摘 要

ABSTRACT

目 录

[摘 要 I](#_Toc8722046)

[ABSTRACT II](#_Toc8722047)

[目 录 III](#_Toc8722048)

[第一章 绪 论 1](#_Toc8722049)

[1.1 选题背景与意义 1](#_Toc8722050)

[1.1.1 从人工智能到深度学习 1](#_Toc8722051)

[1.1.2 股票时间序列的研究 2](#_Toc8722052)

[1.2 研究现状 2](#_Toc8722053)

[1.2.1 股票市场的相关研究 2](#_Toc8722054)

[1.2.2 深度学习在股票市场的相关研究 3](#_Toc8722055)

[1.3 本文主要内容 5](#_Toc8722056)

[1.4 本文的组织结构 5](#_Toc8722057)

[第二章 深度学习理论基础 6](#_Toc8722058)

[2.1 人工神经网络 6](#_Toc8722059)

[2.2全连接神经网络 10](#_Toc8722060)

[2.2.1 前馈神经网络 10](#_Toc8722061)

[2.2.2 全连接层和全连接神经网络 12](#_Toc8722062)

[2.2.3 损失函数 13](#_Toc8722063)

[2.2.4 全连接神经网络的训练方法 13](#_Toc8722064)

[2.3 循环神经网络 14](#_Toc8722065)

[2.3.1 循环神经网络的结构 14](#_Toc8722066)

[2.3.2 循环神经网络的训练 15](#_Toc8722067)

[2.4 长短期记忆网络 18](#_Toc8722068)

[2.4.1 长短期记忆网络的结构 18](#_Toc8722069)

[2.4.2 长短期记忆网络的训练 20](#_Toc8722070)

[2.6 神经网络训练的优化方法 24](#_Toc8722071)

[2.6.1 梯度下降算法的缺陷 24](#_Toc8722072)

[2.6.2 解决方法 24](#_Toc8722073)

[第三章 模型构建 26](#_Toc8722074)

[3.1 使用工具 26](#_Toc8722075)

[3.1.1 Python 26](#_Toc8722076)

[3.1.2 Numpy 26](#_Toc8722077)

[3.1.3 PyTorch 26](#_Toc8722078)

[3.1.4 Pandas 27](#_Toc8722079)

[3.1.5 Matplotlib 27](#_Toc8722080)

[3.2 基于LSTM的神经网络模型构建 27](#_Toc8722081)

[3.4 数据处理 28](#_Toc8722082)

[3.5 训练模块 29](#_Toc8722083)

[第四章 股票数据的选取 30](#_Toc8722084)

[4.1 数据来源 30](#_Toc8722085)

[4.2 上证指数 30](#_Toc8722086)

[4.3 深证成指 30](#_Toc8722087)

[4.4 个股股价 30](#_Toc8722088)

[4.4.1 大盘股与小盘股 30](#_Toc8722089)

[4.4.2 成分股 30](#_Toc8722090)

[第五章 实验分析 32](#_Toc8722091)

[5.1 参数设置 32](#_Toc8722092)

[5.2 整体效果 32](#_Toc8722093)

[5.2.1 上证指数预测 32](#_Toc8722094)

[5.2.2 深证成指预测 33](#_Toc8722095)

[5.2.3 个股股价预测 34](#_Toc8722096)

[5.3 评估指标 42](#_Toc8722097)

[5.3.1 均方误差 42](#_Toc8722098)

[5.3.2 泰尔不平等系数 42](#_Toc8722099)

[5.3.3 平均偏差 43](#_Toc8722100)

[5.3.4 数据准确性 43](#_Toc8722101)

[5.3.5 平均绝对百分误差 43](#_Toc8722102)

[5.3.6 相关系数 43](#_Toc8722103)

[5.4 综合评价 43](#_Toc8722104)

[5.5 投资方案 48](#_Toc8722105)

[第六章 结 论 49](#_Toc8722106)

[致 谢 50](#_Toc8722107)

[参考文献 51](#_Toc8722108)

第一章 绪 论

1.1 选题背景与意义

股票是股份制公司出于募集资金的需要发行给股东的一种有价证券。股东每拥有一支股票就代表对公司有一个基本单位的所有权。人们将已经发行的股票聚集在一起，设立一个专门的交易、流通股票的场所，这就是股票市场。股票市场有着严格的管理，这也使得人们放心地在股票市场中进行交易。17世纪荷兰和英国成立了海外贸易公司。这些公司的建立是为了筹集资本。在经历了4个多世纪的今天，股票市场已经进入了大多数国家。而且在当今的世界经济格局中，各个国家的股票市场已经拥有了不可或缺、举足轻重的地位。对于在股票市场中投资的股民来讲，赚钱是他们的首要目的。但是股票市场中的股票又有着高风险性，一句“股市有风险，入市需谨慎。”劝退了很多想进入股票市场分一杯羹的人。对于投资公司来讲，如果他们有办法预测股票未来的走向，毋庸置疑，他们就可以获得利润。所以，股票价格、走势的预测就成了上百年来人们追求的目标。

1.1.1 从人工智能到深度学习

人工智能是一种使用计算机模拟人类的智能的方法和技术，它属于计算机科学。现有的图像识别技术、机器人、语音识别技术等都属于人工智能的范畴[[1](#_ENREF_1)]。

人工智能是在1956年的达特茅斯会议中由几个计算机科学家提出的。他们预言人工智能会在不久的将来和人类一样智能，但是他们的预言显然没有成为现实。

随着时代的发展，计算机科学家们提出了机器学习的概念。机器学习属于人工智能的范畴，是使计算机拥有智能的最有效的途径。机器学习使用计算机来执行一个没有给出明确指令的特定任务，它要求计算机学习人类的智能行为。

因为机器学习的提出、发展并解决了一系列问题[[2](#_ENREF_2)]，使人们再次关注到人工智能，许多研究者投入研究。在经历了半个多世纪的今天，因为GPU的发展，运算速度变得更快，并且因为数据量的提升和存储设备的发展，使得深度学习的热潮爆发。

深度学习[[3](#_ENREF_3)]是机器学习的一部分。深度学习主要通过搭建有层次的人工神经网络来解决问题，其中包括常用于目标检测的卷积神经网络[[4](#_ENREF_4)]和常用于处理时间序列的循环神经网络[[5](#_ENREF_5),[6](#_ENREF_6)]。

1.1.2 股票时间序列的研究

股票数据和其它的类似于图片、文本等的数据不一样，它是一种时间序列数据，前面的数据会影响到后面的数据。股票数据与自然语言类似，是一种时间序列数据。

在深度学习理论成熟之前，人们在股票市场预测领域主要采取一些传统统计学、小波变换、时间分析等方法预测股票价格以及走势。但由于影响股票价格的因素过多（政治、经济发展状况、新闻等），使这些传统办法有局限性[[7](#_ENREF_7)]。

在深度学习领域，针对这种时间序列数据，可以使用循环神经网络、长短期记忆网络等神经网络结构来处理这些数据的分析和研究问题。所以，有很多计算机科学家开始使用循环神经网络来分析时间序列数据，，所以使用深度学习来研究股票市场的时间序列数据是非常有前景的。

