分　类　号：　0242.1　　单位代码：10183
研究生学号：　2017312038　　密　　级：公开

吉 林 大 学

硕士学位论文

（学术学位）

基于 RNN 在文本分类中的改进及应用

Improvement and application of text classification based on RNN

作者姓名：杨钧淯

专　　业：计算数学

研究方向：计算数学

指导教师：罗宏文 副教授

培养单位：数学学院

2020 年　4 月

基于 RNN 在文本分类中的改进及应用

Improvement and Application of Text Classification Based on RNN

作者姓名：杨钧淯

专业名称：计算数学

指导教师：罗宏文

学位类别：理学硕士

答辩日期：　2020 年　6 月　3 日

吉林大学硕士学位论文原创性声明

本人郑重声明：所呈交学位论文，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。除文中已经注明引用的内容外，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。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标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：杨明清

日期：　2020　年 6　月　3　日

关于学位论文使用授权的声明

本人完全了解吉林大学有关保留、使用学位论文的规定，同意吉林大学保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权吉林大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

（保密论文在解密后应遵守此规定）

论文级别：☑硕士 □博士

学科专业：计算数学

论文题目：基于 RNN 在文本分类中的改进及应用

作者签名：杨明清　　指导教师签名：

2020 年　6 月　3 日

摘 要

基于 RNN 在文本分类中的改进及应用

所谓文本分类，就是针对一段文本信息，在所给定的类别中，选出与该文本相匹配的类别作为输出的一个重要手段。文本分类属于自然语言处理领域的一个基本问题，是机器学习等领域中非常活跃的研究方向，并有许多重要的实际应用。因此，研究具有较高精度与较强鲁棒性的文本分类算法有着重要的理论意义与实际意义。

本文选择经典 RNN 的变体 LSTM（Long Short-Term Memory）作为文本分类的基础工具有以下原因：一方面，LSTM 模型由于引入新的“门”结构，可以很好的解决文本训练过程中样本长度过长学习能力不足的问题，使得与关键词距离较远的词语在学习过程中也可以得到很好的保留。当数据集较大时，就可以更好的对原文本想表达的意思进行学习，从而增强该算法的鲁棒性，并有效地提高了模型的泛化能力。另一方面，该模型在实验过程中可以表现出较高的准确精度，使我们的预测过程从一开始就更加接近事实情况。本文主要针对神经网络方面有监督学习对比研究了 one-hot 模型、word2vec 模型等词嵌入模型、textCNN、BiLSTM 等神经网络框架、注意力模型等，最后部分还对 Google 最新提出的一些无监督学习模型，如 BERT 算法进行简述。

本文的创新点如下：词嵌入的选取对于神经网络的泛化能力有着很大的影响。目前比较常用的词嵌入模型是 one-hot 模型，该模型在预处理文本时由于本身的设计缺陷，会导致维度过高从而占用过大的内存空间，在训练过程中无法表达词语之间的关系等问题。针对这个困难，本文提出应用 word2vec 模型来解决。该模型的原理是数据在经过 one-hot 模型处理的基础上，将其进行降维处理，并将意思相近的词语映射到向量空间中位置相近的地方，从而完美的解决了原模型存在的维度灾难和词汇鸿沟的劣势。同时，为了更进一步使预测精度得到提高，本文在基础模型中加入了注意力模型，与原模型及 textCNN 模型进行对比实验。

最终实验表明，尽管在训练时长方面，新模型较 textCNN 模型用时更长，但是通过更换词嵌入模型以及增加注意力模型的方法可以使得计算机在文本分类任务中，既解决数据集较大时，设备条件有限的情况下导致的内存爆炸问题，也达到最终的实验结果上取得相比于原模型和 textCNN 模型更高的精度的实验目标。除此之外，实验中发现适当的增加训练周期以及隐藏层尺寸会提升准确率。但是学习率过大时，会导致模型出现不收敛的结果。

关键词：

机器学习，自然语言处理，文本分类，神经网络，词嵌入，RNN，LSTM，注意力模型

ABSTRACT

Improvement and Application of Text Classification Based on RNN

Text classification refers to the selection of a category matching a text as an important means of output for a given piece of text information. Text classification is a basic problem in the field of natural language processing. It is a very active research direction in the field of machine learning and has many important practical applications. Therefore, it is of great theoretical and practical significance to study text classification algorithms with high accuracy and strong robustness.

This paper selects LSTM (Long Short-Term Memory), a variant of classical RNN, as the basic tool of text classification for the following reasons: On the one hand, the LSTM model, due to the introduction of a new "door" structure, can solve the problem of insufficient learning ability due to too long sample length in the text training process, which makes words that are far away from keywords retain well in the learning process. When the dataset is large, the original text can be better learned to express the meaning, which enhances the robustness of the algorithm and effectively improves the generalization ability of the model. On the other hand, the model can show high accuracy in the experimental process, making our prediction process closer to the facts from the beginning. This paper mainly studies one-hot model, word2vec model and other word embedding models, textCNN, BiLSTM and other neural network frameworks, attention model, etc. for supervised learning of neural networks. Finally, some unsupervised learning models recently proposed by Google, such as BERT algorithm, are briefly described.

The innovations of this paper are as follows: The selection of word embedding has a great impact on the generalization ability of the neural network. At present, one-hot model is a commonly used word embedding model. Due to its design flaws, it will lead to excessive dimension, occupy too much memory space, and can not express the relationship between words during training. In order to solve this problem, the word2vec model is proposed. The principle of this model is to reduce the dimension of the data after one-hot model processing, and map words with similar meaning to similar locations in vector space, which perfectly solves the disadvantage of dimension disasters and lexical gaps in the original model. At the same time, in order to further improve the prediction accuracy, this paper adds the attention model to the basic model and compares it with the original model and the textCNN model.

The final experiment shows that although the new model takes longer than the textCNN model in terms of training time, the method of replacing the word embedding model and increasing the attention model can make the computer solve both the memory explosion problem caused by large datasets and limited device conditions in text classification tasks, and achieve the final experimental results compared with the original one.Experimentation objectives for higher accuracy of models and textCNN models.In addition, it is found that increasing the training cycle and the size of the hidden layer will improve the accuracy.However, when the learning rate is too high, the model will not converge.

Key Words:

Machine learning, Natural Language Processing, Text Classification, Neural Network, Word-Embedding, RNN, LSTM, Attention

目录

第 1 章 绪论

1.1 选题背景

自然语言处理是目前人工智能领域中的一个重要发展方向。它研究的是人和计算机之间实现真正交流的理论及方法，是一门集合了数学，计算机科学，语言学的学科。之所以这么重要，是因为人类与其他任何动物的根本区别在于人类可以通过语言系统进行相互之间的沟通。所有的生物中，只有人类具备体系的语言交流能力。正因为这样，与人类相关的多种发明都与语言有着紧密的联系。语言不仅能反映人类的逻辑思维，今天人类掌握，学习的绝大部分知识也是通过语言文字的形式传承下来的。因此，可以说人工智能的一个核心部分是自然语言处理。

自然语言处理一般分为以下多个方面：语音识别[1]，机器翻译[7]，信息检索，文本分类，机器阅读理解[2]，语义识别[8]，信息过滤，信息抽取，文本挖掘，情感分析，机器写作等。其中如机器翻译，就是将输入的文本按要求翻译成另外一种语言，目前市面上常用的谷歌翻译，有道翻译等都是这类分支的实现及应用；语音识别，则是通过机器学习，将我们的语音输入转化为我们需要的文本或其他格式进行输出，讯飞则是这类应用的成功代表之一；除此之外，相对复杂的问答系统中遇到的问题也逐渐被攻克，尽管可能现在让机器理解各种长篇名著依然存在很大问题，但是就给出一篇相对容易的阅读文章，并要求其回答提出的几个问题，计算机通过学习的过程可以逐渐达到人类的水平。当然，自然语言处理也一步步的由英语的局限得到了更好的拓展，目前各种汉语相关的自然语言处理任务也得到了较好的发展。

而我们今天要研究的则是其中的两个重要的方面：文本分类及情感分析。总的来说就是通过网络的学习，将我们的豆瓣电影影评文本进行二分类，即积极和消极。具体的细节将在后面的章节详细说明。

1.2 研究意义

一方面，文字是千百年来，传承人类文明的重要载体。通过对文字的学习，使人类一步步的拨开了历史的面纱，并可以很好的将知识传承下去。玛雅文明，古希腊文化，乃至中国闻名于世的诗词歌赋得以多年后被世人知晓，都离不开文字的功劳。如今社会，我们的生活更是一刻也少不了文字：学习新的知识，日常交流，都需要文字的帮助。另一方面，计算机的出现，彻底改变了我们的生活方式。从刚刚面世的笨重迟缓到如今的轻便快捷，从过去的无人问津到今天的家家必备，计算机的迅速发展已经让人类社会有了翻天覆地的变化。因此，让计算机学习文字信息，理解文字信息，最终实现人工智能化，在现代化的今天有着深远、非凡的意义。

文本分类属于分类问题的一个重要应用，是人工智能中自然语言处理领域的一个基础问题。文本分类，是将复杂的数据进行很好的归纳总结，从而更加高效的对事物进行判断

和改进。这一过程在繁多信息资源形成的信息化时代的今天，如果都由人类完成，毫无疑问是一种严重浪费资源的表现。因此，利用计算机远超过人类的计算能力以及节约资源等优点，通过机器学习的过程，让计算机进行文本分类，则是一种跟紧时代步伐的表现。通过改进神经网络使计算机在文本分类过程中达到更高的准确度也因此意义重大。如图 1.1 所示：

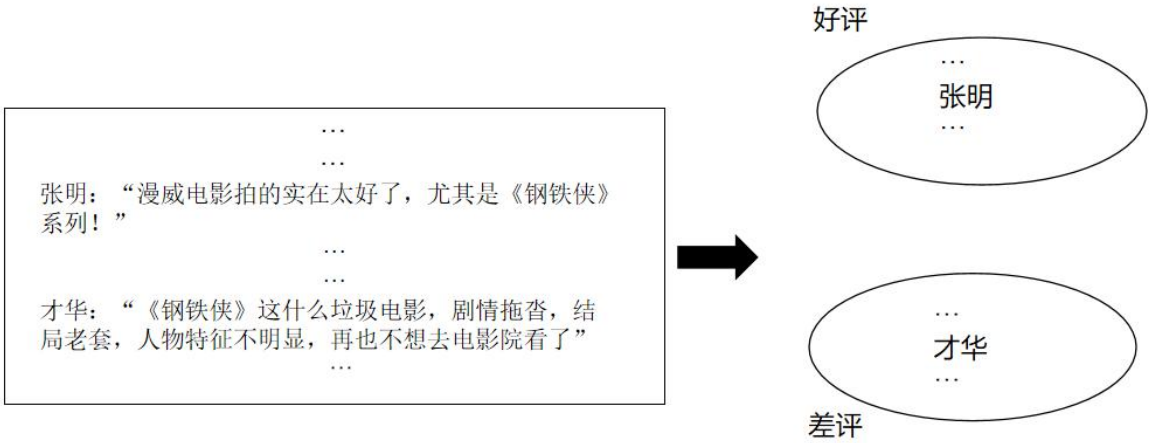

图 1.1 **文本分类**

如果可以更精准的对于文本信息进行分类，那么不仅对于对应事物的判断将更加准确，后续产生的影响也十分重大。举个例子，当实现对于电影院上映的电影影评进行很好的分类的时候，我们就可以高效地实现对电影质量的把控，以及更加合理化，利益最大化的排片选择。这不仅仅只是精度上的提高，而是实实在在将人类的生活品质提高了档次。因此，该研究在实际应用过程中也有着卓越意义。

1.3 本文的主要研究内容

本文主要分为以下六个章节进行具体研究：

第二章对文本分类中的预处理环节中的最重要的部分——词嵌入方法进行深入研究，主要对当前应用最为广泛的 one-hot 模型进行学习，并在处理文本分类问题遇到内存爆炸的时候引入 word2vec 模型，使精度得到了一定程度上的提高。

第三章主要针对所采用的神经网络框架，即循环神经网络 RNN 及其相应的经典变体 LSTM 进行学习。在了解基础框架利弊的前提之下，对于 LSTM 能更好解决的问题进行了深入思考及实践，掌握 LSTM 特有的“门”结构的优势所在。

第四章内容主要介绍注意力模型及其初级版本 Encoder-Decoder 模型。

第五章主要是对该框架的最后数据处理部分进行讨论。详细介绍了分类问题以及数据处理过程中遇到的一些问题。

第六章是对实验所选取的数据集以及参数等设置的具体说明，同时对数据结果统计。对基础模型及更改过的新模型在文本分类问题上进行了实验，在不同参数下找到精度更高

的结论，不仅如此，还在相同参数条件下应用 textCNN 模型进行了对比实验。尽管后者的精度表现不如本文的模型，但是在花费时间的表现上存在明显优势。

最后一部分是对研究生三年期间工作的一个总结，并对近期人工智能领域谷歌团队提出的 BERT 算法等无监督学习方法进行简述。同时，对于下一阶段的学习提出目标和计划。

第 2 章 词嵌入层

2.1 词嵌入的方式

我们日常使用的语言是自然语言，而这些语言不能直接应用于计算机进行机器学习，因此需要对自然语言进行预处理，将它们转化成向量的形式，这个过程就是词嵌入[3]。这些预处理的样本不仅仅局限于文本，还包括语音片段等。

