基于反馈的深度学习方法在情感分析中的应用

学生：唐东格

指导老师：荣文戈

摘要

情感分析是当下十分热门的一个研究领域，包括文本情感分析，音频情感分析和视频情感分析等，虽然类别很多，但是研究方法却有很多相通之处，本文主要研究文本情感分析。

随着网络信息的发展，网上的信息量越来越大，我们经常需要对一些文本信息进行统计分析，文本情感分析就是其中十分重要的一类。显然面对上百万的文本信息如果采用人工处理的方法是不可行的，因此从上世纪九十年代以来，用机器学习的方法进行情感分析一直是一个热门课题。

从最初的传统的机器学习方法，到现在的深度学习方法，一代代的研究学者们构建了各种语言模型，使得情感分析的正确率在一步步地提升。本文主要研究基于反馈的深度学习方法应用于文本情感分析，主要包括RNN（也就是循环神经网络）和LSTM（也就是长短时间记忆模型），实验表明基于词向量的长短时间记忆模型对大语料的训练具有很好的效果。

关键词：情感分析，深度学习，反馈

Deep learning based on feedback in sentiment analysis

Sentiment analysis is a very popular field of research, including sentiment analysis of text, sentiment analysis of audio, video analysis and so on. Though there are many categories in sentiment analysis, the method of research is similar. This paper surveys the sentiment analysis of text.

With the development of network information, the amount of information online becomes larger and larger. we often need some text information for statistical analysis, sentiment analysis is one of a very important class. Clearly, it’s impossible for people to handle millions of text messages one by one. Therefore, since the 1990s, using the method of machine learning in sentiment analysis has been a hot topic.

In the initial stage, it’s the traditional machine learning method, and now it’s the deep learning that appeals to the people. many researchers have constructed a mass of models in languages, and the accuracy becomes higher and higher. This paper surveys Deep learning based on feedback in sentiment analysis, including RNN (i.e. Recurrent Neural Network) and LSTM (i.e. Long Short Term Memory). Experiment result shows that the model LSTM has a better performance in sentiment analysis.

Key words: sentiment analysis, deep learning, feedback

# 1 绪论

## 情感分析课题的研究背景和研究目的

自从计算机发明以来，人类社会发展日新月异，各种新科技、理论不断涌现，给人们的物质文化生活都带来了极大的便利，尤其是进入21世纪之后，电脑普及到千家万户，手机等数码产品成为每个人身边都必不可少的物品。我们每天都可以发现千万级的信息更新，从最简单的生活琐事到轰动世界的科学大事，这都说明了如今是一个信息大爆炸的时代。情感分析是NLP（也就是自然语言处理）中的一种方法，NLP是一个十分复杂的领域，也是当今计算机科学领域十分热门的一个课题，NLP设计的学科十分广泛，包括计算机学、心理学、语言学、逻辑学等，NLP的研究成果现在已经广泛应用于机器翻译、语音识别、人工智能、模式识别等领域，取得了很大的进步。文本情感分析，简单点来说，就是判断一个句子要表达的意思是积极的还是消极的；当然，现在对它的研究不止于此，我们不仅可以辨别一句话是积极还是消极，还可以对此情绪进行分级，可以对一句话进行喜、怒、哀、乐的划分，得出更细致的文本处理结果。

对文本进行情感分析具有重要的意义，面对网上数以万计的信息，商家希望统计出大家的喜好倾向，以便做出更好的商业策略调整，以期取得更好的商业收益；面对一个电影的上万条评价， 显然无法一条条地进行阅读并统计评论的情感倾向，因为那样不仅费时费力，还会造成资源和金钱的浪费，而基于机器学习的情感分析成果就提供了一个简单、快捷的处理方法。到目前为止，为大家所认可的文本情感分析的研究开始时间是在2002年，当时pang提出了对文本进行处理的N元语法等著名论点[1]。目前情感分析的应用领域绝不仅局限于文本，许多学者在音频、视频、图像的处理上也获得了很好的成果，而且他们之间的方法是相通的，这也体现了NLP应用的广泛性。传统的通过调查问卷进行客户喜好研究分析的调查方式早已经被人们所抛弃，基于网上的信息评论不仅可以得到更大的调研数据，同时这些数据的获取也是相当的方便，对这些数据进行又快又好的处理就是情感分析的研究内容；情感分析当然并不局限在对客户的评论进行统计分析上，比如垃圾邮件的识别与自动拦截丢进垃圾箱也是它的研究成果，来自各种渠道的广告基本上每天都在向我们的邮箱发送垃圾邮件，客服人员永远不可能人工处理完那么多邮件，情感分析就可以自动地帮助他们进行邮件的归档处理。由于在预测进行信息过滤、市场走向、进行民意统计调查分析等方面具有极其重要的作用，文本倾向分析的研究获得了很大的商业利益的驱动，再加上世界各地学者孜孜不倦地工作，该领域表现出十分广阔的发展前景。

## 文本情感分析研究的一般步骤

根据大量专家学者的研究工作，目前一般认为进行文本情感分析需要四个方面的处理步骤，包括进行语料的选择、对我们要分析的文本进行预处理、对文本特征进行标注和选择、对文本进行情感分类[2]。

### 进行语料选择

语料就是一些研究人员收集的来源于真实语言的整合起来的文本。语料库因为来源于真实生活，所以具有很高的研究价值，语料库是进行语言研究的可利用的基本资源。由于各个地区的语言千差万别，而且即便是同一个地区也存在着不同的方言，所以针对每一种语言都可以建立一个语料库[3]，语料库的种类自然会很多，目前中国学者普遍研究的语料库就是中文和英文。除了地区的差别就之外，语料库也有着学科的差别，不同的学科的术语是不同的，不同的词语在不同的学科中出现的频率也是千差万别的，所以研究不同的学科利用的语料库也应该是不同的，这样才能更准确地对我们要研究的对象进行正确的评估分析。此外，不同的语料库的语料规模也是不同的，目前最大的语料库是谷歌的新闻数据库，大约有一千亿个单词，一些小的语料库仅有几万个单词，在一定意义上来说，语料库的大小对实验结果会产生影响，一般语料库越大包含的信息就越丰富，那么我们进行文本处理得到的结果就会越准确。目前，网上有各种各样的语料库供大家选择利用，我们实验选择的是文本8（一个100M大小的语料）。

### 1.2.2对我们要分析的文本进行预处理

对文本进行预处理是文本情感分析中十分重要的一部，一方面预处理能够降低原来的中的噪点（即一些来自文本本身的错误），提高文本分析的正确率；另一方面对文本进行预处理之后能够进一步构建深度学习模型，是进行文本深度学习训练的必不可少的一步。

对我们要分析的文本进行预处理一般需要进行一下几步：

首先，我们得到的直接的数据文本是来自网上的一些爬虫处理的文本，我们需要把所有的小文本整合到一个大文本中以便于我们对文本进行遍历处理。

其次，我们要将我们的文档进行切分成一个一个的句子，然后对每一个句子进行分词处理，文本分词以后可以得到一个包含文本中所有单词的字典，这在之后的模型构建中是非常重要的。为了排除标点符号对实验结果的干扰，我们还要对文本进行符号处理，把逗号、句号、省略号、制表符、换行符等一些常见的标点符号去掉，对于原文中由于乱码等产生的特殊符号，我们也要进行过滤处理。

由于我们处理的是英文文本，所以接下来我们还要对原文中的单词进行拼写方面的检查，拼写检查也是为了降低数据本身产生的噪声，使我们的训练结果更加精确。

我们还有进行大写字母小写化。在文本处理的过程中，我们不想区别单词的大小写，因为单词的大小写在大部分情况下只和单词出现的位置有关，和该单词要表达的意思并没有什么特别的联系，所以我们要把所有的单词都变成小写。

