱标注信息的方法以及基于深度学习的方法. 其中基于词典方法的模式是“词典+规则”, 即以情感词典作为判断情感极性的主要依据, 同时兼顾评论数据中的句法结构, 设计相应的判断规则; 基于机器学习的方法主要依赖特征工程, 通过 n-gram 方法、Part-of-Speech 方法、句法特征方法、TF-IDF 方法提取出一定数量的满足条件的特征, 再放到情感词典中进行匹配, 进而推断出情感的性质; 将词典与机器学习融合起来的方法可以将“词典+规则”视为简单的分类器或者将词典信息作为特征与现有特征进行结合, 选择最优的特征组合进行情感分类; 基于弱标注信息的情感分类方法试图从用户产生的数据中挖掘有助于训练情感分类器的信息, 比如评论评分、微博表情符号等, 加入情感词典进行分类.

前面几种方法都属于传统的文本情感分类方法, 在文本分类的表现上仍然有很多不足的地方, 因为在分类过程中需要加入很多人为因素来对某种场景下的情感进行分类, 尤其在语料的训练和情感词典的建立方面体现的更明显, 如果不是语言学家或者拥有浓厚的语言背景知识, 那么整个情感分类的结果就会受到很大的影响. 而基于深度学习的情感分类方法克服了传统方法的不足, 首先要求从大量评论数据中学习出语义词向量, 然后通过不同的语义合成方法用词向量得到对应句子或文档的特征表达, 再通过构建的神经网络模型导入相关的句向量就可以让模型自动通过前向传播算法和反向传播算法不断对参数进行迭代更新以减少 loss 函数值来提升分类准确率, 这个过程中我们只需要设置一些参数即可, 包括输入层数量、隐藏层数量、各层之间的权重、隐藏层及之前需要的偏置值、学习率、激活函数等就可以训练出一个情感分类的模型.

1.1.2 深度学习情感分类方法发展

传统的情感分类方法包括朴素贝叶斯方法、最近邻方法、支持向量机方法等, 准确率受到了很大限制, 随着深度学习的兴起以及数据处理能力的不断提升, 深度学习技术广泛用于图像处理、语音识别、自然语言处理等方面. RNN 循环神经网络方法作为自然语言处理的一种方法因为不能记住太前或太后的内容, 所以仍然有一定的局限性, 而 LSTM 神经网络 (Long Short Term Memory) 作为 RNN 网络的一种变种, 它通过在普通的 RNN 基础上, 在隐藏层各神经元中增加记忆单元, 从而使时间序列上的记忆信息可控, 每次在隐藏层各个单元间传递时通过几个可控门 (遗忘门、输入门、候选门、输出门), 来控制之前信息和当前信息的记忆和遗忘程度, 从而具有长期记忆功能[2]. 本文采用的正是 Word2Vec 与 LSTM 相结合的情感分析方法.

1.2 关键技术

1.2.1 Word2Vec

Word2Vec 摒弃了传统方法的 one-hot 编码方式, 将一个词语对应一个多维向量, 通过该多维向量允许我们用变化较小的数字来表征词语, 如果用 20 维向量理论上就可以表征 2^{20}=1048 576 个词语了, 通过一定方法, 比如欧氏距离或者余弦相似度就可以把相近意思的词语放在相近的位置, 而且一般用的是实数向量[3]. 通过将所有的词向量化, 词与词之间就可以定量的度

量他们之间的关系.

1.2.2 LSTM 简介

LSTM, 是为了解决长期以来问题而专门设计出来的它是在 RNN 的基础上衍生来的, 当相关的信息和要预测的词的位置之间的间隔很小时, RNN 可以学会使用先前的信息, 但是当相关信息和当前预测位置相隔较远时, RNN 会丧失学习到很远信息的能力. LSTM 通过刻意的设计来避免长期依赖问题. 记住长期的信息在实践中是 LSTM 的默认行为, LSTM 重复模块中包含四个交互的层. 其中遗忘门决定我们从细胞状态中丢弃什么信息, 输入门决定多少新信息加入到细胞状态中来, 输出门决定输出什么值[4]. LSTM 结构图如图 1 所示.