在本文中，实验尝试了两种词嵌入方式：one-hot 以及 word2vec。

2.2 One-hot

在机器学习过程中，我们会最先遇到特征分类。举个例子，人的性别有男性和女性；语言有汉语；英语；德语；日语；法语等等。而这些特征值并不是连续的，而是无序的、离散的。因此，需要对其进行特征数字化。还是用上面的例子说明：

性别特征：[“男性”，“女性”]；

语言特征：[“汉语”，“英语”，“德语”，“日语”，“法语”]；

身高特征：[“160cm”，“165cm”，“170cm”，“175cm”，“180cm”]

假设有一个样本（人）满足男性、说汉语、身高 180cm 的特征，则可以由向量表示为[0，0，4]，但是由于类别之间是没有联系的，即无序性，所以这样处理得到的特征向量并不能直接用于词嵌入模块当中，因而才有了以下的编码方式。

One-Hot 编码，主要是采用相同数目的状态寄存器来对该数目的状态进行编码，这样的编码方式可以使得每个状态都有它们自己独立的寄存器位置，并且无论在什么时候，都满足只有一位有效。这种编码方式是将分类的变量转化为二进制向量表示的形式。过程首先需要将特征的分类数映射为整数；然后，每一个整数值被表示为二进制向量，除了本身被标记为 1，其他位置都记为 0。这种编码方式是将离散特征取值延伸到欧式空间，因此，每一个离散特征的取值就一一对应着欧式空间的每一个点个点。

举个例子，假设语料库中有以下三句话：

例句 1：我喜欢游泳。

例句 2：爸爸妈妈喜欢游泳。

例句 3：爸爸妈妈喜欢我。

首先将语料库中出现的词语进行编号：

1.我 2.喜欢 3.游泳 4.爸爸 5.妈妈

然后用 one-hot 的编码模式提取特征向量（被引用到的为 1，没有的则为 0），可以得到图 2.1--2.3：

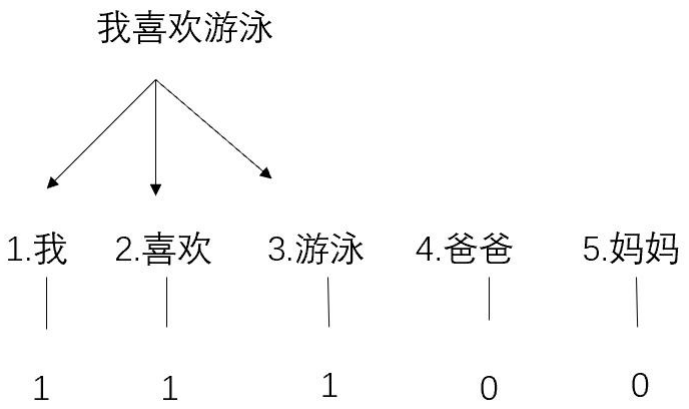

图 2.1 例句 1 中的 one-hot 编码图

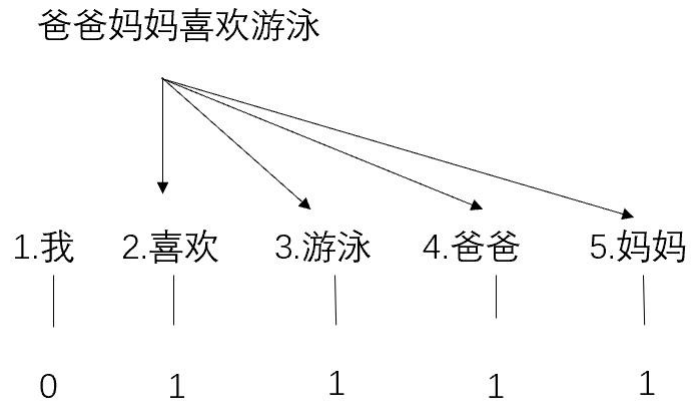

图 2.2 例句 2 中的 one-hot 编码图

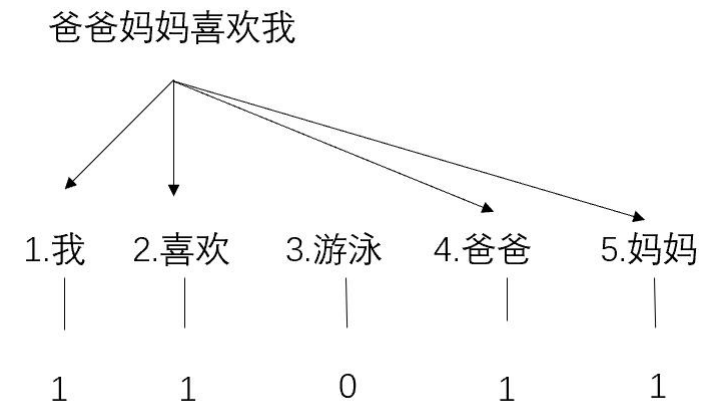

图 2.3 例句 3 中的 one-hot 编码图

经过这样的处理，最终可以得到每句话的特征向量为：

我喜欢游泳：　　（1，1，1，0，0）

爸爸妈妈喜欢游泳：（0，1，1，1，1）

爸爸妈妈喜欢我：　（1，1，0，1，1）

这样处理数据的优势：可以更方便的扩充要补充的特征；增强了模型的非线性能力；不需要对变量进行归一化；加速参数的更新速度；降低了特征值扰动对模型稳定性的影响。

然而这个预处理的方式有个缺点，就是它对文本的刻画方式仅仅是存在或不存在，因此对于词与词之间的相似性关系，这个处理方式所产生的结果均为线性无关，从而不能进行很好的描述。除此之外，还有一个很大的问题，就是当数据集足够大时，对于整个语料库的特征提取组成的包则会很占内存，即维度会过高。当最开始对几个句子进行预处理的

时候，这种方式可以取得很好的效果，但是随着句子的增加到最后整个语料库的输入时，这种词嵌入的方式导致了计算机内存不足。（p.s.实验时使用的电脑配置是 gtx1060，16g）因此，对其改进介绍一种内存更小并效果更好的词嵌入方式，即 word2vec。

2.3 Word2vec

Word2vec 这种词嵌入方式是 Tomas Mikolov 在 2013 年首次提出的[4][5]。在介绍这种预处理方法之前，要先介绍一些预备知识。而由于这种方法也属于神经网络的一个应用，相关的内容会在下一部分进行更详细的解释。

2.3.1 Sigmoid 函数

Sigmoid 函数[11]是神经网络经常要用到的函数之一。其定义为 $\sigma(x)=\frac{1}{1+e^{-x}}$。

该函数的定义域为 $(-\infty,+\infty)$，值域为 $(0,1)$。当 x 趋近于负无穷时，$\sigma(x)$ 趋于 0；当 x 趋近于正无穷时，$\sigma(x)$ 趋于 1；当 $x=0$ 时，$\sigma(x)=0.5$。下图 2.4 给出了该函数的图像：

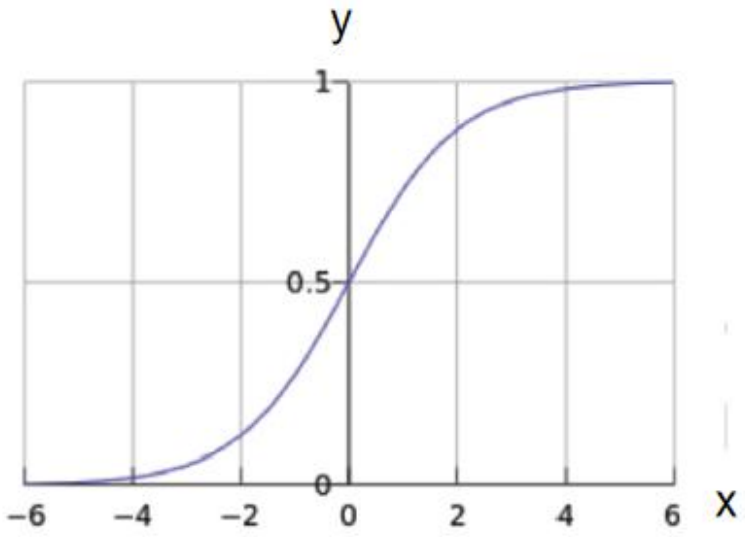

图 2.4 sigmoid 函数图像

尽管这个函数本身容易出现梯度消失的问题，而且对称点不是 $(0,0)$，但是该函数的优点在于其单调连续性，有良好的对称性，输出范围稳定，而且导数具有较好的性质，即：

$$\sigma'(x)=\sigma(x)(1-\sigma(x)) \qquad \cdots\cdots\cdots (2.1)$$

$$\begin{aligned}\left(\log\sigma(x)\right)' &= 1-\sigma(x)\\ \left(\log(1-\sigma(x))\right)' &= -\sigma(x)\end{aligned} \quad \cdots\cdots (2.2)$$

2.3.2 Bayes 公式

Bayes 公式也称贝叶斯公式[10]，简单而言，是用来预测某个条件下，一件事情发生的概率是多少。

假设有两个事件 A，B。事件 B 发生时，事件 A 发生的概率为：

$$\mathrm{P}(\mathrm{A}|\ \mathrm{B})=\frac{\mathrm{P}(\mathrm{A},\mathrm{B})}{\mathrm{P}(\mathrm{B})} \quad \cdots\cdots (2.3)$$

同理，当事件 A 发生时，事件 B 发生的概率为：

$$\mathrm{P}(\mathrm{B}|\ \mathrm{A})=\frac{\mathrm{P}(\mathrm{A},\mathrm{B})}{\mathrm{P}(\mathrm{A})} \quad \cdots\cdots (2.4)$$

则容易得到：

$$\mathrm{P}(\mathrm{A}|\ \mathrm{B})\cdot\mathrm{P}(\mathrm{B})=\mathrm{P}(\mathrm{A},\mathrm{B})=\mathrm{P}(\mathrm{B}|\ \mathrm{A})\cdot\mathrm{P}(\mathrm{A})$$

其中$\mathrm{P}(\mathrm{A})$和$\mathrm{P}(\mathrm{B})$表示 A，B 事件发生的概率。$\mathrm{P}(\mathrm{A},\mathrm{B})$表示事件 A，B 同时发生的概率。

当$\mathrm{P}(\mathrm{A})\neq 0$时，可以得到贝叶斯公式为：

$$\mathrm{P}(\mathrm{B}|\ \mathrm{A})=\frac{\mathrm{P}(\mathrm{A}|\ \mathrm{B})\cdot\mathrm{P}(\mathrm{B})}{\mathrm{P}(\mathrm{A})} \quad \cdots\cdots (2.5)$$

这里，称$\mathrm{P}(\mathrm{A})$为先验概率，在计算前假设的某个概率，比如抛硬币正面向上的概率为 50%；$\mathrm{P}(\mathrm{B}|\ \mathrm{A})$)为后验概率，这是看到数据后计算得到的；$\mathrm{P}(\mathrm{A}|\ \mathrm{B})$是通过先验概率和后验概率计算得到的，称为似然度。

2.3.3 Huffman 树及编码

（1）Huffman 树

用有 n 个结点（都做叶子结点并且都有权值）试图构建一棵树时，如果构建的这棵树的带权路径长度最小，称这棵树为“最优二叉树”，有时也叫“赫夫曼树”。其中路径是一棵树中一个结点到另一个结点的通路；结点的权是指给每个结点赋予一个新的数值；而结点的带权路径长度指的是从根结点到该结点之间路径长度与该结点的权的乘积。所构造的 Huffman 树则是满足所有叶子结点的带权路径长度和最小的那棵树。

（2）如何构造一棵 Huffman 树

对于给定的有各自权值的 n 个结点，构造如下：

1) 在所有结点中选择出权值最小的两个结点，将它们组成一个二叉树，并且使新的二叉树的根结点权值等于左右两孩子结点的权值之和。

2）在之前的所有结点权值的集合中删除已经组成二叉树的两个结点，取而代之的是新形成的二叉树的根结点的权值。

3）重复步骤 1 和 2，直到所有的结点都被用到，生成一棵终极二叉树。这棵树就是我们想要构造的 Huffman 树。

（3）Huffman 编码

Huffman 编码，又称霍夫曼编码，是可变字长编码的一种。该方法完全是根据每个字符出现的概率来构造异字头的平均长度最短的编码。之所以应用这种编码方式，想通过频率的使用来最大化的节省储存空间。