股票市场研究领域也因为这些网络结构的兴起而有着强大的生命力。虽然深度学习在股票市场预测的研究中相比一些传统方法有优势，但深度学习算法未被应用于更广泛的股市预测领域。如今的股票市场研究领域，大多在研究、预测标准普尔指数和纳斯达克指数。这些新提出的深度学习算法是否能同样适用于中国股市未可知。所以，本文以这作为落脚点和出发点，深入探讨如今越来越先进的深度学习算法，是否能很好地预测中国股市未来的发展。

1.2 研究现状

近年来，金融市场在我国发挥着的作用越来越显著、地位不断提升，随着国民经济的发展和金融服务业的完善，在金融市场中起着关键总用的股票市场已经引起了国内外学者和投资者的关注。他们定期提出各种可应用于实践的理论，试图预测市场趋势。在如今深度学习发展的基础上，神经网络在模式识别[[8](#_ENREF_8)]、金融证券等领域得到了广泛的应用。

1.2.1 股票市场的相关研究

James B. Ramsey在1999年发表的论文中[[7](#_ENREF_7)]综述了小波技术在金融时间序列分析领域的应用，他得出了小波特别适用于处理金融市场变化莫测的时间序列数据的结论。

Yakov Amihud在2002年发表的论文中[[9](#_ENREF_9)]通过研究股票的非流动性发现股票的流动性不足会对股票收益有一个正向的影响。

Qiang Song和Brad S. Chissom在论文中[[10](#_ENREF_10)]提出了模糊时间序列模型的概念并讨论了模糊时间序列模型的性质。

Kunhuang Huarng和Hui-Kuang Yu在论文中[[11](#_ENREF_11)]针对[[10](#_ENREF_10)]的模糊时间序列模型提出了Type 2模糊时间序列模型，他们认为在预测方面Type 2模糊时间序列模型在预测方面表现的效果比[[10](#_ENREF_10)]中的模糊时间序列模型更好。

Salim Lahmiri在其2015年的论文中[[12](#_ENREF_12)]提出了一个新模型：变分模式分解（Variational mode decomposition, VMD）。他使用了2008年经济危机的前、中、后的纳斯达克股指数据对模型的效果进行了评价，

1.2.2 深度学习在股票市场的相关研究

深度学习与股票市场的结合，最早可以追溯到1988年，White和Helbert第一次使用神经网络预测股票市场时间序列，他们使用IBM公司股票日收益作为研究对象[[13](#_ENREF_13)]。

在Kamijo和Tanigawa的研究中已经使用RNN代替了波动性预测模型来预测股价[[14](#_ENREF_14)]。

Tsung-Jung Hsieh、Hsiao-Fen Hsiao和Wei-Chang Yeh在2011年的论文中[[15](#_ENREF_15)]，使用了小波变换和循环神经网络结合的方法，并使用人工蜂群算法对循环神经网络进行优化，实现了对道琼工业指数、伦敦FTSE100指数、东京Nikkei225指数、台湾证券交易所资本化加权股票指数的预测。

Lawrence Takeuchi和Yu-Ying (Albert) Lee在2013年的论文中[[16](#_ENREF_16)]使用深度学习制作了一个分类器，这个分类器可以提供股票购买策略。使用这个分类器提供的购买策略可以使股票收益达到45.93%。Takashi Huremoto在2014年的论文中[[17](#_ENREF_17)]再次使用了深度信念网络和限制玻尔兹曼机结合的方法对CATS基准时间序列数据进行了预测，并且加入了粒子群算法。

Luca Di Persio和Oleksandr Honchar在2016年的论文中[[18](#_ENREF_18)]使用了多层感知器（MLP）、卷积神经网络、长短期记忆网络来预测股票的涨跌。他们把股票市场的预测看成一个分类问题，使用标准普尔指数作为训练和测试的数据。他们的模型可以达到56.9%的准确度。

Matthew Dixon、Diego Klabjan和Jin Hoon Bang在他们2016年的论文中[[19](#_ENREF_19)]使用了一个深度神经网络并配合SGD算法实现了对于金融数据的预测，他们的优势点是预测速度更快。

Jacinta Chan Phooi M’ng和Mohammadali Mehralizadeh在他们2016年的论文中[[20](#_ENREF_20)]使用了小波将噪声配合人工神经网络的方法实现了对东亚股指的预测，他们分别预测了香港、日本、韩国、新加坡的股指。他们对比了小波主成分分析和神经网络结合算法（WPCA-NN）、小波和神经网络结合算法（WNN）、神经网络算法（NN）三种算法得出WPCA-NN算法更优的结论。

Wei Bao、Jun Yue和Yulei Rao在他们2017年的论文中[[21](#_ENREF_21)]提出了一个使用小波变换来对股票数据降噪声，使用栈式自编码器处理股票数据，最后用长短期记忆网络进行股价预测的模型，他们预测的是六个股票市场股指的收盘价。通过他们的模型，一年的收益低于40%。

Hao Chen、Keli Xiao、Jinwen Sun和Song Wu在2017年的论文中[[22](#_ENREF_22)]提出了一个双层神经网络模型对100支标准普尔指数中的股票进行预测。他们对比了算法对于信息、健康、工具、金融、消费、能源、材料、通信几个领域的效果。

Yanhui Chen、Kaijian He、Geoffrey K.F. Tso在2017年的论文中[[23](#_ENREF_23)]使用了深度信念网络和长短期记忆网络对石油价格的涨跌进行了预测。

Thomas Fischer和Christopher Krauss在他们2018年的论文中也使用了长短期记忆网络对金融市场进行预测。在他们的研究中，使用了一个逻辑回归分类器，使用LSTM算法，年收益率可以达到43%左右。

Hongju Yan和Hongbing Ouyang在2018年的论文中[[24](#_ENREF_24)]同样使用了小波分析和长短期记忆网络结合的方法对金融时间序列数据进行预测。他们使用数据的是上证指数。他们对比了LSTM、MLP、SVM、KNN几个算法求得的Theil值和MAPE值。

Xiongwen Pang和Yanqiang Zhou等人在2018年的论文中[[25](#_ENREF_25)]使用了一个新的方法。他们分别使用嵌入层结合长短期记忆网络和自编码器结合长短期记忆网络的方法，它们的准确率分别达到了57.2%和52.5%。

1.3 主要研究内容

通过了解国外研究现状，我们发现大部分的研究和文献中，均涉及或使用了LSTM的方法，所以在本文中同样以当前备受关注的深度学习算法为基础，深入研究LSTM对股票市场的研究效果如何。而且大部分文献中都分析了标准普尔指数和纳斯达克指数，所以本文决定分析国内的上证指数、深证成指和一些个股。分别考察输入数据的长度、训练次数对算法预测精确度的影响。通过改变参数，提升算法的精确度。

1.4 文章结构

本文分为6个章节，具体结构和章节主要内容如下：

第一章为绪论，主要阐述股票时间序列分析以及深度学习领域的研究现状、背景和意义以及文章的主要内容。

第二章为深度学习理论基础，主要内容是细致阐述本文中涉及到的几种深度神经网络结构，包括全连接神经网络、循环神经网络、长短期记忆网络。同时还会细致介绍它们的训练方法，优化方法等。

