**深 圳 大 学**

本 科 毕 业 论 文（设计）

题目: **基于深度学习的情感分类技术研究**

姓名: **郭超学**

专业: **计算机科学与技术**

学院:  **计算机与软件学院**

学号:  **2013150371**

指导教师: **黄哲学、陈小军**

职称: **教授、讲师**

2017年4月25日

**深圳大学本科毕业论文（设计）诚信声明**

本人郑重声明：所呈交的毕业论文（设计），题目《基于深度学习的情感分类技术研究》是本人在指导教师的指导下，独立进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明。除此之外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。本人完全意识到本声明的法律结果。

毕业论文（设计）作者签名：

日期： 年 月 日

目 录

[摘要（关键词） 1](#_Toc481985743)

[1 引言 2](#_Toc481985744)

[1.1 研究背景及意义 2](#_Toc481985745)

[1.2 国内外研究现状 2](#_Toc481985746)

[1.3 本文主要工作 3](#_Toc481985747)

[2 深度学习概述 4](#_Toc481985748)

[2.1 人工神经网络 4](#_Toc481985749)

[2.2 人工神经元 4](#_Toc481985750)

[2.3 多层神经网络 5](#_Toc481985751)

[2.4 反向传播算法 6](#_Toc481985752)

[2.5 多分类模型 7](#_Toc481985753)

[3 基于深度学习的文本情感分类模型 9](#_Toc481985754)

[3.1 词嵌入 9](#_Toc481985755)

[3.2 循环神经网络 9](#_Toc481985756)

[3.3 长短期记忆人工神经网络(LSTM) 10](#_Toc481985757)

[3.4 LSTM文本分类模型 14](#_Toc481985758)

[3.5 注意力机制 15](#_Toc481985759)

[4 基于目标的情感分类模型设计 16](#_Toc481985760)

[4.1 词表示 16](#_Toc481985761)

[4.2 LSTM-ATT模型 17](#_Toc481985762)

[4.2.1 双向LSTM网络 18](#_Toc481985763)

[4.2.2 注意力层 18](#_Toc481985764)

[4.2.3 情感分类预测 19](#_Toc481985765)

[4.3 防止过拟合 19](#_Toc481985766)

[4.4 模型训练 20](#_Toc481985767)

[5 实验验证 21](#_Toc481985768)

[5.1 实验环境 21](#_Toc481985769)

[5.2 实验设计 21](#_Toc481985770)

[5.2.1 数据集 21](#_Toc481985771)

[5.2.2 数据预处理 22](#_Toc481985772)

[5.2.3 模型核心代码 23](#_Toc481985773)

[5.2.4 模型检验标准 24](#_Toc481985774)

[5.2.5 超参数设置 24](#_Toc481985775)

[5.3 实验结果和分析 24](#_Toc481985776)

[6 总结和展望 27](#_Toc481985777)

[6.1 本文总结 27](#_Toc481985778)

[6.2 展望 27](#_Toc481985779)

[参考文献 28](#_Toc481985780)

[致谢 29](#_Toc481985781)

[Abstract(Key words) 30](#_Toc481985781)

基于深度学习的情感分类技术研究

计算机与软件学院计算机科学与技术 郭超学

学号：2013150371

**【摘要】**情感分类是文本情感分析的基本任务，是要对文本进行极性分类。分类的作用是判断某文本表述的观点或情绪是积极的、消极的、还是中性的。而在实际情况中，同一文本的情绪表达在不同的方面会有不同的极性。例如，“这家餐厅的味道很好，但是服务态度太差了！”，句中对“味道”的评价是正面的，而对“服务”的评价是负面的。这里我们使用文中出现的关键词作为目标对象，以考量文本表达的情绪极性，我们将之称为基于目标的情感分类问题。近年来，深度学习在图像处理和语音识别领域取得着重的成效，也给情感分类问题带来新的方法。以词嵌入技术将词语映射到实向量特征空间，作为情感特征表示，结合循环神经网络，如长短期记忆神经网络，不但能充分利用词语之间的相关性特征，更考虑了词语在文本中的序列信息。这种方法取得了优于传统分类算法的效果。在处理基于目标词的情感分类问题时，Tang的TD-LSTM[1]将原文本从目标词位置一分为二，使用两个不同方向的LSTM网络处理两个子文本，使得网络输出与目标词有更大的相关性。本文将在此模型的基础上引入注意力机制，并将文本中词语相对于目标词的距离和词语的词性作为扩展特征，以改进原模型的词表示。我们使用开源深度学习引擎Tensorflow实现模型，使用Twitter和微博文本数据集进行训练和检验，最后分析实验结果，检验模型的有效性。

【**关键词**】情感分类；深度学习；长短期记忆人工神经网络；注意力机制

1. 引言
   1. 研究背景及意义

自然语言处理是利用计算机来处理、理解和运用人类语言，它是计算语言学的分支学科，也是人工智能领域的一个重要方向。让计算机拥有处理和理解人类语言而非形式化语言对于人工智能的发展极为重要，具备处理自然语言的能力是计算机实现智能的关键。情感分析作为自然语言处理的一个重要任务，指利用自然语言处理、文本分析、计算语言学来系统地识别、提取、量化和研究情绪状态和主观信息。情感分析的目标在于识别各种文本材料对于某些主题的态度倾向，或者对某些事件的情绪反应。

随着智能设备的普及，人们生活的方方面面都已经离不开网络和智能。各种智能技术的应用已经极大地方便了人们的日常生活，如语音识别技术在导航、信息查询和文本输入等应用上发挥着重要的作用。而在像SIRI[[1]](#footnote-0)或智能对话机器人等应用领域，除了需要识别语音信息并将其转换为文字之外，更重要的是机器必须理解人类语言和感知其中的情感和情绪信息。

除了实现对人类语言所表达的情绪进行感知之外，情感分析技术在商业领域也有着极为重要的价值。互联网的发展进入Web2.0[[2]](#footnote-1)时代，社交网络媒体的数量和规模都在激增。其中几个体量巨大的社交网络应用如Twitter、Facebook和新浪微博等，已经成为大部分人日常获取和分享信息的重要渠道。这些社交网络应用拥有着巨量的用户，每天都有不可计数的短文发布到网络中，这些信息不但传播迅速广泛，而且拥有着极大的社会影响力，社交网络的舆论倾向甚至可以轻易地左右重大社会事件和政治事件的发展和走向。

这些社交网络媒体数据中，还散布着大量可以为商业所利用的信息，如用户对某品牌或某商品的评论文字。利用情感分析技术可以归纳出用户在社交网络应用和电子商务应用上对商品和服务的评价，通过分析出这些评论文字所表达的情绪和态度，企业不但能对此做出相应的商业决策，还能利用社交网络的活跃性保持对市场变化的敏感，从而能及时对变化做出响应。有更多的事实表明，隐藏这些文本数据中的情感信息，有时比实实在在的统计数据更能揭示一些商业运行的实际问题。在这些巨量的社交网络平台的文本数据中，无疑隐藏着巨大的商业价值，挖掘这些数据的情感信息对于企业来说是未来的重点。

应对社交网络应用的快速发展和社交数据的爆炸性增长，情感分析技术不管在个人、商业或是政治领域都有着巨大的应用潜力。

* 1. 国内外研究现状

目前普遍认为，自然语言处理的历史起源于二十世纪50年代，阿兰·图灵对机器智能的猜想。1950年，他在《思想》（Mind）杂志上发表的一篇名为“计算的机器和智能”（Computing Machinery and Intelligence）的文章中提出了一种验证机器是否具有智能的方法：让人与机器交流，如果人无法分辨交流出与自己交流的对象是人还是机器，那么就能认为这个机器具有智能了。在最初的数十年里，研究者对自然语言的认识都局限在人类学习语言的方式上，也就是像人类一样，必须先清楚理解语言的语法和词语的含义，而语义的理解必须建立在此基础之上。但是这种思路并没有带来多少成果，基于文法分析和语义分析的方法在面对有歧义和不规范等输入时遇到极大困难，而不断庞大的语法规则集合也给相关研究带来巨大的人工成本和计算成本。这导致自然语言处理的发展在很长一段时间内处于停滞状态。直到二十世纪七十年代，由于隐马尔可夫模型的引入，基于统计概率模型的方法在语音识别和机器翻译上获得重大突破，才让自然语言处理研究又重获新生。

进入二十世纪90年代，基于机器学习的方法逐渐开始取代传统方法。基于机器学习的方法能够实现文本特征的自动提取，这减少了大量人工特征提取的工作。在文本分类领域，也出现了许多基于机器学习算法的分类器，如朴素贝叶斯、决策树、KNN、支持向量机(SVM)和神经网络等，这些算法在文本分类领域取得了优于传统算法的效果。

深度学习近年来在计算机视觉等领域展示了其强大的特征提取能力，各种其他领域的研究者看见了深度学习的巨大潜力。人们开始尝试将深度学习方法运用到自然语言处理的研究中，发现循环神经网络(RNN)在处理序列化问题上效果突出。随后Hochreiter和Schmidhuber提出的长短期记忆人工神经网络(LSTM)[2]，解决了传统循环神经网络对长期依赖信息处理的困难，在多种自然语言处理问题上得到广泛应用。如机器翻译领域，得益于LSTM等深度学习方法，机器翻译由从前的刻板和无法处理上下文语义和逻辑变得流畅、合理和自然，使用体验得到大幅提升。自此，深度学习成为自然语言处理研究的主流方向，许多基于RNN和LSTM的改进模型不断被提出。

