

学校代码:10730

分类号:O213

密级:公开

兰州大学

硕士学位论文

(专业学位)

论文题目(中文)

基于贝叶斯优化的 CNN-LSTM

股价预测模型

论文题目(外文)

CNN-LSTM Stock Price Prediction Model

Based on Bayesian Optimization

作者姓名

陈晓媛

类型领域

应用统计

研究方向

深度学习

教育类型

学历教育

指导教师

焦桂梅 副教授

合作导师

论文工作时段

2022 年 3 月至 2023 年 4 月

论文答辩日期

2023 年 5 月

校址:甘肃省兰州市城关区天水南路 222 号

# 基于贝叶斯优化的 CNN-LSTM 股价预测模型

## 中文摘要

随着金融市场的发展,人们越来越关注股票市场的走势.股价预测不仅可以给投资者带来巨大的利润,而且也可以作为一种有效的风险管理手段.伴随着人工智能时代的来临,以机器学习、深度学习等为代表的新技术在各行业中得到了迅速的发展,极大地提高了业务效率,改善了用户体验.机器学习和深度学习技术能够通过分析海量数据,快速有效地发现股票波动规律,从而实现预测市场趋势、提供个性化服务等功能,为企业和个人带来了显著的经济效益.

本文将深度学习技术应用到股票价格预测中,构建了新的 BO-CNN-LSTM 组合模型.首先,在数据处理方面,选用基础指标、技术指标构成股票特征体系,并将经主成分分析(PCA)消除各指标间冗余信息后的数据作为初始数据.其次,通过融合 CNN 与 LSTM 两个模型的优点搭建了股价预测模型,同时针对常见的过拟合现象以及人工试错效率低的问题,采用贝叶斯优化(BO)算法优选模型超参数,并探讨了当前使用较多的遗传算法(GA)以及粒子群优化(PSO)算法的预测性能.最后,利用构建好的 BO-CNN-LSTM 模型预测具有代表性的沪深 300 指数收盘价.结果表明,相比于基准模型,BO-CNN-LSTM 模型具有更好的预测性能.同时,与 GA、PSO 算法相比,BO 算法在神经网络超参数的搜索上效率更高,模型训练时间更短,且模型的预测效果相近,表明本文构建的 BO-CNN-LSTM模型在股价预测的精度与耗时上具有一定的优势.

**关键词:** 股价预测, 贝叶斯优化算法, 神经网络模型, 沪深 300 指数

# **CNN-LSTM STOCK PRICE PREDICTION MODEL BASED ON BAYESIAN OPTIMIZATION**

## **Abstract**

With the development of the financial market, people are increasingly paying attention to the trends of the stock market. Stock price forecasting can not only bring huge profits to investors, but also serve as an effective risk management tool. With the advent of the artificial intelligence era, new technologies such as machine learning and deep learning have rapidly developed in various industries, greatly improving business efficiency and user experience. Machine learning and deep learning technologies can quickly and effectively discover stock volatility patterns by analyzing massive amounts of data, thereby achieving functions such as predicting market trends and providing personalized services, bringing significant economic benefits to enterprises and individuals.

This article applies deep learning technology to stock price prediction and constructs a new BO-CNN-LSTM combination model. Firstly, in terms of data processing, basic and technical indicators are selected to form the stock feature system, and the data after eliminating redundant information between each indicator through principal component analysis (PCA) is used as the initial data. Secondly, the stock price prediction model is built by combining the advantages of CNN and LSTM models. At the same time, in view of the common overfitting phenomenon and the low efficiency of manual trial and error, Bayesian optimization (BO) algorithm is used to optimize the model hyperparameter, and the prediction performance of genetic algorithm (GA) and particle swarm optimization (PSO) algorithm, which are widely used at present, is discussed. Finally, use the constructed BO-CNN-LSTM model to predict the representative closing price of the Shanghai and Shenzhen 300 Index. The results indicate that the BO-CNN-LSTM model has better predictive performance compared to the benchmark model. At the same time, compared with GA and PSO algorithms, the BO algorithm is more efficient in the search of neural network

hyperparameter, the training time of the model is shorter, and the prediction effect of the model is similar, which indicates that the BO-CNN-LSTM model constructed in this paper has certain advantages in the accuracy and time consumption of stock price prediction.

**Key words:** Stock price forecast, Bayesian optimization algorithm, Neural network model, CSI 300 index

# 目 录

第一章 绪论.....	1
1.1 研究背景.....	1
1.2 研究意义.....	2
1.3 国内外研究现状.....	2
1.4 研究内容及创新点.....	5
1.4.1 研究内容.....	5
1.4.2 创新点.....	6
1.5 论文组织框架.....	6
第二章 相关理论介绍 .....	8
2.1 深度学习算法.....	8
2.1.1 卷积神经网络.....	8
2.1.2 循环神经网络.....	10
2.1.3 长短期记忆网络.....	11
2.2 参数优化算法.....	13
2.2.1 贝叶斯优化算法.....	13
2.2.2 遗传算法.....	15
2.2.3 粒子群算法.....	17
2.3 主成分分析.....	19
2.3.1 主成分的确定.....	19
2.3.2 主成分的选取.....	20
第三章 数据来源与数据预处理 .....	22
3.1 股票指标选择.....	22
3.2 实验环境及数据介绍.....	25
3.2.1 实验环境.....	25
3.2.2 实验数据.....	25
3.3 数据预处理.....	25
3.3.1 缺失值处理.....	25
3.3.2 归一化处理.....	26
3.3.3 相关系数的计算.....	26
3.3.4 PCA 降维 .....	27
第四章 基于 BO-CNN-LSTM 模型的构建.....	29
4.1 BO-CNN-LSTM 模型网络结构 .....	29
4.2 超参数优化.....	31
4.2.1 优化目标函数及优化算法.....	31

4.2.2 超参数范围域及结果.....	32
4.3 模型效果评估指标.....	33
第五章 实证分析 .....	34
5.1 对比一：各模型预测效果对比 .....	34
5.2 对比二：不同参数优化算法的预测性能对比 .....	35
5.2.1 实验参数设置.....	36
5.2.2 实验结果分析.....	37
第六章 总结与展望 .....	38
6.1 本文总结.....	38
6.2 研究展望.....	39
参考文献.....	40

# 第一章 绪论

本章将介绍股票价格预测的研究背景、研究意义以及国内外研究现状,并提出本文所研究的内容及创新点,最后附上了组织框架.

## 1.1 研究背景

金融业在国民经济中所占比重逐年增加,是一国的关键核心竞争力.证券市场是我国金融行业对实体经济的支撑,是我国未来发展的必然选择.在中国证券市场的发展进程中,其制度体系也在逐步完善.随着国家经济的高速发展,政府的支持以及普通民众对证券投资的认识也在不断提高,越来越多的企业和个人开始加入到证券市场的买卖活动中,与此有关的金融服务也随之产生.股价波动趋势的准确预测已经被看作是经济领域中一个很重要的问题.而股价的波动与多种内外部的因素有关,比如行业前景、国际形势、公司财务数据、国内外经济环境、股市的运作等,因此对股价进行预测的方法也各不相同.

传统的分析方法是以前经济学和金融学为基础,主要采用的是基本面分析法与技术分析法<sup>[1]</sup>.基本面分析法更注重股票的内在价值,定性分析影响股票的外部因素,如利率、汇率、通货膨胀、上市公司的融资、国际关系等经济和政治因素,目的是通过长期地分析股票的内在投资价值,以便于投资者能够合理地购买股票.技术分析法主要是关注股票价格的走向、交易量和投资者的心理预期等因素<sup>[2]</sup>,该方法通过使用 K 线图和其他工具来分析个别股票或整个市场的股指走势,从而发现股票价格波动的普遍规律,为投资者能在股票市场中做出正确决策提供参考.当前,很多企业和个人投资者使用的仍是以上这两种方法.传统基本面分析法的准确性并不令人信服,这不仅是因为政治、经济等外部影响因素处于长期的周期性当中,而且分析员的专业能力也具有差异,这种方法在当今日益庞大和复杂的股票市场中并不适用,因为股票数据作为一个金融时间序列,具有随机漫步的特征<sup>[3]</sup>.基于统计学和概率论的思想,一些学者开始利用时间序列模型来预测股票价格,如广义自回归条件异方差模型 GARCH<sup>[4]</sup>、自回归积分滑动平均模型 ARIMA<sup>[5]</sup>等.然而,鉴于金融时间序列的不确定性和高噪音,自变量与因变量之间存在的某种关系,会随着时间的推移而发生动态变化,这也很大程度上限制了时间序列模型的进一步应用和扩展.

由于高频交易数据的增加和人工智能的普及, 股票价格预测领域的技术分析得到了充分地发展, 研究人员试图将深度学习方法应用到股价预测研究中. 比如利用卷积神经网络进行特征提取, 利用递归神经网络处理时间序列, 以及利用数据挖掘进行股票数据分析等, 这些技术都为股价预测带来了良好的效果. 尽管会有很多的因素对股价波动产生影响, 使得准确地预测变得非常困难, 但机器学习、深度学习等方法在如今的股市运用中仍然具有重要的作用.

## 1.2 研究意义

将机器学习和深度学习方法运用到股市分析中, 不仅具有一定的理论价值, 也具有一定的实践价值. 从理论价值上讲, 中国的量化投资技术<sup>[6]</sup>相对于西方发达国家来说还比较落后, 投资观念、方法与决策等还不够成熟, 尚处在起步阶段, 能够将量化投资技术运用到投资决策中的公司寥寥无几. 目前, 国内对量化投资的研究虽然已经取得了一定的进展, 但是相关的经验和成果还比较欠缺. 而如今在大数据的背景下, 伴随着人工智能的兴起, 深度学习毫无疑问成为了一个重要的研究方向, 其非参数和非线性的特性, 能够对复杂、动态、不确定的金融市场进行有效的拟合. 而实践价值主要体现在, 在金融市场中投资者利用较为成熟的方法对股票价格进行预测, 能够及时、高效地获取收益, 并规避投资风险. 同时, 将机器学习、深度学习等方法引入到投资决策中, 可以降低对股市走势的人为影响, 提高投资决策的准确性.

## 1.3 国内外研究现状

因为金融数据具有高噪声、非线性、高复杂度等特性<sup>[7]</sup>, 所以在对股价进行预测时, 会受到许多因素的影响. 随着机器学习和深度学习技术的出现, 更多的学者试图将各种机器学习和深度学习算法模型用于股价预测, 并在参数优化、模型融合等方面进行拓展研究. 结果表明, 基于机器学习和深度学习方法改进的模型能够对传统的统计模型进行有效的补充, 这种方法不但能够对大量的输入数据进行有效地处理, 同时还能很好地刻画股市中的非线性关系, 具有较高的计算效率和良好的预测效果.

最早利用人工神经网络(Artificial Neural Networks, ANN)来对股票价格进行预测的研究, 可以追溯到上个世纪 80 年代. White(1988)使用 BP 神经网络模型, 对 IBM 普通股日收益率进行了预测, 这为金融资产的趋势预测提供了一个重要的思路. 但是, 在实验模型的训练过程中存在梯度爆炸的情况, 以至于模型陷入



了局部极小值,模型最后的预测效果不佳<sup>[8]</sup>。之后国内外学者对神经网络在股票价格预测方面的应用做了许多探索性的研究。德国学者 Gencay(1996)使用前馈神经网络(Feedforward Neural Network, FNN)对 1963-1988 年道琼斯工业平均指数数据进行实证分析。结果表明, FNN 预测能力要显著优于统计建模方法<sup>[9]</sup>。吴微等(2001)使用 BP 神经网络来预测上证综合指数的涨跌情况,证实了 BP 神经网络在预测中国股市方面是有效的<sup>[10]</sup>。Zhang(2003)在研究中指出,神经网络模型在股票价格趋势上的预测效果优于传统的 ARIMA 模型<sup>[11]</sup>。孙延风等(2005)提出了一种将最优分割算法(OPA)与径向基函数(RBF)神经网络相结合的股票价格预测方法,并通过了数值模拟验证了该方法的有效性<sup>[12]</sup>。王波等(2005)选用 ANNs 和 ARIMA 模型分别对上证 A 股复华实业和标准普尔 500 指数收盘价进行了预测,结果表明, ANNs 对股票价格走势的预测准确度比 ARIMA 模型要高<sup>[13]</sup>。