第三章为模型构建，主要介绍本文采用的适用于对股票市场时间序列分析研究的深度学习模型，通过PyTorch神经网络框架、Pandas时间序列分析模块、Matplotlib图表绘制模块、Numpy科学计算模块等构建模型。

第四章为股票市场数据的选取，主要内容是分析选取的股票时间序列数据，主要是中国上证指数、大盘股、小盘股、成分股等，以及选取这些数据的原因。

第五章为实验分析，主要展示实验的结果，并对结果进行分析、对各种可以应用到股票市场时间序列分析研究的深度学习算法进行比较，分析这些算法的优劣势。

第六章为结论，对整个研究进行总结，分析本文的不足，以及展望以下的研究内容。

第二章 深度学习理论基础

2.1 人工神经网络

在生物学中，生物获取到的外界信息是通过它们的神经网络系统传递到大脑的。神经网络由神经细胞通过复杂的连接方式组合而成。生物神经网络是通过传递电信号来完成运转的。生物神经细胞传递电信号的部件是树突和轴突两个部分。树突负责接收电信号，轴突负责发出电信号。图2-1是一个生物神经元。

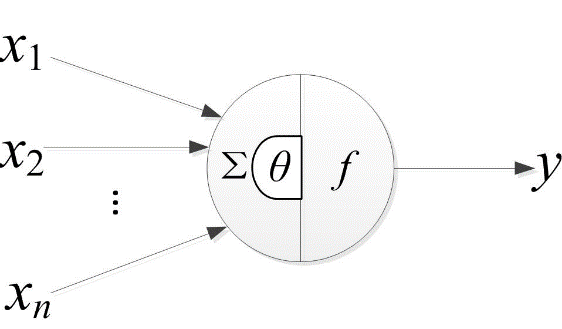
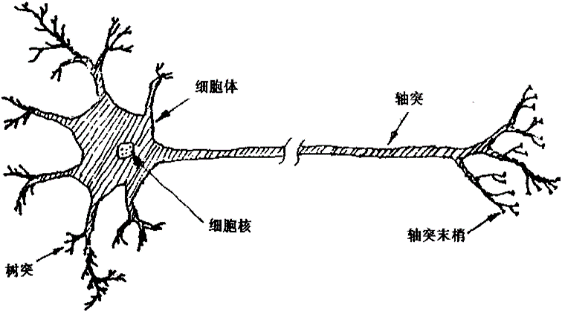


图2-1 生物神经元[[26](#_ENREF_26)] 图2-2 人工神经元[[27](#_ENREF_27)]

生物神经网络非常的复杂，但是把整个生物神经网络简单化、模型化就形成了人工神经网络。人工神经网络通过计算机构建类似生物神经网络的模型。通过构造可以模拟生物神经元的电信号传递过程的人工神经元，再将大量的人工神经元组合起来，形成一个网络结构，这个网络结构就是人工神经网络。与生物神经网络不同的是，人工神经网络之间传递的并不是电信号，而是一些具有实际意义的值或计算结果。在人工神经元中，输入为一系列的特征值，输出为一个结果。图2-2为一个最简单的人工神经元。在一个最简单的神经元中，它要接收前一层神经元的输出，记为，其中为前一层第个神经元的输出。当它接收到前一层神经元的输出后，要将乘以一个权值，将所有乘积求和再加上一个偏移量得到，即（假设前一层共有个神经元）。我们计，，那么上式可以写成。在得到了之后，这一个神经元的输出值，其中为一个激活函数。所以一个神经元的计算过程为。

本文使用的长短期记忆网络中使用的激活函数为Sigmoid函数和Tanh函数。

1）Sigmoid函数

Sigmoid函数是一个S型函数，它将变量映射到区间中，如图2-3。它的函数表达式是，

(2-1)

2）Tanh函数

Tanh函数是双曲函数中的一个，如图2-3，它的函数表达式是，

(2-2)

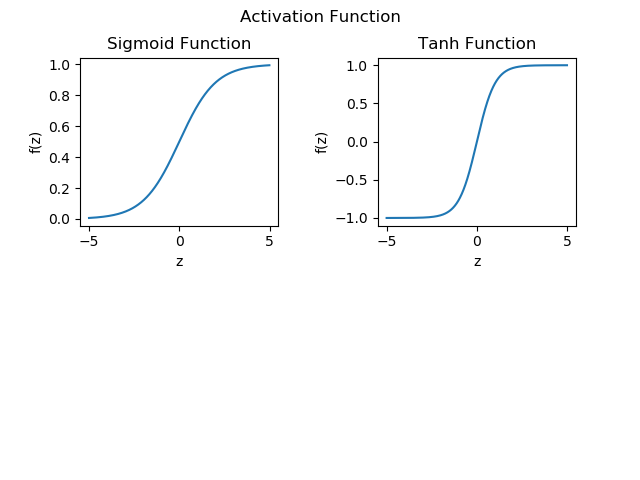


图2-3 激活函数

2.2全连接神经网络

2.2.1 前馈神经网络

前馈神经网络是典型的人工神经网络模型。生物神经网络的结构错综复杂，而前馈神经网络的结构非常简单，它有严格的分层，同一层之间的神经元没有连接。前馈神经网络中只能从低层神经元连接高层神经元，而且不能跨层连接。前馈神经网络中的每一个神经元都接收前一层的输出，经过计算处理后，输出给下一个神经元。它是很多复杂神经网络的基本组成部件。卷积神经网络、循环神经网络中都是它的变型。它分为单层前馈神经网络和多层前馈神经网络单层前馈神经网络就是只有输入层和输出层的神经网络，如图2-4所示。多层前馈神经网络中存在隐藏层，即在输入层和输出层之间的层，隐藏层可能不止一个。多层前馈神经网络的结构就是将多个单层前馈神经网络的神经元依次连接。如图2-5所示。

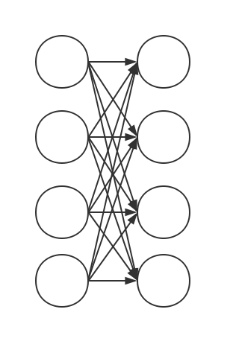
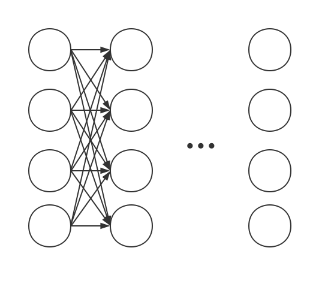
 

图2-4 单层前馈神经网络 图2-5 多层前馈神经网络

2.2.2 全连接层和全连接神经网络

全连接层是指神经网络中的这一层的所有神经元都和上一层的所有神经元相连接，即上一层的所有神经元的输出都会影响到这一层的所有的神经元的输出。

全连接神经网络是仅由全连接层组成的前馈神经网络。全连接神经网络是现阶段使用最广泛的一种神经网络，它可以是很多神经网络模型的重要组成部分，包括深度信念网络、自动编码器等。