1.2.3 LSTM 情感分类处理流程

先使用 Word2Vec 模型将词向量的集合对应成句子, 这样就能够得到表征该句子的句向量, 然后将句向量作为神经网络模型的输入部分, 向量的每个分量对应神经网络的输入层节点, 隐藏层可以根据经验进行设置, 输出层的节点个数就是我们要分类的类别数. 情感分类流程图如图 2 所示.

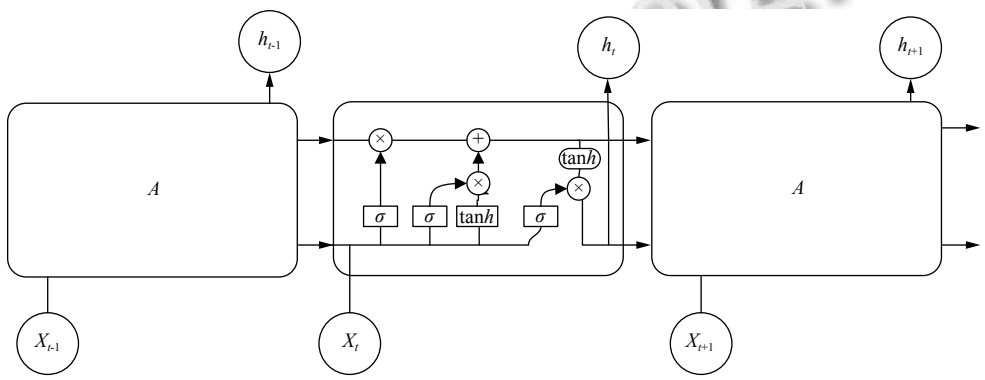

图 1 LSTM 结构图

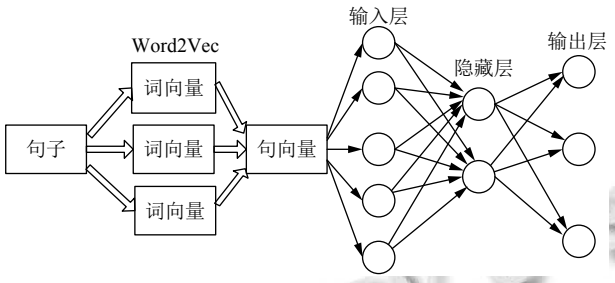

图 2 情感分类流程图

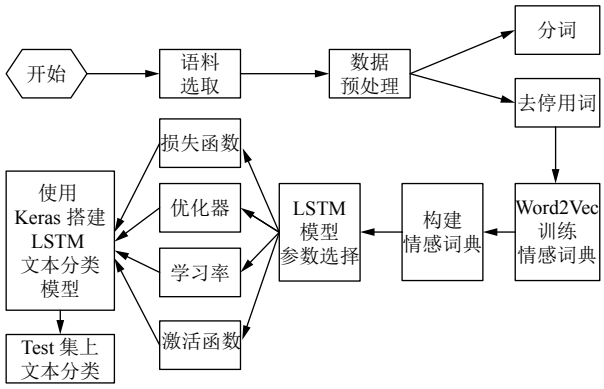

图 3 本文文本分类处理流程图

2 实验流程分析与参数设置

2.1 本文分类流程

选取了一定数量的语料后, 就进入了数据预处理阶段, 包括分词、去停用词, 接着使用 Word2Vec 模型训练出一定数量的情感词典, 然后使用 Keras 模型搭建出一个 LSTM 结构, 设置好网络的输入层、隐藏层、输出层, 然后进行调优, 包括设置学习率、激活函数、损失函数, 选取优化器等. 实验处理的流程如图 3 所示.

2.2 语料选取

本文分别选取了网上关于书籍、酒店住宿、手机的评论数据作为实验用的语料, 一共 21 090 条, 每一类数量均为 7030. 现从每一类随机抽取一些数据集, 包括积极、中性、消极 3 类, 每一类的数据集均划分成训练集、验证集、测试集 3 部分, 比例都为 6:2:2, 数据存储在 csv 格式的数据文件中. 各部分数据的数量如表 1 所示.

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[5]. 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[6].

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×100, 标签数据对应的维度均为词汇大小×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 HidenLayer 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[7]. 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[8]. 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[9]. 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[9], 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[9]. 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.