Hochreiter 和 Schmidhuber(1997)提出的长短期记忆(Long Short-Term Memory, LSTM)网络是一种具有良好预测性能的深度学习算法<sup>[14]</sup>。LSTM 神经网络是循环神经网络<sup>[15]</sup>(Recurrent Neural Network, RNN)的一种特殊结构类型,它在传统 RNN 的基础上,还增加了三个门控单元,分别是遗忘门、输入门和输出门,门控单元会使符合记忆需求的信息被留下,反之不符合记忆需求的信息则会被遗忘,这能够有效处理时间序列数据的长时间依赖性。LSTM 不仅可以提高网络的存储能力,把过去的信息保留到下一个时间点,它还具备收敛速度快、非线性预测能力强等优势,能够很好地解决诸如梯度消失和梯度爆炸等问题。近些年来,围绕着 LSTM 神经网络在金融时间序列预测方面的研究日渐增多。Nelson 等(2017)整理了股票价格和技术指标等数据,并利用 LSTM 模型进行建模以预测股票价格的未来趋势。实验证明,相比于传统的机器学习方法如随机森林、多层感知机等, LSTM 模型有着更高的预测准确率<sup>[16]</sup>。邓凤欣等(2018)为深入探讨 LSTM 模型处理股票数据的可行性,选择友邦保险、微软、长和以及亚马逊四家上市公司作为研究对象,利用四家公司在一年内的所有交易日数据预测其收盘价,证明了 LSTM 模型在个股的价格趋势预测中有较高的精度和较稳定的预测效果<sup>[17]</sup>。在股市综合指数的预测中, Fischer 和 Krauss(2018)用 LSTM 模型与随机森林 RF、深度神经网络 DNN 和逻辑回归分类器 LOG 方法进行了对比,发现 LSTM 模型在短期内有更好的预测效果。因此他们根据这个结果设计了一种短期的投资策略,以便更好地控制股票投资的风险与收益<sup>[18]</sup>。彭燕等(2019)利用 LSTM 方法对苹果股市的收盘价进行了预测,并对 LSTM 的层数以及隐藏层神经元数目进行了适当的选择,从而使该模型的预测准确率提高了约 30%<sup>[19]</sup>。Lanbouri 等(2020)从 Kaggle 获取标准普尔 500 指数交易数据,然后基于价格历史数据和技术指标建立 LSTM 模型,分别预测股票未来一分钟、五分钟、十分钟后的价格,实验结果表

明, LSTM 方法是有效的<sup>[20]</sup>.

卷积神经网络<sup>[21]</sup>(Convolutional Neural Network, CNN)自上个世纪 80 年代开始就被广泛应用于图像识别领域. 近些年来, 大数据时代的来临使得可训练的数据越来越多, CNN 神经网络也开始被广泛地运用于股票价格预测. 当 CNN 进行图像识别与分类时, 整个图像会用到  $1 \times 1$ 、 $3 \times 3$ 、 $5 \times 5$  等大小的滤波器进行扫描. CNN 由不同的层组成, 具体包括了输入层、卷积层、池化层、全连接层和输出层等. 如今金融时间序列数据能以图像数据的方式进行股价预测, CNN 可以自动地学习其中的指标变量, 并将有效信息提取出来. Persio 和 Honchar(2016)利用多层感知机和卷积神经网络对标准普尔 500 指数涨跌趋势情况进行了预测, 重点关注使用特定学习算法时输入特征的选择及预处理过程, 结果表明 CNN 神经网络能够预测金融时间序列的运动<sup>[22]</sup>. Gudelek 等(2017)利用二维卷积神经网络 Conv2D 进行股票趋势分析, 把时间和技术指标分别作为第一维度和第二维度, 形成了一个可以用来模拟图像的矩阵, 同时对 2000-2014 年间 17 只不同的 ETF 进行了实证检验, 结果表明所提出的 Conv2D 模型具有较高的预测精度, 并且当把 Conv2D 作为交易模型的一部分时, 显著优于买入持有策略<sup>[23]</sup>. 陈祥一(2018)通过神经网络库 Keras 搭建了卷积神经网络, 对沪深 300 指数的涨跌趋势进行了分析, 并将其与 Logistic 回归、决策树、支持向量机等常用的监督性方法进行了比较, 结果表明, CNN 模型在预测性能上要优于传统的机器学习模型<sup>[24]</sup>. Tang 等(2020)表明使用 Conv1D 模型处理时间序列分类任务时, 卷积核大小的选择对于确保模型能够从长时间序列中学习到有效特征非常重要, 并提出一种新的网络架构 OS-CNN, 以在模型学习期间捕捉适当的卷积核大小<sup>[25]</sup>. Chen 等(2021)同时考虑了股票市场信息和个股信息, 提出了一种利用图的卷积特性进行股价趋势预测的卷积神经网络 GC-CNN, 用六只随机选择的中国股票来证明所提出的基于 GC-CNN 模型的优越性能, 实验结果表明, 所提出的方法优于其他几种股票趋势预测和股票交易策略<sup>[26]</sup>.

尽管基于 RNN 的模型和基于 CNN 的模型取得了较好的预测效果, 但是由于实际问题中影响股票价格的因素十分复杂, 仅仅凭借单一的预测模型, 难以对真实的股票价格进行精确的拟合. 因此, 为了提高模型的性能使得预测结果更加准确, 越来越多学者开始使用混合模型. 混合模型是一种能够将各个模型的优点进行有效融合的技术, 可以有效地学习到时间序列的多尺度、复杂动态等特征, 帮助人们更加准确地预测金融时间序列的走势, 从而更好地控制风险和提供决策支持. Rather 和 Agarwal(2015)提出了一种新的 ARMA 与 RNN 混合建模方法. 在实验中, 将 6 支股票作为实验对象预测股票收益, 结果显示, 与单一的 RNN 相比, 混合模型的预测误差有所降低<sup>[27]</sup>. Liu 等(2017)利用 CNN 制定用于判断股票

趋势的量化选股策略,利用 LSTM 制定用于提高利润的量化时序策略,于是搭建了 CNN-LSTM 模型并构建出一个股票交易策略,从而得到了更好的收益.通过试验发现,CNN-LSTM 模型能较好地应用于量化选股策略<sup>[28]</sup>.罗向龙等(2018)提出了一种将 LSTM 模型与 K-最近邻相结合的方法,它将道路网络中的交通流的流量序列与时空关联性的时间依赖关系进行了全面地分析,并对其进行了改进,使该方法平均预测准确率与现有的交通预测模型相比提高了 12.28%<sup>[29]</sup>.Kim 等(2019)结合 CNN 和 LSTM 对标普 500 指数 ETF 的时间序列和 K 线蜡烛图提取特征并预测股价,通过实证发现,结合同一数据的图像特征和时间特征可以有效降低预测误差<sup>[30]</sup>.刘恒等(2019)利用贝叶斯正则化算法,对常规 BP 神经网络模型进行了改进,从而克服了 BP 神经网络的初始权值由于获得的随机性而极易陷入到局部最优的难题,通过对试验数据进行分析,得出了改进后的 BP 方法的预测精度比传统的 BP 方法高 42.81%<sup>[31]</sup>.乔若羽(2019)使用注意力机制分别和 RNN、LSTM、GRU 模型进行了结合,通过试验发现了注意力机制的加入,不仅能够降低过拟合,而且能够提升学习效率,同时还能增强预测的精确性和稳定性<sup>[32]</sup>.包振山等(2020)采用 LSTM 模型研究中证 500 指数日内分钟收盘价数据,然后用遗传算法 GA 调整参数,结果表明改进后的 LSTM-GA 模型在实验中的表现效果均优于单独的 LSTM 模型<sup>[33]</sup>.胡聿文(2021)采用 PCA 与 LASSO 方法降维筛选平安银行的 5 类技术指标,并利用 LSTM 模型对其收盘价进行了预测,实验结果表明相比于 LASSO-LSTM 模型和 LSTM 模型,PCA-LSTM 模型可以更好地降低数据冗余,同时预测精度也更高<sup>[34]</sup>.

## 1.4 研究内容及创新点

### 1.4.1 研究内容

本文旨在通过对股价及其影响因素的分析,构建一种基于股价相关指标的股票价格预测模型,从而对股价的未来趋势做出较为精确的预测.其中,由于 CNN 具备较强的特征提取效果,LSTM 在缓解梯度消失的基础上,具备较强的长期时序特征提取效果.因此本文将 CNN 和 LSTM 模型的优点相融合,构建了 CNN-LSTM 模型.同时为了进一步优化模型超参数,针对常见的过拟合现象及人工试错效率低的问题,采用贝叶斯优化<sup>[35]</sup>(Bayesian Optimization, BO)算法对 CNN-LSTM 模型超参数进行优化,从而让模型更好地挖掘出时间序列数据中蕴含的潜在信息,进一步提高股票价格预测的准确性.因此本文提出了 BO-CNN-LSTM 模型.

本文研究思路如下: 首先, 为了提高特征的有效性, 选取股票多类别指标体系, 然后利用主成分分析<sup>[36]</sup>(Principal Component Analysis, PCA)消除各指标间的相关性, 将降维后的数据作为模型输入的初始特征; 其次, 为了提高模型的特征提取能力以及预测精度, 本文提出了 CNN-LSTM 股票价格预测模型, 充分发挥 CNN 模型综合特征提取以及 LSTM 模型时序特征提取能力的优势, 解决了单一模型对噪声数据处理能力差的问题, 大致方法为采用一维卷积神经网络(Conv1D)对数据进行初始特征提取, 然后通过一个 Flatten 层将经 CNN 层提取的特征进行维度转换输入到 LSTM 层中, 进行时间序列特征提取, 最后将学习到的信息传给全连接层. 与此同时, 本文采用 BO 算法, 对 CNN-LSTM 模型中的 CNN 层、LSTM 层和全连接层中的神经元个数等网络超参数进行搜索, 从而获得最优的 CNN-LSTM 模型. 与此同时, 探讨了当前使用较多的遗传算法<sup>[37]</sup>(Genetic Algorithm, GA)以及粒子群优化<sup>[38]</sup>(Particle Swarm Optimization, PSO)算法的预测性能. 最后, 为了验证 BO-CNN-LSTM 模型的有效性, 使用具有代表性的沪深 300 指数数据集作为实验对象来进行两个对比实验, 实验一: 将 BO-CNN-LSTM 模型与基准模型 CNN 模型、LSTM 模型、未经参数优化的 CNN-LSTM 模型作对比; 实验二: 将 BO-CNN-LSTM 模型与 GA-CNN-LSTM 模型、PSO-CNN-LSTM 模型作对比. 实验结果表明, 本文构建的 BO-CNN-LSTM 模型在预测精度与耗时上具有一定的优势.

#### 1.4.2 创新点

本文的主要创新点如下:

(1) 与其他的研究实验相比, 本文除了选用在大部分股价预测研究中都会用到的四价一量等基础指标以外, 还加入了乖离率、动量指标、心理线等较为前沿的技术指标, 将这些股价指标综合在一起, 能够从多个方面涵盖股票价格波动的潜在信息, 对股票价格预测有较强的解释作用.

(2) 建立了基于沪深 300 指数的 BO-CNN-LSTM 模型, 利用 BO 智能优化算法选取 CNN-LSTM 模型的超参数, 从而可以优化模型结构, 提高预测精度, 并避免了人工调整参数带来的局限性. 同时, 为深入对比研究典型参数优化算法, 通过实验探讨了 BO 算法、GA 算法以及 PSO 算法的预测性能.

### 1.5 论文组织框架

第一章为绪论部分. 主要是对本论文的研究背景和意义进行了阐述, 并对当前国内外在股票预测方面的研究状况进行了探讨. 在此基础上, 梳理了本文的研

究内容、创新之处和组织框架.

第二章为论文中涉及到的有关技术理论. 主要有 RNN、LSTM、CNN 三种神经网络模型, BO、GA、PSO 三种参数优化算法, 以及 PCA 方法的相关理论.

第三章为数据来源与数据预处理. 首先对本文选取的股票基础指标、技术指标的金融含义做简要说明; 其次介绍实验环境、数据集来源及其选取原因; 紧接着描述原始数据的预处理过程, 如缺失值、归一化处理等; 最后, 介绍了数据的降维过程.

第四章为股价预测模型的构建. 首先介绍了本文构建的 BO-CNN-LSTM 模型网络结构, 同时描述了利用 BO 算法改进预测模型的参数寻优过程, 帮助神经网络更快地找到全局最优解; 其次介绍了超参数优化范围域及结果; 最后给出了模型效果的评估指标及其计算方法.

第五章为实际数据分析. 利用构建好的 BO-CNN-LSTM 模型预测沪深 300 指数的收盘价, 并通过两个对比实验验证模型性能. 实验一为将其与基准模型 CNN 模型、LSTM 模型、未经参数优化的 CNN-LSTM 模型作对比; 实验二为将其与经过 GA、PSO 算法优化后的 CNN-LSTM 模型作对比.

第六章为总结与展望. 总结全文的研究内容, 并对下一步的研究方向进行展望.