2.2.3 损失函数

在深度学习中，损失函数是用来描述神经网络预测结果与实际结果的差距的一个函数。它有着不同的形式，损失函数越小，代表着神经网络预测结果与实际结果越接近。在本论文中，我们用来表示损失函数。在下文公式中代表神经网络预测结果的集合，代表样本真实结果的集合，表示样本集合中元素的个数。

本文实验中使用的损失函数为损失，又称为均方误差（MSE）

损失函数的数学表达式是：

(2-3)

2.2.4 全连接神经网络的训练方法

对于全连接神经网络来讲，它所有神经元中的权值在最初都是随机分配的。人工神经网络之所以称为“神经网络”是因为它有训练和学习的能力。简单地说，它的所有神经元中的权值是在学习的过程中不断变化的。

2.2.4.1 梯度

梯度（gradient）是表示函数上升（或下降）趋势的量。考虑神经网络中的权值，那么针对函数，梯度是

(2-4)

梯度有着以下性质：1）梯度的方向就是函数上升的方向，2）梯度的范数越大，函数上升地越快

2.2.4.2 梯度下降算法

梯度下降算法是一个迭代的算法。算法的目的是调整神经网络中的权重值，使这个权重值计算出的结果更贴近我们想要的结果。这个目的使用过让损失函数的计算结果不断变小实现的。开始时，算法随机指定一个权重值，当然这个权重值不是我们想要的。接下来时刻，我们对损失函数求导，得到梯度。这样我们就得到了损失函数下降的方向，通过计算就得到了新的权重值。在上式中，代表了学习率，它的大小大部分情况下决定了学习效率的大小。在我们不断重复这个过程的时候，损失函数不断变小，也就是神经网络计算出的结果越来越接近我们想要的结果。这样就达到了训练、学习的目的。

2.3 循环神经网络

循环神经网络是众多神经网络中的一种。循环神经网络擅长描述与时间有关的行为，例如自然语言处理[[5](#_ENREF_5),[6](#_ENREF_6)]。在全连接神经网络中，样本之间相互独立，但是在循环神经网络中，样本之间可以相互影响，所以循环神经网络和全连接神经网络、前馈神经网络不同。循环神经网络更加适用于处理时间序列数据。最早利用循环神经网络成功的是手写识别。

2.3.1 循环神经网络的结构

图2-6是循环神经网络的基本结构，它是一个最简单的循环神经网络，它的输入层、输出层、隐藏层各有一个。我们可以从图2-6中看出相比于全连接神经网络，循环神经网络在隐藏层中有一个指向自身的线，这就是循环神经网络的“循环”。因为在循环神经网络中，隐藏层的值不只依赖于前一层给的输入，还依赖于这层神经元在前一个时刻的值。在本文中，我们使用表示输入层的值，表示隐藏层的值，表示输出层的值。均为向量形式。表示输入层到隐藏层的权重矩阵，是隐藏层到输出层的权重矩阵，表示隐藏层在时刻的值到时刻隐藏层的权重矩阵。我们把图2-6展开，如图2-7。

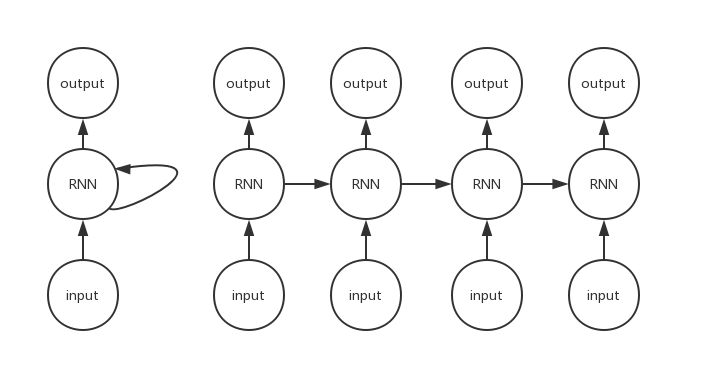


图2-6循环神经网络 图2-7循环神经网络展开

循环神经网络的计算公式：

1）隐藏层计算公式：。这一层是循环层，是激活函数。

2）输出层计算公式：。这一层是全连接层，是激活函数。

所以我们得到：

其中，是时刻隐藏层的值，是时刻输入层的值，是时刻输出层的值。

2.3.2 循环神经网络的训练

循环神经网络循环层的训练方法：BPTT（Back Propagation Trough Time）算法。首先，我们记

BPTT算法包含四个步骤：1）前向计算，2）反向计算误差项，它是损失函数对神经元中的中间计算结果的偏导数，3）计算梯度，4）更新权重值。

(2-5)

2.3.2.1 前向计算每个神经元的输出值

利用计算公式得到循环神经网络中循环层的所有。

2.3.2.2 误差项计算

首先，我们记，所以，因此，

(2-6)

公式(2-6)给出了递推公式中的一项，通过公式(2-6)就可以求得任意时刻的

(2-7)

2.3.2.3 权重梯度计算

权重矩阵在时刻的梯度为、，是所有时刻的和、是所有时刻的和，所以

(2-8)

(2-9)

2.3.2.4 循环神经网络的问题

循环神经网络在处理较长的时间序列数据时，易发生梯度爆炸或梯度消失，它的原因是，时间序列较长时它的误差项就会有比较大的增加或缩小，导致无法训练出想要的结果[[28](#_ENREF_28)]。所以循环神经网络无法处理较长的时间序列数据。

2.4 长短期记忆网络

长短期记忆网络（Long Short-Term Memory, LSTM）属于一种循环神经网络，LSTM弥补了循环神经网络的不足，它比较适合分析较长的时间序列。在大部分情况下，LSTM的处理效果都要由于循环神经网络。原因是LSTM加入了一个用于“遗忘”的“处理器”。

2.4.1 长短期记忆网络的结构

长短期记忆网络（如图2-8所示）是在原本的循环神经网络中加入了一个新的结构—遗忘门。循环神经网络的隐藏层中只有一个状态（在LSTM中记为）。对于较短的时间序列，循环神经网络可以处理，可是股票数据往往有很多，循环神经网络可能会发生梯度消失或梯度爆炸。在LSTM中新增了一个用来保存过去一段时间对现在结果的影响的量（在LSTM中记为，称为单元状态），这使得LSTM处理的时间序列长度更长。循环神经网络中的问题就迎刃而解了。

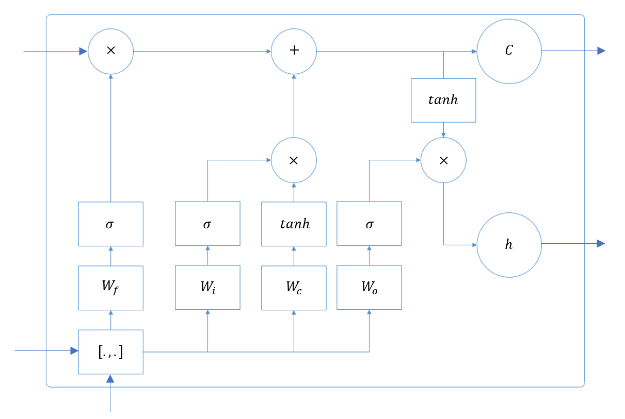


图2-8 长短期记忆网络

LSTM的输入分别是、、，输出分别是、。在LSTM神经元中有三个控制开关，如图2-8所示：1）控制是否保持上一时刻的，2）控制是否把写入中，3）控制是否把作为参考加入到输出中。这三个开关由三个全连接层实现，门可以表示为：。为激活函数，在LSTM神经元中使用了Sigmoid函数和Tanh函数。