1）将字符集合中的所有字符使用过得频率作为权值构造一个 Huffman 树。

2）从根结点开始，对每个叶子结点一一赋值，左侧为 0，右侧为 1（也可左 1 右 0），从根结点到某叶子结点的值相连，得到的即为该点的 Huffman 编码。让我们来看个例子，这样可以帮助我们更好地理解这种编码方式：

假设有 5 个字母 N，O，R，E，G 分别出现次数为 1，2，3，4，5，则可以构造 Huffman 树为下图 2.5 所示：

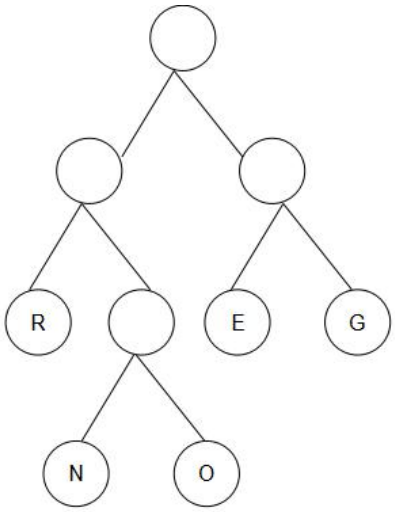

图 2.5 Huffman 树

这样可以得到对应的 Huffman 编码为：

N：010

O：011

R：00

E：10

G：11

2.3.4 NNLM 统计模型

Word2vec 本意为 word to vector，即将词语转化成向量的意思。词作为连续向量的表示有着悠久的历史[12][13][14]。首先，看一个神经网络 Feedforward Neural Net Language Model（NNLM）[6]。给出一个统计语言模型，它可以对于之前给定的所有单词，用下一个单词的条件概率来表示，因为

$$\hat{\mathrm{P}}\left(w_1^T\right)=\prod_{t=1}^{T}\hat{\mathrm{P}}\left(w_t\left|w_1^{t-1}\right.\right) \qquad \cdots\cdots\cdots (2.6)$$

其中，w_t 是第 t 个单词，子序列写成 $w_i^j=\left(w_i,w_{i+1},\cdots,w_{j-1},w_j\right)$。

这样的统计模型已经应用在很多领域，如对话识别，语言翻译，信息检索等中被认为是有用的。因此，统计语言模型的提升自然对于这类应用影响巨大。

在建立自然语言的统计模型时，可以利用语序大大降低建模问题的难度，与此同时，还注意到一个事实，即在时间上更接近的单词在单词序列中的统计上更具有相关性。因此，n 元模型对下一个单词构成了一些条件概率表，即之前 n-1 个词的组合：

$$\hat{\mathrm{P}}\left(w_t\middle|w_t^{t-1}\right)\approx\hat{\mathrm{P}}\left(w_t\middle|w_{t-n+1}^{t-1}\right) \qquad \cdots\cdots\cdots（2.7）$$

训练集是由单词 $w_t\in V$ 组成的序列 $w_1w_2\cdots w_T$，其中词库 V 很大但是有界。目标是经过训练得到一个优秀模型 $f\left(w_t,\cdots,w_{t-n+1}\right)=\hat{\mathrm{P}}\left(w_t\middle|w_1^{t-1}\right)$，从这个意义上说它给出了很高的样本外的可能性。之后取的几何平均数，即复杂度，也被称为平均负对数似然的指数为 $1/\hat{\mathrm{P}}\left(w_t\middle|w_1^{t-1}\right)$。关于这个模型，仅有的限制条件是不管如何选择 w_1^{t-1}，都可以得到 $\sum_{i=1}^{|V|}f\left(i,w_{t-1},\cdots,w_{t-n+1}\right)=1$，且 $f>0$。通过对这些条件概率做乘法运算，最后得到词序列的联合概率模型。将函数 $f\left(w_t,\cdots,w_{t-n+1}\right)=\hat{\mathrm{P}}\left(w_t\middle|w_1^{t-1}\right)$ 分解为两部分：

（1）一个映射 C 将 V 中的任意元素 i 映射到一个实向量 $C(i)$ 表示为与词汇中每个单词相关联的分布特征向量。

（2）用 C 表示的单词的概率函数：用函数 g 映射上下文中单词的特征向量的输入序列 $\left(C\left(w_{t-n+1}\right),\cdots,C\left(w_{t-1}\right)\right)$ 到一个 V 中的对下一个词 w_t 条件概率分布。输出 g 是一个向量。其第 i 个元素估计 $\hat{P}\left(w_t=i\middle|w_1^{t-1}\right)$ 的概率。如图 2.6：

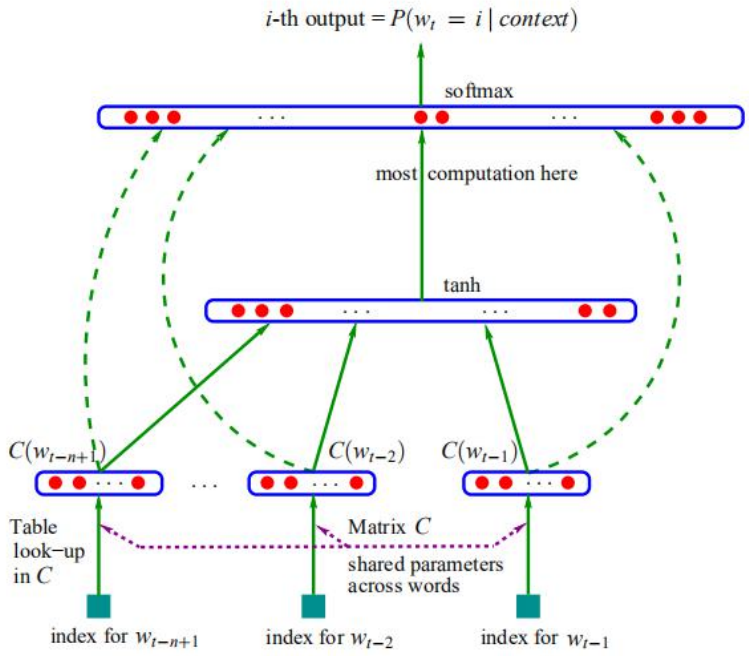

图 2.6 NNLM 流程图

函数 f 是这两个映射的组合，其中 C 被上下文中的所有词共享。这两个部分中的每一个都与一些参数相关联。映射 C 仅仅是特征向量本身，由 $|V|\times m$ 维矩阵 C 表示，其行 i 是关于词 i 的特征向量 $C(i)$。函数 g 可通过参数 ω 在前馈或 RNN 或其他参数化函数中实现。总体参数集为 $\theta=(C,\omega)$。

训练的实现是通过寻找 θ，使训练语料库的对数似然取最大得到的：

$$L=\frac{1}{T}\sum\log f(w_t,w_{t-1},\cdots,w_{t-n+1};\theta)+R(\theta) \qquad \cdots\cdots\cdots (2.8)$$

其中 $R(\theta)$ 是正则项。

2.3.5 Word2vec 中的两个具体模型

Huffman 树给频繁的单词分配了简短的二进制代码，这进一步减少了需要评估的输出单元的数量：虽然平衡的二叉树需要 $\log_2(V)$ 个输出来评估，基于 Huffman 二叉树的分层 softmax 只需要 $\log_2(Unigram-复杂度(V))$。利用 CBOW(Continuous Bag-of-Words)与 Skip-Gram 两种模型将计算复杂度降到最低。

（1）CBOW 模型

这个模型的结构与前馈 NNLM 相似，只是删除了其中的非线性隐藏层，投射层参数不仅仅在投射矩阵中共享，而是共享于所有单词。如图 2.7。因此，所有单词被投射到相同的位置。之所以称为词袋模型，是因为之前的单词语序不影响投射。不仅如此，还用到了该词后面的词语。建立一个前四个后四个词的对数线性分类器使得到了最好的表现，训练的标准是将中间的词进行正确分类。与标准词袋模型不同，它用连续分布表示上下文。输入层与投射层之间的权重矩阵在所有词位置共享。

（2）Skip-Gram 模型

这个模型与 CBOW 模型类似，只是它是给出当前词来预测上下文。如图 2.8 它试图基于同一句话的另一个词语来将该词的分类最大化。更具体的说，是用每一个当前词语作为输入，到一个具有连续投射层的对数线性分类器中，从而预测当前词语前后一定范围的词。由于距离较远的词语不如近的词语与当前词语相关性高，因此，对于远距离词通过更少的采样来降低其权重。

本文中，对于这两种文本预处理都进行了实验，由于样本量比较小的缘故，CBOW 模型略好于 Skip-Gram。但是二者都能解决内存爆炸的问题。接下来的部分，将进行介绍神经网络。

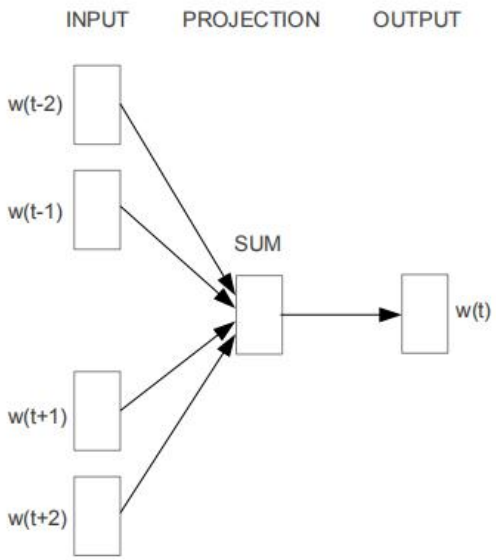

图 2.7 CBOW 模型

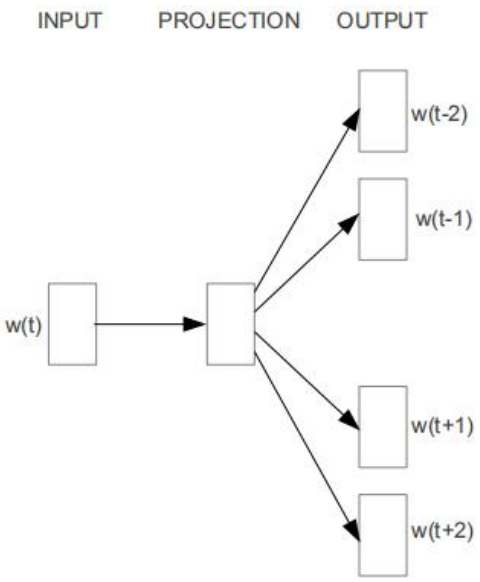

图 2.8 Skip-gram 模型

第 3 章 神经网络

3.1 概述

截至今日，计算机已经取得了很大的成就。不仅可以用它实现日常的休闲娱乐办公等基本要求，也开始利用它做更高级的尝试，即尝试让其学习人类的思考模式。人工智能的名词也因此应运而生。人工智能发展至今，已经逐渐的融入了我们的生活，扫地机器人，自动泊车，已经慢慢的取代了当初的模式，2016 年的 Alpha Go 机器人，也已经在短短几年的时间里，实现了在围棋领域战胜人类的目标。

当然，这其中的一些领域的突破，少不了神经网络的功劳。1943 年，人工神经网络的概念由心理学家 Warren McCulloch 和数学家 Walter Pitt[15]首次提出。因为计算机上的二进制很好的可以表示生物神经元的兴奋或抑制的状态，因而有了神经网络的概念。

随后的时间里，出现了可以模拟人类感知能力的“感知机”[16]，也可以说是初代神经网络。然而，由于其本身的结构缺陷，很大的制约了它的未来发展。感知机中特征提取层的参数需要人手工调整，不仅如此，过于简单的单层结构也限制了它的学习能力，很多复杂的函数都超出了它的学习范畴，不能很好的拟合，从而达不到最终的实验目的。

神经网络迎来又一个重大的革新是在 1986 年，Geoffrey Hinton 等人实现将反向传播算法（BP 算法）[14]引入多层（而非单层）感知机的革新。网络层数的增加，也使神经网络有了深度，从而更好地进行学习复杂的函数。1989 年，卷积神经网络（CNN）[18]在银行支票的手写字体识别应用上取得了成功，但是由于硬件水平的限制，训练时间过长的问题也阻碍了其进一步的前进脚步。不仅如此，对训练集的要求，网络层数也由于算法限制不能太多等原因也，也阻挡着它们的发展。

之后的时间，尽管逐渐的调整，使网络层数可以达到一个很深的水平。然而有一个问题一直得不到解决，即 CNN 不能解决时间序列上的建模问题。然而，对于语音识别，自然语言处理等问题，数据提供的时间顺序所包含的信息非常重要。因此，出现了一种新型的神经网络结构，即循环神经网络，也就是 RNN。

3.2 RNN 网络

RNN 网络（Recurrent Neural Network）[20]，也被称为循环神经网络。这个神经网络的提出，就是为了解决时间序列上所遇到的 CNN 和简单的人工神经网络解决不了的问题。前面这两者，是将信息元素视为相互独立的，相互之间没有任何联系。然而，在现实生活中，信息之间不仅仅有联系，而且存在着十分密切的复杂的联系。