最后，我们把文本中经常出现的一些词根据停用词表进行删除，因为停用词在各种句子中都经常出现，增加了语句的复杂度，但对于我们进行模型训练并没有太大的影响，有时候反而会造成不必要的干扰。我们还要把一些词进行还原，比如has、haven、had、have 统一返回have，是每一个单词的意思不会因为词形的变化受到影响。

经过以上这些步骤，我们就会得到一个较为干净、明了、易于处理的文本了，文本中每两个词之间都以空格相间，接下来我们就可以利用我们得到的文本进行模型构建了。

### 1.2.3对文本特征进行标注和选择

对本文进行特征标注就是对文本中出现的词语的情感进行做标记，通常分为中性词、褒义词、贬义词，当然也可以示研究内容和目的采用其他的极性标注方法。目前常用的标注方法可以两种，一种是利用机器进行自动标注，另一种是采用人工的方法进行标注。用机器进行自动标注时最简单的就是利用一些关键词进行标注，我们可以把文本中出现的词语与标准情感词典（常用的词典是HowNet和WordNet等）进行比较检索，此外在自动标注中常用的方法还有根据情感词的频率进行标注，在实际运用中目前是把这几种自动标注的方法融合运用的。人工标注就是研究者认真阅读每一个句子，然后人工对这些句子中的词语进行标注。采用机器方法进行标注的优点是简单易行、操作迅速，但是标注的准确度有待提高；人工进行标注的优点是标注的准确度很高，标注内容也较为全面，但是缺点也是显而易见：人工标注太过费时费力，面对百万级的语言文本，其工作量可想而知。因此，现在研究者们所采用的一般是在自动标注的基础上进行人工辅助标注，这样既能在一定程度上提高标注效率，又能在一定程度上调高标注的准确性。

在对文本特征进行选择的时候，目前的结构和方法有很多。经常用到的一些特征比如：文本中的一些关键词、出现频率比较高的词、表示情感的词、否定词、主观词等都是进行特征选择时常用到的选择特征。特征选择一方面集中了文本信息，相当于对噪音进行了有效地过滤，另一方面通过特征选择的方式，我们降低了要研究的文本的大小，在空间上降低了模型的维度，降低了计算的时间和复杂度；但是另一方面在对文本进行特征选择的时候，也人为地减少了一部分原有的信息，如果特征选择的不好的话，就会造成信息丢失，对最后的模型构建和情感分析造成影响，降低我们分析的正确率。目前常用到的进行特征提取的算法有很多，从前人的研究成果来看，主要可以分为三类：第一类是Embed（也就是嵌入式方法），之所以成为是嵌入式，主要原因是选择过程的算法被嵌入了学习的算法之中，比如决策时方法就是最经典的该类算法。第二类进行特征选择的算法是Wrapper（也就是封装式方法），封装式特征选择算法相对于其他的特征选择算法的效果更好一些，但是这种算法相对于其他算法也更为复杂，而且算法学习效率也比较低，典型的SBS（序列后向选择算法）就是它的代表，在进行特征评价时，该算法选择的评价标准是利用交叉验证的准确度。第三类进行特征选择的方法是Filter（也就是过滤式方法），该方法中比较常用的算法包括：DF（也就是文档频率方法），DF方法提出的模型是，在文档中，出现频率比较低的词，对文本信息的贡献率比较小，在处理的工程中就可以忽略这种低频度的特征词，从而过滤掉一些无用的信息，提取出来比较主要的特征，但是DF方法遗漏了特征词在特定文本中频度可能很高的情况，所以DF一般适用于规模比较大的文本处理，对于短数据集的效果并不理想；CHI（也就是卡方检验方法），该算法利用统计学原理中的地独立性检验的方法，判断文本中的特征词与每个类别的独立程度，一次来确定是将该特征进行选择还是舍弃；IG（也就是信息增益方法），是一种以信息熵为基础的模型，该模型通过计算信息熵增益，比较每一种特征对文本分类的贡献程度，然后按照由大到小的顺序进行特征选择，信息增益比较小的特征将会被舍弃。特征选择是在进行文本情感分析的过程中非常重要的一部分，构建一个好的特征选择模型，可以使我们的情感分析过程事半功倍，相反如果我们的算法选择的不合理，轻则会降低正确分类的概率，重则导致维度爆炸问题（尤其是对于大文本而言），使得我们对学习模型的训练无法进行。

### 1.2.4对文本进行情感分类

经过以上的步骤，我们基本就完成了文本分类的处理阶段，接下来就是针对我们选出来的特征构建分类模型，然后在我们的分类模型上进行文本的训练，最后输出我们的分类结果。为了对我们的分类结果进行正确的评价，我们还需要与原始的分类数据进行对比计算，得出我们学习模型的分类准确率。

对文本进行情感分类的研究经过了那么多年，也产生了许多种方法，并且每种方法都在不断地更新与改进之中，从最原始的基于语义词典的情感分类方法，到基于传统机器学习的方法，再到基于深度学习的方法，我们构建的模型基本上是越来越复杂，但是在进行文本处理的过程基本是越来越简单，而且我们可分析的文本规模的大小也在逐渐增大，最后的分类结果的准确度也在逐步提高。

用情感词典进行分类是比较原始的分类方法，情感词典中含括了各种词语（如否定词、感叹词、主观词、表情等），对于每一类不同的词根据其对语句情感的影响程度添加权重，然后进行加权加和，最后比较负向情感得分与正向情感得分的大小，选择得分较大的一方作为我们的分类标准。这种方法原理简单易懂，操作上也并不十分复杂，但是它遗漏了很多重要信息（比如无法区别一词多义，对权重的设置存在缺陷等），使得最终得到的分类结果不理想。现在我们常用的方法一方面是利用传统机器学习方法构建的学习模型，另一方面是利用深度学习方法进行构建的模型。

## 如今文本情感分类的两大方法

现在，比较流行的用于进行文本情感分类的方法总体上可以分为两大类，上面我们已经提过，一类是利用传统的机器学习方法进行分类，另一类是利用深度学习的模型进行分类。

### 1.3.1传统机器学习方法

传统机器学习中可用来进行文本分类的算法主要有SVM（support vector machine 也就是支持向量机）算法，NB（naïve bayes 也就是朴素贝叶斯）算法，KNN（k-nearest neighbors algorithm也就是K邻近算法）等。

SVM在传统机器学习中使用十分广泛，它是一种二分类器[4]。SVM通过寻找一个空间中的超平面来对输入的样本进行分割，分割的原理就是使超平面两侧的样本到该平面的距离取得最大值。对于线性可分的数据集我们可以很容易的构造一个线性SVM分类器模型；对于线性不可分的数据集我们需要进行转化，也就是通过映射，把原来的数据集映射到一个线性空间里，然后再对映射后的数据进行分类，映射过程中需要用到核函数，在实际应用中使用频率比较高的函数[5]包括“多项式核函数、字符串核函数和高斯核函数”。为了防止出现过拟合的情况，或者说为了排除原始数据中的一些噪声对模型训练的影响，在实际使用的过程中我们设置一个可容忍误差（也就是松弛变量），该参数的大小对实验结果的准确度影响很大。实验表明，利用SVM进行文本情感分类要比情感词典的方法好很多[6]。