LSTM中的三个门分别为：

1）遗忘门：

(2-10)

2）输入门：

(2-11)

3）输出门：

(2-12)

2.4.2 长短期记忆网络的训练

长短期记忆网络的训练方法与循环神经网络类似，主要有下面四个步骤：1）前向计算，2）反向计算误差项，3）计算梯度，4）更新权重值。

2.4.2.1 前向计算

对于LSTM来讲，前向计算就是计算的值

2.4.2.2 反向计算误差项

和循环神经网络不同，我们定义

(2-13)

这里我们假设误差项是损失函数对输出的导数。接下来我们记

(2-14)

(2-15)

下面我们计算,

(2-16)

在上文中可以看出，都是的自变量，利用全导数公式可得,

(2-17)

接下来，我们可以得到：

(2-18)

由此我们得到，

(2-19)

同时根据公式(2-16)，可知，

(2-20)

定义层的误差项：

(2-21)

第层的输入由公式计算，其中是第层的激活函数。

我们用全导数公式将上式化为：

(2-22)

2.4.2.3 权重矩阵梯度的计算

根据公式(2-15)，接下来很容易求出的梯度：

(2-24)

各个时刻的梯度加和，求得最终梯度，

(2-25)

偏置项梯度，

(2-26)

最后更新权重值，LSTM神经网络的训练就完成了。

2.5 神经网络训练的优化方法

2.5.1 梯度下降算法的缺陷

在上文中，我们利用来更新权值矩阵，这个算法有些明显的缺陷：1）最基本的梯度下降算法很容易求得局部最小值或者陷入驻点，2）最基本的梯度下降算法很容易在最优点附近震荡而无法求得最优点。

2.5.2 解决方法

针对第一种缺陷我们引入“动量”来解决，针对第二种缺陷我们通过调整学习率来解决。

2.5.2.1 动量

动量就是将历次梯度和最新的梯度做线性叠加。通常，动量的表达式可以是，其中，和是两个预设的常数，用来更新：

2.5.2.2 调整学习率

为了让循环开始时的权重矩阵变化尽可能大，到快接近最优点时权重矩阵变化尽可能小，我们可以在调整权重矩阵时使用不同的学习率，让学习率开始时很大，在多次循环之后逐渐减小。

2.5.2.3 Adam算法

Adam算法[[29](#_ENREF_29)]是一种随机优化的算法。在Adam算法中，实现了上文提及的两个方面，也补足了梯度下降算法的两个不足。下面是Adam算法的计算步骤：

1）初始化：初始化

2）计算：

其中，，一般为，一般为，一般为，一般为。

第三章 模型构建

本文主要的研究目的是使用LSTM神经网络模型成功预测股票价格，尽量提高预测准确率。基于这个目的，本文使用现阶段比较擅长构建神经网络的Python语言进行网络搭建以及数据分析。

3.1 使用工具

3.1.1 Python

Python语言在1991年由Guido van Rossum创立，它是一个解释性的高级编程语言。Python语言极其强调代码的可读性以及代码的优美，在Python语言中，使用强制空格的方式划分模块。这种方法代替了C语言中的花括号、Matlab中的关键词等，使得代码可读性更高，更加优美[[30](#_ENREF_30)]。

因为Python语言是一个解释型语言，所以Python在运行时需要解释器。Python的解释器可以在很多平台上运行（几乎是所有平台），这使得Python可以在更广泛的范围内传播。如今Python语言已经是世界上常用语言之一。如Linux操作系统中就默认装有Python语言模块，在Linux中也有部分功能使用Python实现。

在本文中，我们使用Python3.5.2（Ubuntu中自带的Python版本），我们使用pip安装以下几个模块：

1）Numpy

2）PyTorch

3）Pandas

4）Matplotlib

3.1.2 Numpy

Numpy是Python语言的科学研究一个库。它支持大型的、多维的向量和矩阵的运算，同时也实现了很多数学函数，我们用起来特别方便[[31](#_ENREF_31)]。

在Numpy中实现了维数组的结构，它是用“ndarray”实现的。它和Python中的list、tuple等不太相同的是：Numpy数组的数据必须是同一数据类型的，在Python数组中数据可以是不同数据类型的。

3.1.3 PyTorch

PyTorch是Python的一个库，它是一个开源的机器学习库。它最初是由Facebook的人工智能研究团队开发的。PyTorch的显著特点是可以利用GPU的快速运算来加速。

3.1.3.1 Tensor

在Python中，Tensor是一个最基本的单位，一切数据都存储在Tensor中，它类似于Numpy的ndarray。Tensor同样可以存储多维数组、矩阵、向量。Tensor的中文为“张量”，张量可以是维的。

3.1.3.2 PyTorch中的模块

1）Autograd模块

PyTorch使用自动微分技术。在这个模块中使用一个记录器来记录函数的表现，然后进行计算梯度。这个模块在神经网络的训练中特别有用，因为它可以实现神经网络中的梯度下降法。

2）Optim模块

PyTorch中的Optim模块实现了很多神经网络中的优化算法，包括上文提到的Adam算法，我们在进行训练时，只需调用Optim模块，而不需要重新构建优化算法。

3）nn模块

nn模块中实现了现如今主流的神经网络模型，同样我们也可以利用nn模块来重新搭建一个我们想要的神经网络模型，nn模块非常好用且高效。

3.1.4 Pandas

Pandas是Python的一个库，它的作用是数据处理和数据分析，它同Numpy类似，都是用来处理数据的库。不同的是，Numpy更加注重数学运算，而Pandas更擅长处理时间序列，因为本文的数据是股票市场的数据，它是一个时间序列数据，所以我们将Pandas和Numpy结合来处理股票数据。

3.1.5 Matplotlib

Matplotlib是Python的一个库，它的功能是将数据以图表的形式展示出来。它结合Numpy可以轻松地画出股票市场的曲线图。它的图表形式与Matlab类似。虽然它是Python的一个库，但是它的使用风格与Matlab相似。

3.2 基于LSTM的神经网络模型构建

在构建神经网络前，我们需要设定一个窗口，它代表我们预测股票市场时间序列所基于的数据，我们想要预测某一天的股票收盘价，我们需要知道它前天的数据，这段数据就是我们要设定的窗口，我们称它为。

在神经网络中就是我们神经网络的输入的维度，即，在PyTorch中需要将input\_size设定为这个值。接下来，我们需要确定在LSTM中的输出的长度，在PyTorch中需要将hidden\_size设定为这个值。最后，我们需要确定PyTorch中的num\_layers，它代表循环层的层数，也就是我们使用多少个LSTM神经元。因为LSTM的输出的维度是hidden\_size，所以我们需要一个全连接层来将这个输出变为维度为的值。我们使用torch.nn中的Sequential。Sequential是一个容器，我们可以将神经元加入到其中，整合成一个神经网络。这里我们在Sequential中加入一个全连接层，使得数据通过全连接层后，变成我们想要的股价预测数据。