举个例子，下面有一道填空题：我最喜欢向日葵，如果生日能收到一束（ ），那我会非常开心的。这个问题，对于我们而言十分简单，可能小学三年级的孩子都可以回答出答案“向日葵”。然而如果不能对前面信息“最喜欢向日葵”及“一束”进行分析，那这道问题的答案将无从得知。

所以，可以看出时间序列上的信息对于计算机真正实现拟人化是至关重要，必不可少的。下图 3.1 是 RNN 的网络结构。

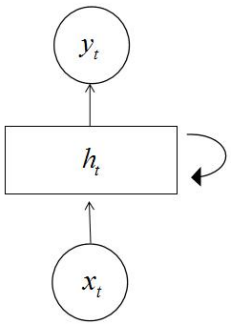

图 3.1 **RNN 网络结构**

上图这个神经网络由输入层，隐藏层，输出层三个部分组成。分别用图中的 x_t， h_t，y_t 表示。值得注意的是右侧有一个表示循环的箭头，也正是这个箭头，实现了学习语序信息的想法。将这个结构图展开后可以得到图 3.2：

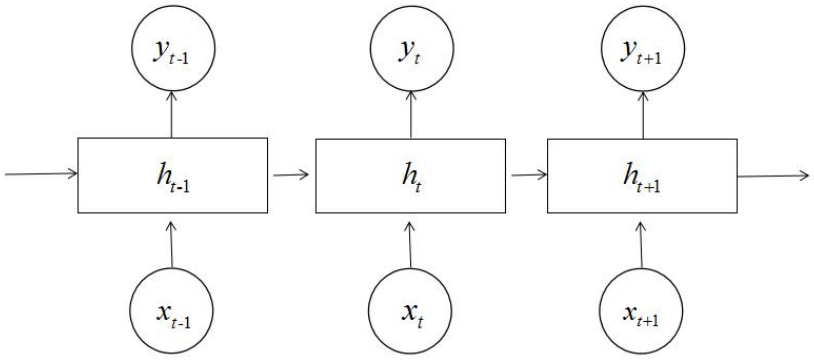

图 3.2 **RNN 展开图**

即

$$h_t = f\left(h_{t-1}, x_t; \theta\right) \qquad \cdots\cdots\cdots (3.1)$$

也就是说，t 时刻的隐藏状态都是由当前的输入 x_t 以及上一个时间步骤的隐藏状态 h_{t-1} 共同决定的。在 NLP 应用中，将句子或段落表示成单词的序列，并将每个词语转换成向量，这个过程也就是前面一章提到的词嵌入，即 $x = x_1, x_2, \cdots, x_n \in R^d$，则 $h_t \in R^d$ 可以用来对上下文 $x_{1:t}$ 信息进行建模。上述的过程也可以表示为：

$$\begin{aligned} m_t &= Ux_t + Wh_{t-1} \\ h_t &= f\left(m_t\right) \\ y_t &= g\left(Vh_t\right) \end{aligned} \qquad \cdots\cdots\cdots (3.2)$$

其中，f 和 g 都是激活函数，f 可以是 tanh，Relu 或 sigmoid 函数。g 通常是 softmax 函数。值得注意的是，这里的权重 U，V，W 在每个时刻都是不变的，即共享的。

RNN 神经网络虽然解决了词序之间的关系问题，但是又出现了新的问题，就是这个简单的链式网络在处理距离过长的词之间的关系时，或处理过长句子或段落的时候，会出现梯度消失或梯度爆炸。如下图 3.3 所示，一个多句话组成的段落中，第一句，第二句的输入对第 $t-1$ 句输出的影响最大，这样的情况，很难在学习过程中让计算机很好的学习到其中的关系。

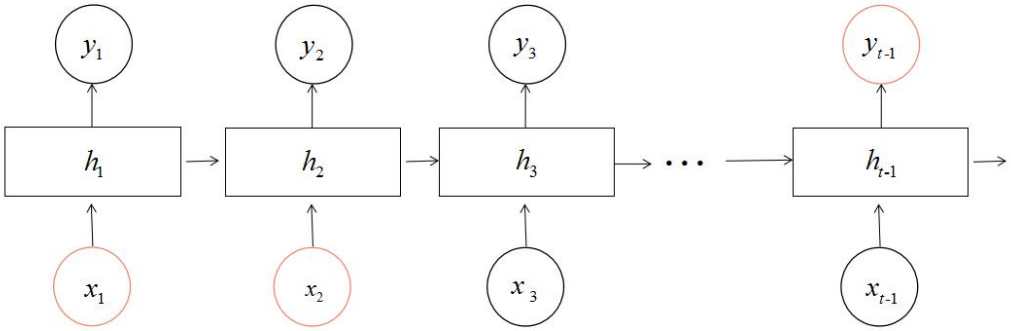

图 3.3 **RNN 在段落中学习**

因此，本文提出一个基于经典 RNN 的变形结构来解决这个问题。

3.3 RNN 神经网络的变形 LSTM

这个部分，要介绍一种经典的 RNN 神经网络的变形，长短依赖记忆（Long Short-Term Memory），简称为 LSTM。LSTM 由 Hochreiter 和 Schmidhuber (1997)[9]首先提出，Felix Ger[21]在 2001 年的博士论文进一步改进了 lstm 的网络结构，并在后来被 Alex Graves[8]进行了改良和推广。之所以提出这个结构，是在处理文本的实践中，随着篇幅的增长，会出现梯度消失或者梯度爆炸的问题，而且句子长度过大的时候，也会导致计算机不能很有效的提取词与词之间的关系。而这些往往是链式结构难以避免的，因为对于第 n 项，假设第 1 项最重要，但是经过了 $n-1$ 次循环，权重自然变得微不足道了。利用 LSTM 是可以有效地克服这个问题的。下图 3.4 是 LSTM 的总体结构图：

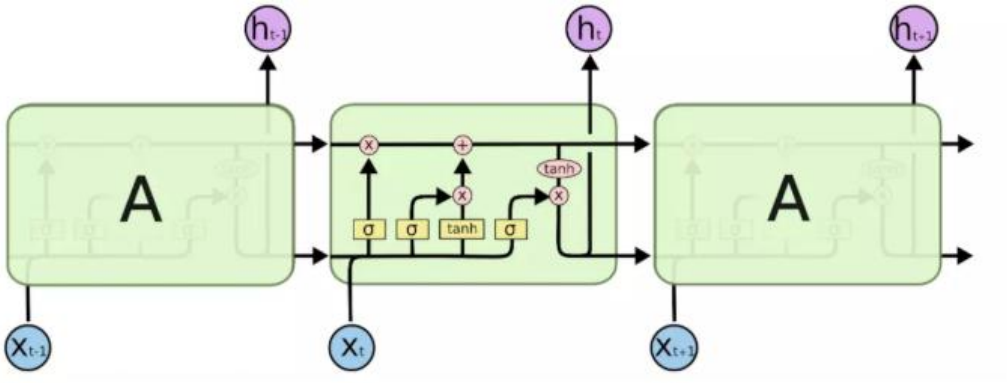

图 3.4 LSTM 结构图

图中的黄色框代表神经网络层，粉色的圈代表数值运算，箭头表示向量在这个网络结构中的传输过程，线的合并代表连接，线的交叉代表将当前的内容进行复制。

LSTM 的关键是长依赖记忆 c_t，也被称为细胞状态，表示细胞状态的这条线横向穿过图的顶部，如图 3.5。细胞的状态的更新在这条链上按时间步骤运行。即对上一个时间的细胞状态进行一些数值计算，得到当前时间步骤的。

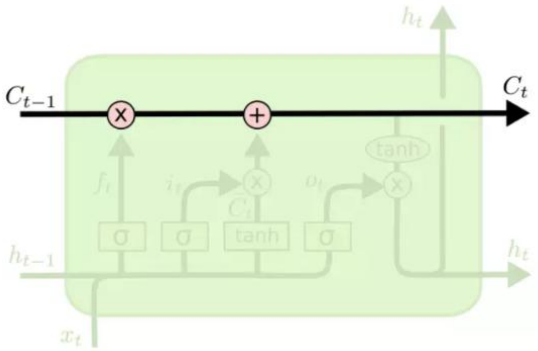

图 3.5 细胞状态

LSTM 之所以可以处理长距离文本信息，是因为在它的神经网络框架中引入了一种叫“门”的结构。在这个模块中，一共有三个门，即“输入门”，“输出门”及“遗忘门”。神经网络可以通过自主控制这三个门来做到对之前信息的筛选，通过赋予较重要信息较大的权重，达到对远距离重要信息的提取的目的。

下图 3.6 展示的就是“门”结构。它是由一个 sigmoid 神经网络层和一个点乘运算所组成：

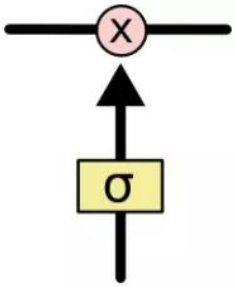

图 3.6 “门”结构

如图 3.7 所示为遗忘门部分：

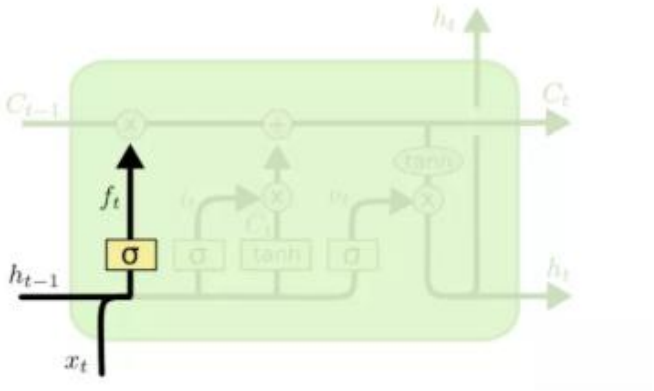

图 3.7 **遗忘门结构**

这一部分的作用是决定上一时刻的单元状态c_{t-1}有多少信息保留到当前时刻c_t。通过对当前输入x_t以及前一个隐藏层输出h_{t-1}的信息整合，通过 sigmoid 函数σ，为上一个细胞状态c_{t-1}输入一个$[0,1]$之间的数，其中 0 表示将之前信息完全删除，而 1 则表示完全保留。也就是：

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \qquad \cdots\cdots\cdots (3.3)$$

其中，W_f是遗忘门的权重矩阵，$[h_{t-1}, x_t]$表示把两个向量连接成一个长一点的向量，b_f是遗忘门的偏置项。这之后的类似的符号都是相同含义，之后不再做解释。

接下来是第二部分，这部分由输入门和一个 tanh 层共同组成，输入门部分是决定当前时刻网络的输入x_t有多少保存到细胞状态x_t，即

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \qquad \cdots\cdots\cdots (3.4)$$

而 tanh 层，则是根据上一个隐藏状态输出和当前的输入描述当前输入的单元状态，即

$$\widetilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \qquad \cdots\cdots\cdots (3.5)$$

其中，$\widetilde{C}_t$表示当前的记忆，即当前的单元状态。下图 3.8 所示就是第二部分的过程。

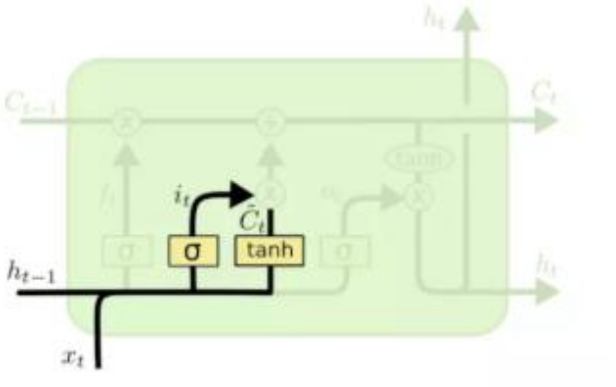

图 3.8 输入门及 tanh 层

完成了以上两部分，就可以实现细胞状态的更新了。既可以对之前信息进行选择性输入，也可以添加关于当前状态的新信息。这个步骤如下图 3.9 所示：

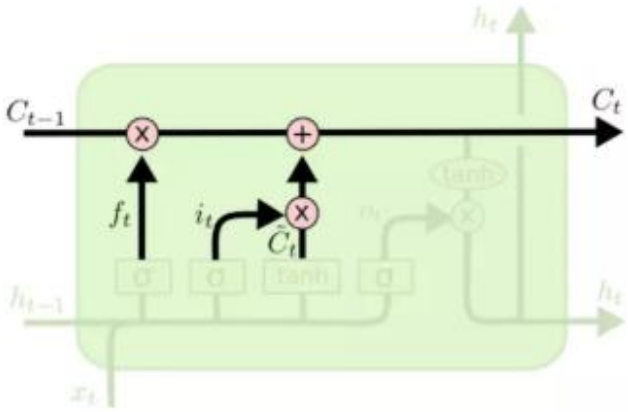

图 3.9 细胞更新过程

这个过程可以表示为：

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \qquad \cdots\cdots\cdots (3.6)$$