* 1. 本文主要工作

本文通过研究深度学习在分类任务中的典型方法和模型，介绍了深度神经网络的结构原理和训练方法，浅析了循环神经网络在序列数据中的应用和局限性。通过详细解析LSTM网络模块的内部结构原理，结合词嵌入技术的应用，介绍了应用于情感分类的LSTM模型。并在Tang的TD-LSTM模型的基础上，引入注意力机制，并增加词性嵌入和位置嵌入两种词表示方法，设计了基于注意力机制的情感分类模型，用于目标依赖的情感分类问题。最后设计了对比实验，检验模型的可行性和有效性。

1. 深度学习概述

深度学习(Deep Learning)是机器学习(Machine Learning)的一个分支，它试图使用包含复杂结构或多重非线性变换构成的多个处理层对数据进行高层次抽象和表征学习。对于拥有多层次抽象特征的数据，深度学习使用多层次结构在各个处理层上对数据进行线性和非线性变换，来进行特征抽象。通过多层次结构的学习，我们就能够将数据转化为其特征表示，而这个过程完全不需要人工的特征提取工作。深度学习基于多层人工神经网络，称为深度神经网络，其网络结构中具备至少一个隐层神经网络。深度神经网络和浅层神经网络一样具备拟合复杂非线性数据的能力，但其隐层神经网络提供了高层次抽象能力，因此能更好的学习到数据的本质特征。自深度学习出现以来，基于深度学习的机器学习系统在很多领域，尤其是计算机视觉和语音识别中，取得了领先的效果。同时为一些其他领域的研究，如自然语言处理、文本分类等领域提供了新的思路和方法。

* 1. 人工神经网络

人工神经网络(Artificial Neural Network, ANN)是一种模仿生物神经网络的结构和功能的数学模型和计算模型，用于对函数进行估计和近似。神经网络通过大量的人工神经元结构联结进行计算，能对非线性统计数据进行建模。神经网络结构由大量的神经元组成，每个神经元按一定权重接受多个输入分量，并按照特定的偏置和激活函数给出输出。神经网络的网络拓扑结构根据具体问题进行定制，通过基于定义目标函数和反向误差传播进行优化。现代的深度神经网络通过建立深层次的网络结构，来模拟人脑的机制进行解释和分析图像、语音和文本等数据。与传统机器学习方法相比，深度神经网络通过多层网络结构来储存多层次的抽象特征，通过增加网络的广度以抽取出不同水平、不同维度的有效表示，以提高在不同抽象层次上对数据的解释能力。目前已经有多种神经网络模型在相关领域取得了极好的效果，如卷积神经网络(Convolutional Neural Network, CNN)被广泛应用于计算机视觉，循环神经网络(Recurrent Neural Network, RNN)被应用于语音识别、自然语言处理、生物信息学和序列预测等。

* 1. 人工神经元

人工神经元(Artificial Neuron)是模仿生物神经系统神经元结构的数学模型，是人工神经网络的组成单元（以下简称神经元）。神经元模型接收多个输入，并按照不同的权重(Weights)对这些输入进行求和，加上一个常数项作为偏置(Bias)后，通过一个非线性的激活函数(Activation Function)得到神经元的输出。单个神经元的作用是把一个维向量空间用一个超平面区分成两个部分（称为判断边界），当给定一个输入向量时，神经元可以判断出这个向量位于超平面的哪一边。

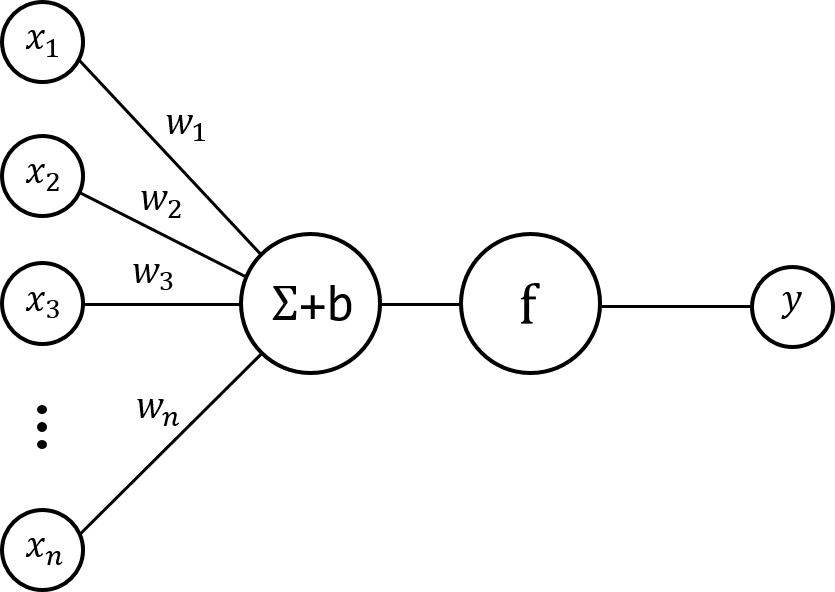


图 1：人工神经元模型

图 1为人工神经元模型示意图，输入向量，权重向量，另外需要一个常数项作为偏置，则表示激活函数。设表示神经元4结构的输出，则有：



激活函数(Activation function)

神经元使用一类非线性函数，用于计算单个神经元结构的输出值。在多层神经网络模型中，激活函数引入了非线性因素。若神经元结构仅仅对输入进行线性组合，那么多层网络将变得毫无意义，这时神经网络的输出仅仅是输入的线性组合。而线性模型的表达能力极其有限，因为大部分数据并不是线性可分的。激活函数使得神经网络在每一层都能对输入进行非线性变换，这样的神经网络才能将输入空间转换成一个线性可分的向量空间。以下为几个常用的激活函数。

Sigmoid函数 其取值范围是，定义为：



Tanh函数 它是Sigmoid函数的一种变体，取值范围是，定义为：



ReLU函数 一个简单而有效的激活函数，被广泛应用于神经网络单元，其定义为：



* 1. 多层神经网络

深度学习通过多层的神经网络结构来增强模型的学习容量。单层或几层的简单网络结构在特征表示和非线性划分上有局限性。在许多问题上，不同抽象层次可以提取出不同的特征表示，多层次的神经网络结构提供了对应于不同抽象层次的学习能力。如多层卷积神经网络(CNN)在处理图像时，先是识别不同像素、不同颜色组成的局部特征或局部结构，再由这些基本特征组合出更高层次的抽象特征，每一次抽象都是对上一层特征的总结，不同层次的网络结构可能在对图像的颜色、纹理或者宏观结构等特征进行识别。从数学角度上看，神经网络的学习就是学习如何利用多次的线性变换和激活函数的非线性变换，将原始输入空间投射到线性可分的稀疏空间，以达到分类或回归的目的。增加单层网络结构的神经元数量相当于增加维度，以增强表征能力和线性变换的能力，而增加网络层数则相当于增加非线性变换的次数，也就是提高了网络结构对输入特征的抽象能力。

我们以一个简单的三层前馈网络(Feedforward Network)为例，介绍神经网络前馈传播的计算过程。如图 2所示，网络从左到右分别为输入层(Input Layer)、隐藏层(Hidden Layer)和输出层(Output Layer)。记输入向量的维度为，隐藏层向量的维度为，输出向量的维度为。输入向量为，第一层网络的连接权重可表示为一个权重矩阵，其中表示第个输入节点到第个隐藏层节点的连接权重，第一层网络的偏置向量为。

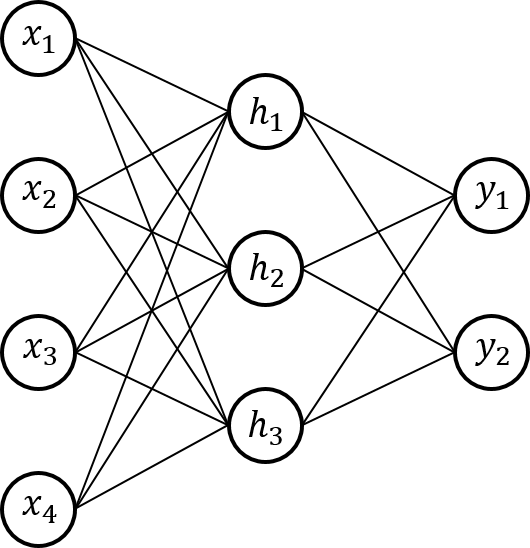


图 2：多层神经网络

假设两个处理层都使用Sigmoid函数作为激活函数，则第一层网络的输出由以下式子给出：



同理，第二层网络的权重矩阵为，偏置向量为。则第二层网络的输出由以下公式给出：



* 1. 反向传播算法

反向传播(Backpropagation)[3]是一种与最优化方法(如梯度下降法)结合使用的，用来训练人工神经网络的常见方法。反向传播算法先对网络中的所有权重计算损失函数(Loss Function)的梯度，然后将梯度反馈给优化方法，用来更新网络权值以最小化损失函数。损失函数用来衡量网络预测与真实样本的偏离程度。在线性回归模型中，常以网络预测与真实样本的均方差作为损失函数，而在二分类问题和多分类问题中，常使用交叉熵(Cross Entropy)作为损失函数。