NB模型是一类典型的概率模型[7]，首先它提出了一个规则，也就是假设我们要研究的文本特征的独立性[8]，根据这一假设，在进行文本分类时我们先计算输入和输出数据集的联合分布概率，详细的过程就是先计算它的先验概率分布，然后计算它的条件概率分布[9]，通过这两种学习过程，我们就可以得到P（X，Y）。在接下来进行分类的时候，我们把输入代入我们模型，计算出数据的后验概率分布，然后从中选择后验概率最大的作为我们数据的实际输出。

K临近算法[10]是文本分类中一种十分简便的模型，该模型十分适合用于无监督学习过程，因为它不需要进行训练，也不需要进行参数估计。在该算法中我们通过计算每一个样本点距离分类中心的距离远近来对数据集进行分类，它进行多分类的效果一般比较好，在利用它进行文本分类建模的过程中，有两个问题需要考虑，一个是分类个数的初始化，一个是分类中心的初始化。分类中心的选择根据实际情况一般要稍大于正确值，但也不宜多大，否则会增加计算复杂度；分类中心可以进行多次随机初始化，最后对每一次的结果进行平均，作为我们实验的最后分类，这又叫做K-means algorithm。

### 1.3.2深度学习方法

基于深度学习的情感分类模型是近几年十分火的一个课题，深度学习模型最基本的就是神经网络，深度指的就是神经网络的层数不止是输入和输出层，通过模拟生物大脑的神经系统，我们在入和出之间加上若干隐藏层，隐藏层可以对输入的信息进行复杂的学习过程，最终生成最优的分类模型。深度学习需要大量的计算过程，也正因为它的计算量大，最后才会有很好的结果，当然它被大家欢迎一方面得益于计算机的处理数据的性能越来越好，一方面得益于我们的数据集越来越大，对于这些大数据集来说，使用这个模型会取得很好的分类效果。常用的deep learning模型有：MLP，DA，RBM，CNN，RNN，LSTM等。

MLP（multilayer perceptrons）是最简单也是最基本的深度学习模型，它通过把输入层一步步地向前传播，让计算机自动探究并学习其中包含的关系，并对这些关系进行正确处理，学习到一个最优的处理模型。在对MLP进行训练的过程中，我们用到的是BP算法[11]。

DA（Denoising Autoencoders）就是自动编码器[12]。DA模型的原理是特征提取，也可以说是降维处理。DA模型总体上来说构建了一个从输入到输出的恒等映射，也就是所输入层的数据和输出层的数据是完全相等的，它的目的是训练一个隐藏层出来，使得隐藏层的数据维度小于输入层的数据维度，但是它又能够很好的表示我们原始的数据信息，也就是尽可能减小信息丢失的情况。但是我们原始数据中难免会有噪声，所以在训练的过程中要进行正则化处理，这是其一，其二就是为了防止出现过拟合的情况，我们可以认为地随机添加一些噪声，比如把一些数据直接随机的初始化为零。单纯的DA是无法完成对文本的分类任务的，它只是利用深度方法对特征进行了提取和降维，要想完成最终的分类，还需要连接一个文本分类器，比如SVM或者NB分类器。

RBM（受限玻尔兹曼机）模型只能用于二分类问题，也就是说它的输出只能是0和1。RBM模型中构造了一个函数，我们称之为能量函数，该函数共走反应系统的稳定状态，在文本分析中能够对数据特征进行考量，当能量函数达到最低值时，我们的模型达到稳定状态，这时候我们得到的特征最有最优贡献率。在对RBM模型进行训练的工程中，我们如果选择传统的BP算法，就会发现运算十分复杂，运算量非常大，而且很多情况下不可解，因此我们实际上采用的是CD（**Contrastive Divergence）方法[13]。**

**CNN又叫做卷积神经网络，该模型经常被用于图像识别和视频处理领域。CNN模型是一种基于词向量模型的神经网络结构，在进行CNN模型构建之前，我们需要先对输入的文本进行处理，把它们转化为机器可以识别的语言，CNN的高级之处就在于它的输入是有确定维度的词向量，而且这个维度我们可以认为地进行设定，这样就可以避免词典过大造成的维度爆炸，计算无法运行的问题。把CNN运用于文本处理问题是最近许多学者进行的一项工作，因为文本的长度大小不一，造成了我们在处理上的障碍，目前大多数CNN文本分类时进行的处理过程是，选择长度最长的一个句子作为输入的维度[14]，对于那些长度比它短的句子，我们在它们的空缺上认为地补上0，这样处理的结果可想而知，虽然能够进行运行，但是结果只能差强人意，并不能像视频处理一样得到很好地结果。**

**我们本篇论文的模型基于反馈的，主要就是RNN和LSTM。**

## 文本情感分析的研究现状

文本情感分析经过国内外专家学者几十年的深入研究，已经达到了非常好的效果，不管是最初的基于情感词典的方法，还是后来的基于传统机器学习的方法，一直到现在的深度学习的方法，每种方法都没有说绝对的落后，每一种方法都在被专家学者们研究，每一种方法都在被反复优化。新的方法层出不穷，而且很多新的方法是基于一些原来的基本方法稍加改造之后的结果。把不同种的方法进行融合也是大家普遍进行的一个研究方向，比如利用DA进行特征提取之后采用SVM进行分类；利用RBM进行特征训练，利用RNN进行分类等。针对英文语料库的分类现在已经很成熟了，而且很多分类模型都可以达到很高的精度，比如本文中的LSTM就被公认为目前最好的文本分类模型，它对英文文本训练分类的正确率已经达到了93%。与英文情感分类相比较，中文的情感分类发展的稍微缓慢一些，一方面是因为中文远比英文表现力丰富，更加复杂，另一方面中文的语言规则更加繁琐，上下文之间的联系也远比英文要紧密，更涉及到很多成语、习语、典故等等，不过目前国内外的学者们都在努力研究。

## 论文构成及研究内容

我的论文主要包括五部分，第一部分是两个基于反馈的深度学习方法，也就是RNN和LSTM；第二部分是我的模型构建过程，包括文本的处理和词向量的训练；第三部分是我的实验数据，也就是实验过程中的数据集；第四部分是我的实验结果；第五部分是我对情感分类的一些方法的对比，还有对实验的一些分析。我的研究内容是反馈方法，所谓反馈包括两种情况，一种是前面的信息向后面的反馈，另一种是后面的信息向前面的反馈，因为对于一个文本来说，他们上下文语义之间肯定是有联系的，所以，反馈的目的就在于构建这种上下文之间的联系，让计算机去学习，大量实验表明反馈对于实验结果具有很好的提升。我研究了词带模型和词向量模型作为输入的情况下，他们对于LSTM的表现力如何；针对于词向量的维度，我做了很多实验，最后发现一般情况下词向量维度越高，实验结果就越好，但是维度高于满足需要之后，再增加维度，对实验结果的影响并不大，反而会大大增大实验运行的时间，甚至造成数据溢出，使实验无法进行；在训练的过程中我们采用的是随机梯度下降的方法，但是实验过程中很可能会出现过拟合的问题，导致最后生成的模型不具有代表性，在测试集上的运行结果较低，因此我一方面采用了交叉验证的方法，也就是把数据集分成三部分，用于训练、验证和测试，另一方面我们运用了early stopping的方法，让训练过程提前结束，针对随机梯度下降过程中的batch-size，我也做了实验，实验表明batch-size太大有可能导致出现局部最优解，导致实验结果并不理想。