在这个神经网络中有一个LSTM神经元和一个普通神经元，它们分别在循环层和全连接层中。

3.4 数据处理

在我们获取到股票数据时，我们需要进行标准化（normalization）处理。目的是使得数据更加的正常或者说更加的规则。我们使用的标准化方法是z-score标准化方法：

(3-1)

其中，是标准化前的数据，是标准化后的数据，是数据整体的均值，是数据整体的标准差。

在我们把数据标准化之后，我们就要考虑如何把数据输入至神经网络中了。当我们把数据输入神经网络时，并不是一条一条的输入进入神经网络的，我们需要把数据重新编排，把数据的输入特征和标签分开（标签就是我们想要的数据的输出结果）。这里我们使用Pandas工具将数据进行重新编排。我们在预测第天的股票数据时，需要第天至第天的数据作为窗口。在Pandas中，我们需要先创建一个DataFrame，接下来我们需要设置这个DataFrame的列，其中，前条数据代表窗口，第条数据代表标签。

之后，我们把这个DataFrame转化为Numpy的Array，再转换为PyTorch中的Tensor。

最后，需要用到PyTorch中的两个模块：Dataset和DataLoader。它们都在torch.utils.data中。Dataset是一个抽象类，所以我们在使用Dataset时，需要继承这个类，然后重写Dataset类中的\_\_getitem\_\_()方法和\_\_len\_\_()方法，它们分别实现了取出数组中第个元素和求数组长度的功能。DataLoader类是用来加载数据的，DataLoader类会返回一个迭代器。在DataLoader的构造方法中，需要一个必须的参数，就是Dataset类的一个对象。在我们的实验中，还使用了两个可选参数，batch\_size和shuffle，batch\_size代表一次传入多少组数据，shuffle接收一个布尔型变量，代表是否需要将数据顺序打乱。DataLoader使训练变得十分方便，在训练过程中，我们只需要使用一个循环就可以完成一次梯度下降的过程。

3.5 训练模块

在这个模块中，我们主要使用PyTorch。我们在网上下载的数据是csv格式的。我们需要调用数据处理模块将数据传入到Python进程中。然后数据处理模块会将数据改为可以进行训练的数据（将输入特征和输出标签分开）。之后，我们创建一个LSTM神经网络的实例，创建一个优化器optimizer，创建一个损失函数实例。最后，我们需要将数据加载到前面构建的LSTM神经网络中，使用以上的几个模块进行神经网络的训练。

第四章 实验分析

4.1 股票数据的选取

本文中实验的数据有两部分，第一部分是股指数据，分为上证指数的数据，和深证指数的数据，第二部分是个股股价的数据。本文用这两种数据来进行训练和预测，以及对算法的评估。

4.1.1 数据来源

两部分数据都是从RESSET金融研究数据库获取的。RESSET数据库是一个金融研究数据库，其中包含了股票、外汇、基金、债券、期货、黄金等金融领域数据。本文中主要使用RESSET数据库中的与股票、股指相关的数据。

4.1.2 上证指数

上证指数指的是上海证券交易所的成分股指数，指数代码为000001。它是通过上海证券交易所的成分股的上涨或下跌

它是用来反应上海证券交易所的成分股平均上涨或下跌情况的。因为成分股是一些具有代表性的股票（下文会提到），所以上证指数可以反应上海证券交易所大部分股票的上涨或下跌情况。

4.1.3 深证成指

深证成指值得是深圳证券交易所的成分股，指数代码为399001。它于上证指数作用相同。深证成指通过计算深圳证券交易所的成分股的平均上涨或下跌情况来反映深圳证券交易所中大部分股票的上涨或下跌情况

4.1.4 个股股价

个股股价的预测比上证指数更难做到，因为存在着多方因素的影响，比如政策、人们的心理、一些机构的操纵等等。在本文中，我们选取了一些具有代表性的股票，包含大盘股、小盘股、成分股等。

4.1.4.1 大盘股与小盘股

大盘股是指流通股份达到20亿股的公司发行的股票。而小盘股一般指流通股份较小的公司发行的股票，一般不超过3000万股。

4.1.4.2 成分股

成分股是指股票交易所在计算股票价格指数时使用的股票，它们是一些具有代表性的股票。在股票价格指数的计算过程中，因为股票个数过多，不可能每只股票都计算进去，所以选择具有代表性的成分股，股票交易所使用这些股票计算股票价格指数，从而反应整个股票的大部分股票价格上证或下跌情况。

本文在上证A股成分股列表和深证A股成分股列表中选取了一些股票数据进行预测，从而评估算法的准确性。

4.2 参数设置

在本文的所有实验中使用到的超参数如下：

表4-1 超参数设置

|  |  |
| --- | --- |
| 参数名称 | 参数设置 |
| 学习率（） |  |
| 训练次数（） | 1000 |
| 窗口（） | 5 |

4.3 整体效果

4.3.1 上证指数预测

本文选取了2009年1月至2014年12月的数据进行上证指数的预测，其中前1350天的数据进行训练使用，后面的数据进行模拟预测。

预测结果可以从图4-1中看出。在2014年11月中旬以前，预测数据比较贴合真实数据，但是在2014年11月中旬以后，预测数据走势虽与真实数据走势大致相同，但是，预测精度方面有所下滑。究其原因，可能是神经网络训练的数据是2014年8月以前的数据。所以在经历了3个多月之后，大盘环境已经有了变化，但是预测使用的神经网络没有变化，所以造成了神经网络并不能很好的拟合上证指数变化过程。

图4-1

4.3.2 深证成指预测

本文选取了2009年1月至2014年12月的数据进行深证成指的预测，其中前1350天的数据进行训练使用，后面的数据进行模拟预测。

预测结果可以从图4-2中看出，在2014年12月以前，预测的股指数据比较贴合真实的股指数据，但是，在2014年12月以后，预测的股指数据与真实的股指数据相差甚远。究其原因，可能同上证指数的预测相同，用于神经网络训练的数据是2014年8月以前的股指数据，在经历了3个月后，股票市场已经发生变化，所以预测的效果并没有2014年8月至2014年11月下旬好。

图4-2

4.3.3 个股股价预测

在这一部分中，我们选取了一些个股进行预测，以测试算法是否可以被炒股的股民真正的使用。我们从股民比较关心的几种股票类别中各选取了一些。

4.3.3.1 成分股预测

在这一部分，我们选了上证A股成分股中选取了浦发银行、武钢股份、东风汽车、中国石化。

1）浦发银行

本文选取了浦发银行2011年1月至2015年12月这段时间每天的最高价进行分析，其中2011年1月至2015年8月的股价数据进行训练，2015年8月至2015年12月的股价数据进行预测。预测结果可以从图5-3中看出。对比上证指数和深证成指，个股的股价具有更强烈的波动，所以，理论上讲，个股股价的预测比指数行情的预测更困难一点。从图中也可以看出，神经网络的预测相对保守，它的预测结果比起真实股价数据来讲显得平滑一些，股票价格基本走势与真实的股价相差不大，但是浦发银行这支股票存在暴涨暴跌的情况，神经网络并没有很好的预测出来。