反向传播算法主要有两个阶段：激励传播和权重更新。

1. 激励传播 在每次迭代中，传播环节包含两个步骤：
2. 前向传播阶段：将训练输入送进网络以获得网络预测。
3. 反向传播阶段：将网络预测与训练输入的目标求差，并将误差逐层反向传播到网络各层。
4. 权重更新 对于每个突触上的权重和神经元偏置，按照以下步骤进行更新：
5. 根据反向传播的误差项计算出对应于该权重的梯度项。
6. 将该梯度取反并乘上一个比例加到权重上。

控制权重更新的比例会影响到训练过程的速度和训练效果，这个比例称为学习率(Learning Rate)。因梯度的方向指向误差扩大的方向，因此在权重更新的时候需要对其取反，从而让权重向误差变小的方向偏移。

* 1. 多分类模型

在多分类问题中，我们需要神经网络在接收一个输入样本后，输出相应的分类预测。对于多分类问题，神经网络最后一层神经元的数量与分类类别数量相等。我们可以直观得想到，将最后一层的每个神经元都对应到一个类别，则输出值最大的神经元所对应的类别即是神经网络的预测类别。但我们通常会在网络的输出层增加一个Softmax层，将输出向量归一化。

Softmax函数

Softmax函数也称为归一化指数函数，它能将一个含任意实数的维向量压缩到另一个维实向量中，使得每一个元素的范围都在之间，并且所有元素的和为1。在神经网络的输出层上，设神经元的输出是一个维向量，则Softmax层的输出向量由以下函数给出：



交叉熵(Cross Entropy)

交叉熵是信息论中的概念，而在神经网络中，我们可以用交叉熵来衡量两个概率分布的偏离程度，也就是作为损失函数。在多分类问题中，我们通常把样例的类别表述成一个概率分布。如三分类问题中，若某样例的类别是，则此样例的类别表述成概率分布就是。将此样例的输入送入神经网络，神经网络通过Softmax层输出了另一个概率分布，假设为，作为神经网络对该样例的类别预测。设训练样例的实际分布为，神经网络的输出分布为，和都是维向量，则根据交叉熵的定义，有：



综合以上，在多层神经网络的输出层上增加Softmax层，使用交叉熵作为损失函数，通过梯度下降和反向传播对网络权重进行优化，我们便得到了一个简单而典型的基于神经网络的多分类模型。在自然语言处理中，我们也使用这种基本架构建立文本分类器，但简单的前馈网络结构并不能很好地处理文本序列数据，所以在此基础上，我们需要一些适用于自然语言处理的技术。

1. 基于深度学习的文本情感分类模型

文本情感分类也就是文本情感极性分析，其基本步骤就是对文本进行两极性或多极性分类。分类的作用是判断出此文字中表述的观点是积极的、消极的，还是中性的情绪。在文本情感分析领域，早期的研究贡献来自于Turney[4]和Pang[5]，他们使用了多种方法来探测商品评论和电影评论的两极观点。其中，基于统计机器学习的情感分类方法被广泛使用，其中包括词袋模型(Bag-of-words, BOW)、朴素贝叶斯(Naive Bayes)、隐马尔科夫模型(HMM)、支持向量机(SVM)等。

随着深度学习在计算机视觉和语音识别领域的成功应用，越来越多的深度学习方法也被应用于情感分析等自然语言处理问题上。自词嵌入(Word embedding)[6]技术和长短期人工神经网络(LSTM)出现后，它们被广泛应用于自然语言处理的各种任务中，基于深度学习的情感分类方法开始大量出现，成为主流。

* 1. 词嵌入

词嵌入(Word Embedding)是自然语言处理中语言模型与表征学习技术的统称。它指把一个维数为所有词的数量的高维空间嵌入到一个维数低得多的连续向量空间中，每个单词或词组被映射为一个实数域上的向量。词嵌入的主流方法包括人工神经网络(ANN)和概率模型。词嵌入技术最早出现在2000年，Bengio等人在一系列论文中使用了神经概率语言模型(Neural probabilistic language models)[6]使得机器“习得词语的分布表示(learning a distributed representation for words)”，从而降低了词语表示空间的维度。2013年，谷歌一个由Tomas Mikolov领导的团队发明了一套工具word2vec来进行词嵌入[7]，训练向量空间模型的速度比以往的方法都快。在神经网络模型中，使用词嵌入来表示词组的方法极大得提升了包括情感分析在内等自然语言处理任务的效果。在词嵌入技术之前，在自然语言处理中，词的表示通常使用one-hot向量，即高维度的词语同现向量。但one-hot形式的词语表示只体现了词语之间的区分，而丢失了词语之间语义上的联系。而词嵌入是将词语投射到了一个低维的连续向量空间，每个维度相当于描述了词语在不同方面或角度下的区别和联系，所以词嵌入能更好地使用数值来表征了词语在上语义的信息。

* 1. 循环神经网络

循环神经网络(Recurrent Neural Network, RNN)是一种通过在各网络单元之间建立连接从而形成内部循环的神经网络模型。典型的全连接神经网络结构仅仅将不同的输入特征做组合，而不考虑这些特征的顺序信息，也就是说在时间维度上这些输入是相互独立的，这对于序列数据的处理来说具有很大的局限性。对于自然语言处理而言，词语的顺序信息对于语义理解是非常重要的。就如我们阅读一篇文章一样，对句意的理解要考虑前后词的词义，对文章的理解要考虑上下文的线索，我们不会脱离上下文信息去独立地判断某个词语或某段文本的含义。传统的神经网络很难处理这类任务，而循环神经网络解决了这个问题，其内部的循环结构使得网络可以保存之前的记忆，让信息持久化。

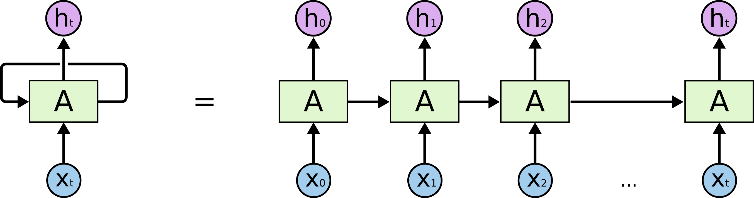


图 3：展开的循环神经网络

如图 3所示，输入是一组序列数据，按照顺序依次输入到网络中，每一次输入除了对应一次输出之外，还会产生对应的中间信息，这个中间信息会传递给下一步。循环神经网络可以被看作是同一神经网络经过多次复制得到，每一份神经网络的副本在接收前一步骤的中间信息后，结合当前步骤的输入给出对应的输出，并再产生新的中间信息传递给下一个副本。

* 1. 长短期记忆人工神经网络(LSTM)

长短期记忆(Long Short-term Memory)是一种特殊的循环神经网络，于1997年首次由Hochreiter和Schmidhuber提出[2]。与传统的循环神经网络不同，LSTM非常适合于处理和预测时间序列中间隔和延迟较长的重要事件。LSTM在设计上针对传统循环神经网络可能出现的梯度消失和梯度爆炸问题做出了相应的改进，使得网络能轻易地保留长期的依赖信息。

长期依赖信息

在某些情况下，如我们想要预测句子“汽车行驶在[路上]。”中最后一个词，我们不需要太多的上下文信息便能预测出这个词语。在这种情况下，当前处理位置与需要的相关信息所在位置距离较小，普通的循环神经网络足以应付这样的任务。但在更多的文本处理问题中，我们期望的预测可能与更早或更晚的上下文信息相关。比如我们想要预测句子“我从小在中国长大，...，我能讲一口流利的[中文]”中的最后一个词。附近的上下文信息告诉我们，这个词可能是某种语言，然而我们想确定它到底是哪一种语言时，我们就需要“中国”这个更早以前的上下文信息。在实际问题中，相关上下文信息的位置与当前处理位置的距离可能非常远。

梯度消失与梯度爆炸

传统循环神经网络在处理长序列时，随着相关信息的距离不断增大，其处理这些信息的能力就会不断减弱。当相关的信息处于较远距离时，由于多层网络的多次变换和新信息的不断输入，这些处于较远距离的信息对目前位置的影响变得非常小。从循环神经网络的结构原理及其训练方法的角度分析，信息丢失的原因是由于网络在训练时遭遇了梯度消失(Vanishing Gradient Problem)或梯度爆炸(Exploding Gradient Problem)。

在深层网络的训练中，影响梯度大小的因素主要有网络权重和激活函数的偏导。多层网络中各层的梯度使用链式法则连乘计算得到，当这些激活函数的偏导数小于1的时候，那么梯度随着距离的增加会很快变成0，也就是距离较远的网络遭遇了梯度消失；相反，如果这些激活函数的偏导数大于1，那么相应就会出现梯度变得极大，也就是梯度爆炸。不管是梯度消失还是梯度爆炸，都会使得计算出的梯度失去其原本的训练意义，这将使得网络无法按照预期进行训练。