# 基于反馈的深度学习方法

我的研究是基于两个基本的模型RNN和LSTM，这两种模型都以词向量作为直接的机器语言的输入。

## 文本作为机器语言的输入

在用计算机做文本处理，或者音频处理，或者视频处理的过程中，我们不可能把文字，声音或者视频直接作文数据进行输入，因为这些都是人类的语言，计算机是没法理解的。因此在做这种数据处理或者分析的时候，我们第一步需要做的就是把这些人类可以理解的数据转为成计算机类型的数据。在做文本情感分析的时候我们进行的处理是把每个词语转化成一个空间向量，这样就可以把向量作为输入直接用于机器学习，进而进行分析处理得到我们需要的分类结果。把文本加工成向量，有两种思路，一种是bag of words，也叫作是One-hot Representation，中文叫做“词带”模型，另一种就是Distributed Representation，也叫作是Word Embedding，中文叫做“词向量”。

词带模型就是把每个单词展成一个条带一样的向量，但是条带的大小并不是确定的，因此这个向量的维度是不确定的。在实际处理中我们把要处理的文本集中在一起，然后逐一遍历文本中的每个单词，确保每个单词都被统计在内，最后生成一个由文本中所有的单词构成的字典，字典的大小就是我们最后要构建的词带模型的大小，接着我们对数据集中的每一句话进行处理。我们初始化每个单词都是一个零向量，向量的维度就是地点的大小，如果单词在词典中的某个位置出现，那么我们就把该位置的0改为1，比如说一个词典的大小是9，一个单词“like”在词典中的第七个位置出现，那么我们这个所对应的向量表示就是[0 0 0 0 0 0 1 0 0]。扩展到每一句话，我们就是要把每一句话表示成一个词典大小的向量，我们把这句话中的每一个单词在我们建立的词典中进行遍历，单词出现就将该位置设置为1，否则就保持原来的0。比如说，我们要处理的原文本中包括三句话“This book is good”，“Is it Sunday”，“I like reading”，因为在文本处理中不区分大小写，所以我们在构建词袋模型的时候为了简便统一采用小写形式，那么我们这里构建的词典就是[‘this’，‘book’，‘is’，‘good’，‘it’，‘Sunday’，‘i’，‘like’，‘reading’]，根据词带模型的原理，我们第二句话就可以表示为[0 0 1 0 1 0 0 0 0]；当然在构建词袋模型的时候，我们视情况也会考虑词频，当一个单词在一句话中出现两次时，我们就可以把它设置为2。词袋模型构建起来非常简单，使用起来也极其方便，但是它存在几个问题。一方面我们的词带模型的维度是随着文本大小而变化的，它的不确定性会给训练过程增加难度，而且，当处理一个大文本的时候，我们的词带模型会达到几千维的大小，这对计算机的运算能力也是一个不小的挑战；另一方面我们在构建词典的时候是无法考虑单词在文中出现的先后顺序的，我们只能记录该单词的有和无，结果我们得到的词带其实是每句话的一个无序向量排列状态，这样就没法考虑到单词之间的联系，以及上下文直接的联系，我们把每个单词都看成了孤立的，这显然不符合正常的语言逻辑。所以，词带模型训练出来的分类效果是有限的。

词向量模型产生于1986年，它的创造者是Hinton，我们知道在平面直角坐标系中，我们可以通过测量两个点之间的距离来表示他们的相对位置关系，由此受到启示，专家学者们提出我们可以用空间中的点来作为词语的向量表示，于是，自然而然地，我们可以用表示两个单词的点之间的距离来衡量这两个单词在语义上的相似性，当这两个向量的距离越短，那么他们之间的关系越紧密，当这两个向量的距离越远，那么他们之间的关系也就越疏远。比如，“China”和“Beijing”这两个词在一个训练效果不错的词向量模型中所对应的向量在空间中的距离就十分接近，而“cat”和“moon”分别对应的空间向量在位置关系上就会比较远。当然了，词向量并不像词带模型一样可以直接从词典中简单的得到，词向量是需要机器进行训练学习才能得到的。首先我们要设定一个维度，也就是我们想把单词训练成一个多大的空间向量，维度的大小在一定程度上影响着训练的精度和训练的复杂度（一般就是训练的时间长度），训练词向量需要用到神经网络，而且在训练的过程中，我们的直接目的并不是要得到词向量，词向量只是在训练的过程中顺便产生的附加产物，我们直接的训练目的其实是语言模型。语言模型的训练，到目前为止，公认的最为经典的就是Bengio大师在2001——2003年构造的三层神经网络。后来的专家学者基于Bengio的思想模型提出了更加先进、准确度更高的模型。在2008年，C&W（Ronan Collobert 和 Jason Weston）提出了SENNA方法[15]；同年，M&H（Andriy Mnih 和 Geoffrey Hinton）发表了HLBL的方法；在2010年，Mikolov用循环神经网络对Bengio的模型进行改进，产生了RNNLM词向量模型，既提高了精确度，又缩短了训练时间。现在比较先进的产生词向量的方法包括两种，一种是Word2Vec，一种是Doc2Vec。Word2Vec的研发来源于谷歌团队，它其实包括了两种模型，第一种模型是CBOW（Continuous Bag of Words），在利用该模型进行语言学习的过程中，它是通过学习上文和下文的包含信息，来对当前词语进行预测，预测的结果就是一个概率分布；第二种模型就是Skip-gram，这个模型与第一个相比，可以说是做了一个反转，它的输入其实是第一种模型的输出，它的输出其实是第一种方法的输入，也就是说Skip-gram学习的是当前的单词，然后对可能出现的上文和下文进行预测，预测的结果也是一个概率模型。通过前面的两个模型，我们最终就可以把我们的文本信息转化成一个特定维度大小的空间向量，然后我们就可以利用该向量作为机器学习的输入语言，进一步构建我们的分类模型。Doc2Vec在训练原理和训练的方法上和Word2Vec并没有很大的差别，它只是把段落向量纳入了训练的过程中。

## RNN（循环神经网络）

RNN与其他的神经网络模型相区别，因为它的模型是一个环，如图2-1所示，MLP、RBM、CNN等神经网络都没有用到循环结构，图中的x表示输入层，s表示隐藏层，o表示输出层。

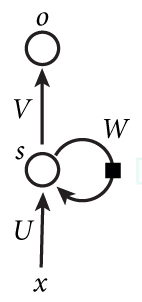
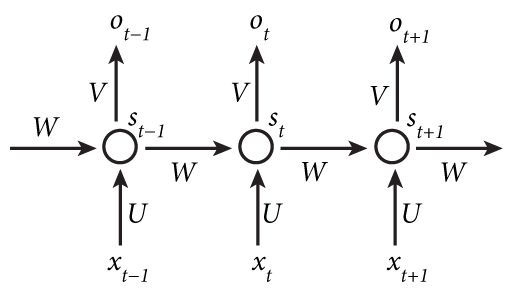


图2-1 图2-2

隐藏层的循环结构表示，上一次的隐藏层的数据可以传递到下一次的训练过程中来。在语言学中我们知道相互的两个单词之间，或者上下文之间都是有着表达上的联系的，比如说“小明想约我去吃饭，可是我刚吃过饭”，这句话中上下句放在一起才能说明“我不能和小明一块去吃饭了”这个事实，如果抛开上下句，那就变成了单纯的小明想约我吃饭和我刚吃过饭，并没有表达出这句话的主人真正想表达的意思。RNN就是基于这一点出发，为了便于理解，我们把RNN沿着时间轴展开，如我们的图2-2所示，我们可以清楚地看到信息沿着时间轴一步步地向前传播，当我们输入第一个单词的时候，我们的神经网络对其进行了学习，学习得到的信息很大一部分储存在我们的隐藏层（也就是s层）中，我们把t-1时刻的隐藏层传到t时刻的隐藏层，就相当于把上个单词所表达的含义与下个单词结合了起来，我们在这两层之间设置一个权重W，表示把上文中的信息传到下文中的比例。当输入就这样沿着时间轴向前传递的时候，信息也被累积在一起，所以最后所有的信息都汇总在了最后一个单词的输出上了，最后一个单词的输出也就是我们最后的分类结果，从理论上来说它是文本中所有信息的整合，具有很好的表现力。RNN的公式[16]如下：