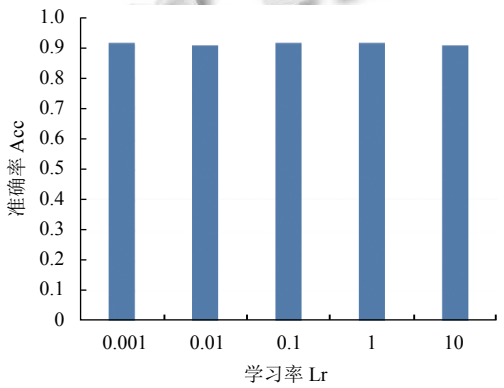

图 4 不同学习率对应的准确率图

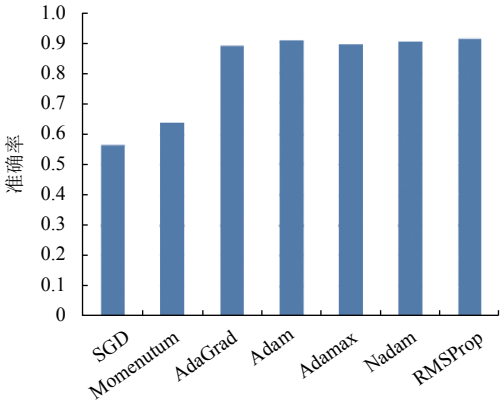

图 5 不同优化器对应的准确率图

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[10], 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 Intepreter, 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[11]. 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.

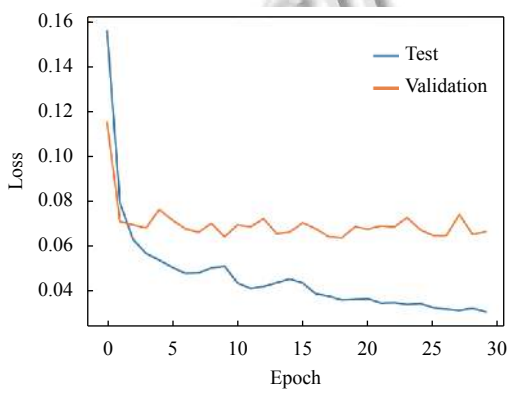

图 6 验证集与测试集损失函数值随训练轮数变化图

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.

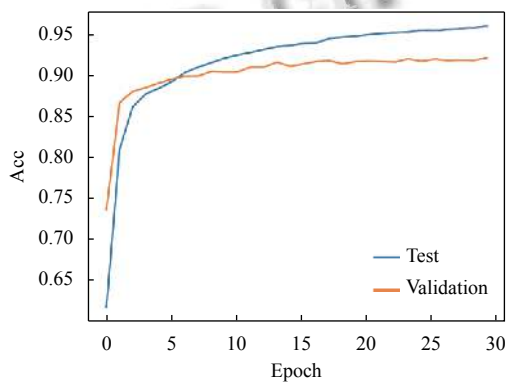

图 7 验证集和测试集准确率随训练轮数变化图

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.

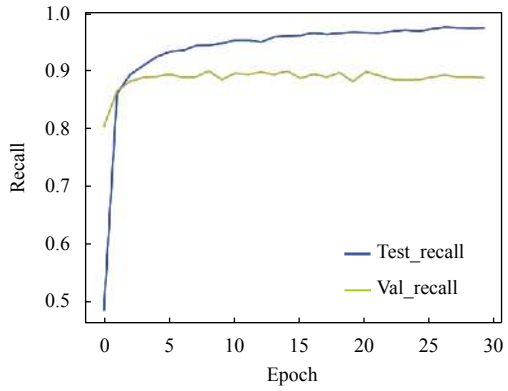

图 8 验证集和测试集召回率随训练轮数变化图

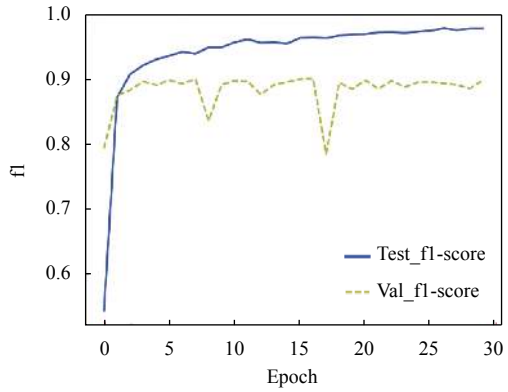

图 9 验证集和测试集 f1 值随训练轮数变化图

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.