LSTM得益于其特殊的网络模块设计，轻松地避免了以上问题。循环神经网络由重复的神经网络模块形成链状结构，传统循环神经网络的重复模块结构较为简单，例如只有一个Tanh层。LSTM同样拥有这样的链状结构，但其重复模块的结构与传统循环神经网络有所不同，在最经典的LSTM模型中，它的每个重复模块中有4个神经网络层，这些网络层之间的交互方式非常特别。

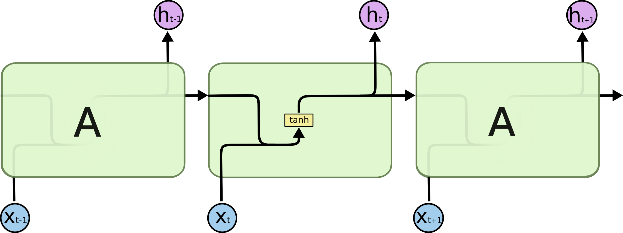


图 4：传统RNN网络模块的内部结构

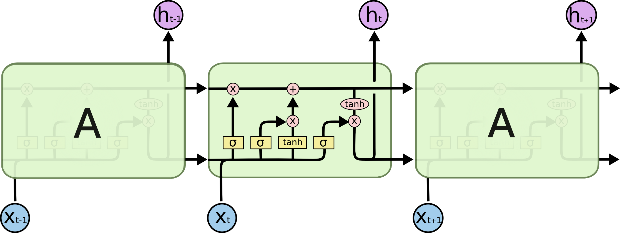


图 5：典型LSTM网络模块的内部结构

图 5显示了典型LSTM重复模块的内部结构。线条代表了信息的传递，在这里，信息被表示为一个固定维数的向量。粉红色圆圈表示对向量的元素级操作，如相加或者相乘。黄色方框代表神经网络层。线条的合并表示向量的级联，线条分叉表示向量复制。

细胞状态

LSTM能够轻易保留长期信息的关键在于它并不对序列信息直接进行重复的变换，而是引入一个叫细胞状态(Cell State)的概念。细胞状态通常也被表示为一个相同维数的向量，这个向量将直接贯穿整个LSTM链，每个重复模块只会对它进行一些线性交互，所以细胞状态上携带的某些重要信息可以很容易地流过而不发生改变。

门结构

LSTM通过一种叫门(Gate)的结构对细胞状态上的信息进行删除或添加。使用门结构是一种让信息选择性通过的方法，它们由一个Sigmoid神经网络层和一个元素级相乘操作组成。Sigmoid层会输出值在之间的向量，其中每个值代表对应的信息应该通过多少，0值表示该信息不能通过，1值表示让该信息完整通过。

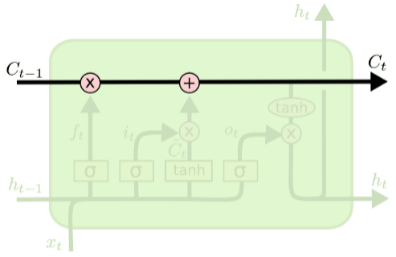


图 6：细胞状态

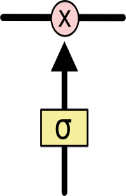


图 7：门结构

LSTM网络模块详解

每个输入对应的LSTM模块首先接收来自上一次输入所得到的细胞状态和输出信息，当得到当前输入之后，LSTM根据和来决定要从细胞状态中去掉哪些信息，这一步由一个Sigmoid层控制，这一层称为“遗忘门(Forget Gate)”。

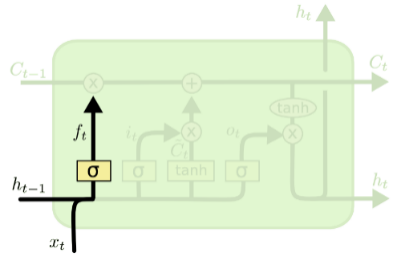


图 8：遗忘门网络层

记该门结构的控制向量为，由以下式子给出：



接下来，LSTM将决定把哪些信息添加进细胞状态中。这一步需要计算出可能添加进细胞状态的候选信息和用于决定哪些候选信息能最终添加进细胞状态的“输入门(Input Gate)”的控制向量。

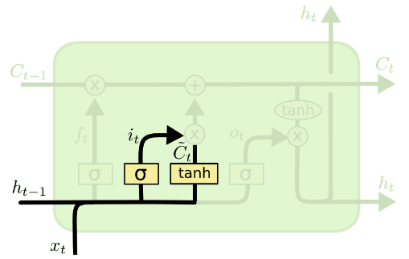


图 9：输入门网络层

和由以下式子给出：





现在我们可以将旧细胞状态更新到新状态了。我们把旧状态乘上（注意这里指各元素对应相乘，即），忘掉那些我们需要忘记的信息，然后加上，这样便得到了新的细胞状态，这个新的细胞状态会继续传递到下一步的网络。

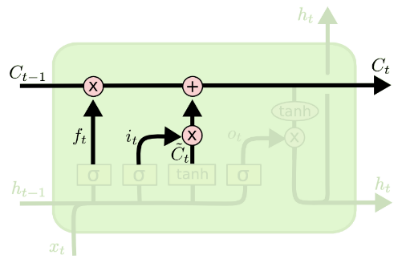


图 10：更新细胞状态

细胞状态的更新公式为：



最后，我们要决定最终的输出。输出会在新的细胞状态的基础上加入一些过滤。在这一步，我们会再使用一个Sigmoid层计算出“输出门(Output Gate)”的控制向量，然后将通过Tanh函数将数值调整至之间，最后将与之相乘即可得到最终的输出。

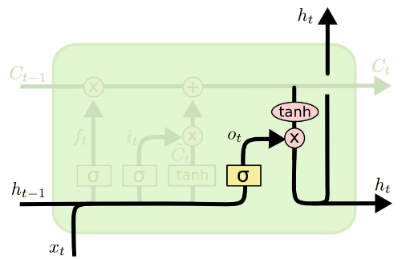


图 11：输出门网络层

输出向量的计算公式：





这种特殊的网络模块设计使得LSTM能够处理长序列数据，如文本数据。对于情感分类问题，使用LSTM能更好地处理上下文语义和情感线索，这使得LSTM逐渐成为情感分析领域的一种主流模型。

* 1. LSTM文本分类模型

结合词嵌入技术和LSTM神经网络，我们就能构建一个可用于处理文本分类问题的神经网络模型了。我们通过词嵌入技术预先训练出一个词典，每一个词语有对应的语义特征向量。文本数据通过查找词典将词语转换为向量作为网络输入数据。设一个文本由个单词组成，转换后的文本数据可表示为，其中的每个元素都是一个固定维数的列向量，设向量的维数为。我们依次将这些向量输入到LSTM单元数量同样为的网络中，LSTM单元依次对这些语义向量进行解析，前文依赖信息通过细胞状态在LSTM单元之间传递，在首个LSTM单元中，细胞状态被初始化为一个零向量。在最后一个LSTM单元给出输出后，这个输出将被作为整个句子的语义特征向量。

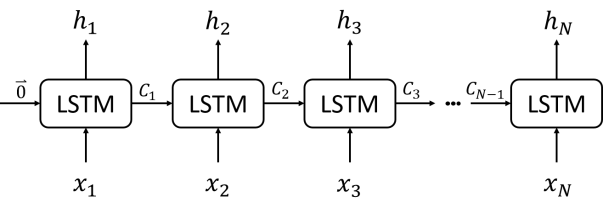


图 12：文本分类模型-LSTM层

最后一步的输出向量包含了对整个句子语义的理解和概括，这相当于我们已经提取出了与词语顺序无关的语义特征，这时我们再把这个特征向量当作全连网络层的输入，如送入两层全连网络，再通过Softmax对网络输出归一化，便能用前文所述的方法对文本进行分类了。

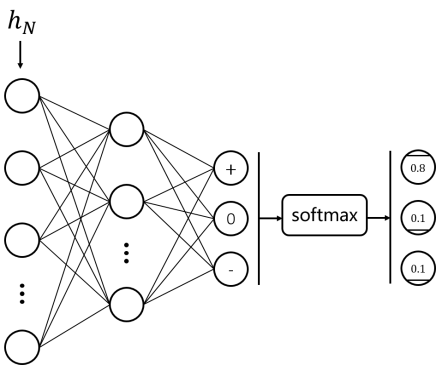


图 13：文本分类模型-全连接层与Softmax层

上述的模型只从一个方向对文本进行解析，这代表这网络时基于正序对文本进行语义理解和推理。但有些情况下，自然语言的语义理解和推理可能基于逆向的语序，或者在文本区域的语义是基于后文的线索。这时，如果只考虑一个方向的LSTM解析将会丢失另一些角度语义理解。所以我们在此基础上，再加入一个结构相同的LSTM链，但它将对文本进行反方向的解析。这样，我们就会得到文本的两个语义向量，我们可以用许多方式将这两个向量组合起来，如求和、平均或级联，然后再送入全连网络。我们把这样的模型称为双向LSTM(Bidirectional LSTM/Bi-LSTM)。