 （2-1）

（2-2）

在上面这个公式中，W是对应于隐藏层向下传播的权重，V是输入层向输出层传播的权重。b是一个偏差项，是为了防止欠拟合而加的一项偏置，对于输入和输入，他们分别对应的偏置是不同的。在对RNN模型进行训练的过程中，采用的是BP算法，但是，在此过程中存在着一个问题，当我们的句子过长的时候，我们的传播层就会很多，我们进行反向传播的层数也就会很多，我们每反传一次，就会对因式进行一次指数层级的累加，最终会导致一个很严重的问题，如果我们的因式大于1，那么就会产生一个很大的梯度下降，造成参数波动极大，显然无法完成训练任务，如果我们的因式小于1，就会产生一个很小的梯度下线，近乎为0，那么我们的参数就几乎没有被更新，很显然也没法完成训练任务。所以，RNN模型只适用于短文本类型的情感分类，对于长文本无法进行处理。有学者对RNN的模型进行了改进，提出可以构建一个Multiplicative RNN（多层循环神经网络），然后利用HF optimizer进行训练，就可以对长文本进行分类。

## LSTM（长短时间记忆模型）

长短时间记忆模型是在RNN模型的基础上改造的，因为我们的原始的RNN模型在比较长时间的序列传递上无法进行正确信息的传递。LSTM模型的根本原理和RNN是一样的，只是它增加了三个门结构，这三个门起到了一个控制中心的作用，这点是仿照人的神经系统的工作原理创造的。在生物的神经调节的过程中，低级神经要收到高级神经的控制，比如我们的体温调节系统，首先我们皮肤上的感受器感受到温度变化，然后把这个信号传入，我们并不会直接进行穿衣活动，因为这个传入信号要收到大脑的控制，我们的大脑对传入信号进行处理，然后发出调节体温的命令，“穿衣或者不穿”，当然实际中的体温调节远比这里讲的要复杂得多，但是LSTM就是仿照了这个原理，设计出了类似于大脑神经中枢的门枢纽，门的开闭意味着信息的传递与舍弃，在具体计算的过程中，门的状态值十分接近于0，表明该门枢纽对应的信息基本被舍弃，即我们认为该信息是无用的，门的状态值十分接近于1，表明该门枢纽对应的信息是十分有用的，是我们要全部保留进行传递的。输入门决定是否要把输入信息进行训练，以及输入信息被纳入计算的程度；非常重要的一个门结构是遗忘门，遗忘门与上一个时间节点的隐藏层信息（在LSTM里就是中心记忆室cell）直接相连，上文的信息是否对下文产生影响，产生多大的影响，这在很大程度上是通过遗忘门来进行调控的；最后一个门是输出门，掌管着我们输出信息的强弱。LSTM的计算过程[17]如下所示：

首先是临时记忆中心的计算： （2-3）

然后我们要计算输入门值：  （2-4）

接着是我们的遗忘门：  （2-5）

再接着是我们的记忆中心：  （2-6）

下面是我们的输出门：  （2-7）

最后是我们的输出值：  （2-8）

X代表我们的输入，也就是我们在此时此刻的单词，当然了，单词是无法直接作为机器语言进行计算了，我们前面已经讲过，我们这里把单词转换成了词向量，因此这里的x是一个空间N维向量，这里的N值大小由我们进行决定；w代表的是权重，顾名思义，w就是衡量每个值所占的信息比重的参数，对于不同的控制门我们的权重是不一样的，权重的大小不是人为设定的，它是通过机器进行学习最后得到的，w是一个N\*N的矩阵，它与x其他的参数相乘之后得到的是一个N维的向量；b是一个非线性参数，我们称之为偏差，b也是一个N维大小的向量，b的作用就是优化训练方程，防止出现欠拟合的情况，b的值同样是要经过机器自己学习的，我们不能进行人为设定。公式中的是一个函数，该函数是我们构建神经网络进行数据分类时经常用到的，一般称为逻辑斯蒂函数（也叫作S函数）。上面的公式中最重要的一项就是c，这个值存储了我们输入的信息，最简单的LSTM只有一个记忆中心，处理起来也不麻烦，但是很多情况下我们选择多个记忆中心，这些记忆中心连接着相同的输入和输出，不同的是他们作用的权重大小，使用多个c可以更好的对信息进行储存和传递[18]，当然，也会使得计算过程变得更加麻烦。

# 实验步骤

在我的这个实验的过程中，我采用的实验步骤基本就是按照传统的文本情感分类的步骤，因为我们的具体模型是基于反馈的深度学习模型，也就是LSTM，所以在具体的处理过程中也有一些区别。

## 实验数据及处理

我在这个实验中采用的数据来自文本分类中十分常用的两个数据集，他们都是关于电影评论的，叫做mouvie review。其中一个数据集是长集，包括50000条评论，其中认为电影积极的评论是25000条，认为电影消极的评论也是25000条。另一个数据集是短集，总共包括10659条评论，其中认为电影积极的评论是5329条，认为电影消极的评论是5330条。

在进行词向量的构建之前，我们需要对这些文件进行一些简单的处理。首先这些文件是由 50000个小文件组成的，每一个小文件是一个人的评论，我们需要把这些评论整合到一个文本之中，最后这个大的文本包含50000个数据，其中25000个评论是积极的，25000个评论是消极的。之后我们先利用正则表达式把文本中的换行符、制表符等去掉，接着把逗号、句号、引号等标点符号去掉，之后把所有的单词进行小写化处理，最后对文本进行分词处理，这样我们就得到了一个只含有小写字母的文本数据集。

接着我对得到的上类数据文本进行两种处理，也就是构建两种模型，这两种模型在上面我们已将说过了，一个是词向量，另一个是词带模型。

首先是词袋模型的建立，这个很简单，我对文本中的每个单词进行逐个的便利操作，如果遇到的单词不在我们已经建立好的词典当中，我们就把它统计进去，如果遇到的单词在我们的词典中已经存在过了，我们就把它出现的次数进行统计，正常情况下词典中的单词出现的位置都是随机的，所以前后两个单词之间基本是独立的，相互之间不会有影响，但是在这里我进行了一个简单的排序，我把把词典中的所有单词按照他们在数据集中出现的次数高低进行排列，出现次数比较高的，我们放在词典的最前面，出现次数低的，我们放在词典的最后面。之所以这么做，是因为我们要训练的模型是LSTM，而LSTM结构最基本的就是时间序列，也就是单词的上下文之间存在着相互影响的联系，所以，我们提出一个假设：假设出现频率越高的词语，在文中传递的信息越远，因此，放在词典最前免得词语的信息将会一直被传递到最后，对它沿时间经过的单词都会产生或大或小的影响，而排在词典最后的单词，我们认为它出现的次数太少，对文本信息的贡献率也是最低，所以它的信息对其他单词机会没有什么影响。

至于词向量的构建，我们用Word2Vec进行处理，这点和前面一模一样，我们只需要对向量的维度进行初始化就可以了。

## 将处理好的数据代入模型进行计算

根据我构建的模型（LSTM模型），直接将上面得到的数据带进去，进行计算。在计算的时候，需要对对变量进行初始化处理，我们把权重w采用random进行随机初始化，把偏差b全部初始化为0。对变量进行初始化的方法有很多，这里只是其中的一种。初始化之后就可以直接代入极性计算了。这个时候计算出来的结果只是为了能够进行接下来的训练，并没有其他实质性的代表意义。