## 第二章 相关理论介绍

正如第一章所述, CNN 具备较强的特征提取效果, 能够有效地挖掘时序数据的深层信息, 提高股价预测的准确率; 同时基于传统 RNN 改进的 LSTM, 具备较强的长期时序特征提取能力. 因此本文首先介绍了 CNN、RNN、LSTM 的基本结构及特点等基础理论. 其次, 为了避免人工调整参数带来的局限性, 基于一类新的参数优化算法, 因此本章还介绍了 BO、GA、PSO 的基本原理及算法流程. 最后简要介绍了数据处理部分会用到的 PCA 方法, 如载荷向量、方差贡献率、碎石图等概念.

### 2.1 深度学习算法

#### 2.1.1 卷积神经网络

Yann LeCun 等在 1988 年针对手写数字识别问题提出了 CNN 模型<sup>[39]</sup>. CNN 是一种前馈神经网络, 它采用了多个检测层来提取和训练特征. 目前 CNN 已经在图像处理, 自然语言处理以及语音识别等领域中有了很好的发展前景, 并且能够有效地应用于时间序列.

一个典型的 CNN 模型主要由五个部分组成: 输入层、卷积层、池化层、全连接层和输出层. 在这些层中, 卷积层的作用是利用卷积核对样本进行卷积运算, 提取输入数据的空间特征. 通常来讲, 大部分的卷积层的第一层都只能够学习到诸如边缘、线条等较低层级的特征. 但是, 多层数网络能够从这些低层级的特征中, 通过一次次迭代提取出相对复杂的特征; 而池化层可以减小数据的空间大小, 从而降低运算的复杂性, 这在一定程度上控制了过拟合; 全连接层和输出层共同构成了网络的后半部分, 全连接层可以将卷积层以及池化层所提取到特征进行综合, 而输出层主要被用来输出模型的训练结果.

CNN 网络的主要特点具体有以下几个方面, 分别是局部连接、权值共享和池化. 局部连接是指由于从空间的角度上来看, 相邻点之间有着更加密切的关系, 因此并不需要每一个神经元去感知全局特征, 只需要让每一个神经元去感知局部特征, 然后更高层的神经元再去合成局部信息, 以达到提取更高层次信息的目的<sup>[41]</sup>, 这种方法能够有效地减少参数的数量. 权值共享是指 CNN 在进行卷积运算时, 其中提取特征的滤波器参数不依赖于所处的位置, 使用同样的过滤器扫描

图像中各个位置, 即权重是一样的, 于是就构成了卷积层. 卷积层会采用一个尺寸大小为  $n \times n$  的卷积核来对输入信息进行卷积操作, 而卷积层中通常包含了多个卷积核. 计算公式如式(2-1)所示.

$$C_m = \sigma(W_m \otimes X + b_m), \quad (2-1)$$

式中:  $C_m$  为卷积层中输出的第  $m$  个特征图;  $\sigma(\cdot)$  为激活函数;  $W_m$  为卷积层中第  $m$  个卷积核的权重矩阵;  $\otimes$  为卷积运算;  $b_m$  为卷积层中第  $m$  个卷积核的偏置;  $m$  为第  $m$  个卷积核. 卷积运算示意图如图 2-1 所示.

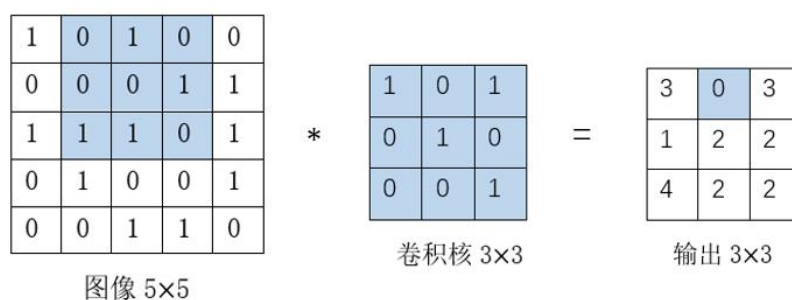


图 2-1 卷积运算示意图

在输入数据通过卷积层完成特征提取后, 会对其进行池化操作, 目的是为了将卷积层输出的特征信息进行抽象化, 从而使该模型的感受野变得更强并且具有更好的泛化性能. 池化层通常也被称作下采样层, 其功能主要是数据压缩、特征降维、保留显著特征、降低过拟合等. 池化层通常出现在卷积层之后, 常见的池化方法是最大值池化与平均值池化, 即通过求出某一局部区域内的最大值或平均值来作为该区域池化后的值. 池化层具有以下三个特点: 首先将特征进行降维, 减少了网络运算的复杂性; 其次, 池化层不需要对权值进行更新, 也就是说在这个过程中不会有任何附加的运算; 最后, 池化层单元还具有在输入数据稍有轻微偏差时的平移不变性, 这时池化操作仍然会给出同样的计算结果, 从而提高了模型的稳健性和精确性. 通过运用多个卷积层对数据提取特征, 随后池化层对得到的信息作抽象处理, 将得到的特征再做一个 Flatten 处理, 于是会形成一个一维向量, 然后将这个向量经过全连接层的计算得到分类或预测的结果, 通过将全连接层输出的结果和标签进行比较, 并在此基础上不断进行迭代以更新权重值, 从而实现反向传播过程. 最大池化和平均池化过程如图 2-2 所示.

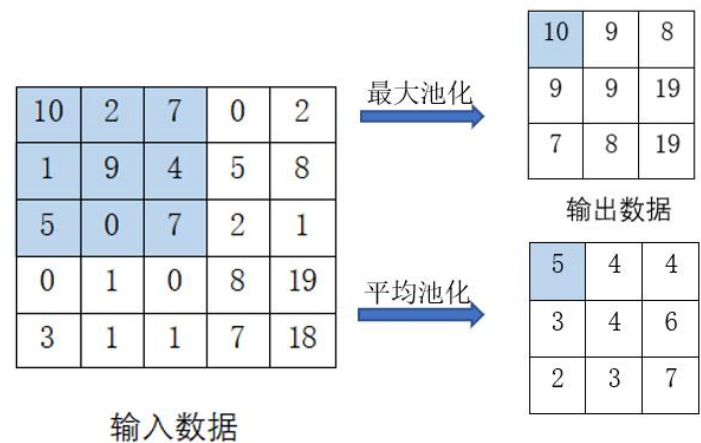


图 2-2 最大池化和平均池化

2. 1. 2 循环神经网络

RNN 大多数情况是被用来处理时序相关的问题. 许多研究人员利用它从数据中提取时序关系, 在时序预测研究上取得了非常大的突破. 与常见的神经网络如 CNN、全连接神经网络(Fully Connected Neural Network, FCNN)等不同的是, RNN 处理并存储前一刻的信息, 使得网络具有了“记忆”的功能. RNN 之所以被称作是递归神经网络, 是因为序列当前的输出结果与之前的输出是相联系的, 即 RNN 隐藏层之间的节点不再是无连接的而是有连接的, 且隐藏层的输入不但包含了输入层的输出, 还包含了前一刻的隐藏层的输出. 因此, 存储在网络中的记忆会对当前的输出产生影响. 从理论上讲, RNN 可以对任意长度的序列数据进行处理, 但是在实际操作中人们通常会假定目前的状态只是和之前的几个状态有关以减少复杂度. 此外, 在 RNN 中, 各个时间点上的循环神经网络结构参数也具有共享性, RNN 的隐藏层结构如图 2-3 所示<sup>[40]</sup>.

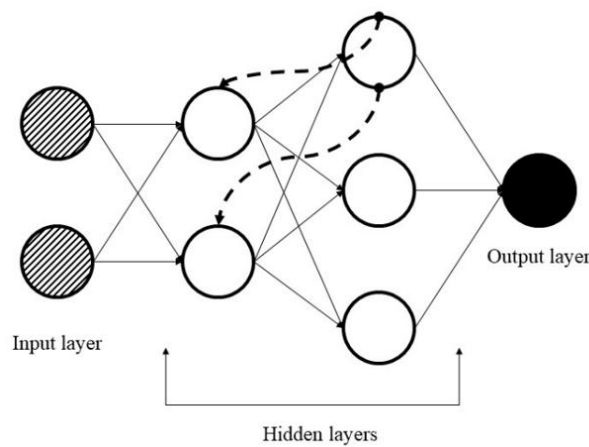


图 2-3 RNN 隐藏层结构



从网络模型的结构上来看, RNN 具有短时记忆能力, 因此 RNN 可以挖掘数据中的时序信息以及语义信息, 从而使输出结果能够体现出时序性. 相对于其他网络结构, 它能更好地反映出股票市场的动态特征. 虽然 RNN 处理时间序列问题的效果很好, 但时间间隔较长时, 该模型就会出现梯度消失或者梯度爆炸的问题, 这就会导致模型不能很好地进行梯度反向传播. 例如, 随着时间的推移, 第一个时间步的特征影响度逐渐变小, 而到了第七个时间步时, 第一个时间步的特征就会无法体现出来, 这将导致权值无法正确更新, 从而对预测结果产生影响. 为了解决这一问题, 诞生了 LSTM 神经网络, 它是一种在 RNN 的基础上所改进的具有门控机制的特殊循环神经网络.

### 2.1.3 长短期记忆网络

LSTM 是 Sepp Hochreiter 和 Jürgen Schmidhuber 等于 1997 年提出的一种时间递归神经网络<sup>[4]</sup>. 当在时间序列问题中需要预测较长间隔的事件时, 可以应用 LSTM 模型来进行处理. 在 LSTM 的结构上, 它与一般的 RNN 结构相似, LSTM 同样是链式结构. 与 RNN 不同的主要是 LSTM 存在控制存储状态的结构.

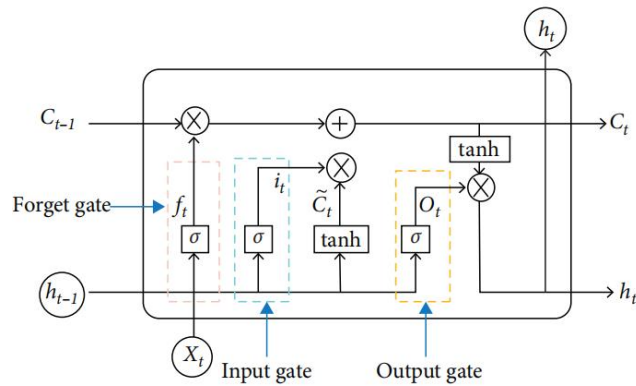


图 2-4 长短期记忆网络结构图

图 2-4 显示了 LSTM 的结构图<sup>[41]</sup>, 与传统的 RNN 相比, LSTM 采用了多种门控结构, 其中包括输入门 $i_t$ , 遗忘门 $f_t$ 和输出门 $o_t$ 三种. 这些门控结构通过递归方程持续地更新  $t$  时刻的细胞状态 $C_t$ .

LSTM 的工作原理是, 首先通过 $f_t$ 来确定哪些信息需要从 $C_t$ 中丢弃掉, 从而减少模型在训练过程中的数据量.  $f_t$ 的值是由  $t$  时刻的输入 $x_t$ 与  $t-1$  时刻的输出 $h_{t-1}$ 共同确定的, 表达式如式(2-2)所示:

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f), \quad (2-2)$$

其中, 由于是以一定的概率通过上一层的隐藏状态, 在这里选择的激活函数为 sigmoid, 目的是为了确保其取值范围为 0~1 之间. 当接收到当前的新信息时, 它

会以一定的概率将上一刻的部分信息叠加起来, 形成新的输入信息. 之后跟  $C_{t-1}$  逐个相乘, 若  $f_t$  对应的数值为 0, 则丢弃该单元相关信息; 若数值为 1, 那么就将其单元的相关信息保存下来.

LSTM 的下一个步骤是确定在  $C_t$  中要存储什么信息. 这包含两个部分: 首先是门控结构  $i_t$ , 它决定我们将更新哪些信息; 其次是一个  $\tanh$  层创建候选值的向量  $\tilde{C}_t$ , 它决定哪些信息能够被加到  $C_t$  中, 计算公式如式(2-3)和(2-4)所示:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i), \quad (2-3)$$

$$\tilde{C}_t = \tanh(W_c h_{t-1} + U_c x_t + b_c), \quad (2-4)$$

$i_t$  也是以一定概率对输入信息进行过滤, 因此这里  $i_t$  的取值也是在 0 和 1 之间, 激活函数选择  $\text{sigmoid}$  函数, 当接收到新的信息时, 把  $i_t$  和  $\tilde{C}_t$  相乘, 在将其加上  $f_t$  的信息, 从而获得新的  $C_t$ .  $f_t$  与  $i_t$  是以一定的概率方式对上一时刻和当前时刻的信息进行选择, 从而改变了当前的  $C_t$ . 将细胞状态从原先的  $C_{t-1}$  更新为当前  $C_t$  的过程可以用式(2-5)来表示:

$$C_t = C_{t-1} \odot f_t + i_t \odot \tilde{C}_t. \quad (2-5)$$

最后需要决定输出信息, 它取决于  $C_t$ . 首先生成一个输出矩阵  $o_t$ , 在这里用到的是  $\text{sigmoid}$  层, 通过输出门来决定输出当前  $C_t$  的哪些部分, 接着将  $C_t$  经过  $\tanh$  激活函数处理过后, 得到一个介于 -1 和 1 之间的数值, 同时将其与  $\text{sigmoid}$  门的输出矩阵  $o_t$  进行相乘, 于是就得到了需要输出的那部分内容  $h_t$ . 式(2-7)中的  $\odot$  表示矩阵的 Hadamard 乘积, 其含义是将两个矩阵相同位置的元素做乘积运算.