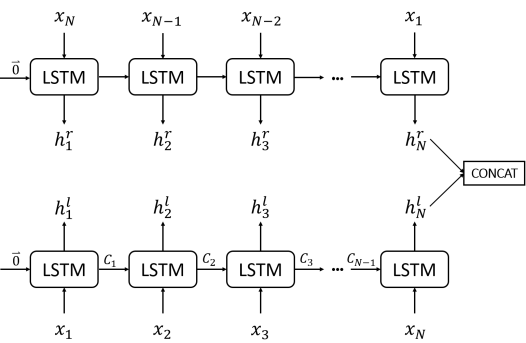


图 14：双向LSTM网络

* 1. 注意力机制

注意力机制(Attention Mechanism)源于大量对人类和动物视觉系统的研究。对人类和动物视觉系统的研究发现，我们的视觉系统仅会将注意力集中在一个特定的区域便能提供足够的信息以做出相关的响应。被视觉系统关注的区域图像会更加的清晰，而周围的图像会变得模糊。我们的视觉系统会根据相关信息控制注意力区域的位置，并且随着时间不断改变位置。这个机制对于计算神经科学有很大影响，因为面对大部分任务，我们需要正确地选择出相关性更高的局部信息，抛去无关信息的干扰，才能更好的做出推断和反应。

注意力机制在人工神经网络上的应用已经有一段较长的历史，特别是在计算机视觉领域，已经有许多问题利用注意力机制提升了神经网络的性能。直到近期，才有相关的工作将注意力机制结合到RNN中，将其应用到NLP领域[8][9]。令人兴奋的是，注意力机制在NLP领域同样带来的显著的效果提升。

1. 基于目标的情感分类模型设计

基于目标的情感分类不仅考虑文本内容，而是根据给定的目标对象来识别与之相关的情感信息来做出情感极性判断。对于描述对象单一的简单文本来说，情感极性通常仅有一个。而在实际情况中，一些文本常常描述了不止一个对象，对不同对象的情感倾向也不同。如文本“这家餐厅味道不错，就是服务态度很差！”中出现了两个情感表达的对象：“味道”和“服务态度”，文本对“味道”的情感倾向是正面的，而对“服务态度”的情感倾向是负面的。考虑目标对象使得情感极性的判断更加准确。在实际应用当中，我们也常需要文本对不同目标对象的情感分类结果。

在基于词嵌入和LSTM的基础上，Tang提出的TD-LSTM[1]方法将文本从目标词位置处切为两段文本，将这两段文本分别使用两个不同方向的LSTM网络处理，两个网络都将在最后阶段处理目标词，使得LSTM网络的预测对目标词具有更高的相关性。实验结果表明，通过考虑目标词的出现位置来设计双向LSTM的网络结构能得到优于传统机器学习方法和双向LSTM模型(Bi-LSTM)的分类效果。除此之外，Tang在文章中提出将目标词的向量表示连接到文本的向量表示的方法TC-LSTM[1]来增强网络预测与目标的关联性，并取得了更好的分类效果。

基于目标的情感分类需要更多的考虑目标词语部分，我们不能简单的考虑文本的整体语义，我们希望模型能在文本中自动挑选某些与之相关的重点信息，而抛弃不相关和有干扰的无关信息。这与人们在阅读时有着相同的机制，我们会将注意力集中在某些关键词上。而通常，词语的词性(Part-of-Speech, POS)也扮演着重要的角色，我们通常更加关注于形容词、副词和动词。这些信息对于我们的情感极性判断有着重要的影响。

所以，本文在以上基础上，引入注意力机制，提出基于注意力机制的LSTM网络模型。注意力机制在考虑目标词的情况下，将帮助网络在各步骤的LSTM输出中选择相关性更高的信息。我们在LSTM网络之上增加注意力层，用于计算LSTM各步骤输出的影响权重。除此之外，本文还在词的向量表示中增加词性嵌入(POS embedding)和位置嵌入(Position embedding)的向量表示，结合Tang的情感词向量SSWE[10] 数据集作为网络模型的输入。在数据处理阶段，我们先使用NLTK[[3]](#footnote-2)工具对词语的词性进行标注，我们将标注出名词、形容词、副词、动词和其他，这几种词性。同时还会标注出词语的位置信息，这个位置信息是文本词与目标词的相对位置，它描述了词语与目标词的距离和前后位置信息。随后在训练中，我们将三种词向量连接，形成维度更高的词向量表示，它包含了更多与目标相关的信息。

* 1. 词表示

模型的第一步是将文本中的每个单词转换为对应的词嵌入向量，作为模型的词表示。词表示包括三个部分：SSWE情感嵌入向量、词性嵌入向量(POS embedding)、位置嵌入向量(Position embedding)。

SSWE情感嵌入

使用前文所述的SSWE情感向量数据集，建立情感嵌入矩阵，其中为情感向量的维度，是一个固定大小的词表，记录着所有可能在文本数据中出现的单词。中每一列都是某个单词的对应情感嵌入向量。设一个句子由个单词组成：。我们通过对矩阵查表把每个单词转换成一个对应的实向量。单词的情感嵌入向量表示为：



其中是一个one-hot向量，代表着单词对应的嵌入向量在矩阵中的位置，如向量代表着单词对应的嵌入向量在矩阵的第3列。可作为网络参数参与训练，作为超参数调整。

词性嵌入

使用NLTK对文本词进行词性标注，我们只区分名词、副词、形容词、动词和其他，这几种词性。建立词性嵌入向量矩阵，其中为词性嵌入向量的维度，是一个固定大小的词性类别集合。和上面转换一样，我们通过查表方式得到单词的词性嵌入向量：



位置嵌入

对于每个句子，我们先标定每个单词与目标词的相对位置。如文本“I love this moive so much !”，其中“movie”为目标，则对应的原文本中每个词的位置编码为。建立位置嵌入向量矩阵，其中为位置嵌入向量的维度，由于使用定长模型，所以为一个固定大小的位置编码集合。单词的位置嵌入向量同样通过查表得到：



单词的词表示由以上三个嵌入向量连接而成，则单词的最终词表示向量为：



当句子的所有单词都完成转换后，我们便得到了整个句子的词嵌入向量组。SSWE情感嵌入矩阵使用Tang的数据集作为初始化值，而词性嵌入矩阵和位置嵌入矩阵使用随机数值进行初始化。

目标词与文本词作相同的处理，将目标词三种嵌入向量连接得到目标词的词表示。当目标对象由多个词组成是，我们简单地对他们的词表示取平均。

* 1. LSTM-ATT模型

词嵌入向量表征了词语的语义信息和情感相关的信息，接下来我们要对整个句子进行语义和情感表征，得到用于表征句子语义和情感信息的句子表示。句子的语义和情感信息解析使用LSTM网络，LSTM网络对每个词向量按照顺序进行解析，最终输出一个代表句子语义和情感的向量。但普通LSTM或者Bi-LSTM在解析句子时，没有考虑目标词的语义和情感信息。而Tang的TD-LSTM和TC-LSTM仅简单的从目标词的位置考虑，使得语义解析于目标词的语义具有更高的相关性。但事实上，句子中对情感极性判断有关键影响的词可能出现在任何位置，而不一定出现在目标词的附近。为了解决模型在这方面的局限性，我们在LSTM网络中引入注意力机制。

* + 1. 双向LSTM网络

模型的情感预测不仅与前文的语义线索有关，还与后文的语义线索有关。所以我们使用双向LSTM网络作为网络模型的基础。其中一个LSTM网络由左至右解析输入向量组，并在每个LSTM步骤中输出该步骤的输出向量，最终该LSTM网络得到一个输出向量组。而另一个LSTM网络将输入倒置，即，然后每个LSTM步骤解析出输出向量，最后得到另一个输出向量组。

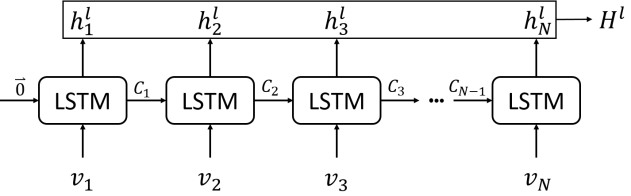


图 15：其中一个方向的LSTM网络

* + 1. 注意力层

注意力机制的核心在于计算出每个LSTM输出的影响力权重，权重是一个概率分布，也称为注意力权重(Attention Weights)。然后使用该权重多所有的LSTM输出进行按权加和，得到最终的句子表示向量。注意力权重的计算需要考虑各步LSTM输出和额外的参考信息，在这里，这个参考信息就是目标词的词表示向量。我们用目标词向量和各步骤LSTM输出进行交互计算，每个步骤得到一个实值，最后通过Softmax函数将所有实值归一化成概率分布。其中目标词向量和LSTM输出向量的交互计算有多种方式，以其中一个方向的网络输出向量组为例，介绍以下两种常用的计算方式：