图5-3

2）武钢股份

本文选取了武钢股份2011年1月至2014年12月的股价数据进行测试，其中2014年最后一季度的股价数据用于测试，2011年1月至2014年8月的股价数据用于训练。预测结果在图5-4中显示。可以看出使用神经网络可以基本预测股价走势，但神经网络预测的结果相比真正的股价数据较为平滑。

图5-4

3）东风汽车

本文使用东风汽车2011年1月至2014年12月的股价数据进行预测分析，其中2011年1月至2014年第三季度的数据用于训练神经网络，2014年第四季度的股价数据用于测试神经网络的预测效果。预测效果可以从图5-5中看出。图中显示，在东风汽车股价的预测中，神经网络预测的股价数据与真实股价数据相差略大，这可能是东风汽车这支股票的波动性太大造成的，可能当时公司受到了一些因素的影响，使得股价波动变大。但神经网络预测的股价数据与真实的股价数据走势相吻合。

图5-5

4）中国石化

本文选取中国石化的2011年1月至2014年12月的股价数据进行分析，其中2011年1月至2014年8月的股价数据用来训练使用，2014年9月至2014年12月的股价数据用于测试神经网络的预测效果。预测效果由图5-6显示。我们看出在2014年12月以前神经网络的预测与真实数据较为贴合，但是在2014年12月时，中国石化这支股票存在这强烈的波动，神经网络没有预测出来。神经网络比较适合于预测波动性小的股票数据。

图5-6

4.3.3.2 大盘股预测

本文选择了一些大盘股进行实验分析，以测试神经网络对大盘股的预测效果如何。本文选取的大盘股有：中国平安、中信证券。

1）中国平安

本文选择了中国平安2011年1月至2014年12月的股票单日最高价进行对算法的分析评估，其中2011年1月至2014年7月的数据用来训练，2014年8月至2014年12月的数据用来测试。测试效果如图5-7所示，中国平安这支股票在2014年8月至2014年11月中旬属于平稳期，股票价格波动很小，在2014年11月下旬时进入波动期，这段时间股票上涨和下跌的幅度都很大。对比神经网络的预测结果，在平稳期时神经网络可以预测到它的股价，在波动期时，神经网络预测的上涨幅度没有实际的上涨幅度大，而且没有预测到股票下跌的情况。

图5-7

2）中信证券

本文选取了中信证券2011年1月至2014年12月的股票单日最高价进行对神经网络的实验与分析，其中2011年1月至2014年7月的股价数据用来训练，2014年8月至2014年12月的股价数据用来测试。测试结果在图5-8中显示。从实际股价数据中可以看出，中信证券在2014年8月至2014年11月中旬的股价较为平稳，在2014年11月下旬至2014年12月上涨和下跌幅度都较大，波动性较大。从预测的股价数据中可以看出，神经网络可基本预测股票价格走势，但是遇到突然下跌的情况，预测效果不好。

图5-8

4.3.3.3 小盘股预测

本文选择了一些小盘股进行实验分析，以测试神经网络对小盘股的预测效果如何。本文选取的小盘股有：上工申贝、南京化纤。

1）上工申贝

上工申贝属于专用设备制造业，主要业务是研发、生产、维修缝纫设备及零部件。我们选取上工申贝2011年1月至2014年12月的股票数据进行分析和研究，其中2011年1月至2014年7月的股票单日最高价格用来训练，2014年8月至2014年12月的单日最高价格用来测试神经网络的预测效果。预测效果在图5-9中显示。从图中可以看出，上工申贝这支股票的上涨和下跌的幅度都很大，而且神经网络可以基本预测它的走势。

图5-9

2）南京化纤

南京化纤属于化学纤维制造业，主要业务是化学纤维制品。在本文中我们选择南京化纤2011年1月至2014年12月的股票数据对算法的效果进行检验。其中，我们用2011年1月至2014年7月的股票数据训练，2014年8月至2014年12的股票数据用来测试。图5-10显示了神经网络对这支股票的预测效果。从图中看出，南京化纤的波动略大，神经网络的预测结果基本上贴合南京化纤的真实股票数据走势。

图5-10

4.4 评估指标

4.4.1 均方误差

均方误差（Mean Square Error, MSE）同上文的损失，它的数学表达式是：

(5-1)

均方误差评价算法的重要指标，它描述了预测数据与真实数据的差距。

4.4.2 泰尔不平等系数

泰尔不平等系数（Theil`s inequality coefficient, Theil U）的数学表达式是：

(5-2)

4.4.3 平均偏差

平均偏差（Mean Absolute Deviance, MAD）同上文的损失，它的数学表达式是：

(5-3)

4.4.4 数据准确性

数据准确性（Data Accuracy, DA）是描述预测算法的正确率的。因为股票价格通常只有上涨和下跌两种情况（本文不考虑股票价格不变的情况），所以数据准确性就是用来描述算法预测的成功率。它的数学表达式是：

(5-4)

其中，

(5-5)

4.4.5 平均绝对百分误差

平均绝对百分误差（Mean Absolute Percentage Error, MAPE）的数学表达式是：

(5-6)

4.4.6 相关系数

相关系数（Correlation Coefficient, R）的数学表达式是：

(5-7)

4.5 综合评价

在上文中，我们使用不同的股票市场时间序列数据对算法在不同方向的表现情况进行了对比。而且，我们还提及了现有的对比算法性能的指标。所以在这一部分中，我们定量地分析神经网络算法的性能。

表5-2 均方误差

|  |  |
| --- | --- |
| 股票或股指名称 | 均方误差 |
| 上证指数 |  |
| 深证成指 |  |
| 浦发银行 |  |
| 武钢股份 |  |
| 东风汽车 |  |
| 中国石化 |  |
| 中国平安 |  |
| 中信证券 |  |
| 上工申贝 |  |
| 南京化纤 |  |

表5-3 泰尔不平等系数

|  |  |
| --- | --- |
| 股票或股指名称 | 泰尔不平等系数 |
| 上证指数 |  |
| 深证成指 |  |
| 浦发银行 |  |
| 武钢股份 |  |
| 东风汽车 |  |
| 中国石化 |  |
| 中国平安 |  |
| 中信证券 |  |
| 上工申贝 |  |
| 南京化纤 |  |

表5-4 平均偏差

|  |  |
| --- | --- |
| 股票或股指名称 | 平均偏差 |
| 上证指数 |  |
| 深证成指 |  |
| 浦发银行 |  |
| 武钢股份 |  |
| 东风汽车 |  |
| 中国石化 |  |
| 中国平安 |  |
| 中信证券 |  |
| 上工申贝 |  |
| 南京化纤 |  |