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o), \quad (2-6)$$

$$h_t = o_t \odot \tanh(C_t), \quad (2-7)$$

将以上两个式子联立可得:

$$h_t = o_t \odot \tanh(C_{t-1} \odot f_t + i_t \odot \tanh(W_c h_{t-1} + U_c x_t + b_c)), \quad (2-8)$$

$$C_t = C_{t-1} \odot f_t + i_t \odot \tilde{C}_t. \quad (2-9)$$

从推导的式(2-8)和(2-9)可以看出, 与传统的 RNN 相比, LSTM 可以有效地避免梯度消失与爆炸的问题. 由于三个门控结构的激活函数都是  $\text{sigmoid}$ , 即输出结果接近于 0 或者接近于 1. 因此, 当门输出结果为 1 时, 可以很好地传递 LSTM 中的梯度, 极大地降低了梯度消失的可能性; 当门输出结果为 0 时, 则表示前一

时间点的信息不会对当前时间点产生任何影响, 这样就不需要将梯度进行返回以更新参数了.

## 2.2 参数优化算法

参数优化算法的目标是为了对给定的机器学习模型寻找最优超参数, 使其在已有的验证数据集中得到最优的结果. 与模型参数不同的是, 超参数必须在模型训练前确定. 优化目标方程表达式如式(2-10)所示.

$$\theta^* = \operatorname{argmin}_{\theta \in X} f(\theta). \quad (2-10)$$

式中,  $f(\theta)$  表示评估验证集需要最小化的目标分数, 例如模型的错误率或均方根误差等.  $f(\theta)$  也被称为需要优化的模型损失, 这里的损失可以与神经网络训练过程中的损失函数不同, 两者相互独立. 影响  $f(\theta)$  的变量为模型的超参数  $\theta$ , 而  $\theta^*$  是使损失最小的超参数,  $\theta$  可以取超参数范围域  $X$  中的任何值. 优化的最终目的是要在验证集度量上, 找出使损失最小的超参数. 然而在优化的过程中, 评估目标函数要找到分数的成本很高, 每尝试不同组合超参数时, 都需要在训练数据上训练一个模型, 来预测验证集并计算其度量值. 比如, 对于一个具有大量超参数的 LSTM 网络来说, 每个超参数都会有很多的选择空间, 因此需要一次又一次地手动调参, 得到相应模型验证集的错误率, 根据结果再更换超参数组合, 如此反复地进行, 这需要花费很高的时间成本与人力成本.

除了人工调整超参数之外, 还有两种常见的超参数优化方式, 即网格搜索法和随机搜索法, 其中网格搜索法是在模型超参数域的基础上建立一个超参数的网格, 自动地运行训练-预测-评估循环过程; 随机搜索法也是在超参数网格的基础上进行随机选择, 其选择效率略高于网格搜索法. 但这两种搜索方法在进行评估时不受过去评估的影响, 导致这些方法效率较低.

为了实现高效的全局寻优, 目前已有多种方法将确定与随机两类信息有机地融合在一起, 形成了一类新的参数优化算法, 最常用的参数优化算法有 BO、GA 和 PSO 等.

### 2.2.1 贝叶斯优化算法

BO 算法是一种十分有效的全局优化算法, 目标是找到全局最优解. BO 算法根据对未知函数获取的信息, 找到下一个评估位置, 从而最快地达到最优解. 给定优化的目标函数, 可以是广义的函数, 只需指定输入和输出即可, 无需知道内部结构以及数学性质. 通过不断地添加样本点来更新目标函数的后验分布, 直到

后验分布基本贴合于真实分布,简单的说,就是考虑了上一次参数的信息,从而更好的调整当前的参数.

BO 算法结合了手动调参中拥有的人工经验和网格搜索、随机搜索自动选择两者的优点,跟踪过去的评估结果并用于建立概率模型,寻找使损失最小化的超参数组合.如式(2-11)所示,式中 $f$ 表示目标函数或概率代理模型中的参数; $D_{1:t}$ 表示已观测的集合大小为 $t$ ;  $p(D_{1:t}|f)$ 表示在参数 $f$ 条件下观测数据的似然分布;  $p(f)$ 表示模型参数 $f$ 的先验概率分布;  $p(D_{1:t})$ 表示 $f$ 的边际似然分布;  $p(f|D_{1:t})$ 表示经过观测数据 $D_{1:t}$ 修正后 $f$ 的后验概率分布<sup>[42]</sup>.

$$p(f|D_{1:t}) = \frac{p(D_{1:t}|f)p(f)}{p(D_{1:t})}. \quad (2-11)$$

在 BO 算法超参数问题中涉及四个部分的内容:

(1)目标函数:即机器学习模型中需要优化的对象,通常是超参数在验证集上的损失.

(2)域空间:需要搜索的超参数的取值范围,与网格搜索和随机搜索方法使用的超参数域相似.

(3)优化算法:BO 算法中最核心的部分,它包括了两个部分,一个是概率代理模型,另一个是采集函数.概率代理模型被用来取代未知的目标函数,它需要先进行假设先验,接着不断地增加信息量,对先验进行修正,于是就可以得到较为精确的概率代理模型.主要的概率代理模型有 Gaussian 过程、基于树状结构 parzen 密度估计(Tree-structured Parzen Estimator, TPE)和随机森林回归等<sup>[42]</sup>;采集函数是对要选择的下一个超参数值进行评估所依赖的函数,常见的采集函数包括三种:概率提升函数(Probability of Improvement, PI),期望提升函数(Expected Improvement, EI)和置信上界函数(Upper Confidence Bound, UCB)<sup>[43]</sup>.

(4)结果历史记录:根据目标函数进行评估后所存储的历史结果,其中包含了超参数组和在验证集上的损失值.大致来看,BO 算法框架是由以下三个步骤组成的一个迭代式流程:首先,基于最大化采集函数,选取下一个“潜力”评估点 $x_t$ ;其次,根据上一步骤所选取的评估点对目标函数值 $y_t = f(x_t) + \varepsilon_t$ 进行评估;最后,将重新得到的输入与观测值对 $\{x_t, y_t\}$ 加入到历史观测集合 $D_{1:t-1}$ 中,同时更新概率代理模型,准备下一轮的迭代过程.算法 1 为 BO 算法框架的伪代码<sup>[42]</sup>.

---

**算法 1** BO 算法框架

---

```
1: for  $t=1, 2, \dots$ , do  
2:   最大化采集函数, 从而获得下一个评估点:  $\mathbf{x}_t = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x} | D_{1:t-1})$ ;  
3:   评估目标函数值  $y_t = f(\mathbf{x}_t) + \varepsilon_t$ ;  
4:   整合数据:  $D_t = D_{t-1} \cup \{\mathbf{x}_t, y_t\}$ , 并且更新概率代理模型;  
5: end for
```

---

### 2.2.2 遗传算法

GA 又被称作进化算法, 它是一种受到达尔文的进化论启发并参考了生物的演化过程而提出的搜索算法. 它的主要特征是可以对结构对象进行操作, 所以它与其它寻找最优解的算法有很大的区别, 它没有求导, 也没有对函数的连续性进行限制, 它使用了一种概率化的寻优方法, 可以在没有明确的规则的情况下, 自动地获得并引导最优的搜索空间, 并自适应地调整搜索方向. 算法中涉及到的相关术语如下:

- (1)个体: 指染色体具有特征的实体.
- (2)种群: 由个体组成的集合, 集合内的个体数目即为种群的大小.
- (3)进化: 种群逐渐适应其生活的环境, 品质也在不断提高. 物种的演化是通过种群来完成的.
- (4)适应度: 衡量一种生物对其所生活的环境的适应程度.
- (5)基因型: 性状染色体的内部表现.
- (6)表现型: 依据基因型而形成的个体的外部表现.
- (7)选择: 在种群中, 按照一定的概率进行多个个体的筛选, 这是一个以适应度为基础的适者生存的过程.
- (8)复制: 当一个细胞发生分裂时, 遗传物质 DNA 经过复制进入到新产生的细胞中, 此时新细胞便会继承旧细胞的基因.
- (9)交叉: 在两条染色体的某个相同位置处切断 DNA, 而前面和后面的两条又分别相互结合, 从而构成两条新的染色体.
- (10)变异: 在复制过程中, 小概率会出现一些复制差错问题而发生变异, 变异会产生表达新性状的新染色体.
- (11)编码: 在 DNA 中, 遗传信息会以特定的方式排列在一条长链上, 每一条链上都有一个遗传编码. 可以把遗传编码看作是表现型到基因型的映射, 它决定了某一种性状的表达.

(12)解码: 基因型到表现型的映射.

GA 可以仿真自然选择和遗传过程中所要进行的复制、交叉和变异等现象, 从任意一个初始种群出发, 运用随机选择、交叉和变异运算, 生成一群与其所生活的环境更相适应的个体, 由此将种群进化到一个更优化的搜索空间. 以此为基础, 经过反复的繁衍和进化, 最终将其收敛至一群最适应所处环境的个体. 于是获得了问题的最优解. 具体操作步骤如下:

(1)随机生成一个种群, 并将其作为问题初始解. 一般情况下, 初始解与最优解之间可能存在很大的差异, 这是可以接受的, 但前提是初始解是随机生成的, 从而保证个体基因是具有多样性的.

(2)为群体中的个体找到一种适当的编码方式, 可以选用一些常见的编码方式, 比如二进制编码或者浮点数编码, 而这些编码方式的不同将会对后面的基因操作产生很大的影响.

(3)通过建立适应度函数, 以此对个体进行适应度计算, 得到的适应度作为下一步个体筛选的依据.

(4)按照适应度的大小来挑选参加繁衍的父类和母类, 所按照的挑选原则就是适应度高的个体更有可能被挑选出来, 这样就可以将适合度低的个体不断地剔除出去.

(5)对被选中的亲代进行遗传操作, 即复制亲代的基因, 同时使用交叉和变异等操作算子以产生出后代, 在尽可能多地保留优良基因的同时, 增加了基因多样性, 提高了寻得最优解的概率.

(6)按照特定的准则通常是给定的解的阈值或是迭代次数等, 来决定是否要继续执行该算法, 在所有的后代中找到一个适应度最高的个体, 然后将该个体作为解进行返回并终止该程序.

GA 流程图如图 2-5 所示.

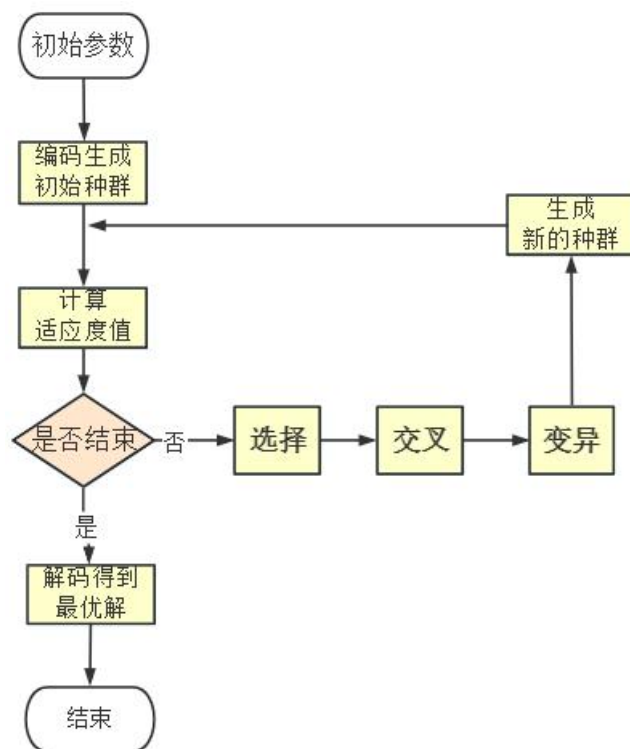


图 2-5 GA 流程图

### 2.2.3 粒子群算法

PSO 是美国学者 Kennedy 与 Eberhart 于 1995 年提出的一种基于群体合作模拟鸟类捕食习性的优化算法<sup>[38]</sup>。PSO 是一种启发式算法，又称为智能优化算法，它的基本思想是利用群体中的个体进行互相合作、分享信息从而找到最优解。

PSO 设计了一种无重量的粒子，它具有两种属性，一个是位置，另一个是速度。位置表示粒子的运动方向，而速度表示粒子的运动速度。首先将一组随机粒子进行初始化作为问题的随机解，每个粒子都会在搜索空间中独立地寻找最优解，以减少搜索时间，同时会标记当前自身的极值，与其它粒子进行共享，从中找出最好的个体极值，将其视作为当前粒子群体中的全局极值。然而也可以只使用当中一部分群体，从而找到局部极值。在这个过程中，粒子会根据个体极值、全局或局部极值来调整自身位置和速度<sup>[31]</sup>。算法中涉及到的相关基本概念如下：

- (1)粒子：优化问题的候选解。
- (2)位置：候选解所在的位置。
- (3)速度：候选解移动的速度。
- (4)适应度：评价粒子优劣的值，一般设置为目标函数值。

(5)个体最佳位置: 单个粒子至今找到的最优解, 即个体极值.