输出门的结构是用来控制单元状态 C_t 有多少输出到 LSTM 的最终输出值 h_t：

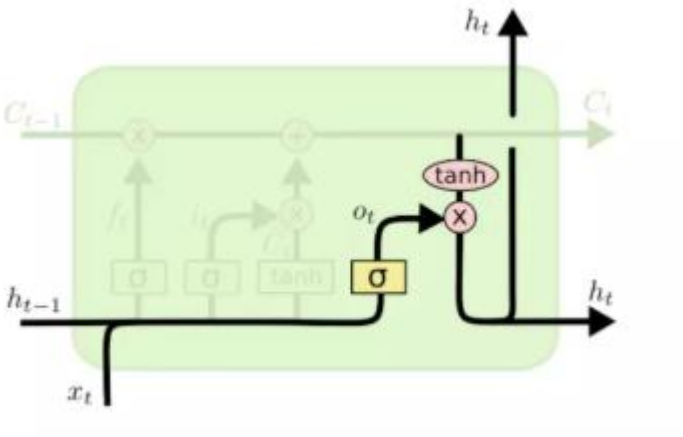

图 3.10 **输出门结构**

如上面的步骤一样，这个过程可以表示成：

$$\begin{aligned} o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right]+b_o\right) \\ h_t &= o_t * \tanh\left(C_t\right) \end{aligned} \qquad \cdots\cdots\cdots (3.7)$$

正是这个经典的 LSTM 结构，使得通过 RNN 实现了突飞猛进的进展。本实验应用的模型为 BiLSTM[17]，这只是在原有的 LSTM 结构上，连接了一个反向相同的 LSTM。因为对于文本来说，有用的信息可能不止来自于目标信息的前面，也可能在后面。所以我们这样设置是为了能让准确率达到更高得水平。为了模型达到更高的精度呢，下一部分介绍另一个模型，即注意力模型。

第 4 章 注意力模型

这一章介绍编码-解码模型，即 Encoder-Decoder 模型[22]以及本实验应用的注意层 Encoder-Decoder 模型的提出，主要是针对解决 sequence to sequence 问题，也就是 seq2seq 问题。seq2seq 指的是根据一个输入序列x，来生成另一个输出序列y。这种问题具有很广泛的实际意义，比如将文本翻译成另一种语言的机器翻译；机器阅读理解中的问答系统也是这类问题，即通过输入一篇文章让机器输出针对提出问题的答案等。

下图 4.1 所示就是一个 Encoder-Decoder 模型的示意图。

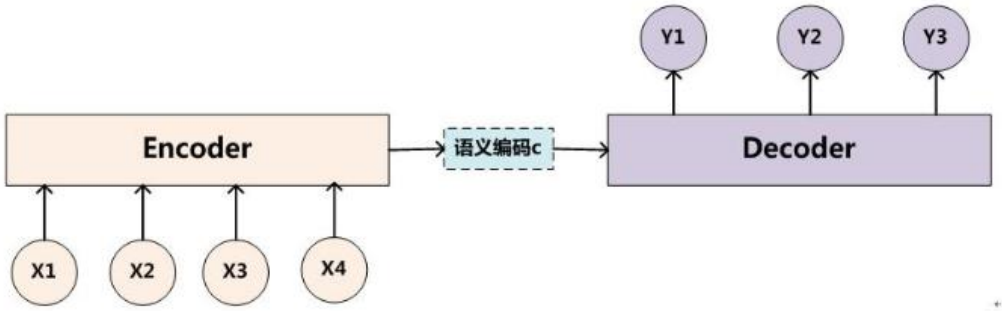

图 4.1 Encoder-Decoder 模型

Encoder 模块对输入序列先进行编码，之后通过函数变换为中间的语义编码向量 c，之后通过 Decoder 模块，根据 c 和之前生成的y_{i-1}，来生成y_i。在实际应用中，经常应用 RNN 来对句子编码：$h_1,h_2,\cdots,h_n$，同时用最后一步的h_n来预测任务：

$$P(Y=y)=\frac{\exp(W_y h_n)}{\sum_{y'}\exp(W_{y'}h_n)} \qquad \cdots\cdots\cdots (4.1)$$

但是这需要这个模型可以将所有句子中的重要信息都压缩为固定长度的向量，这个过程会导致丢失很多信息。不仅如此，当文本长度很大时，由于设计问题，也会导致后面的信息覆盖前面的，从而导致信息的不完全传递。而注意力模型，则很好的解决了这个问题。

如图 4.2 是注意力模型的示意图：

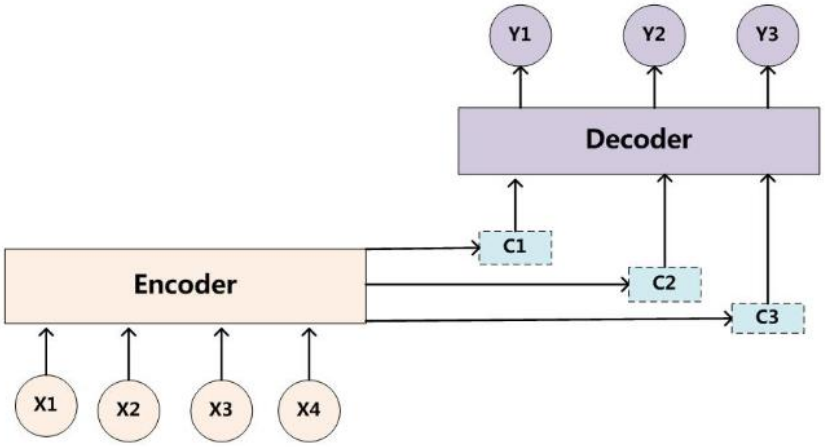

图 4.2 **注意力模型**

注意力模型不是将所有信息压缩到最后一个隐藏向量中，而是应对与每一个时间步骤的隐藏向量，并且有选择的选择这些向量集合中的子集：

$$\alpha_i = \frac{\exp\left(g\left(h_i, w; \theta_g\right)\right)}{\sum_{i'=1}^{n} \exp\left(g\left(h_{i'}, w; \theta_g\right)\right)}$$
$$c = \sum_{i=1}^{n} \alpha_i h_i \qquad \cdots\cdots (4.2)$$

参数 w 可以根据不同的任务经过训练得到，参数方程 g 可以有不同的选择方式，如点积，双线性积等。本文中通过输出向量的加权和来表示句子 r，并通过最终步获得分类的句子对表示：

$$M = \tanh\left(H\right)$$
$$\alpha = soft\max\left(w^T M\right)$$
$$r = H\alpha^T \qquad \cdots\cdots (4.3)$$
$$h^* = \tanh\left(r\right)$$

简单来说，注意力机制就是对每个 h_i 计算一个相似分数，之后通过一个 softmax，得到所有时间步骤下的一个离散概率分布。而对于分类任务，则是根据最后的 $[0,1]$ 之间的数来做分类。

注意力神经网络最近在回答问题、机器翻译、语音识别、图像字幕等一系列广泛的任务中取得了成功[25][7][26][27]。目前的注意力模型已经不仅仅在 RNN 或 CNN 中应用。最新的 BERT 等模型，都是采用了词嵌入＋注意力的框架结构，少了 RNN，CNN 的结构约束，这种框架使用的参数更少，能力鲁棒性强，而且并行能力也高，因此已经取得了相当出色的成绩。

第 5 章 分类及正则化

5.1 分类问题

采用机器学习的目的是想让计算机代替我们进行决策。对于人类而言，我们可以根据事物的特征来进行判断事情的属性。举个例子，夏天去超市买橘子，可以根据其大小，软硬，气味来判断这个西瓜的甜度。这个过程中，这些判断的依据统称为特征。与之类似，将这些特征信息化后输入计算机，通过其对已有信息的学习，从而得出这个橘子是甜还是酸的判断我们称之为分类问题。说到分类问题，不得不提的是回归问题。两者的区别，是在于最后得到的判断标准是离散的还是连续的。如果输出是连续的，则是回归问题；反之，则是分类问题。另外举一个例子，对于天气情况，如果我们是预测明天的温度情况，那么就是个回归问题，因为温度是一个连续的变化区间，可能是 13 度，也可能是 13.1 度，没有办法进行一个准确的判断；但是如果我们预测的是明天是晴天，阴天，下雨还是下雪的时候，这个问题就变成了分类问题，因为这几种天气情况是离散的。

5.1.1 线性回归

线性回归是回归问题的一种，是假设实验目标与特征关系呈线性关系。换句话说，如下图 5.1，对于坐标系中的各种代表特征的点，目的就是找出一条直线，使其可以尽可能好的拟合这些特征点。

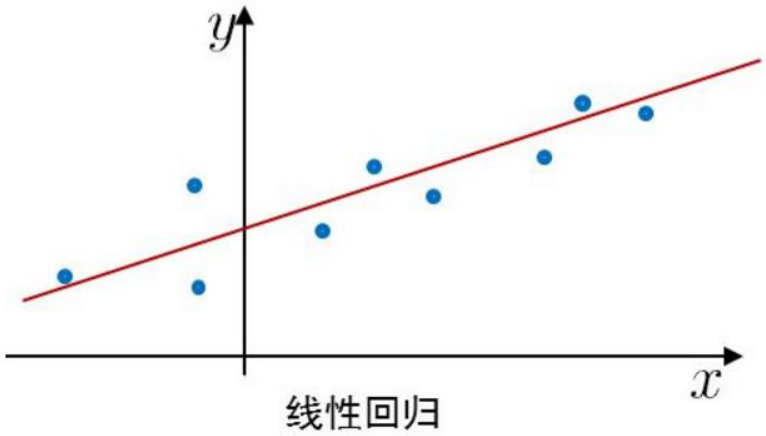

图 5.1 **线性回归**

当然，这种直线可以有很多很多条。那么，实验目的是来判断哪条直线拟合的最好。在一维特征空间中，可以假设拟合的直线为$h_\theta(x)=\theta_0+\theta_1 x$，其中，$h_\theta$为直线根据已知的特征$x$得到的预测值，$y$为特征$x$的条件下的真实值。$x$，$y$是已知的。这样，可以根据损失函数的大小来判断拟合效果，反过来说，通过找到使损失函数最小的

θ_0，θ_1值，使得我们找到这无数条直线中拟合效果最好的那条。损失函数定义如下：

$$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x)-y_i)^2 \qquad \cdots\cdots\cdots（5.1）$$

即预测值与真实值之间的均方误差，将线性函数代入，得

$$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(\theta_0+\theta_1\ x-y_i)^2 \qquad \cdots\cdots\cdots（5.2）$$

因此我们的目的是找到$(\theta_0，\ \theta_1)$，使得

$$(\theta_0，\ \theta_1)=\arg\min\sum_{i=1}^{m}(\theta_0+\theta_1\ x-y_i)^2 \qquad \cdots\cdots\cdots（5.3）$$

同理，当预测值需要多个特征综合进行判断的时候，有

$$h_\theta(x)=\theta_0+\theta_1x_1+\cdots+\theta_nx_n \qquad \cdots\cdots\cdots（5.4）$$

此时，上式可以表示为

$$h_\theta(x)=\sum_{i=0}^{n}\theta_i\ x_i \qquad \cdots\cdots\cdots（5.5）$$

将上式代入损失函数，可以得到

$$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2 \qquad \cdots\cdots\cdots（5.6）$$

因此，通过满足以下条件的θ成了解决问题的关键

$$\theta=\arg\min J(\theta) \qquad \cdots\cdots\cdots（5.7）$$

接下来，介绍两种解决这种优化问题的方法：即最小二乘法及梯度下降法。

最小二乘法：其代数解法就是对θ_i求偏导，并令偏导等于 0，之后再解方程组，最后得到θ_i。设$h_\theta(x)=\theta_0+\theta_1x_1+\cdots+\theta_nx_n$的矩阵表达式为：

$$h_\theta(x)=X\theta \qquad \cdots\cdots\cdots（5.8）$$

而损失函数我们将其定义为

$$J(\theta)=\frac{1}{2}(X\theta-Y)^T(X\theta-Y) \qquad \cdots\cdots\cdots（5.9）$$

其中，Y为样本的实际值。而$\frac{1}{2}$是为了使我们的后续计算更加容易。因此，对θ求偏导，取 0，可以得到：

$$\frac{\partial}{\partial\theta}J(\theta)=X^T(X\theta-Y)=0 \qquad \cdots\cdots\cdots（5.10）$$

则θ容易得到解，为：

$$\theta=(X^TX)^{-1}X^TY \qquad \cdots\cdots\cdots（5.11）$$

但是，当需要处理任务的特征数量大于样本数量的时候，θ解不唯一。而且，逆矩阵的过程往往消耗大量时间。不仅如此，如果需要拟合的不是直线而是曲线的时候，最小二乘法也并不适用。因此，下面我们介绍梯度下降法。