点积 通过计算目标词向量和各步LSTM输出的点积得到一个标量，再通过Softmax函数将其归一化：



Bilinear 通过一个可训练的参数矩阵，将中各列向量做一次线性变换后，再与计算点积，最后通过Softmax函数归一化：



计算出注意力权重之后，就能对中的各列向量进行加权求和了，求和之后的到一个维度与输出向量相同的句子表示向量，由以下公式给出：



以同样的方法我们可以计算出另一个网络的注意力权重和它的句子表示向量。通过连接两个句子表示向量，得到完整的句子表示向量：



注意力层在模型中的运行方式如下图所示：

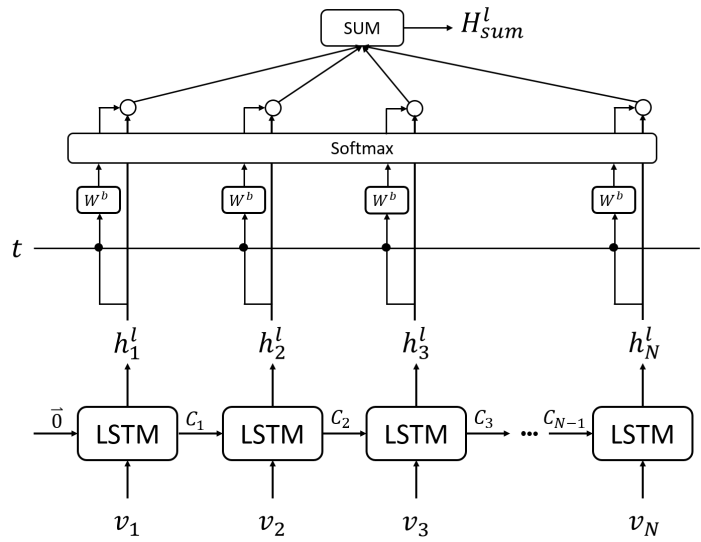


图 16：LSTM层和注意力层

* + 1. 情感分类预测

我们把送入最后的全连网络层，全连网络层的输出层输出一个维度为3的输出向量，分别对应三个类别。输出向量最后经过一个Softmax层给出网络预测。我们使用随机梯度下降法最小化交叉熵来训练网络参数。

* 1. 防止过拟合

过拟合(Overfitting)

当我们设计一个模型的参数数量过多时，常常会出现过模型被某些噪声干扰而导致模型失去了对数据普遍特性的概括能力，也就是失去了一般化的能力。一个足够复杂的极端模型可以完美地适应训练数据，但在测试数据上的表现则非常差，不能做一般化扩展。而神经网络往往拥有成千上万的参数，拥有如此庞大数量参数的复杂模型，极容易遭遇过拟合现象，神经网络的过拟合过程也被称为过度训练(Overtraining)。诚然，数据的质量也是影响模型一般化能力的因素，若数据集不够大或者噪声太多，也会使得训练出的模型出现过拟合问题，但数据集的质量我们一般不好把控，数据集的大小也受限于人工标注的效率。另一种可能的途径是减小网络复杂度，降低网络参数规模，但通常我们不愿意这么做，因为复杂的大型网络比小型网络具有更大的潜力。

所以，应对神经网络过拟合的现象，我们通常需要使用一些手段来优化神经网络模型以减少过拟合现象的发生。本实验对模型使用了两种较为常用的手段来防止过拟合现象。

L2-正则化

正则化(Regularization)是通过在模型中增加额外的正则化项来防止模型过拟合的手段。有些正则化手段是在模型上添加限制，有些是对模型参数做平滑处理，而有些则是在目标函数上添加正则化项。本实验模型使用的正则化手段是L2-正则化(L2-Regularizaton)，这一种在目标函数上添加正则化项的方法，也称为权重衰减(Weight Decay)。我们模型的目标函数使用交叉熵，梯度下降法中计算交叉熵的公式为：



而添加了正则化项的目标函数则为：



其中是正则化项，他是网络中所有权值（不包括偏置）的平方和，正则化参数控制权重衰减的程度。正则化项的作用是让网络偏好学习较小的权重，而不只是一味地最求最小化损失。直观上来说，网络权重较小的情况下，样本中的噪声更不容易对网络产生太大的影响。网络权重较大时，模型能更容易的适应训练数据，也更容易学到局部的噪声。

Dropout层

Dropout[11]是另一种用于防止神经网络过拟合的有效方法。这个方法是在每次训练时，都随机地让一些神经元失效（不包括输出层神经元），也就是在前向激励时，这部分神经元将不输出激励信号。这相当于在训练时，对网络结构进行适当的剪纸，降低网络的结构复杂度。随机失效的过程由参数控制，给出神经元保留输出的概率。而在测试网络性能时，我们仍然以完整的网络结构来计算输出。

本实验在模型现实时，将加入以上两种方法用于防止过拟合。L2-正则化将应用于网络中除偏置以外的所有网络权重，而实现Dropout的方法时在隐层网络神经元的输出后面增加一层随机过滤层。

* 1. 模型训练

神经网络的训练使用梯度下降和反向传播。当我们使用训练集进行训练时，可以每次向神经网络送入单个训练样本，但这样单样本训练会带来问题。神经网络在训练时无法得到训练样本的一般特性，而是每次随着各自训练样本的梯度方向修正，导致神经网络的训练失去一个稳定的方向，目标函数难以收敛。要使神经网络学习到训练数据的普适特性，训练时可以每次使用全训练集来训练网络，并求得全样本的平均误差，这样计算出的梯度方向会引导神经网络参数向更准确的方向修正。但面对大规模数据集时，进行全数据集训练不仅会带来巨大的内存消耗和更慢的训练速度，而且可能将一些正负样本误差抵消，从而影响神经网络对这些样本的特征学习。

所以，我们在训练时可以选取一个折中的方案，也就是Mini-Batch的训练方式。我们在训练神经网络时，每次在训练集中随机选取一定数量(Batch Size)的样本，对网络进行训练。适当的Batch Size不仅能提升训练速度，也能较好的代表训练数据的普适特性。

1. 实验验证

基于前文的理论基础和模型设计，我们通过深度学习引擎Tensorflow实现和训练多个LSTM模型，包括本文提出的LSTM-ATT模型。通过对比和分析实验结果检验该模型的可行性和有效性。

* 1. 实验环境

Tensorflow

模型的实现使用Google的开源深度学习框架Tensorflow。Tensorflow是一个面向机器学习的开源软件库，最初由Google Brain团队开发，用于Google的研究和生产，后于2015年11月发布开源版本。Tensorflow的前代是Google Brain开发的第一代机器学习系统DistBelief，它已经为Google的多个商业产品提供了技术支持，包括Google搜索、Google语音搜索、广告、Google相册、Google翻译和Youtube。Google的科学家对DistBelief进行简化和重构，使其变成一个更快、更健壮的应用级代码库，形成了Tensorflow。

Tensorflow可以运行在多个CPU和GPU上，支持分布式部署，甚至能运行在智能手机等移动计算平台上。Tensorflow的计算使用有状态的数据流图(Graph)表示，数据流(Data Flow)会在数据流图中传递，每个数据流图的节点代表对数据的计算操作。Tensor表示流动在数据流图中的数据张量，用于装载多维数组和负责数据的计算操作。在实际运行中，Tensor仅作为数据流图的数据容器。Variable用于维护实际的数据状态。在根据模型建立好数据流图之后，Tensorflow使用Session管理和启动，它负责将计算任务和数据分发到不同的CPU和GPU上。由于Tensorflow支持自动微分，所以在建立好模型的数据流图后，直接使用已经封装好的参数优化方法便能快捷方便的实现模型的训练。

使用Tensorflow，我们可以快速便捷地实现和训练深度学习模型。

* 1. 实验设计

我们使用Tensorflow实现并训练了4个情感分类模型：Bi-LSTM、TD-LSTM、TC-LSTM和LSTM-ATT。设置不同的参数组合，通过分析模型在测试集上的预测效果，检验模型的分类性能。

* + 1. 数据集

数据集原文使用Twitter和微博的短文本，每个数据样本包含三个部分：文本、目标和对应于目标的情感类别（分为正面、中性和负面）。本数据集的目标均在原文本中出现，由一个单词或若干个单词组成。

Twitter英文数据

数据原文是通过关键词和Twitter API爬取的英文文本。每个文本样例由人工标注对于关键词对象的情感类别。其中训练集包含6,248个数据样本，测试集包含692个数据样本。训练集和测试集中，正面、中性和负面情感类别的比例均为25%，50%，25%。

微博中文数据

数据原文是通过关键词和新浪微博API爬取的中文文本，由人工标注情感类别。由于中分分词不是本文的研究重点，所以中文数据预先使用Jieba[[4]](#footnote-3)中文分词工具对文本进行了分词。此数据集包含6608个数据样本，其中正面、中性和负面情感类别的比例为21%，57%，23%。我们在此数据集中随机抽取90%的样本作为训练集，剩余的其他样本作为测试集，同时保持训练集和测试集的情感类别比例与原数据集大致相同。

SSWE词向量数据

本文实验使用SSWE(Sentiment-Specific Word Embedding)词向量数据集[[5]](#footnote-4)作为配合Twitter英文数据集使用的词义特征向量数据。

* + 1. 数据预处理

数据集的文本、目标和类别都以原文字符储存在纯文本文件中，在将数据送入模型前，我们需要先将数据集转换为方便操作的数据结构。

每个数据样本占三行，第一行为文本原文，目标对象的描述使用占位符$T$替换，第二行为目标对象，第三行为情感类别，其中1表示正面，0表示中性，-1表示负面。

表格 1：数据文本格式

|  |
| --- |
| I love $T$ as a whole with all its services  google  1 |
| Man I hate that $T$ song and its playing everywhere I go  lady gaga  -1 |
| LONDON - $T$ said the international community would not wait indefinitely for Iran  hillary clinton  0 |