## 训练我的模型

要想对一个神经网络进行学习，也就是把实验中的相关变量参数求解得到最优结果，在LSTM中就是寻找一组w和b的值，使得我们的模型计算出的结果与我们的真实值最为接近。为了达到这个目的，我们首先要定义一个函数，也就是能够准确描述实验值与真实值接近程度的函数，我们称之为代价函数。常用的代价函数包括两种，一种是方差计算方法，一种是交叉熵计算方法，在这里我选择第二种方法，因为第一种方法在进行参数更新时较为缓慢，而我们的数据集又比较大。求最适合参数的过程其实就是使目标函数最优化的过程，我们要使这种误差最小，也就是使目标函数取得最小值，按照数学上的方法就是进行求导，解出它的最小值。但是在用机器进行学习的过程中，我们不可能直接通过求解函数进行参数求解，因此我们采用数学上的逼近思想，通过梯度下降，对参数进行逐步更新，最终一步步靠近我们函数的最小值。梯度下降的速度选择也是一个问题，因为参数选的过小，我们在更新的时候每次只对数据进行了非常小的改动，要想达到我们的最优点，需要的时间就会非常长；但是，如果参数选的太大，虽然会使得我们的参数更新变快，却也会导致我们的实验值在最优值两侧往复摆动，永远达不到最优，形成一个死循环。我在该实验中选择的速度是0.0001，在牺牲了一部分学习速度的情况下，以期取得更好的实验结果。在实验过程中，经常出现的一个问题就是过拟合，也就是说，我们最后训练出了一个和我们的训练集非常非常逼近的模型，但是由于我们的输入并不是完美的，存在着偶然性，有一些噪点，这样就造成我们的模型不能对其他的数据进行分析评估，而我们训练模型的目的恰恰是要对其他的数据进行分类的。所以，我们要降低这种完美的训练拟合程度，也就是对模型进行泛化处理。有一种泛化方法，是对输入的数据进行操作的，我们可以把输入的数据进行随机加噪，就是把一部分输入初始化为0，这样就可以在一定程度上阻止过拟合的发生。在我此次的实验中应用了三种方法去处理，首先是dropout的处理方法，这种方法需要对网络模型进行修改，正常情况下我们应该按照网络结构，一步步地进行计算，知道网络的最后一层，然后输出我们的分类值；但是在dropout的处理方式中，我们要主动地放弃一些神经节点，每次随机地丢掉一些节点，并且多次进行同样的处理，采用每一次输出结果的平均值作为实验结果。因为我们丢弃了部分节点，也就是说我们降低了每个网络的复杂度，但是又增加了需要训练的网络模型的个数，从总体上来说，并没有减弱我们模型的学习能力，由于随机训练了多个网络，使得每个网络的学习路径不尽相同，从而很大程度上避免了过拟合的出现，本次实验中，我是设置的节点随机丢弃率是百分之五十；第二种方法是提前停止，当实验结果趋于稳定的时候，我们就没必要再对所有的数据进行计算，那样一方面会浪费时间，另一方面会造成过拟合，所以我们只需要设定一个适合的值，如果我们的错误率在执行这个值的步骤之后都比现在的要大，我们就有理由相信我们现在的错误率基本就是最优解，那么模型的训练到此为止就可以了，我在这次实验中设定的耐度值是10；第三种方法是交叉验证[19]，在神经网络中这个方法十分常用，过拟合就是模型的适用性太低，它对训练的数据有非常好的拟合效果，但是一旦换了一个数据集就不行了，所以我把实验数据随机分成三部分，训练部分、验证部分和测试部分，通过训练部分和验证部分进行交叉验证就能降低出现过拟合情况的概率，在本次试验中，我的训练部分和测试部分的比例是19:1。

除了上述问题外，在实际训练的过程中还会出现局部最优解的情况。在数学中我们知道，一个函数的最小值只能有一个，但是一个函数的极值可能有很多。于是，在我们的代价函数中就会出现这样的问题，当我们的参数值停留在一个极值点处时，训练就会停止，这是返回的输出值并不是我们要求的最值，这就造成了模型分类效果变差。为了尽可能减少这种情况出现的概率，我们在实际训练的时候采用随机梯度下降的方法。在传统梯度下降过程中，我们的代价函数是所有的样本模型输出值与真实值之差的平均值，也就是说没更新一次参数，我们就要把所用的训练样本进行计算一遍，对于几万条的数据，计算量可想而知，虽然这样得出的代价函数的表达式是最精确的，但是复杂度太高，而且极易造成局部最优，所以我在实验中采用更为简单有效的随机梯度下降。我们只对一条或者一小部分样本进行计算，根据它的计算结果直接对参数进行更新，因为样本是随机挑选的，所以对随机挑选的一部分都能达到很好的分类效果的话，它对整个样本基本上也会拥有很好的效果。

## 3.2 实验介绍

在我的研究中，我针对不同的研究目的做了不同的实验，在实验模型的构成上也会有些不同。

### 3.2.1 实验一：词带模型和词向量模型的比较

第一个实验，我们要研究这两种模型在LSTM模型上的分类准确度。对于词向量，我们已将准确的描述过了，这里只需要把每个单词得到的向量作为LSTM在时间t处的输入就行了，所以时间序列的长度等于句子中包含的单词的个数。对于词袋模型，要想在LSTM中进行训练，我们要基于上文中的假设：假设出现频率越高的词语，在文中传递的信息越远。这是我们的LSTM的长度是固定的，对于每句话来说它的长度都等于词带的大小，比如说其中的几个文本对应的词带是[1 0 0 2 4 0 0 6 9 0 1 0 2 0]，那么这个LSTM的长度就是14，t=1时，输入x=1，t=2时，输入x=0，以此类推，把词带中的每一个值依次作为一个输入向前传递。实验中，我是用的是数据集二，也就是短集，因为数据集太大的话对词带的训练将会变得异常缓慢。

## 3.2.2 实验二：数据集的大小对实验的影响

在这个实验中，我们使用基于词向量的LSTM模型。实验的输入分别是前面的两个数据集，第一个是大小为50000的长集，第二个为大小为10659的短集。除了输入以外的其他参数都一样，验证集占交叉验证的比例都是百分之五，词向量的大小都是128维，提前停止的忍耐度都设置为10，随机丢弃神经节点的比例都是0.5。这个实验我们要研究的是LSTM在不同大小的数据集上的学习能力是不是有差别。

### 3.2.3 实验三：词向量维度

在训练词向量的过程中，向量的维度大小是认为设定的，也就是说我们可以选择任意大小的维度作为LSTM的输入。因此，我们肯定会猜想，是不是存在一个最适合、效果最好的维度能使我们的模型达到最好的学习目的。在这个实验中，我们使用的数据集是50000大小的长集。我设置了十组实验，除了词向量维度以外，他们的其他参数都完全一样。这十组实验的维度分别是15，30，40，60，80，128，180，256，400，500。我们分别记录每一组实验的分类结果，然后进行比较分析。

### 3.2.4 实验四：batch-size的大小

首先我们要说一下什么是batch-size，在避免局部最优解的方法中，我们使用了随机梯度下降，它是分别对每一个输入进行计算输出误差之后，立即进行反向传播，进行参数大小的更新。这种方法虽然简单，运算量也相对来说小很多，而且可以避免局部最优，但是，每次对一个数据进行计算然后更新具有很大的随机性，基本上很难保证收敛性，于是，我们在模型之中反复计算，却始终得不到需要的结果；如果batch-size的值选的太大，就又回到了我们的传统的梯度下降方法，很可能就会造成局部最优解。这次实验过程中，我设置了7组对照实验，他们的batch-size分别设定为1，20，64，80，200，500，1000。我们分别记录每组实验的结果，并进行比较分析。