梯度下降法[23]：易知，梯度的定义就是函数在某点的方向导数沿着这个方向取得极大值，即这个函数在该点处沿这个方向的变化率最大。

因此，多元函数的梯度表达式可以写成：

$$\nabla f(x)=\left(\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2},\cdots,\frac{\partial f}{\partial x_n}\right)^T \qquad \cdots\cdots\cdots（5.12）$$

梯度下降法的基本思想是通过改变θ的取值，使函数沿着负梯度的方向不断改变，通过不断地迭代，最终，当梯度值变为 0 的时候，找到令函数取最小值的θ。因为这个方法的思想对于解决优化问题十分契合，所以被广泛应用于各种各样的机器学习求解的过程之中。

流程如下：第一步，给θ一个值，这个值可以是任意的，甚至可以是零向量；第二步，通过依次增加步长改变θ的值，使J沿着负梯度下降；最后，当梯度不再下降时，对应的θ值，即为我们想要。

假设需要求解目标函数（5.13）的最小值：

$$f(x)=f(x_1,x_2,\cdots,x_n) \qquad \cdots\cdots\cdots（5.13）$$

不妨先假设初始点为$x^0=\left(x_1^0,x_2^0,\cdots,x_n^0\right)$，因为学习率$\alpha$必须大于 0 的前提，可以建立一个迭代过程，即当$i\geq 0$时，有

$$x_1^{(i+1)}=x_1^{(i)}-\alpha\cdot\frac{\partial f}{\partial x_1}\left(x^{(i)}\right) \qquad \cdots\cdots\cdots（5.14）$$

$$x_2^{(i+1)}=x_2^{(i)}-\alpha\cdot\frac{\partial f}{\partial x_2}\left(x^{(i)}\right) \qquad \cdots\cdots\cdots（5.15）$$

$$\cdots$$

$$x_n^{(i+1)}=x_n^{(i)}-\alpha\cdot\frac{\partial f}{\partial x_n}\left(x^{(i)}\right) \qquad \cdots\cdots\cdots（5.16）$$

其中，$x^{(i)}=\left(x_1^{(i)},x_2^{(i)},\cdots,x_n^{(i)}\right)$。一旦达到收敛条件，迭代就停止。因此，对于这个迭代的过程g，有

$$x_0\overset{g}{\rightarrow}x_1\overset{g}{\rightarrow}x_2\overset{g}{\rightarrow}\cdots\overset{g}{\rightarrow}x_n$$

即对于所有的i，都满足$x^{(i+1)}=gx^{(i)}$。因此梯度下降法表达式为

$$g\left(x\right)=x-\alpha\cdot\nabla f\left(x\right) \qquad \cdots\cdots\cdots（5.17）$$

现在，将这个过程代回线性回归损失函数（4.5），θ的更新过程可以写成：

$$\theta_j:=\theta_j-\alpha\cdot\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}-y^{(i)}\right)\right)x_j^{(i)} \qquad \cdots\cdots\cdots（5.18）$$

迭代更新的方法有很多，比如随机梯度下降法（SGD）[24]等。这些方法在应用的时候，优点在于很好的节约时间，效率高，但是另一方面，可能会因为起始点的选择的随机性，以及函数的本身性质，导致迭代结束后只是得到一个局部最小值，而不是全局的最小值。

举个例子，这个寻找最优解的过程被比喻成从高处滚落一个皮球，通过其沿着最快到达最低处的路线，最终找到这个球停止的位置。但是如果这条路线上有多个很深的坑，就容易导致球在某一次迭代后，滚入坑中出不来，在反复迭代后停止在一个局部的最低点结束，而没有到达最终期望的全局最低点。

因此，在实际的实验中，设置了很多大小不一的学习率及起始位置，这样虽然不能保证迭代结果一定是全局最低点，但是也尽可能的绕开了各种局部最低点。从预测结果来看，效果还是不错的。

5.1.2 逻辑回归

逻辑回归名字上是回归，但是实际上也是分类问题。线性回归解决的问题是回归拟合任务，而逻辑回归，同样需要一条直线，只不过不是用来拟合每个样本点，而是把不同类别的样本进行分类。

本次实验中，所做的文本分类问题就是逻辑回归。对于语料库中的文本信息，包含积极和消极两种不同的感情基调。最终的测试集及预测都是一个二分类问题。前面 1.3.1 节介绍的 sigmoid 函数就是针对二分类问题很好的一个激活函数，由于函数性质，很好的关于（0,0.5）对称，且 y 恒大于 0，所以在分类中，只需要对于预测的样本通过激活函数处理，然后将结果与 0.5 进行比较。当预测值大于 0.5，则是积极的评价；若小于 0.5，则为消极评价。

5.2 拟合问题

经过学习，最终得到的模型在预测的结果上可能会出现以下三种情况，即欠拟合、拟合、过拟合情况，如下图 5.2 ：

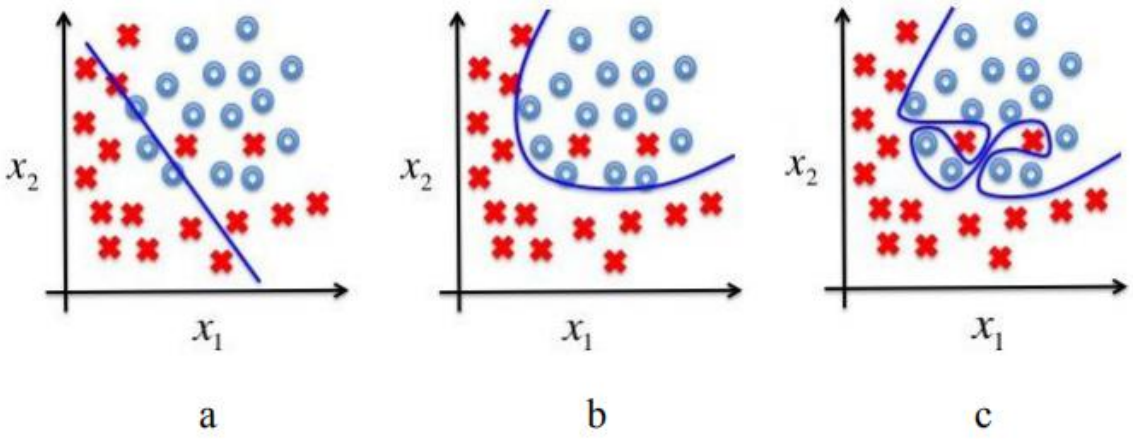

图 5.2 **拟合的三种情况**

图 5.2 中 a 图称为欠拟合，对于二分类的样本集合，通常以一条直线对该集合进行分类,但是通过观察该图以及参考日常实际情况，或许是样本中被提取的特征过少，最终会出现无论对起始点和直线斜率怎么选取，得到的直线都无法非常好的将两种样本分隔开，此时认为模型的偏差较高，即训练模型不能很好的匹配；而图 5.2 中的 b 图情况叫做拟合情况，即在图中选取了一条恰当的曲线对该样本集进行划分，这条曲线不仅能较为精确地将两种样本分隔开，而且对于新给出的样本，该曲线也可以很好的将其进行分类处理。虽然可以看到还是有少数的样本出现了错误区域，但是在这里，由于这些样本出现的位置无法通过任何简单曲线对其进行分类，就将这些点看为噪点，即误差点；图 5.2 中 c 图被称为过拟合，这种分类方式是通过一条十分精细复杂的曲线对样本集合进行彻底分类，两种样本被这条曲线十分精确的分隔开，但是由于过于精确，容易导致当我们改变验证集合或验证集合中有新的样本时，分类错误的可能性会非常大，此时认为模型的方差较高因此，对于任何机器学习模型，最终的目的都是希望可以获得图 4.2-b 图所示的训练结果。

在实验中，采用 softmax 分类器来预测标签 $\hat{y}$ 。该分类器只是将 sigmoid 局限于二分

类的情况进行扩展到多分类，但是需要二分类时，只需要将分类数设为 2，即可成为 sigmoid 的形式。分类器用隐藏状态 h^* 作为输入：

$$\begin{aligned}\hat{p}(y|S) &= soft\max\left(W^{(S)}h^* + b^{(S)}\right)\\ \hat{y} &= \arg\max_y \hat{p}(y|S)\end{aligned} \qquad \cdots\cdots\cdots (5.19)$$

损失函数是正确分类标签 $\hat{y}$ 的负对数似然：

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} t_i \log(y_i) + \lambda\|\theta\|_F^2 \qquad \cdots\cdots\cdots (5.20)$$

其中 y 是用 softmax 对每一类估计概率，λ 是 L_2 正则化的超参数，m 是目标分类数。我们还加入了 dropout[29]来解决过拟合情况。我们的 dropout[29]都用在了嵌入层，LSTM 层以及分类层。

第 6 章 实验

6.1 数据集&实验设置

本文的实验数据集是取自于互联网电影资料库 IMDB（Internet Movie Database）的电影影评。其中一共包含了 50000 条已经做好标记的评论，25000 条是积极评价，另外 25000 条则是消极的评价，评论内容两极化明显。本实验的目的就是让计算机通过学习这些影评文本信息，最终对于给出的预测文本能进行正确分类。

本次实验中，总共计划对四个模型进行了对比实验，分别是

模型 1：one-hot+BiLSTM+Attention 模型；

模型 2：textCNN 模型；

模型 3：Word2vec+BiLSTM 模型以及

模型 4：Word2vec+BiLSTM+Attention 模型。

但是实际实验过程中，由于 one-hot 模型占用内存过大，导致模型 1 出现运行报错的情况，因此这里只给出后三种模型的实验结果对比。

设置情况如下，在训练方面，周期数设置为 1，5，10 进行了三次实验。这里的周期数可以理解为将评论一共进行几次循环。每训练 100 次，我们进行一次精准度方面的输出，分类数设为 2，学习率设置为 0.001 及 0.1，检验点也设为 100。在模型方面，词嵌入尺寸设置为 200，隐藏层尺寸设置为[256,256]以及[128,128]，[512,512],dropout 始终保持在 0.5，训练集占整个样本集的 80%。

6.2 实验结果

以下为实验结果，在实验过程中图 6.1—图 6.3，图 6.4—图 6.6，图 6.7—图 6.9 是仅周期改变，隐藏层尺寸均为[256,256]的情况下，模型 2,3,4 的准确率变化情况。