词向量数据文件中，每个词占一行，每一行中包含一个单词和该单词所对应的词义特征向量。我们需要先为词向量数据的所有单词建立对应序号(ID)，并将单词字符、序号和对应的实向量建立查询字典，然后通过查找字典将数据文本和目标的单词转换为对应的词向量。将类别转换为一个三分类的分布向量，如正面类别使用向量表示，中性类别使用向量，负面类别则使用向量表示。最终每个数据样本是一个由文本词向量组（或使用文本词的序号组）、目标词向量组和类别分布向量组成的一个数据体。这样，我们就完成了数据的数值化，方便后续在神经网络模型上的使用。

* + 1. 模型核心代码

通过对嵌入矩阵查表获取词向量表示

|  |
| --- |
| left\_sentence\_emb = tf.nn.embedding\_lookup(embeddings, left\_sentence\_ids)  right\_sentence\_emb = tf.nn.embedding\_lookup(embeddings, right\_sentence\_ids) |

双向LSTM网络（两个方向代码相似，仅展示一个方向代码）

|  |
| --- |
| # 生成LSTM模块  left\_lstm\_cell = tf.nn.rnn\_cell.BasicLSTMCell(lstm\_size)  # 封装Dropout层  left\_lstm\_cell = tf.nn.rnn\_cell.DropoutWrapper(cell=left\_lstm\_cell,  input\_keep\_prob=keep\_prob)  # 生成LSTM循环链  left\_lstm\_outputs, left\_lstm\_states =tf.nn.dynamic\_rnn(cell=left\_lstm\_cell,  inputs=left\_sentence\_emb,  dtype=tf.float32,  time\_major=True) |

注意力层（两个方向代码相似，仅展示一个方向代码）

|  |
| --- |
| # 注意力权重矩阵  left\_attention\_weights = tf.Variable(  tf.random\_uniform(shape=attention\_weights\_shape, minval=-0.1, maxval=0.1),  dtype=tf.float32  )  # LSTM输出和目标向量交互计算  left\_attention\_outputs = tf.batch\_matmul(  tf.transpose(left\_lstm\_outputs, perm=[1, 0, 2]),  left\_attention\_weights  )  left\_attention\_outputs = tf.batch\_matmul(  tf.transpose(left\_attention\_outputs, perm=[1, 0, 2]),  target\_mean\_emb  )  # Softmax函数归一化  left\_attention\_softmax = tf.nn.softmax(  tf.transpose(left\_attention\_outputs, perm=[0, 2, 1])  )  # 对LSTM加权求和  left\_attention\_lstm\_output = tf.reshape(  tf.batch\_matmul(left\_attention\_softmax, left\_lstm\_outputs),  shape=[-1, self.embedding\_size]  ) |

全连层和Softmax层

|  |
| --- |
| # 连接两个LSTM-ATT模型的输出  lstm\_output = tf.concat(1, [left\_attention\_lstm\_output, right\_attention\_lstm\_output])  last\_layer\_weight = tf.Variable(  tf.random\_normal([self.embedding\_size \* 2, self.num\_class])  )  last\_layer\_biases = tf.Variable(tf.zeros([self.num\_class]))  last\_layer\_output = tf.sigmoid(  tf.matmul(lstm\_output, last\_layer\_weight) + last\_layer\_biases  ) |

* + 1. 模型检验标准

为了评判情感分类模型的分类效果，我们在模型训练完成后，通过计算模型在测试集上的预测准确度(Accuracy)来衡量模型的分类性能。准确度被定义为模型在测试集中正确预测的样本占所有样本的比例。

* + 1. 超参数设置

Twitter英文数据集

词表示中使用50维和100维的SSWE情感嵌入向量、10维的词性嵌入向量、10维的位置嵌入向量。LSTM网络的隐层节点数量与词表示向量维度相同。LSTM网络处理的句子步长固定为40（即固定处理句子的40个单词）。L2-正则化项参数设置为0.001。 Mini-Batch的大小设置为64。学习率设置了0.003、0.03、0.1、0.3和0.6。LSTM网络Dropout层的保留概率参数设置了0.5、0.6和0.8。模型每隔100步训练迭代进行一次模型检验。

微博中文数据集

词表示中使用50维和100维预训练的中文情感嵌入向量、10维的词性嵌入向量、10维的位置嵌入向量。LSTM网络的隐层节点数量与词表示向量维度相同。LSTM网络处理句子的步长设置为32。L2-正则化项参数设置为0.001。 Mini-Batch的大小设置为64。学习率设置了0.003、0.03、0.1、0.3和0.6。LSTM网络Dropout层的保留概率参数设置了0.5、0.6和0.8。模型同样每隔100步训练迭代进行一次模型检验。

* 1. 实验结果和分析

根据以上超参数设置，我们使用了所有参数组合对4种模型进行训练，得到一些模型的检验结果。为了能清楚分析不同超参数如何影响模型的训练效果，我们每次选取一项超参数，计算模型在此项超参数值相同时，其他参数模型的平均效果。

使用50维与100维SSWE词向量训练效果对比

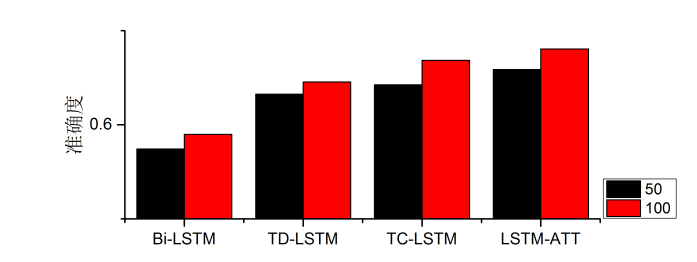


图 17：词向量维度对比(Twitter数据)

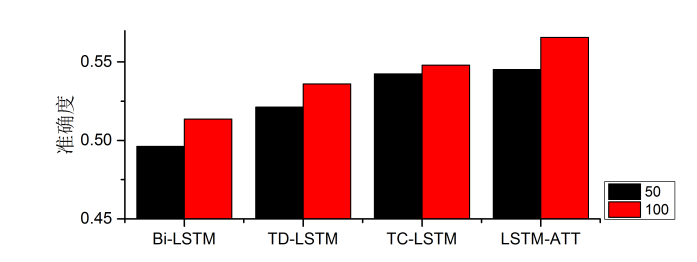


图 18：词向量维度对比(微博数据)

由图 17结果可以看出，情感词向量的维度对模型训练效果的影响较为明显。所有模型在100维情感词向量下均取得更优的训练效果。这说明高维度的词向量能够表征更多词语的语义或情感特征，同时也使得LSTM隐层节点数量增加，这增强了模型对文本的情感语义特征的抽象和提取能力。

使用不同Dropout保留概率的训练效果对比

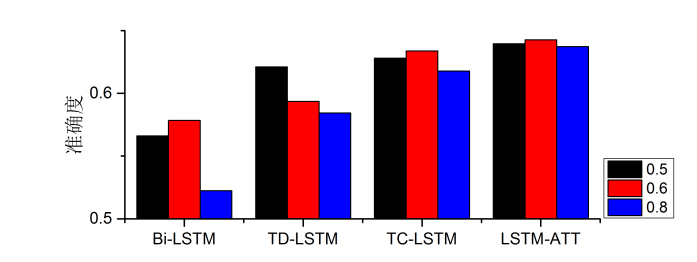


图 19：Dropout参数调整(Twitter数据)

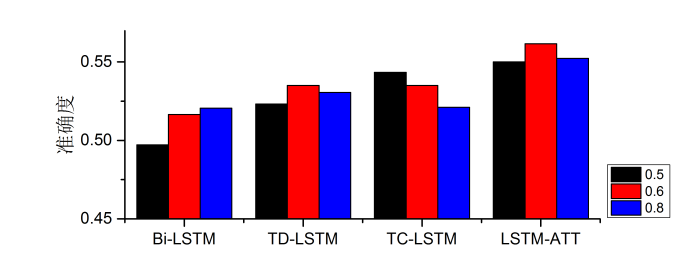


图 20：Dropout参数调整(微博数据)

通过图 19看出，不同的模型分别在不同的Dropout参数下取得最好的训练效果。但总体上，保留概率达到0.8时，所有模型的训练效果都出现了下降。这可能由于在保留概率较大时，基本保留了网络较完整的复杂性，可能导致了过拟合现象的发生。而Dropout参数的最合适数值可能有依赖于网络本身的结构特性。

使用不同学习率的训练效果对比

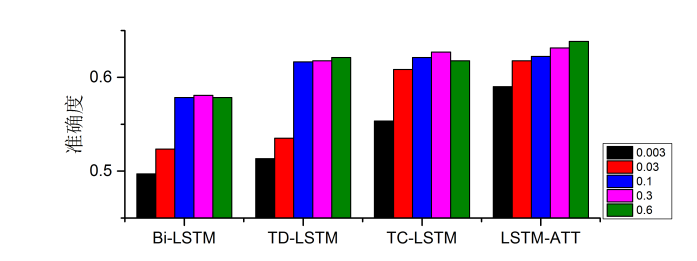


图 21：调整学习率(Twitter数据)