(6)群体最佳位置: 所有粒子至今找到的最优解, 即全局极值.

假定在  $D$  维的搜索空间内有  $N$  个粒子构成了一个群落, 那么第  $i$  个粒子的位置可以用一个  $D$  维向量表示为:

$$X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}), i = 1, 2, \dots, N, \quad (2-12)$$

则粒子速度为:

$$V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD}), i = 1, 2, \dots, N, \quad (2-13)$$

根据式(2-14)和(2-15)更新粒子的位置和速度:

$$X_{id+1} = X_{id} + V_{id} \times T, \quad (2-14)$$

式(2-14)表示粒子第  $d+1$  步所在的位置=第  $d$  步所在的位置+第  $d$  步的速度 $\times$ 运动时间, 每一步运动的时间  $T$  一般取 1.

$$V_{id} = \omega \times V_{id-1} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}), \quad (2-15)$$

式(2-15)由三项构成, 从左至右第一项是惯性部分, 表示上一步速度矢量对粒子的影响, 即粒子运动习性. 第二项是自身认知项, 表示从当前位置指向自身最佳位置的向量, 体现了粒子根据自身经验来调整动作. 第三项是群体认知项, 表示从当前位置指向种群最佳位置的向量, 体现了粒子之间的互相合作及信息共享.  $\omega$ 表示惯性因子, 其数值一般为非负数, 功能是调整求解空间的搜索范围. 与固定值相比, 动态 $\omega$ 可以得到更好的寻优效果.  $c_1$ 、 $c_2$ 表示学习因子, 用于调节学习的最大步长, 通常取值为 2.  $r_1$ 、 $r_2$ 表示  $rand()$ , 即取介于 0~1 之间的随机数. PSO 的具体流程如图 2-6 所示.



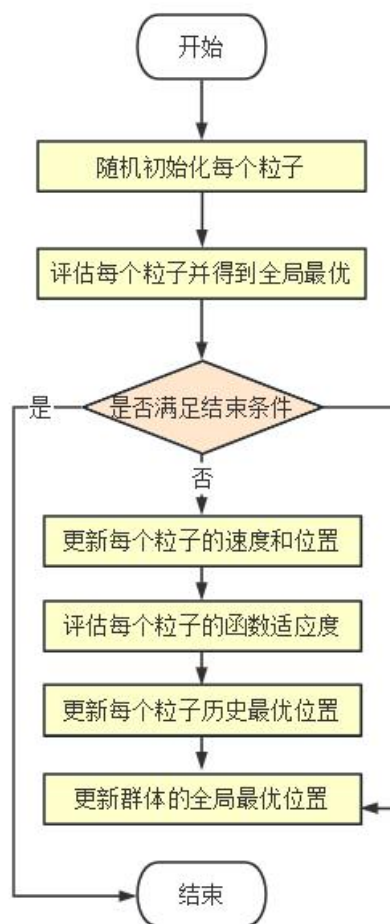


图 2-6 PSO 流程图

## 2.3 主成分分析

PCA 是一种应用最为广泛的线性降维算法，其目的是利用一定的线性投影，把高维数据映射到低维空间，并且希望在所投射的维度中含有尽可能多的信息量即方差最大，从而在保持原有数据特征的前提下，减少数据维数<sup>[44]</sup>。

### 2.3.1 主成分的确定

主成分是指选出几个少于原始变量数目，并且能够对问题中的原始变量做出解释的新变量。假设  $Z_1$  为变量组  $X_1, X_2, \dots, X_p$  的第一主成分，则  $Z_1$  的方差越大，表明  $Z_1$  包含的信息量就越多。则对于变量组  $X_1, X_2, \dots, X_p$  构成的所有线性组合中， $Z_1$  可以表示为如式(2-16)所示：

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p, \quad (2-16)$$

其中 $\phi_{11}, \phi_{21}, \dots, \phi_{p1}$ 表示 $Z_1$ 的载荷, 具体含义是变量 $X_i (i = 1, 2, \dots, p)$ 与 $Z_1$ 之间的相关系数, 于是也就构成了 $Z_1$ 的载荷向量 $\phi_1 = (\phi_{11}, \phi_{21}, \dots, \phi_{p1})^T$ . 为了对荷载的大小进行控制, 避免方差任意变大, 因此将荷载的平方和设定为 1, 即 $\sum_{j=1}^p \phi_{j1}^2 = 1$ . 在提取变量组 $X_1, X_2, \dots, X_p$ 的主成分之前, 需要对原来 $p$ 个变量进行数据中心化处理, 目的是消除变量量纲的影响. 然后再求出满足(2-17)式的线性组合:

$$z_{i1} = \phi_{11}X_{i1} + \phi_{21}X_{i2} + \dots + \phi_{p1}X_{ip}, \quad (2-17)$$

以上问题可视为一个优化问题, 其求解方式为:

$$\max_{\phi_{11}, \dots, \phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^n (\sum_{j=1}^p \phi_{j1}X_{ij})^2 \right\}, \sum_{j=1}^p \phi_{j1}^2 = 1. \quad (2-18)$$

将(2-17)式带入到(2-18)式中, 于是目标函数可转化为 $\frac{1}{n} \sum_{i=1}^n (z_{i1})^2$ . 由于 $\frac{1}{n} \sum_{i=1}^n x_{ij} = 0$ , 则 $z_{11}, z_{21}, \dots, z_{n1}$ 的均值也为零, 故上式中需要最大化的目标函数实际上就是变量 $z_{i1}, i = 1, 2, \dots, n$ 的样本方差.  $z_{11}, z_{21}, \dots, z_{n1}$ 代表 $Z_1$ 的得分, 通过主成分得分的大小, 可以对各个变量在各主成分上的表现进行分析. 如果 $Z_1$ 不能充分地表示出原来 $p$ 个变量的信息, 那么再考虑选择 $Z_2$ , 也就是线性组合中方差第二大的, 并且 $Z_1$ 与 $Z_2$ 不相关, 即满足 $\text{Cov}(Z_1, Z_2) = 0$ , 表示 $Z_1$ 已经存在信息无需再出现在 $Z_2$ 中.

$$z_{i2} = \phi_{12}X_{i1} + \phi_{22}X_{i2} + \dots + \phi_{p2}X_{ip}, \quad (2-19)$$

如(2-19)式所示,  $\phi_2 = (\phi_{12}, \phi_{22}, \dots, \phi_{p2})^T$ 是 $Z_2$ 的载荷向量, 若要让 $Z_2$ 与 $Z_1$ 不相关, 需满足 $\phi_2$ 的方向与 $\phi_1$ 的方向是垂直的. 同理可以得到 $Z_3, Z_4, \dots, Z_p$ .

### 2.3.2 主成分的选取

主成分数目的选取对于正确地使用 PCA 非常重要. 当主成分取得太少时, 无法反映出原始变量所包含的主要信息; 当取得过多时, 很容易引入一些不必要的误差和噪声. 通常使用以下三种方式来选取主成分数目:

(1)通过累计方差贡献率确定主成分数目. 假设变量 $X_i$ 方差为 $\text{Var}(X_i) = \sigma_i^2$ , 通过证明可得:

$$\sum_{i=1}^p \sigma_i^2 = \sum_{i=1}^p \lambda_i, \quad (2-20)$$

其中 $\lambda_i, i = 1, 2, \dots, p$ , 表示 $p$ 个特征值, 式(2-20)表示新得到的综合变量即主成分的总方差等于原始变量的总方差. 于是可以通过以下定义来衡量主成分所包含原有变量的信息, 其中

$$\gamma_i = \lambda_i / \sum_{i=1}^p \lambda_i, \quad (2-21)$$

表示第  $i$  个主成分的方差贡献率, 而

$$\eta_m = \sum_{i=1}^m \gamma_i = \sum_{i=1}^m \lambda_i / (\sum_{i=1}^p \lambda_i), \quad (2-22)$$

表示前  $m$  个主成分的累计方差贡献率. 由此可知累计方差贡献率反映了前  $m$  个主成分从  $p$  个原始变量  $X_1, X_2, \dots, X_p$  中提取了多少信息. 一般情况下, 根据  $\eta_m$  的大小, 可以确定用来替代原始变量的主成分数目. 当  $\eta_m$  大于 80% 时, 可以判断出保留前  $m$  个主成分较为合适, 而剩下的主成分可以省略. 另外, 还可以通过以下准则来确定主成分的数目.

(2) 取特征值  $\lambda$  大于 1 的因子数  $m$ .

(3) 通过碎石图来进行判断. 其中, 纵坐标表示特征值, 横坐标表示主成分的数目. 随着主成分数目的增加, 特征值大小会出现急剧减小, 但到达某一主成分后, 特征值减小幅度不大, 此时可以找到一个转折点, 通常称之为肘(elbow), 肘所对应的主成分数目即为  $m$ .

## 第三章 数据来源与数据预处理

第二章对本文主要涉及到的方法进行了理论介绍. 接下来本章将阐述研究的实验数据及预处理工作. 首先为了使股价预测更加准确, 选取了一些常用的股票基础指标及技术指标, 详细介绍了各指标的基本概念及作用, 并结合本文研究内容等设定了指标参数. 其次介绍了实验环境、实验数据及其选取原因等, 并进行数据清洗. 最后, 为了剔除冗余信息, 提升算法效率, 采用 PCA 方法对选取的 23 个指标变量进行降维, 同时解释了所选取主成分的含义.

### 3.1 股票指标选择

本文除了选用在大部分股价预测研究中都会用到的四价一量等基础指标以外, 还加入了随机指标、动量指标、心理线等较为前沿的技术指标. 技术指标考虑了市场行为的各个方面, 建立了一个数学模型并给出计算公式, 得到一个体现股票市场的某个方面内在实质的数字. 将这些股价指标综合在一起, 能够从多个方面涵盖股票价格波动的潜在信息, 对股价预测有较强的解释作用.

以下是本文选取的股票指标介绍:

- (1)开盘价(open): 某一天交易的股票起始价.
- (2)收盘价(close): 某一天交易的股票最终价.
- (3)最高价(high): 股票当天价格的最大值.
- (4)最低价(low): 股票当天价格的最小值.
- (5)涨跌额(change): 股票当天收盘价与前一日收盘价的差.
- (6)涨跌幅(pct\_change): 股票当天涨跌额与前一日收盘价的比值.
- (7)成交量(volume): 股票当天的买卖数量.
- (8)成交额(amount): 股票当天的成交金额.

(9)异同移动平均线(MACD): 根据短期指数移动平均线 $EMA_{12}$ 与长期指数移动平均线 $EMA_{26}$ 之间的聚合与分离的情况, 来判断何时买卖股票的技术指标. MACD 可以由离差值 DIF 和异同平均数 DEA 求出. 绝对值 DIF 表示短期 $EMA_{12}$ 与长期 $EMA_{26}$ 之间的距离, 这个数值越大, 表明在短时间内, 股票价格将会加速上升或下降. 短期和长期指的是 EMA 的周期, 这里取 12 和 26. DEA 表示 DIF 的平滑移动平均值, 参数通常取 9.  $DEA_{yest}$  为前一日的 DEA 值. 计算公式如下:

$$DIF = EMA_{12} - EMA_{26}, \quad (3-1)$$

$$DEA_9 = DEA_{yest} \times \frac{8}{10} + DIF \times \frac{2}{10}, \quad (3-2)$$

$$MACD = 2 \times (DIF - DEA_9). \quad (3-3)$$

(10)相对强弱指标(RSI): 一条以某一时间段内涨跌点数之和所占比例为依据的技术曲线, 可以反映这段时间内股市的兴衰状况. 由于本文是短期预测, 故选取参数为 6. 计算公式如下:

$$RS_6 = \frac{6 \text{ 日内 } close \text{ 上涨数之和的平均值}}{6 \text{ 日内 } close \text{ 下跌数之和的平均值}}, \quad (3-4)$$

$$RSI_6 = 100 - 100 \times \frac{1}{1+RS_6}, \quad (3-5)$$

RSI 的取值范围介于 0-100 之间, 通常 RSI 以 50 作为分水岭, 超过 50 就是强市, 低于 50 就是弱市. 当小于 20 时为超卖区, 很容易出现短期反弹; 当大于 80 时就是超买区, 很可能会有短期回档.