图 6.1 **模型 2，周期 1**

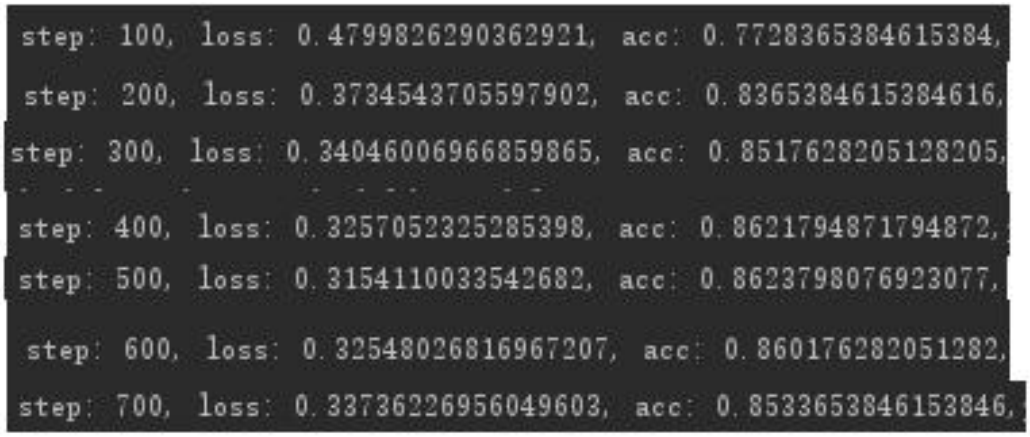

图 6.2 模型 2，周期 5

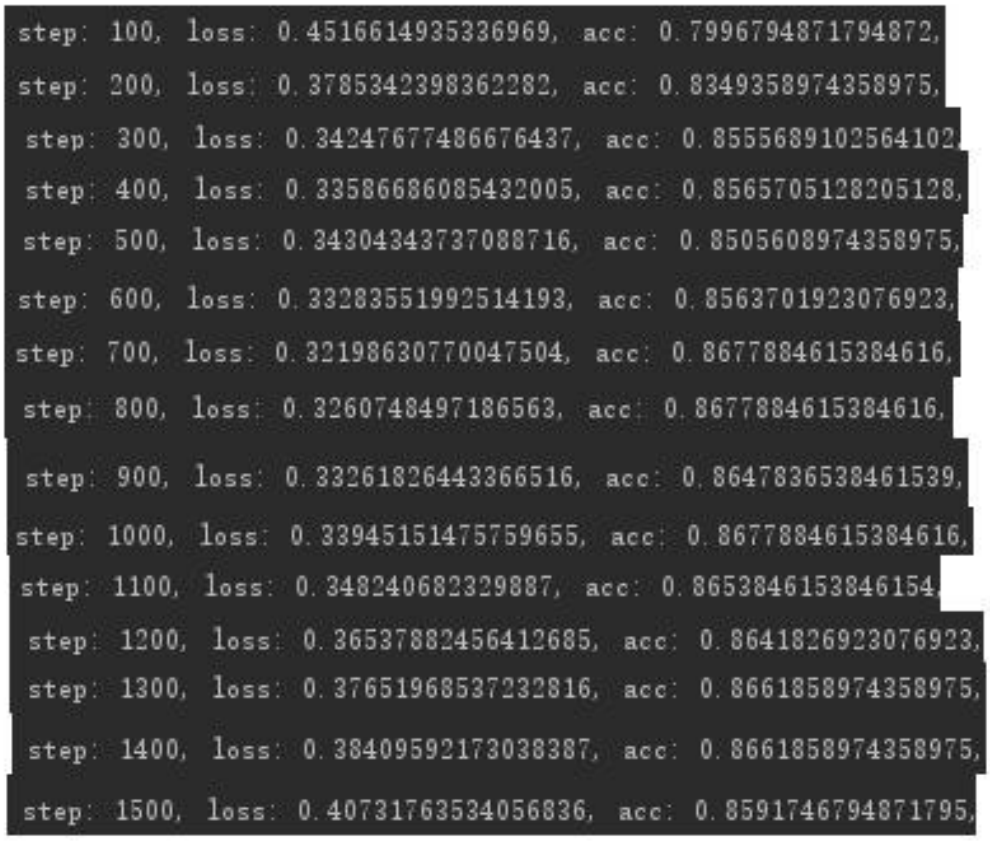

图 6.3 模型 2，周期 10

图 6.4 模型 3，周期 1

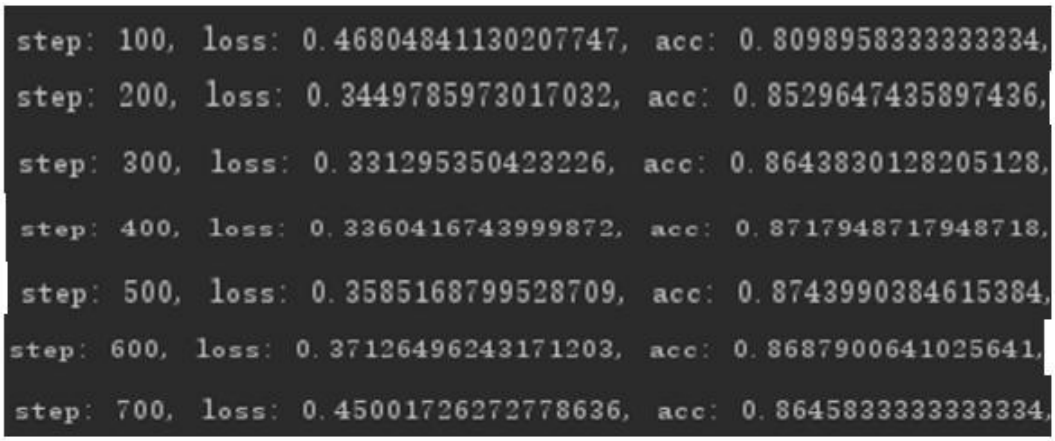

图 6. 5 模型 3，周期 5

图 6.6 模型 3，周期 10

图 6.7 模型 4，周期 1

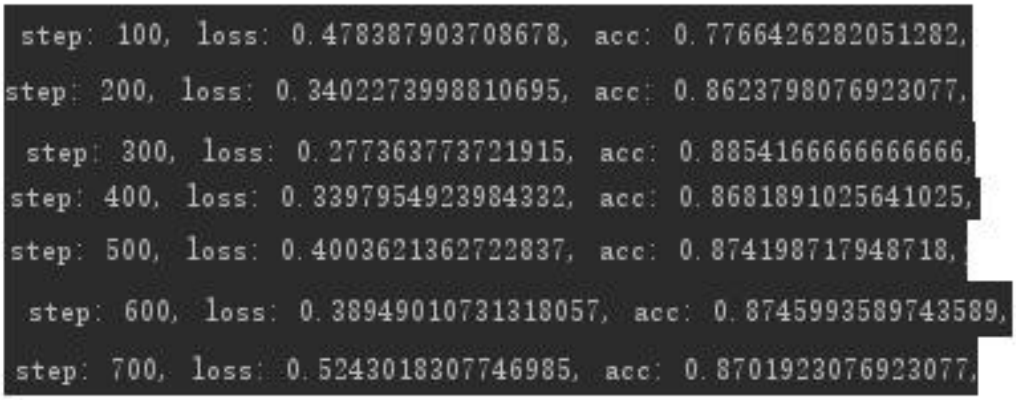

图 6.8 模型 4 周期 5

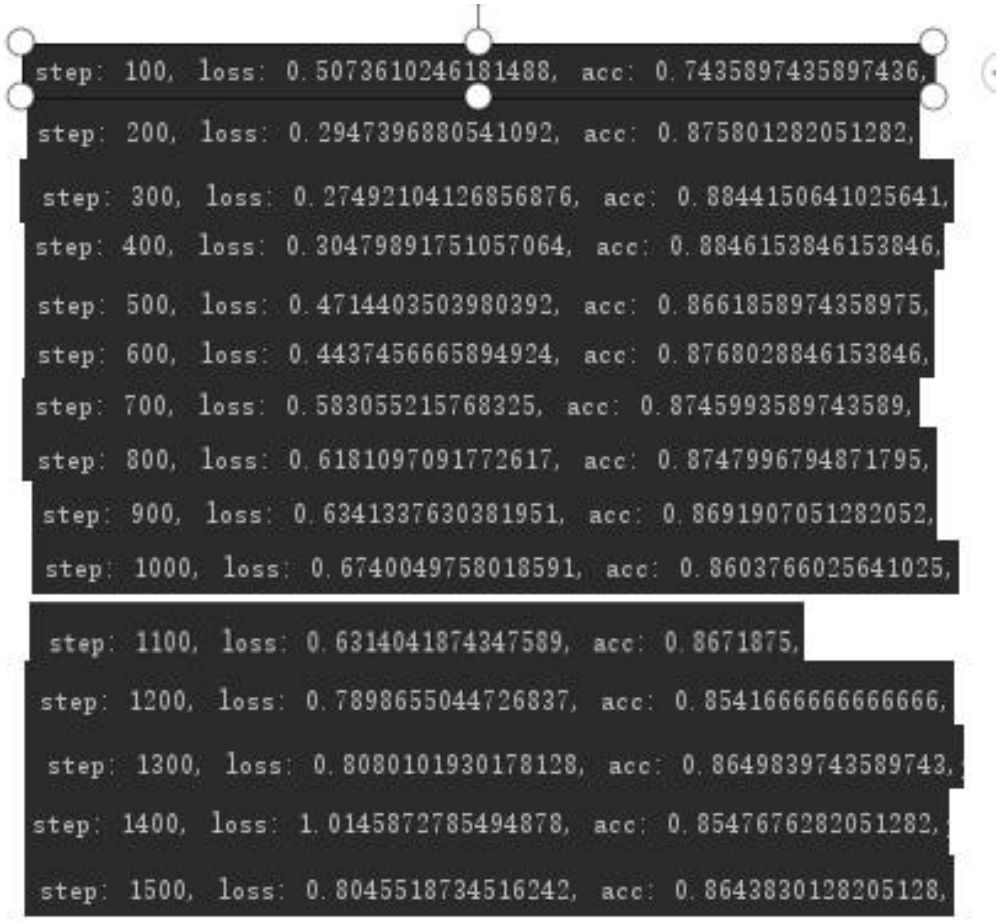

图 6.9 模型 4，周期 10

数据整理后，在表 6.1 中，可以看出一方面，三种模型相比，在相同参数设置的情况下，最终提出的 Word2vec+BiLSTM+Attention 模型在 10 轮周期的循环后最高能达到 88.46%的准确度，比原模型高 1.2%。比 textCNN 模型高 1.7%；在 5 轮周期的循环后，新模型能最高达到 88.54%，比同条件下原模型精度高 1%，比 textCNN 模型高 2.3%；另一方面，不难观察到，适当的增加数据循环的周期数可以提高预测精度，但是过度增加周期数不能在精度方面一直取得提高。不考

虑循环周期过低的情况，尽管 textCNN 在各项实验中准确率都是最低的，但是值得一提的是，当用 textCNN 模型训练时，训练速度要比其他两个模型快至少一倍以上，经过分析总结，得出的结论是在训练过程中，CNN 采用的池化方式相比于 RNN 的链式结构需要上一个时间状态的信息来推出当前时间状态要快捷。因此，在时间较为紧张的情况下，可以采用 textCNN 模型学习，从而适当降低准确度提高效率。

表 6.1 不同模型在不同周期下预测最高准确度

	周期为 1	周期为 5	周期为 10
textCNN	78.02%	86.24%	86.78%
word2vec+BiLSTM	80.11%	87.44%	87.26%
Word2vec+BiLSTM +Attention	75.90%	88.54%	88.46%

接下来我们准队隐藏层尺寸变化进行了对比实验：图 6.10，图 6.11，图 6.12，图 6.13 表示周期设为 5，隐藏层变化为[128,128]，[512,512]及[16,16]时的后两种模型的准确率变化，因为 textCNN 并不具备 LSTM 结构，因此这里没有对其做对比实验。

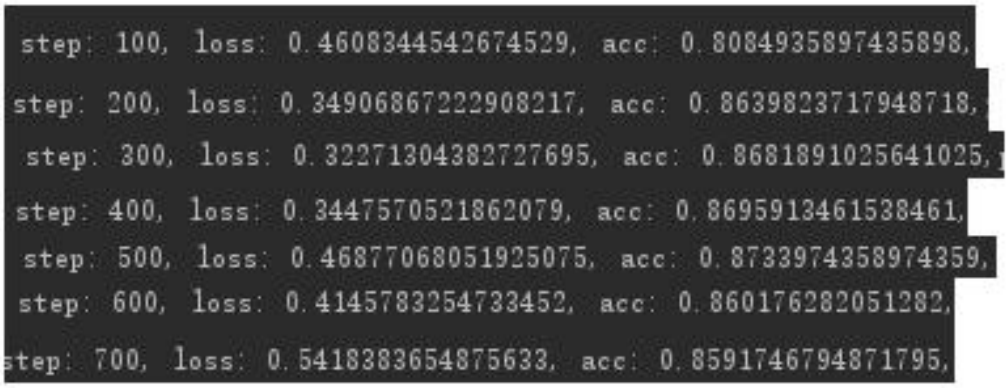

图 6.10 模型 3，隐藏层尺寸[128,128]

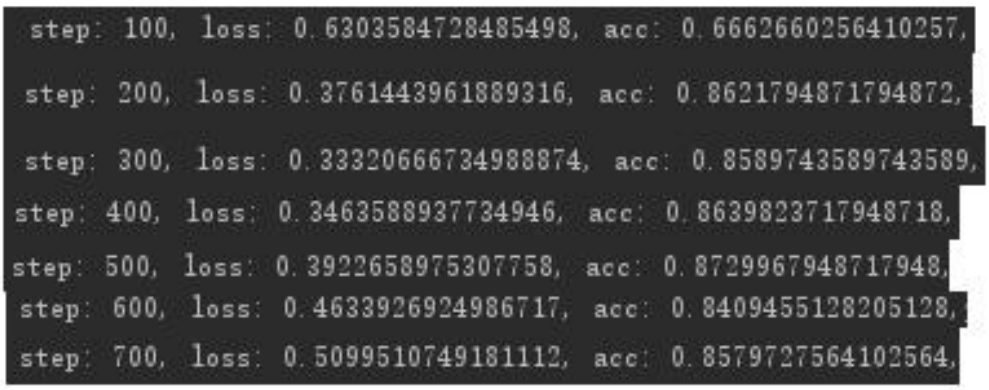

图 6.11 模型 4，隐藏层尺寸[128,128]

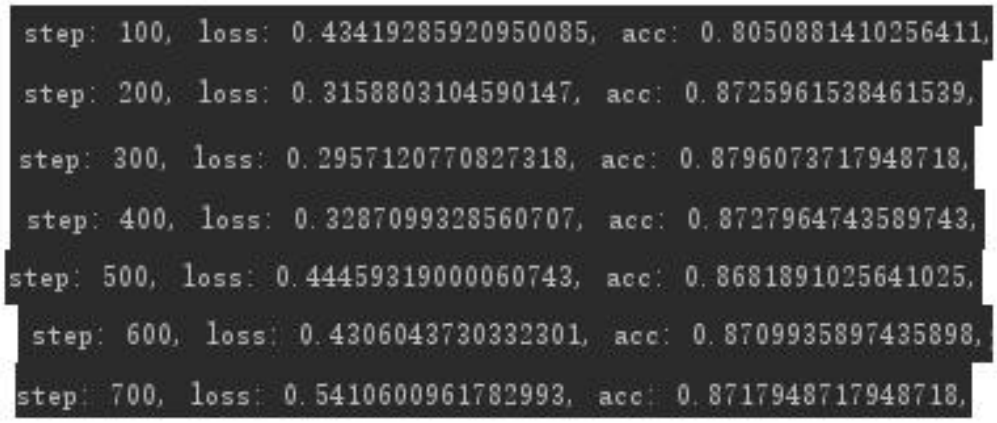

图 6.12 模型 3，隐藏层尺寸[512,512]