### 3.2.5 两种MLSTM模型

其实在最基本的LSTM模型上还有许多变形，这次实验中我要研究的就是其中的两种变形。这两种变形的基本思想都是一样的，都是MLP。MLP是最基本的神经网络，它之所以具有很好的效果，是因为它有不止一个隐藏层，这些隐藏层就好像神经结构，神经结构比较多学到的东西就会越深入、越丰富。第一个模型我们把每个LSTM序列的cell进行多层学习，这里我设置层数为3，就是说我的MLSTM包括一个输入层，3个cell隐藏层（3个cell直接相连，不再连接遗忘门），一个输出层，输入门仍位于第一个cell之前，输入出则位于最后一个cell之后，在这个模型中，我们的原理是，在生物学兴奋传导的过程中，经过每条神经纤维的逐级传到，最后信号被传到效应器，然后发生相应的反应，我们的各级cell就相当于每条神经元的细胞体。第二个模型中我们把第一层的输出作为下一层的输入进行信息传递，在这里我们同样设置了三层学习模型，每层 相当于一个完整的LSTM，在水平方向上依然是时间序列，在竖直方向上则是三个LSTM的组合，每一层都有输入门、遗忘门、输出门，在这里我们是仿照高等生物信息传递收到大脑中枢神经调节的原理，我们假设每一层的输出都要由大脑进行一次信息处理，并决定是否需要记录、或者是否需要遗忘。在这两个实验模型中，我们的输入都是50000大小的数据集，用词向量模型进行处理，向量的维度设为128，作为模型的输入，训练的时候仍然是BPTT。

# 实验数据

## 4.1实验一：词带模型和词向量模型的比较

在这个10659大小的数据集上，我们的LSTM在词带和词向量模型上的结果如下表所示：

|  |  |  |
| --- | --- | --- |
| 模型 | 数据集大小 | 分类准确度（accuracy） |
| LSTM+词带 | 10659 | 75.6% |
| LSTM+词向量 | 82.2% |

表4-1

从上表我们可以看出，LSTM模型在词向量上的效果要比词带好很多。

## 4.2 实验二：数据集的大小对实验的影响

在这个实验中，我们在50000大小和10659大小的数据集上分别进行了测试。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型 | 数据集大小 | 词向量维度 | Dropout | 分类准确度（accuracy） |
| LSTM | 10659 | 128 | 0.5 | 82.3% |
| 50000 | 87.6% |

表4-2

从上表，我们可以看出，LSTM在大数据集上的学习效果要比在小数据集上的效果好很多。

## 4.3 实验三：词向量维度

在这个实验中，我们设置了10组分别进行对比，实验采用的是50000的长集。实验结果如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 数据集大小 | 词向量维度 | 分类准确度（accuracy） |
| LSTM | 50000 | 15 | 86.5% |
| 30 | 86.6% |
| 40 | 87.1% |
| 60 | 87.2% |
| 80 | 87.1% |
| 128 | 87.4% |
| 180 | 87.6% |
| 256 | 87.5% |
| 400 | 87.6% |
| 500 | 87.6% |

表4-3

从上表我们可以发现，虽然数据有一些波动，但是从总体上来说，随着维度的增大，最好的分类准确度也在提高。

## 4.4 实验四：batch-size的大小

在这个实验中，我们设置了7个对照组，分别在128维的向量上训练LSTM模型。实验结果如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 数据集大小 | Batch-size的大小 | 分类准确度（accuracy） |
| LSTM | 50000 | 1 | 90.1% |
| 20 | 88.6% |
| 64 | 87.5% |
| 80 | 87.6% |
| 200 | 86.8% |
| 500 | 85.7% |
| 1000 | 86.1% |

表4-4

通过上面表格中的数据，我们可以发现，虽然数据有一些波动，但是从总体上来说，我们的分类效果随着batch-size的增大是在逐渐下降的。

## 4.5 两种MLSTM模型

最后一个实验，数据集是大小为50000的长集，我们用128维的向量，batch-size为64的LSTM模型进行处理。实验数据如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | 数据集大小 | 词向量维度 | Batch-size大小 | Dropout | 分类准确度（accuracy） |
| LSTM1 | 50000 | 128 | 64 | 0.5 | 89.2% |
| LSTM2 | 90.3% |

表4-5

通过上面的表格，我们可以看出，LSTM2的效果要比LSTM1稍微好一些。

# 实验结果的说明与分析

首先通过实验一，我们发现词袋模型的训练效果并不好，这个结果甚至比贝叶斯分类还低。但是词向量的LSTM效果还是很不错的。其实LSTM最为深度学习方法，并不适合用于词带模型的训练，LSTM模型之所谓被设计出来，就是因为它考虑到了上下文信息的相互关联性，通过时间序列的传播，上文的信息被传播到下文，这样相互之间的语义就被联系了起来，因此训练结果就会得到优化。但是，词袋模型在建立的过程中是一个单词无序排列的随机状态，虽然我们做出了假设：假设出现频率越高的词语，在文中传递的信息越远，但是因为单个文本的大小是有限的，一个文本小的只有短短10个单词左右，大的也只有1000左右，所以，每个单词出现的频率并没有非常明显的差别，而且有些单词虽然出现的频率低，但是他们携带的信息量确实非常大的。

通过实验二，很名言，长集的效果比短集要好。所以有些学者说过，在一定程度上“你的数据集决定了你的实验结果”，当然这句话并不完全正确，只是在一个方面上的描述。大的数据集之所以会有更好的效果，一方面是因为它携带的信息量更大、更全面，机器可以更好的对文本进行深入准确的学习；另一方面是因为文本越大，实验中出现过拟合的概率就会越小，所以实验会得到更好的准确度。

在实验三中，总体上来说，词向量维度越大，实验结果就会越好，因为，维度越大，对应的向量空间越复杂，它包含的信息也就越多，那么我们的向量就可以更好地对每个单词进行表示，向量之间的距离也能更好地反应单词之间的关系。但是，在本实验中，当维度大于128时，进一步增大维度，对结果的提升并没有非常明显，训练时间却会明显的增加，因为维度增大会明显加大机器的计算量，而且当维度太大时对内存要求也很高，一般的计算机无法进行模型的计算。

实验四在batch-size为1时得到了最好的实验结果，因为随机梯度下降中的batch-size设置的越小，我们出现局部最优解的可能性就会越低，所以会有更好的拟合效果。Batch-size设为1000时，结果显然很差，就是因为出现了局部最优。但是，我们前面已将说过当batch-size太小的时候，我们对每一个输入进行学习、训练，很容易出现一直找不到最优解的情况，即使能找到，训练时间也是非常长的。在我的实验中，batch-size为1000时，训练只用了2个小时，然而batch-size为1时，训练用了24个小时。所以我们在实验时还是要对此进行权衡的。

实验五的两个MLSTM模型的训练结果相差并不算很大，而且与普通的LSTM相比，它们的结果是得到了明显的提升的。说明更深层次的学习会改善学习的质量，当然，LSTM2比LSTM1还是好一点的，说明在每层都进行门控制会产生更好的效果，因为每层都可以对信息进行分析学习，可以去决定信息的好坏、去留。