表5-5 数据准确性

|  |  |
| --- | --- |
| 股票或股指名称 | 数据准确性 |
| 上证指数 |  |
| 深证成指 |  |
| 浦发银行 |  |
| 武钢股份 |  |
| 东风汽车 |  |
| 中国石化 |  |
| 中国平安 |  |
| 中信证券 |  |
| 上工申贝 |  |
| 南京化纤 |  |

表5-6 平均绝对百分误差

|  |  |
| --- | --- |
| 股票或股指名称 | 平均绝对百分误差 |
| 上证指数 |  |
| 深证成指 |  |
| 浦发银行 |  |
| 武钢股份 |  |
| 东风汽车 |  |
| 中国石化 |  |
| 中国平安 |  |
| 中信证券 |  |
| 上工申贝 |  |
| 南京化纤 |  |

表5-7 相关系数

|  |  |
| --- | --- |
| 股票或股指名称 | 相关系数 |
| 上证指数 |  |
| 深证成指 |  |
| 浦发银行 |  |
| 武钢股份 |  |
| 东风汽车 |  |
| 中国石化 |  |
| 中国平安 |  |
| 中信证券 |  |
| 上工申贝 |  |
| 南京化纤 |  |

同时我们使用了上证指数的时间序列数据对窗口对算法性能的影响做了实验，实验结果如图5-11所示。从图中可以看出，

4.6 投资方案

第五章 结 论

致 谢

参考文献

[1] Russell S J, Norvig P. Artificial intelligence: a modern approach[M]. Malaysia; Pearson Education Limited, 2016.

[2] Bishop C M. Pattern recognition and machine learning[M]. springer, 2006.

[3] Lecun Y, Bengio Y, Hinton G J N. Deep learning[J], 2015, 521(7553): 436.

[4] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems, 2012: 1097-1105.

[5] Graves A, Mohamed A-R, Hinton G. Speech recognition with deep recurrent neural networks[C]. 2013 IEEE international conference on acoustics, speech and signal processing, 2013: 6645-6649.

[6] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]. Eleventh annual conference of the international speech communication association, 2010.

[7] Ramsey J B. The contribution of wavelets to the analysis of economic and financial data[J]. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 1999, 357(1760): 2593-2606.

[8] Arel I, Rose D, Coop R. Destin: A scalable deep learning architecture with application to high-dimensional robust pattern recognition[C]. 2009 AAAI Fall Symposium Series, 2009.

[9] Amihud Y J J O F M. Illiquidity and stock returns: cross-section and time-series effects[J], 2002, 5(1): 31-56.

[10] Song Q, Chissom B S J F S, Systems. Fuzzy time series and its models[J], 1993, 54(3): 269-277.

[11] Huarng K, Yu H-K J P a S M, Applications I. A type 2 fuzzy time series model for stock index forecasting[J], 2005, 353: 445-462.

[12] Lahmiri S. Long memory in international financial markets trends and short movements during 2008 financial crisis based on variational mode decomposition and detrended fluctuation analysis[J]. Physica A: Statistical Mechanics and its Applications, 2015, 437: 130-138.

[13] White H. Economic prediction using neural networks: The case of IBM daily stock returns[J], 1988.

[14] Kamijo K, Tanigawa T. Stock price pattern recognition-a recurrent neural network approach[C]. 1990 IJCNN International Joint Conference on Neural Networks: 215-221 vol.1-221 vol.1.

[15] Hsieh T-J, Hsiao H-F, Yeh W-C. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm[J]. Applied soft computing, 2011, 11(2): 2510-2525.

[16] Takeuchi L, Lee Y-Y A: Applying deep learning to enhance momentum trading strategies in stocks, Technical Report: Stanford University, 2013.

[17] Kuremoto T, Kimura S, Kobayashi K, et al. Time series forecasting using a deep belief network with restricted Boltzmann machines[J]. Neurocomputing, 2014, 137: 47-56.

[18] Di Persio L, Honchar O. Artificial neural networks architectures for stock price prediction: comparisons and applications[J]. International Journal of Circuits, Systems and Signal Processing, 2016, 10: 403-413.

[19] Dixon M, Klabjan D, Bang J H. Classification-based financial markets prediction using deep neural networks[J]. Algorithmic Finance, 2016, (Preprint): 1-11.

[20] M’ng J C P, Mehralizadeh M. Forecasting East Asian Indices Futures via a Novel Hybrid of Wavelet-PCA Denoising and Artificial Neural Network Models[J]. PloS one, 2016, 11(6): e0156338-e0156338.

[21] Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory[J]. PloS one, 2017, 12(7): e0180944-e0180944.

[22] Chen H, Xiao K, Sun J, et al. A double-layer neural network framework for high-frequency forecasting[J]. ACM Transactions on Management Information Systems (TMIS), 2017, 7(4): 11-11.

[23] Chen Y, He K, Tso G K F. Forecasting Crude Oil Prices: a Deep Learning based Model[J]. Procedia computer science, 2017, 122: 300-307.

[24] Yan H, Ouyang H. Financial Time Series Prediction Based on Deep Learning[J]. Wireless Personal Communications, 2018, 102(2): 683-700.

[25] Pang X, Zhou Y, Wang P, et al. An innovative neural network approach for stock market prediction[J]. The Journal of Supercomputing, 2018: 1-21.

[26] 百度图片-发现多彩世界官网[EB/OL]. <https://image.baidu.com/search/detail?ct=503316480&z=0&ipn=d&word=%E7%94%9F%E7%89%A9%E7%A5%9E%E7%BB%8F%E5%85%83&step_word=&hs=0&pn=39&spn=0&di=152765082690&pi=0&rn=1&tn=baiduimagedetail&is=0%2C0&istype=2&ie=utf-8&oe=utf-8&in=&cl=2&lm=-1&st=-1&cs=1743467864%2C2752520972&os=2992126829%2C4036637801&simid=0%2C0&adpicid=0&lpn=0&ln=1671&fr=&fmq=1557822314372_R&fm=index&ic=0&s=undefined&hd=undefined&latest=undefined&copyright=undefined&se=&sme=&tab=0&width=&height=&face=undefined&ist=&jit=&cg=&bdtype=0&oriquery=&objurl=http%3A%2F%2Fimages2015.cnblogs.com%2Fblog%2F1180580%2F201706%2F1180580-20170611102432700-79279755.png&fromurl=ippr_z2C%24qAzdH3FAzdH3Fooo_z%26e3Bk7k7h5_z%26e3Bv54AzdH3Ftgu51jpwts-d889dl9_z%26e3Bip4s&gsm=0&rpstart=0&rpnum=0&islist=&querylist=&force=undefined>.

[27] 计算思维百科[EB/OL]. <https://wiki.jsswsq.com/index.php?title=Template:%E4%BA%BA%E5%B7%A5%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C%E7%9A%84%E5%9F%BA%E6%9C%AC%E7%89%B9%E6%80%A7>.

[28] Goodfellow I, Bengio Y, Courville A. Deep learning[M]. MIT press, 2016.

[29] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[30] Wikipedia - The Free Encyclopedia[EB/OL]. <https://en.wikipedia.org/wiki/Python_(programming_language>).

[31] Wikipedia - The Free Encyclopedia[EB/OL]. <https://en.wikipedia.org/wiki/NumPy>.