(11)移动平均线(MA): 通过对某一特定时段 close 进行平均, 并将不同时间的平均值连在一起, 就能得到一条 MA. 它能够反映短期内股价平均成本的波动趋势. 本文选取短期操作参数 5 和 10. 计算公式如下:

$$MA_5 = \frac{5 \text{ 日 } close \text{ 之和}}{5}, \quad (3-6)$$

$$MA_{10} = \frac{10 \text{ 日 } close \text{ 之和}}{10}. \quad (3-7)$$

(12)随机指标(KDJ): KDJ 主要研究 close、high 和 low 之间的关系, 可以反映一段时间里股票市场的价格走势情况和超买超卖现象. 本文选定默认参数 9, 计算 9 日内未成熟随机值  $RSV_9$ , 再求出 K 值、D 值、J 值. 其中对于前一天缺少 K、D 两项数据的情况, 取初始值 50. 计算公式如下:

$$RSV_9 = \frac{close_9 - low_9}{high_9 - low_9} \times 100, \quad (3-8)$$

$$K = \frac{2}{3} \times K_{yest} + \frac{1}{3} \times RSV_9, \quad (3-9)$$

$$D = \frac{2}{3} \times D_{yest} + \frac{1}{3} \times K, \quad (3-10)$$

$$J = 3K - 2D. \quad (3-11)$$

(13)顺势指标(CCI): CCI 是以统计理论为基础, 提出了在一段时间内股价偏离平均区间程度的定义. 它主要在超买和超卖区域发生作用, 对急涨急跌的行情检测性相对准确. 本文选定默认参数 14. 计算公式如下:

$$TP = \frac{high+low+close}{3}, \quad (3-12)$$

$$MA = \frac{\text{近 14 日 } close \text{ 的累计之和}}{14}, \quad (3-13)$$

$$MD = \frac{\text{近 14 日 } |MA-close| \text{ 的累计之和}}{14} \quad (3-14)$$

$$CCI_{14} = \frac{TP-MA}{MD \times 0.015}. \quad (3-15)$$

(14)动量指标(MTM): MTM 通过对股票价格波动的速度进行观察, 以此来度量股价波动的动能, 从而发现了股价反转规律. 参数值通常在 6 到 14 之间选取, 本文选定起始参数 6.  $close_6$  表示 6 日前的收盘价. 计算公式如下:

$$MTM = close - close_6. \quad (3-16)$$

(15)乖离率(BIAS): 通过计算 close 与 MA 之间的差距百分比来反映两者的偏离程度, 于是可知当价格剧烈波动时, 因偏离 MA 趋势而会出现反弹或回档的可能性, 和当价格在正常范围内波动时, 从而维持原来趋势的可信度. 本文选定默认参数 12. 计算公式如下:

$$BIAS = \frac{close-MA_{12}}{MA_{12}} \times 100\%. \quad (3-17)$$

(16)能量潮(OBV): 以 close 和 volume 指标为依据来反映股市的兴衰状况. 根据股价的上涨和下跌, 人为地把 volume 分为正和负, 并做累加运算.

(17)心理线(Psy): 通过反映投资者对股市涨跌情况的心理状况, 将在一段时间内, 投资者更偏向于买家还是卖家的心理事实转换成了一个数值, 于是构成了一个情绪指数. 本文选定默认参数 12. 计算公式如下:

$$PSY = \frac{\text{12 日内上涨天数}}{12} \times 100. \quad (3-18)$$

(18)布林线(BOLL): 其工作原理为 BOLL 随着中轨线 MD、上轨线 UP、下轨线 DN 三条线所形成的股价通道宽窄的变动, 以此反映股票的波动情况. 本文选定默认参数 20. 计算公式如下:

$$MD = 20 \text{ 日股票价格移动平均线}, \quad (3-19)$$

$$UP = MD + 2 \text{ 倍的 } 20 \text{ 日股票价格标准差}, \quad (3-20)$$

$$DN = MD - 2 \text{ 倍的 } 20 \text{ 日股票价格标准差}, \quad (3-21)$$

三线形成的股价通道会随着股票波动而收缩,当股价位于 MD 上方时,此时处于强势区;当股价位于 MD 下方时,此时处于弱势区.若三线同时向上,表明此时股价强劲,可买进股票;若三线同时向下,表明此时股价颓势明显,宜观望.

## 3.2 实验环境及数据介绍

### 3.2.1 实验环境

所有的模型训练都是由一台 Xiaoxin Air 15IKBR 来进行的,其处理器配置为 Intel(R)Core(TM)i5 - 8250UCPU@1.60GHz1.80GHz, 8G 存储空间和操作系统为 64 位的 Window10 系统, Python3.8.5 版本.此外,在代码编写部分使用 Jupyter Notebook 作为集成开发环境,神经网络模型的搭建与训练通过 Tensorflow.Keras 框架完成.

### 3.2.2 实验数据

由于股指包含了许多行业的不同只股票,能够更好地反映出经济的势头和总体情况.因此相对于个股而言,股指波动性较低.沪深 300 指数(简称沪深 300,代码 000300)是以沪深市场中流动性好、规模大的最具代表性的 300 只股票为编制对象,同时以这些成份股的自由流通量为权数.因此,为了避免个股随机性或恶意操纵等偶发因素造成的预测不准确问题,本文选取沪深 300 作为研究对象.同时由于神经网络是一种数据驱动模型,它在学习过程中必须要有大量的样本.为此,我们选择了一个更长的时段以增加模型训练样本量,从而提高模型的泛化能力.选用时间跨度从 2006 年 1 月 4 日至 2022 年 9 月 30 日的股票历史数据,数据包括上述介绍的 23 个股价指标变量,按照 3:1 的比例来划分训练集和测试集,也就是将 2006 年 1 月 4 日至 2018 年 7 月 25 日所有交易日的数据作为训练集,而剩余交易日的数据作为测试集.所有实验数据均下载于 wind 金融终端.

## 3.3 数据预处理

### 3.3.1 缺失值处理

针对缺失值数据,常用的解决方法有填充和删除.例如可以采用众数、均值、中位数等统计量进行填充;除此之外,还可以根据缺失值数据的数量考虑将整个

属性或样本删去。由于本实验中获得的数据是完整的，因此这里无需对缺失值进行处理。

### 3.3.2 归一化处理

由于数据的变量间单位是不一样的，于是在对模型进行训练时，会因为变量间存在着量纲的差异，造成了梯度下降的速度较慢，从而对训练速度产生一定的影响。故需要对数据进行标准化处理，即数值范围控制在 0~1 之间。本文通过最大最小值方法对数据进行归一化，便于模型在不同特征之间进行比较。公式如式 (2-44) 所示。

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad (3-22)$$

其中  $\tilde{x}$  为归一化后的数据， $x$  为原始数据， $\max(x)$  和  $\min(x)$  分别为原始数据的最大值和最小值。

### 3.3.3 相关系数的计算

为了更好地反映出各个股票指标间的相关性，本文采用皮尔逊相关系数对各指标间的相关程度进行衡量。从图 3-1 热力图的颜色深浅可以看出，大多数数值都在 0.7 之上，表明所选取的 23 个指标之间具有很强的相关性，因此需采取一定的方法对数据进行降维，从而降低神经网络在训练过程中的复杂性，以提升模型的学习效率。

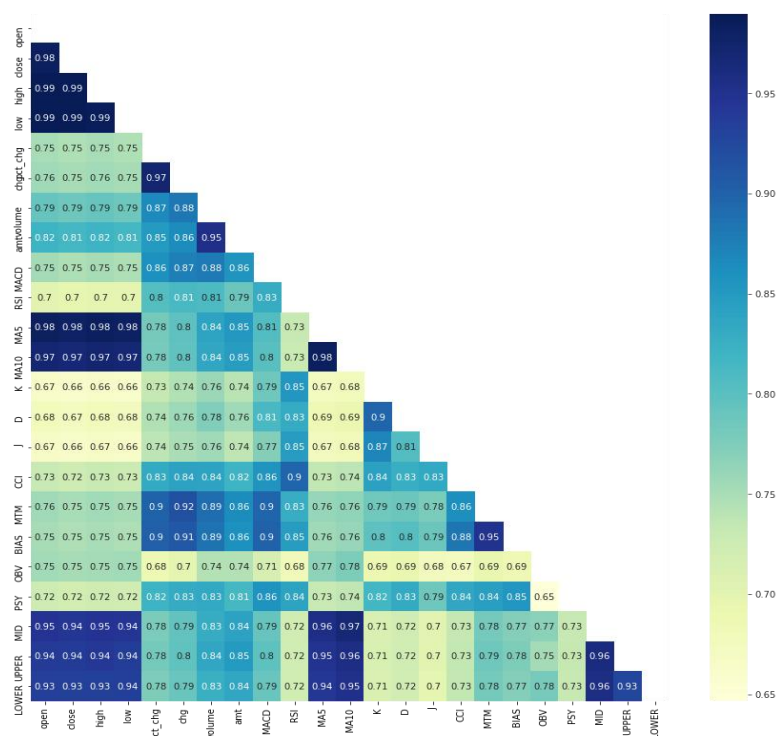


图 3-1 各指标间的皮尔逊相关系数



3.3.4 PCA 降维

PCA 方法一般适用于特征变量较多的模型，主要作用是对变量进行降维，变量过多会使研究的难度加大，影响整体工作进展。由于本文选取的指标变量间是存在较强相关性的，故采用 PCA 方法对 23 个指标变量进行降维处理。经过处理后的数据在保留原始数据信息的同时，降低了数据维度。

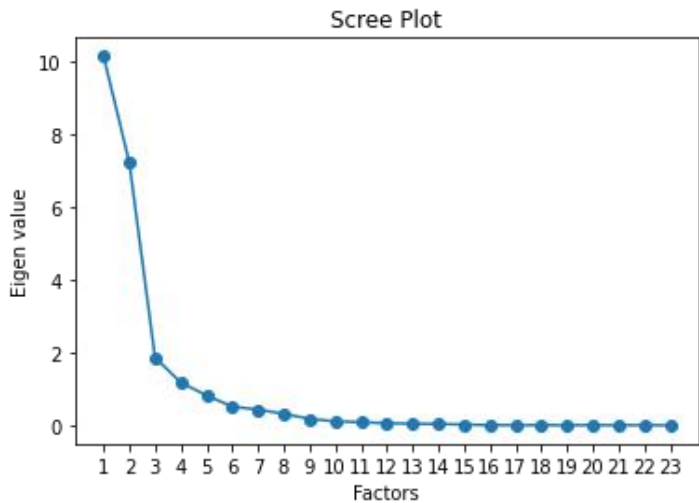


图 3-2 碎石图

表 3-1 PCA 特征值及贡献率

主成分次序	特征值	方差贡献率%	累计方差贡献率%
1	10.146	44.11	44.11
2	7.230	31.44	75.55
3	1.861	8.09	83.64
4	1.175	5.11	88.75
5	0.820	3.57	92.32
6	0.510	2.22	94.53
7	0.428	1.86	96.40
8	0.313	1.36	97.76
9	0.173	0.75	98.51
10	0.105	0.46	98.97

结合图 3-2 碎石图和表 3-1 特征值计算结果可知，前 3 个主成分对应的特征值分别为 10.146, 7.230, 1.861，特征值均大于 1，同时累计方差贡献率高达 83.643%，这一数值已超过 80%，表明前 3 个主成分可以涵盖原始 23 个指标数据

83%的信息,有效反映了原始数据的本质特征.因此,本文将前3个主成分构成的3维数据替代原始的23维数据作为模型的输入.3个主成分的表达式如下:

$$Y_1 = -0.311X_1 - 0.311X_2 - 0.311X_3 - 0.311X_4 + 0.015X_5 + 0.01X_6 - 0.172X_7 - 0.23X_8 - 0.03X_9 + 0.013X_{10} - 0.311X_{11} - 0.311X_{12} + 0.007X_{13} + 0.003X_{14} + 0.011X_{15} + 0.017X_{16} + 0.006X_{17} + 0.019X_{18} - 0.227X_{19} + 0.001X_{20} - 0.309X_{21} - 0.306X_{22} - 0.307X_{23}.$$

$$Y_2 = -0.015X_1 - 0.022X_2 - 0.017X_3 - 0.02X_4 - 0.171X_5 - 0.167X_6 - 0.104X_7 - 0.083X_8 - 0.228X_9 - 0.357X_{10} - 0.006X_{11} + 0.01X_{12} - 0.348X_{13} - 0.313X_{14} - 0.326X_{15} - 0.343X_{16} - 0.31X_{17} - 0.347X_{18} + 0.022X_{19} - 0.285X_{20} + 0.033X_{21} + 0.031X_{22} + 0.034X_{23}.$$

$$Y_3 = -0.005X_1 + 0.024X_2 + 0.007X_3 + 0.01X_4 + 0.619X_5 + 0.625X_6 - 0.106X_7 - 0.079X_8 - 0.235X_9 + 0.026X_{10} + 0.001X_{11} + 0.009X_{12} - 0.171X_{13} - 0.283X_{14} - 0.006X_{15} - 0.026X_{16} + 0.096X_{17} + 0.076X_{18} + 0.049X_{19} - 0.146X_{20} + 0.036X_{21} + 0.033X_{22} + 0.04X_{23}.$$

其中 $Y_1$ 、 $Y_2$ 、 $Y_3$ 分别为第一、第二、第三主成分,  $X_1$ 为收盘价,  $X_2$ 为开盘价,  $X_3$ 为最高价,  $X_4$ 为最低价,  $X_5$ 为涨跌幅,  $X_6$ 为涨跌额,  $X_7$ 为成交量,  $X_8$ 为成交额,  $X_9$ 为异同移动平均线 MACD,  $X_{10}$ 为相对强弱指标 RSI,  $X_{11}$ 为移动平均线 MA<sub>5</sub>,  $X_{12}$ 为移动平均线 MA<sub>10</sub>,  $X_{13}$ 为随机指标 K 值,  $X_{14}$ 为随机指标 D 值,  $X_{15}$ 为随机指标 J 值,  $X_{16}$ 为顺势指标 CCI,  $X_{17}$ 为动量指标 MTM,  $X_{18}$ 为乖离率 BIAS,  $X_{19}$ 为能量潮 OBV,  $X_{20}$ 为心理线 PSY,  $X_{21}$ 为布林线中轨线 MD,  $X_{22}$ 为布林线上轨线 UP,  $X_{23}$ 为布林线下轨线 DN.