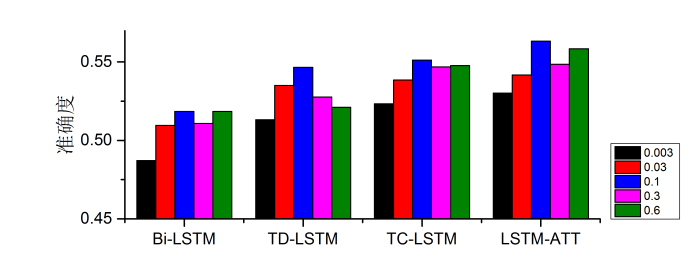


图 22：调整学习率(微博数据)

模型的训练需要寻找最佳的学习率，学习率太小时，会使得模型无法被训练。而学习率过高时，又可能出现模型优化不稳定的情况，同样不好得到较优的效果。我们通过设置了较多的学习率参数，试图寻找对各个模型最优的学习率。部分网络模型通过实验结果得出较好的学习率范围，而另一部分模型仍需要更多参数搜索。

从两个数据集的模型对比实验结果来看，考虑目标词出现位置的TC-LSTM和TD-LSTM模型与普通的双向LSTM网络相比，在情感分类效果上有所提升。这说明目标在句中出现的位置确实影响着文本对目标对象情感极性判断，通过增强目标位置附近词语对情感预测的影响，模型能提高情感预测与目标对象的相关性。

同时，基于注意力机制和多种词嵌入的LSTM-ATT模型在两个数据集上的情感分类效果都较前三种模型有所提升。说明注意力层通过训练后，能根据目标词的语义信息，定位句中与之相关性更高的局部位置。全连网络通过获取更多这些相关位置的特征，能够更加准确的做出情感预测。这表明了注意力机制在LSTM模型上应用的可行性。同时，模型增加的词性嵌入增强了网络的语义解析能力，而位置嵌入发挥了类似TC-LSTM和TD-LSTM模型结构的作用，引导网络定位目标在句子中出现的位置。

实验结果表明了本文设计的情感分类模型的可行性和有效性。

1. 总结和展望
   1. 本文总结

情感分类技术在社交网络发达的现在，有着巨大的价值，可用与信息检索、信息过滤、语义理解和商业分析等领域。随着深度学习技术日渐展现其巨大潜力，在自然语言处理领域，越来越多的研究工作实现了突破性的进展。

本文通过研究深度学习在分类任务中的典型方法和模型，介绍了深度神经网络的结构原理和训练方法，浅析了循环神经网络在序列数据中的应用和局限性。通过详细解析LSTM网络模块的内部结构原理，结合词嵌入技术的应用，介绍了应用于情感分类的LSTM模型。而后在词嵌入和LSTM模型的基础上，引入注意力机制，针对基于目标的情感分类问题，在模型中增加注意力层，设计了基于注意力机制的LSTM情感分类模型。并通过Twitter和微博文本数据集对4种模型进行训练和检验。最终实验结果说明了基于注意力机制的LSTM情感分类模型的有效性。

* 1. 展望

应用于自然语言处理和情感分析领域的深度学习研究仍在初步的探索阶段，大量对模型的改进和尝试都在不断地出现。其中不乏效果拔群的新模型，也有颇具启发性的改进。在目前阶段，任何具有启发性意义的改进或思想都是值得研究者去研究和尝试的。本文针对基于目标的情感分类问题，使用了更加复杂的网络模型。但整体的情感预测效果仍不能达到很高的置信程度，且模型有效的具体机理也仍处于黑盒状态。网络复杂度的提高也带来了性能问题。针对情感分析问题，我们仍然需要更加合理的语义表征方法和网络模型结构。如使用卷积神经网络(CNN)进行文本语义分析，并合理结合CNN和RNN的语义表征方式。同时，也有更多类似LSTM等改进的RNN网络模型，如GRU网络模型。

【参考文献】

1. Tang D, Qin B, Feng X, et al. Effective LSTMs for Target-Dependent Sentiment Classification[J]. Computer Science, 2016.
2. Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735.
3. Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[M]// Neurocomputing: foundations of research. MIT Press, 1986:533-536.
4. Turney P D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews[J]. Proceedings of Annual Meeting of the Association for Computational Linguistics, 2002:417--424.
5. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment Classification using Machine Learning Techniques[J]. Proceedings of Emnlp, 2002:79--86.
6. Bengio Y, Schwenk H, Senécal J, et al. Neural Probabilistic Language Models[J]. Journal of Machine Learning Research, 2006, 3(6):1137-1155.
7. Mnih A, Hinton G. A scalable hierarchical distributed language model[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2008:1081-1088.
8. Luong M T, Pham H, Manning C D. Effective Approaches to Attention-based Neural Machine Translation[J]. Computer Science, 2015.
9. Wang Y, Huang M, Zhu X, et al. Attention-based LSTM for Aspect-level Sentiment Classification[C]// Conference on Empirical Methods in Natural Language Processing. 2016:606-615.
10. Tang D, Wei F, Yang N, et al. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification[C]// Meeting of the Association for Computational Linguistics. 2014:1555-1565.
11. Zaremba W, Sutskever I, Vinyals O. Recurrent Neural Network Regularization[J]. Eprint Arxiv, 2014.
12. Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.
13. Dong L, Wei F, Tan C, et al. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification[C]// Meeting of the Association for Computational Linguistics. 2014:49-54.
14. Liu B. Sentiment Analysis and Opinion Mining[M]. Morgan & Claypool, 2012.
15. Tang D, Qin B, Liu T. Aspect Level Sentiment Classification with Deep Memory Network[J]. 2016.
16. Vo D T, Zhang Y. Target-dependent twitter sentiment classification with rich automatic features[C]// International Conference on Artificial Intelligence. AAAI Press, 2015:1347-1353.
17. Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.

**致谢**

在本篇论文完成之际，本人在深圳大学的本科学习生活也将结束。

首先，我要感谢我的论文指导老师黄哲学老师和陈小军老师。他们在我完成毕业设计的几个月里，不论是选题、开题，还是学习理论基础知识和完成实验，都给予了我极大的帮助和鼓励。他们不仅热衷学术、严格认真，还紧跟着前沿领域的研究，让我有机会接触到令人兴奋的新技术和新工具，让我在深度学习领域有了一定的基础。在此向黄哲学老师和陈小军老师表示衷心的感谢。

同时，我还要感谢不管是高中时期还是大学时期带领我参与算法竞赛的师兄和队友们。是他们给我打开了更宽广世界的大门，见识到了更高层次的东西。在与他们学习和拼搏的几年里，我不但学习到大量知识、锻炼了自己的技能，还让教会了我不断学习、不断挑战自我的精神。这些将是让我在日后的生活和工作中保持自我不断进步动力源泉。

感谢同实验室的师兄师姐在完成毕业设计和论文时给予我无微不至的关照和帮助。

感谢我的父母在二十多年来对我教育和照顾。

最后，感谢各位论文评审老师一丝不苟且辛苦的评审！由于本人能力有限，文章中还存在不少需要改进的不足之处，希望各位评审老师对本篇论文进行指正。

**Attention-based LSTM for Sentiment Classification**

**【Abstract】**As a fundamental task in Natural Language Processing, Sentiment Classification aims to classifying the polarity of a given text which tells whether the expressed opinion within the text is positive, negative, or neutral. But most of the time, there are different kinds of attitude, or opinions towards different targets. Thus, taking the target into account is important when we predict the sentiment polarity of a text. Recently, Deep learning has achieved significant results in image processing and speech recognition, and has brought new approaches to NLP problems like Sentiment Classification. Using the Word embedding technique, we can map words into a vector space with sentiment features embedded and feed them into the Recurrent Neural Network such as the Long Short-term Memory(LSTM) model. The LSTM model not only can make full use of the semantic correlation between words, but also considers the order of words as a significant information. The TC-LSTM method from Tang cut the text into two parts from where the target is located. Then feed them into two LSTM models that process from different directions, bringing more correlation between the target and the sentiment prediction. In this paper, we propose an attention-based bidirectional LSTM model to improve the performance of target-dependent sentiment classification. And we will combine unigram content features, syntactic features and word position features to form the word representations. Finally, we will implement the new model and some other basic LSTM models using Tensorflow. Experiments will be performed to evaluate our model.

**【Key words】**Sentiment Classification; Deep Learning; Long Short-term Memory; Attention Mechanism

指导老师：黄哲学（教授）

陈小军（讲师）

1. SIRI是一款内建在苹果iOS系统中的人工智能语音助理软件。 [↑](#footnote-ref-0)
2. Web2.0指的是一个利用Web的平台，由用户主导而生成内容的互联网产品模式。 [↑](#footnote-ref-1)
3. NLTK是一个基于Python的开源自然语言处理软件包。 [↑](#footnote-ref-2)
4. https://github.com/fxsjy/jieba [↑](#footnote-ref-3)
5. http://ir.hit.edu.cn/~dytang/paper/sswe/embedding-results.zip [↑](#footnote-ref-4)