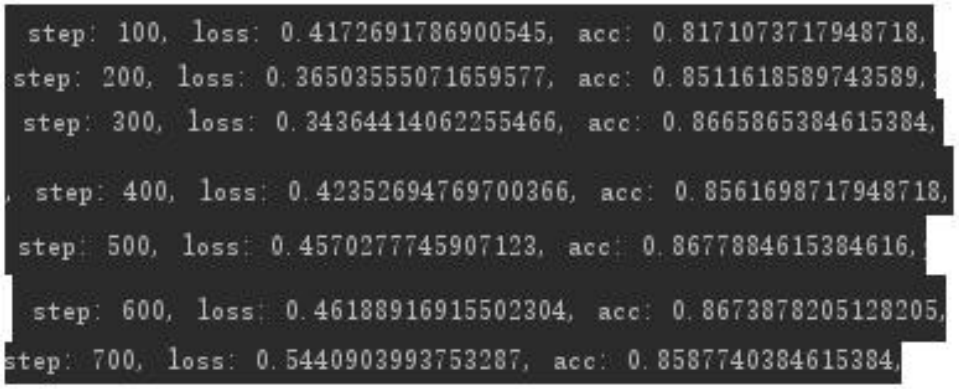

图 6.13 模型 4，隐藏层尺寸[16,16]

由表 6.2，表 6.3 可以观察到，对于两种模型而言，隐藏层尺寸的改变会影响准确度。在当前的设置条件下，隐藏层尺寸的增加均会使两种模型的准确度得到提高。但是在时间花费上，隐藏层尺寸选择[512,512]的时候，要耗时更长一些。

表 6.2 原模型在隐藏层尺寸不同时的最高准确度

	隐藏层为[128，128]	隐藏层为[256，256]	隐藏层为[512，512]
Word2vec+BiLSTM	87.34%	87.44%	87.96%

表 6.3 新模型在隐藏层尺寸不同时的最高准确度

	隐藏层为[128，128]	隐藏层为[256，256]	隐藏层为[16，16]
Word2vec+BiLSTM+Attention	87.30%	88.54%	86.79%

最后，还针对学习率变化做了一组对比实验。将新模型的学习率从 0.001 改为 0.1 后，实验结果如图 6.14 所示：

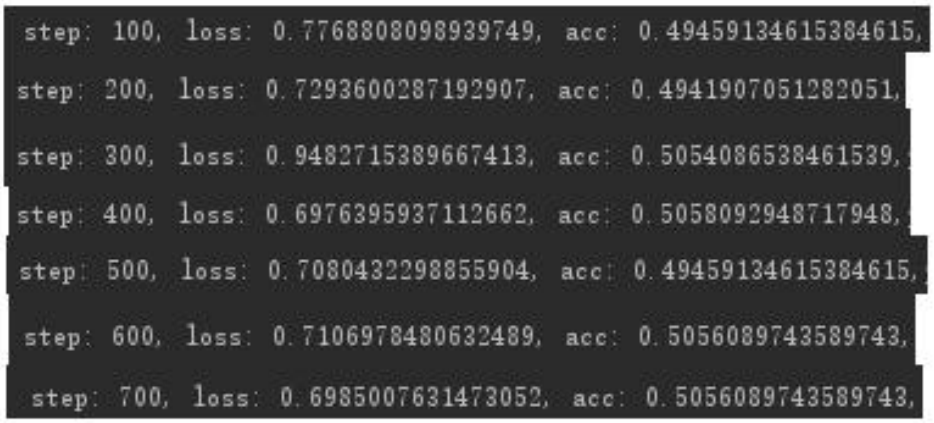

图 6.14 模型 4，学习率为 0.1

可以观察到，过高的提高学习率，会使模型不收敛，准确度一直停留在 0.5 左右。

值得一提的是，当我们在对所有参数改变的实验的过程后的预测环节，即针对给出的一个语料库中的积极含义的句子，应用我们训练完成的模型来判断它的意思是积极还是消极的的过程中，周期为 1 的时候，出现过判断错误。但是在其他的情况下，模型们都成功的预测出了该句子的正确表达意思。因此，周期过小对于机器很好的学习模型是有一些难度的。

第 7 章 结论和展望

研究生三年期间，针对目前机器学习的研究热点领域，自然语言处理 NLP，主要研究了语音检测，机器阅读理解及文本分类，对于该领域有了较深入的了解。最终选择在文本分类方向完成论文。在有限的实验条件下，由于内存爆炸，没有完成 one-hot 为词嵌入方式的模型实验。改换了新的词嵌入模型 word2vec 后，解决了内存爆炸等问题，并在原模型基础上添加了注意力模型后得到了一定程度上精度的提升。在分类问题中，取得了较好的结果。在神经网络方面，得出并不是一味的增加隐藏层尺寸就可以得到更高的准确度的结论及过高的学习速率会导致训练的模型并不收敛。不仅如此，还选择 textCNN 做对照实验，得出在本实验中，该卷积神经网络训练精度不如循环神经网络，但是训练速度确实高于循环神经网络的结论。

目前的 NLP 领域，Google 团队新提出的 BERT 算法[19]等无监督学习的方法很是火热。不光在学习结果上，较这些传统的方法又有了较大的提升；也进一步解决了过去无法处理无标注的数据集，必须人工对数据集进行标注的难题。尽管在学习过程中，因为少了标注，想达到相同的准确度需要在数据集的要求上比原来规模大很多，但是处于如今这样一个网络发达，数字化，信息化的时代，没有标注的数据集的收集也变得更加容易。因此，这也将是未来的一个重要发展趋势。相信在未来的数十年里，神经网络也能得到飞速发展，人类生活水平也会因此得到更大的提升。

在未来的学习过程中，我将更深层次的针对无监督学习方向开展工作。不仅如此，在研究方向上，也会努力涉及更多的之前没有接触过的领域。相信经过钻研，可以对自然语言处理这个方向有更深的认识，在实现真正人工智能的路上，贡献自己的一份力量。

参考文献

[1] WANG Yun. Polyphonic Sound Event Detection with Weak Labeling[D]. 5000 Forbes Avenue, Pittsburgh, PA 15213:Language Technologies Institute School of Computer Science Carnegie Mellon University, 2018.
[2] CHEN Danqi. Neural Reading Comprehension and Beyond[D]. Stanford University, 2018.
[3] Bengio Y., Schwenk H.,等.Neural probabilistic language models[M].Innovations in Machine Learning.137-186.2006.
[4] Mikolov T., Chen Kai, Corrado G.,等. Efficient estimation of word representations in vector space[C]. International Conference on Learning Representations, 2013.
[5] Mikolov T., Sutskever I., Chen Kai,等. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.
[6] Kandola E.J., Hofmann T., Poggio T.,等. A Neural Probabilistic Language Model[J]. Studies in Fuzziness & Soft Computing, 2006, 194:137-186.
[7] Bahdanau D.. Neural Machine Translation By Jointly Learning To Align And Translate[C]. International Conference on Learning Representations, 2015.
[8] Graves A., Mohamed A.R., Hinton G.. Speech Recognition with Deep Recurrent Neural Networks[J]. Acoustics, Speech, and Signal Processing, 1988. ICASSP-88. 1988 International Conference on, 2013, 38.
[9] Hochreiter S., Schmidhuber J.. Long Short-Term Memory[J]. Neural Computation, 9(8):1735-1780.
[10] 程依明 濮晓龙。概率论与数理统计教程[M]。北京：高等教育出版社，2011:48。
[11] Han J., Moraga C.. The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning[C]// International Workshop on from Natural to Artificial Neural Computation. Springer-Verlag, 1995.
[12] Hinton G.E.,McClelland J.L.,Rumelhart D.E.. Distributed representations[J]. Parallel distributed processing: Explorations in the microstructure of cognition. Volume1:Foundations,MIT, 1986.
[13] Jeffrey L.Elman. Finding Structure in Time[J]. Cognitive Science, 14(2):179-211.
[14] Rumelhart D.E., Hinton G.E., Williams R.J.. Learning Representations by Back Propagating Errors[J]. Nature, 1986, 323(6088):533-536.

[15] Mcculloch W.S., Pitts W.. A Logical Calculus of the Ideas Immanent in Nervous Activity[J]. bulletin of mathematical biology, 1943, 52(1-2):99-115.
[16] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain.[J]. Psychological Review, 1958, 65(6):386-408.
[17] Ma Xuezhe, Hovy, Eduard. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF[J].
[18] Lecun Y., Boser B., Denker J.,等. Backpropagation Applied to Handwritten Zip Code Recognition[J]. Neural Computation, 1989, 1(4):541-551.
[19] Devlin J., Chang M.W., Lee K.,等. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J], 2018.
[20] Wojciech Z.,Sutskever I., Vinyals O.. Recurrent neural network regularization[C]. International Conference on Learning Representations, 2015.
[21] Felix G.. Long short-term memory in recurrent neural networks[D]. Verlag nicht ermittelbar, 2001.
[22] Cho K., Van Merrienboer B., Gulcehre C.,等. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. computer science, 2014.
[23] Sutskever I., Vinyals O., Le Q.V.. Sequence to Sequence Learning with Neural Networks[J]. Advances in neural information processing systems, 2014.
[24] Ruder S.. An overview of gradient descent optimization algorithms[J], 2016.
[25] Léon Bottou. Stochastic Gradient Descent Tricks[J], 2012.
[26] Hermann K.M., Koisk Tomá, Grefenstette E.,等. Teaching Machines to Read and Comprehend[J], 2015.
[27] Chorowski J.K., Bahdanau D., Serdyuk D.,等. Attention-based models for speech recognition[J]. Advances in Neural Information Processing Systems, 2015:577-585.
[28] Kelvin Xu, Jimmy Ba, Ryan Kiros,等. Show,attend and tell:Neural image caption generation with visual attention[C]. International conference on machine learning, 2015,2048-2057.
[29] Kim Yoon. Convolutional Neural Networks for Sentence Classification[J], 2014.
[30] Srivastava N.. Dropout:a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014,15.1:1929-1958.

致谢

研究生三年时间过得好快，毕业论文也终于在最后这个学期开始之前完成了。在这里我首先想真心感谢我的研究生导师罗宏文老师。从刚刚入学对于机器学习这个领域的一无所知到今天的有了较深的了解，最终完成这篇论文，可以说这些进步离不开了老师一点一滴的耐心教导。刚刚入学那会老师说过的话还回荡在耳边，当你对于这个领域还不太了解的时候，你就需要去找些这个专业里大牛的文章来看，这样你才能更快更好的从中找到你需要的东西，或许一开始读起来，英语阅读加上专业知识的匮乏会有些困难，但是坚持住，你会受益匪浅。如今的我不仅专业知识，连英文文献阅读能力相比三年前也有了质的飞跃。我坚信在我以后漫长的人生道路上，面对着接踵而来的新问题新挑战，这些话也犹如明灯，指引着我的方向。不仅如此，老师严谨细致，一丝不苟的求学态度也深深的影响着我。当讨论班汇报我们最近的学习情况时，我理解有偏差或者没有深入思考的地方总是第一时间被老师发现，然后在课后默默给我发来相关的论文，帮助我战胜这些困难；改论文的时候，老师也会发现我各种各样的写作中的小问题，帮我细致的标注出来。生活方面，老师一改学术研究时的严肃，变得十分和蔼，平易近人。我比较喜欢运动，当得知我要参加学校的游泳比赛时，老师笑着和我说水平怎么样啊，能不能拿个第一啊？加油加油。去年冬天我因为滑雪不小心手腕骨折，后来去吉大一院讨论开展合作时被老师得知，结束后的中午老师提议师门聚餐去吃酱骨头，说这样能补补钙，利于恢复。之后伤愈后还总是问起我手腕的恢复情况。我想我会一辈子记得老师当时关心的神情。

其次我想感谢我的室友：彭辉，孟祥龙，许景昭。平时看论文推导公式遇到问题，包括最后论文写参考文献过程中我曾遇到找不到格式等问题时，他们耐心的帮我一遍遍解答都让我铭记于心，平时生活上也感谢他们对我的包容。

最后我想对我的妈妈爸爸说声谢谢。谢谢你们对我这么多年来的悉心的照顾。我不是个善于把感谢说出来的性格，但是你们为我做的那些付出，我都看在眼里。谢谢你们。