从主成分的系数矩阵可以观察到, 价格指标、成交量、成交额、MA<sub>5</sub>、MA<sub>10</sub>、BOLL、OBV 在 $Y_1$ 载荷较大, 这说明 $Y_1$ 能够较好地反映这些指标的信息, 故 $Y_1$ 可解释为股票趋势指标. 同理 MACD、RSI、K、D、J、CCI、MTM、BIAS、PSY 指标在 $Y_2$ 载荷较大, 故 $Y_2$ 可解释为股票动力指标. 涨跌幅、涨跌额在 $Y_3$ 载荷较大, 故 $Y_3$ 可反映收盘价的波动能力.

第四章 基于 BO-CNN-LSTM 模型的构建

本章将详细介绍 BO-CNN-LSTM 模型的搭建过程，基本框架为首先展示了模型的网络结构以及 Conv1D 结构图，同时依据模型图对网络层进行了描述；接着确定了 BO 算法的优化目标函数、优化算法以及超参数范围域，并给出 CNN-LSTM 模型中的学习率、卷积核个数及尺寸大小等超参数的优化结果。最后给出了模型的评估指标及计算方法。

4.1 BO-CNN-LSTM 模型网络结构

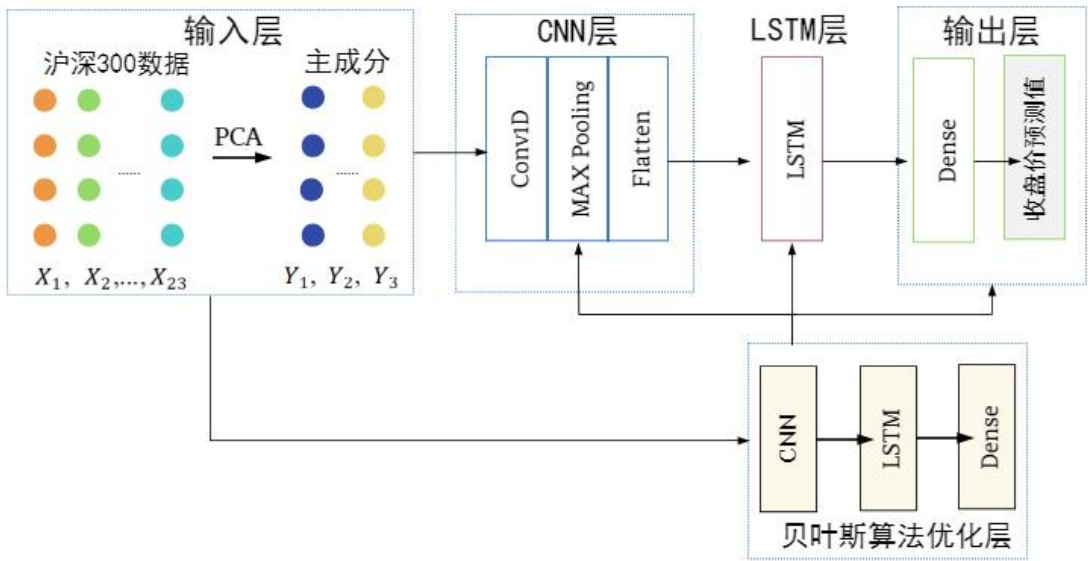


图 4-1 BO-CNN-LSTM 模型网络结构

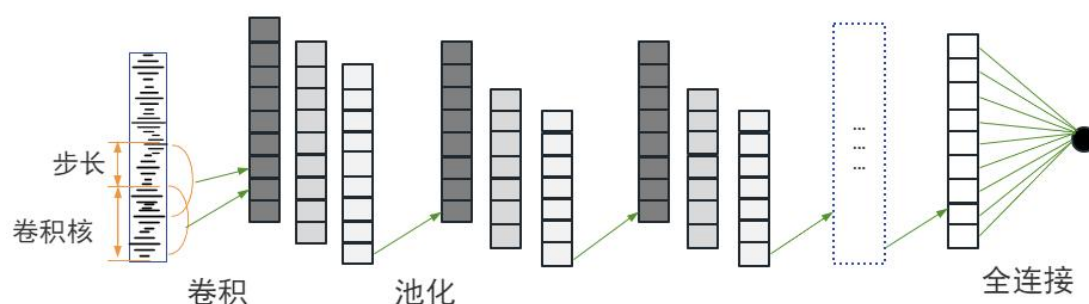


图 4-2 Conv1D 结构

如图 4-1 所示, 本文搭建的 BO-CNN-LSTM 模型主要由输入层、CNN 层、LSTM 层、输出层、贝叶斯算法优化层组成. 模型各层具体解释如下.

第一层为输入层. 对经过缺失值、归一化等数据预处理后的沪深 300 数据  $X_1, X_2, \dots, X_{23}$ , 采用 PCA 方法进行降维, 从而在保持原有数据特征的前提下, 减少数据维数. 同时将选取出来的 3 个主成分  $Y_1, Y_2, Y_3$  作为模型的输入.

第二层为 CNN 层. 它包含了卷积层 Conv1D、池化层 Max Pooling、Flatten 三个部分. 通常在图像识别问题上, 会采用二维卷积 Conv2D 结构进行特征处理. 而股票数据是一类会随着时间的推移而发生变化的时序数据, 故首先采用一维卷积 Conv1D 提取输入数据的特征. 其次利用 Max Pooling 对卷积后的数据作进一步的下采样, 以去除冗余信息、压缩特征. 输入特征在经过卷积和池化后, 特征图维度会明显减少, 从而得到了更有效的特征值, 这在一定程度上缓解了过拟合. 最后使用 Flatten 对数据进行扁平化处理, 将数据“压平”成一维向量, 作为 LSTM 层的输入.

第三层为 LSTM 层. 利用 LSTM 网络提取隐藏在股票数据中的时序特征. 由于 LSTM 的特征提取能力较强, 一般 1 至 2 层就能达到较好的效果. 同时基于 LSTM 结构的复杂性, 每一层网络都会涉及较多的参数需要优化, 所以本模型将 LSTM 层数定为 1 层.

第四层为输出层. 输出层包含了两个全连接层 Dense, 第一个 Dense 用于汇聚 LSTM 网络信息, 转换信息维度; 第二个 Dense 用于输出模型的预测结果.

第五层为贝叶斯算法优化层. 为了提升模型预测效果, 通过 BO 算法对本模型中的学习率 `learning_rate`、卷积核个数 `filters`、卷积核尺寸大小 `kernel_size`、LSTM 层神经元个数 `lstm_units`、Dense 层神经元个数 `dense_units` 五个超参数进行优化确定.

## 4.2 超参数优化

由于本文涉及的超参数数量较多,故而模型的复杂度就会增加,极易发生过拟合现象.通常而言是根据经验调整超参数组合,以期获得较好的泛化水平.但人工调参难以全面地考虑所有相关的影响因素和历史表现,且每选一次新的超参数组合,都需要重新训练模型得出结果,所需要的人力与时间成本很高.即使通过人工调参得到了较好的效果,所构建出的模型也仅限于本文使用的数据集所对应的股价预测问题表现良好,该超参数组合对于其他股票价格的预测能力仍然未知.因此为进一步提高模型的泛化能力,使其对于不同数据集都能达到良好的预测效果,本节将使用 BO 算法选取模型超参数.

### 4.2.1 优化目标函数及优化算法

超参数优化可以看作是一个“黑盒”问题,超参数组合作为“黑盒”的输入,对应的输出就是需要优化至最小值的目标,也称为损失.构建该“黑盒”的过程就是学习超参数组合与优化目标的关系的过程,也是贝叶斯优化算法的关键.首先定义贝叶斯优化算法需要优化的目标,即目标函数.由于本文选取超参数组合的最终目的是为了提高模型在测试集上的预测能力,从而达到更好的泛化能力.因此,本文超参数优化以测试集均方根误差为目标函数.

确定了优化算法的输入与输出,接下来需要确定连接两者的“黑盒”的构建方法,该方法是贝叶斯优化算法的核心.为了学习超参数组合与优化目标之间的“黑盒”,本文将使用 TPE 来构建优化算法.TPE 算法通过在现有的目标函数估计结果的基础上建立一个优化的目标概率模型,根据所建立的概率模型,寻找使目标函数最小化的概率最大的超参数值,然后基于选取出的超参数组合训练出的模型更新输入输出对,继续构造新的概率模型,循环以上过程直到损失降至最低点.

TPE 使用两个密度函数来定义  $p(x|y)$ :

$$p(x|y) = \begin{cases} l(x), & \text{if } y < y^* \\ g(x), & \text{if } y \geq y^*. \end{cases} \quad (4-1)$$

$l(x)$  是使用观测空间  $\{x^{(i)}\}$  来建立的,该观测空间对应的损失  $f(x^{(i)})$  小于  $y^*$ ,使用剩下的观测来建立  $g(x)$ .基于高斯过程的方法偏向更大的  $y^*$ ,基于 TPE 的方法取决于比当前所观测到最好的  $f(x)$  更大的  $y^*$ ,这样一些点就可以用于建立  $l(x)$ .

TPE 选择期望改进(Expected Improvement, EI)作为采集函数,由于无法得到  $p(y|x)$ ,使用贝叶斯公式进行如下转换:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) p(y|x) dy = \int_{-\infty}^{y^*} (y^* - y) \frac{p(x|y)p(y)}{p(x)} dy. \quad (4-2)$$

其中,  $y^*$  代表阈值, 我们令  $\gamma = p(y < y^*)$ , 表示 TPE 算法的一定分位数, 用于划分  $l(x)$  和  $g(x)$ , 范围在  $(0, 1)$  之间. 采集函数用于确定在何处采集下一个样本点.

为了简化上式, 首先对分母进行构造:

$$p(x) = \int p(x|y) p(y) dy = \gamma l(x) + (1 - \gamma) g(x). \quad (4-3)$$

假设我们有 100 个观测值,  $\gamma$  设置为 0.2, 意味着将使用 20 个最佳观测值来构建好的分布, 其余的 80 个用于构建坏的分布.

其次, 对于分子, 我们可以得到

$$\begin{aligned} \int_{-\infty}^{y^*} (y^* - y) p(y|x) dy &= l(x) \int_{-\infty}^{y^*} (y^* - y) p(y) dy \\ &= \gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} p(y) dy \end{aligned} \quad (4-4)$$

最后, EI 可以化简为

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} p(y) dy}{\gamma l(x) + (1 - \gamma) g(x)} \propto \left( \gamma + \frac{g(x)}{l(x)} (1 - \gamma) \right)^{-1} \quad (4-5)$$

最后一个表达式表明, 为了最大化 EI, 希望在点  $x$  下  $l(x)$  概率高的同时  $g(x)$  概率低. 在每次迭代中, 算法返回具有最大 EI 的候选  $x^*$ :

$$x^* = \operatorname{argmax} EI_{y^*}(x) \quad (4-6)$$

#### 4.2.2 超参数范围域及结果

无论是传统的人工调参, 还是使用得较为广泛的网格搜索和随机搜索超参数调优方法, 超参数都应设定一个合理的范围. 故利用自动优化算法寻找最优超参数组合前, 同样需要确定每一个超参数的范围域. 本文超参数范围域由经验及实验确定. 表 4-1 为超参数范围域及经优化算法搜索后的结果.