# 结论

本次课题，我针对基于反馈的深度学习方法（主要是LSTM）做了深入的研究，并将它用于文本情感分析之中。LSTM最为比较前沿的神经网络结构，具有很好的分类效果，它是建立在词向量的输入之上的，对于词带模型并不很适用。文本分析的数据集越大，我们的处理效果就会越好，所以要对一个话题进行情感分析，我们要尽可能的收集有关它的数据，才会取得更好的效果。在构建词向量是，在计算机计算能力允许的范围内提高维度，是可以提高分类准确度的，不要同时也要考虑时间要求，这点和batch-size很相似，理论上batch-size越小越好，但是要以训练时长作为代价，在具体应用中，我们要进行综合分析，确定最佳参数模型。MLSTM是LSTM的一种变形，深层次的网络学习对提高实验结果具有很大的帮助。

情感分析作为目前十分流行的一个领域，很多东西在处在研究之中，每种方法都在被优化、被改进，比如在LSTM模型之上的BLSTM，TLSTM等，相信以后还会有更多更好的模型取代LSTM。

# 致谢

首先我要感谢以Bengio大师为代表的所有在机器学习和文本情感分析领域孜孜不倦的学者专家们，正是有了他们的理论铺垫、方法创新，才有了现在各种各样的新方法。

特别感谢我的毕设指导老师——荣老师，荣老师学识渊博，治学严谨认真、一丝不苟，在机器学习和深度学习上给了我很多的建议。荣老师待人亲和，对我们刚入门的学生耐心教导，给了我们很大的鼓励。

感谢在网上视频教程、代码、数据集等源数据的各位大师。

感谢我的师兄姜楠对我的鼓励，感谢他对我的毕设提出宝贵的意见，他现在是我学习的榜样。

感谢我的实验室的其他师兄师姐，他们给了我很大的关怀，让我的生活变得丰富多彩。

# 参考文献

[1] 赵妍妍，秦兵，刘挺. 文本情感分析[J]. 软件学报，2010,21（8）：1834-1848.

[2] 周立柱，贺宇凯，王建勇. 情感分析研究综述[J]. 计算机应用，2008,28（11）：2725-2728. [3] 徐琳宏,林鸿飞,赵晶.情感语料库的构建和分析[J].中文信息学报,2008,22(1):116−122.

[4] 杨经，林世平. 基于SVM的文本词句情感分析[J]. 计算机应用与软件，2011,28（9）：225-228.

[5] 李航. 统计学习方法[M]. 北京：清华大学出版社，2012: 121-124.

[6] 金鑫，张宪超. 基于朴素贝叶斯的文档级情感分析[D]. 大连：大连理工大学，2013.

[7] M.Govindarajan. Sentiment Analysis of Movie Reviews using Hybrid Method of Naïve Bayes and Genetic Algorithm[J]. International Journal of Advanced Computer Research，2013,3（4）：139-145.

[8] 王淼，孟洛明，刘坤，杨昕. 基于贝叶斯和规则的混合情感分析研究[D]. 北京：北京邮电大学网络与交换技术国家重点实验室，2014.

[9] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J].The Journal of Machine Learning Research, 2003, 3:1137-1155.

[10] 张宁，贾自艳，史忠植. 使用KNN算法的文本分类[J]. 计算机工程，2005，31（8）：171-172.

[11] Williams D E R G E H R J, Hinton G E. Learning representations by back-propagating errors[J]. Nature, 1986,323:533-538.

[12] 秦胜君. 基于稀疏自动编码器的微博情感分类应用研究[J]. 广西科技大学学报，2015，26（3）：37-40.

[13] 高琰，陈白帆，[晁绪耀](http://xueshu.baidu.com/s?wd=author%3A%28%E6%99%81%E7%BB%AA%E8%80%80%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson)，[毛芳](http://xueshu.baidu.com/s?wd=author%3A%28%E6%AF%9B%E8%8A%B3%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson). [基于对比散度-受限玻尔兹曼机深度学习的产品评论情感分析](http://xueshu.baidu.com/s?wd=paperuri%3A%287c96cf5bfc543db6f60cc588ab689af5%29&filter=sc_long_sign&tn=SE_xueshusource_2kduw22v&sc_vurl=http%3A%2F%2Fwww.cnki.com.cn%2FArticle%2FCJFDTotal-JSJY201604032.htm&ie=utf-8&sc_us=6995737214285838195)[J]. 计算机应用，2016，36(4)：1045-1049.

[14] [X Ouyang](http://xueshu.baidu.com/s?wd=author%3A%28Xi%20Ouyang%29%20%3BSch.%20of%20Electron.%20Inf.%20%26%20Commun.%2C%20Huazhong%20Univ.%20of%20Sci.%20%26%20Technol.%2C%20Wuhan%2C%20China%20%3B&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson)，[P Zhou](http://xueshu.baidu.com/s?wd=author%3A%28Pan%20Zhou%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson)，[CH Li](http://xueshu.baidu.com/s?wd=author%3A%28Cheng%20Hua%20Li%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson)，[L Liu](http://xueshu.baidu.com/s?wd=author%3A%28Lijun%20Liu%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson). [Sentiment Analysis Using Convolutional Neural Network](http://xueshu.baidu.com/s?wd=paperuri%3A%2885de7657a7c03d0445e5e08f06b023bf%29&filter=sc_long_sign&tn=SE_xueshusource_2kduw22v&sc_vurl=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D7363395&ie=utf-8&sc_us=2372497843587184249)[A]. IEEE International Conference on Computer & Information Technology[C]. IEEE,2015:2359-2364.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu and Pavel Kuksa.Natural Language Processing (Almost) from Scratch[J]. Journal of Machine Learning Research (JMLR), 2011, 12:2493-2537.

[16] Ilya Sutskever,James Martens,Geoffrey Hinton. Generating Text with Recurrent Neural Networks[A]. International Conference on Machine Learning[C].USA: L. Getoor, T. Scheffer, editors, 2011: 1017–1024.

[17] Martin Sundermeyer，Hermann Ney，Fellow. From Feedforward to Recurrent LSTM Neural Networks for Language Modelong[J]. IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING，2015，23（3）：517-529.

[18]Sepp Hochreiter，Jurgen Schmidhuber. LONG SHORT-TERM MEMORY[J].NEURAL COMPUTATION,1997,9(8):1735-1780.

[19] [杨静](http://xueshu.baidu.com/s?wd=authoruri%3A%283c62851ac47a1bbc%29%20author%3A%28%E6%9D%A8%E9%9D%99%29%20%E5%B1%B1%E8%A5%BF%E5%A4%A7%E5%AD%A6%E6%95%B0%E5%AD%A6%E7%A7%91%E5%AD%A6%E5%AD%A6%E9%99%A2&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson&sort=sc_cited)，[王瑞波](http://xueshu.baidu.com/usercenter/data/author?cmd=authoruri&wd=authoruri%3A%28723c7589b7021874%29%20author%3A%28%E7%8E%8B%E7%91%9E%E6%B3%A2%29%20%E5%B1%B1%E8%A5%BF%E5%A4%A7%E5%AD%A6%E8%AE%A1%E7%AE%97%E4%B8%AD%E5%BF%83)，[李济洪](http://xueshu.baidu.com/usercenter/data/author?cmd=authoruri&wd=authoruri%3A%28962600ceb78614c5%29%20author%3A%28%E6%9D%8E%E6%B5%8E%E6%B4%AA%29%20%E5%B1%B1%E8%A5%BF%E5%A4%A7%E5%AD%A6%E8%AE%A1%E7%AE%97%E4%B8%AD%E5%BF%83). 一种均衡的RHS交叉验证[J]. 南京大学学报（自然科学），2015，51（4）：842-849.