表 4-1 超参数范围域及结果

超参数	范围域	参数结果
<b>learning_rate</b>	[0.0001, 0.01]	0.0035
<b>filters</b>	[2, 100]	34
<b>kernel_size</b>	{1, 2}	2
<b>lstm_units</b>	[2, 100]	68
<b>dense_units</b>	[2, 500]	201

### 4.3 模型效果评估指标

为了比较各个模型的预测准确性,本文选用均方根误差(Root Mean Square Error, RMSE)、平均绝对误差(Mean Absolute Error, MAE)和决定系数( $R^2$ )作为模型效果的评估指标.在下文中 $y^{(i)}$ 为收盘价的真实值,  $\hat{y}^{(i)}$ 为模型对收盘价的预测值,  $\bar{y}$ 为收盘价真实值的平均值,  $n$ 为预测值的个数.

(1)均方根误差: RMSE 是预测值与真实值偏差的平方,除以预测个数 $n$ 后的平方根,它对于预测值误差的大小极为敏感. RMSE 的计算公式为:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2}. \quad (4-7)$$

(2)平均绝对误差: MAE 表示预测值和真实值之间的距离平均值.若预测值和真实值之间的误差没有得到有效控制,就会出现误差正负抵消的情况,而 MAE 可以有效地避免这种情况的发生. MAE 的值越小,表示模型预测效果越好. MAE 的计算公式为:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y^{(i)} - \hat{y}^{(i)}|. \quad (4-8)$$

(3)决定系数:  $R^2$ 用于判断模型的拟合效果,  $R^2$ 值越接近于 1,则表示模型的预测效果越好.  $R^2$ 的计算公式为:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^n (y^{(i)} - \bar{y})^2}. \quad (4-9)$$

## 第五章 实证分析

在本章中, 基于第四章搭建的 BO-CNN-LSTM 模型, 将第三章处理好的数据作为模型的输入, 完成模型训练, 并将训练好的模型在测试集上进行验证. 为了验证 BO-CNN-LSTM 模型的有效性, 在相同实验环境下搭建了不同的模型进行对比, 最后根据 RMSE、MAE、 $R^2$ 、模型耗时等评估指标, 对比分析各模型预测效果. 本章设计了两个对比实验来做进一步的说明.

### 5.1 对比一: 各模型预测效果对比

在本节中, 选取了单一的 CNN 模型、LSTM 模型作为基准模型, 同时为了体现 BO 算法的作用, 又加入了未经过参数优化的 CNN-LSTM 模型进行对照. 综上所述本文用到以下四个模型: CNN 模型、LSTM 模型、CNN-LSTM 模型、BO-CNN-LSTM 模型, 在模型对比上具有一定的说服力.

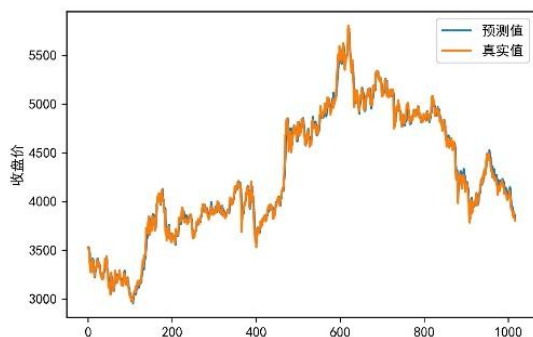


图 5-1 BO-CNN-LSTM

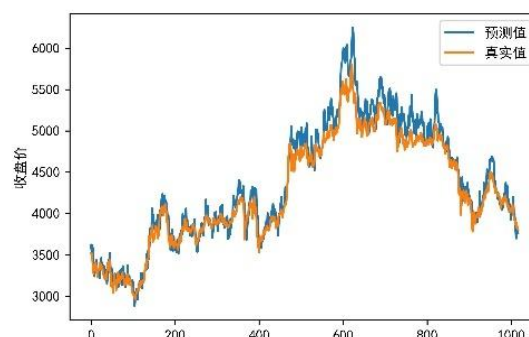


图 5-2 CNN-LSTM

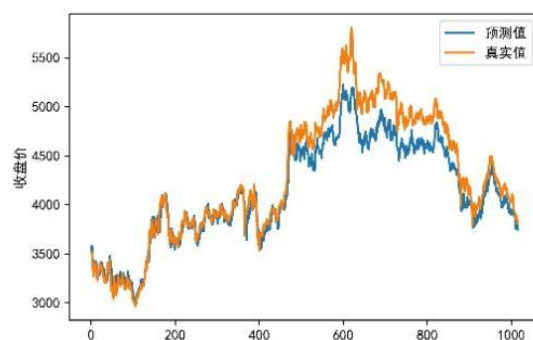


图 5-3 LSTM

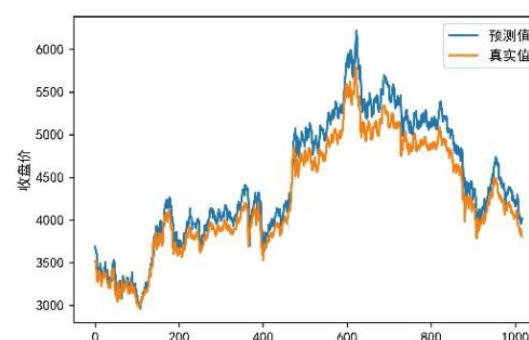


图 5-4 CNN



表 5-1 各模型评估指标结果

模型	RMSE	MAE	R <sup>2</sup>
CNN	220.406	189.283	0.8858
LSTM	211.295	161.154	0.8951
CNN-LSTM	165.111	122.528	0.9359
BO-CNN-LSTM	60.399	44.684	0.9914

通过图 5-1, 5-2, 5-3, 5-4 和表 5-1 的预测结果可以看出, CNN 模型的 RMSE、MAE 最大, R<sup>2</sup>最小. BO-CNN-LSTM 模型的 RMSE、MAE 最小, R<sup>2</sup>最大且最接近 1. CNN 模型与 LSTM 模型的各评估指标结果相对接近但误差值都较大, 模型效果表现一般. CNN-LSTM 模型在 RMSE、MAE 和 R<sup>2</sup>三个指标上都优于单一的 CNN 模型、LSTM 模型, 同时对局部峰值处的趋势判断相对较好, 这表明通过将 CNN 模型和 LSTM 模型的优势进行融合, 有效地抓取了数据的全局和局部特征. 而 BO-CNN-LSTM 模型在各项指标上结果均为最优. 整体来看, 相较于单一的 CNN 模型、LSTM 模型以及未经参数优化的 CNN-LSTM 模型, BO-CNN-LSTM 模型的 RMSE 分别降低约 72.6%, 71.41%, 63.42%; MAE 分别降低约 76.39%, 72.27%, 63.53%; R<sup>2</sup>分别提高 11.90%, 10.76%, 5.93%. BO-CNN-LSTM 模型的预测效果有明显的提升. 在时间不太长的情况下, 各模型的预测结果与实际情况吻合良好. 但随着预测时效的不断延长, CNN 模型的预测精度明显下降, LSTM 模型也受到一定程度的影响, CNN-LSTM 模型相对单一的 CNN 模型、LSTM 模型来说表现较好, 但预测值与真实值之间也出现了较大的偏离. 而 BO-CNN-LSTM 模型的预测精度仍然能够保持稳定, 对股价波动趋势预判基本正确. 这说明 BO-CNN-LSTM 能够有效地处理短期股价预测问题.

上述结果表明, 相比于 CNN 模型、LSTM 模型以及 CNN-LSTM 模型, BO-CNN-LSTM 模型有着更高的预测精度, BO 算法可以有效地提高 CNN-LSTM 模型的预测性能. 这也验证了本文构建的 BO-CNN-LSTM 模型在沪深 300 指数预测上的有效性, 该模型可以应用于金融时间序列预测领域, 其预测结果可为股市投资提供一定的参考依据.

5.2 对比二：不同参数优化算法的预测性能对比

为了进一步验证本文构建的 BO-CNN-LSTM 模型在股票价格预测上的优越性, 本节将 GA、PSO 算法分别用于 CNN-LSTM 模型的超参数寻优, 同时对三种

参数优化算法进行预测性能对比. 优化算法的主要方法为, 将 CNN-LSTM 模型的 RMSE 作为优化算法的适应度函数, 从而找到 CNN-LSTM 模型的最佳超参数, 并用于 CNN-LSTM 预测模型中.

5.2.1 实验参数设置

GA 种群规模通常在 10 到 200 之间, 这样的规模取值能够使得算法效率较高, 同时也能够保证算法在运行过程中, 避免出现过多的染色体, 从而保证种群质量. 由于本文的数据量比较大, 但问题相对来说并不复杂, 所以种群规模选取为 20, 即一个种群中染色体的数量为 20. GA 终止条件一般有两种, 一种是阈值终止, 即当进化到一定程度时, 必须停止进化; 另一种是迭代次数终止, 即每经过一次迭代(或称为一次循环)就对染色体进行操作. 本文采用迭代次数设置的方法作为终止条件, 并设定迭代次数为 15, 这样可以保证算法能够及时停止进化过程. 交叉概率是一种用于确定两个个体是否会发生交叉的概率, 一般大于 0.9; 变异操作概率用于允许少数个体的变异, 以避免局部最优解的约束, 一般小于 0.1. 本文设定交叉概率为 0.9, 变异概率为 0.05.

PSO 初始种群数目同样设定为 20, 迭代次数为 15, 学习因子取推荐值  $C_1=C_2=2$ , 惯性权重  $w$  为 0.9. GA、PSO 算法参数设置如表 5-2 所示.

表 5-2 算法参数设置

优化算法	参数	取值
GA	种群规模	20
	迭代次数	15
	交叉概率	0.9
	变异概率	0.05
PSO	学习因子	$C_1=2, C_2=2$
	种群规模	20
	迭代次数	15
	惯性权重	0.9

为了使得实验结果具有可比性, 超参数范围域的设定与 BO 算法保持一致, 即  $U=[500, 100, 2, 100, 0.01]$ ,  $L=[2, 2, 1, 2, 0.0001]$ . 其中:  $U$  为超参数上限;  $L$  为超参数下限, 从左到右分别代表全连接层神经元个数、卷积核个数、卷积核尺寸大小、LSTM 层神经元个数、学习率五个超参数.

### 5.2.2 实验结果分析

在对 BO-CNN-LSTM 模型、GA-CNN-LSTM 模型、PSO-CNN-LSTM 模型进行训练之后,统计了三个模型在测试集上预测结果的 RMSE 和训练时间,结果如表 5-3 所示:

表 5-3 各参数优化方法结果对比

模型	RMSE	模型训练时间
<b>BO-CNN-LSTM</b>	60.399	<b>4min15s</b>
<b>GA-CNN-LSTM</b>	<b>56.884</b>	44min16s
<b>PSO-CNN-LSTM</b>	57.352	46min02s

由表 5-3 可知,三种优化算法的 RMSE 相差不大,但相比于 BO 算法,GA 和 PSO 算法的 RMSE 较小,表明经 GA、PSO 优化后的模型预测效果略好.但经 GA、PSO 优化的模型训练时间是 BO-CNN-LSTM 模型的十倍多,与 BO 算法的单点搜索相比,GA、PSO 等群体智能算法的每一次迭代都要更新超参数的点集,而每一个更新的点都对应着训练模型和计算目标函数值,这是很耗时的.相比之下,BO 采用主动搜索最优策略,因此省去了很多不必要的采样点的操作.由于本文实验数据是以日为单位的,如果采用更小的时间尺度(比如每小时),则需要更多的训练样本,GA、PSO 使得模型运行时间过长的缺点将会更加明显.在此基础上,建议当训练数据量较小时,可以用 GA 和 PSO 来优化模型超参数,以达到较小的预测误差;而对于训练数据量较大时,建议用 BO 优化超参数以节省时间.

## 第六章 总结与展望

在第五章中,重点探讨了 BO-CNN-LSTM 模型在股票价格预测问题中的应用,实验结果证明了该模型的有效性.本章将对此次研究工作做一个全面的总结,并指出在后期实验中可探索的方向.

### 6.1 本文总结

本文首先阐述了股票市场与国民经济的关系,点明股票价格预测的研究背景和意义;接着从国内外股价预测的理论演进及实践发展入手,对各个阶段的研究结果进行了说明;最后,基于以往的探讨,给出了本文的研究思路与方法,并对所要用到的相关理论进行了详细地描述;继而构建了新的 BO-CNN-LSTM 股票价格预测组合模型.本文的主要工作总结如下:

(1) 数据预处理.首先分析了数据的缺失值处理、归一化处理.其次,采用皮尔逊相关系数衡量了 23 个股价指标之间的相关性,通过热力图直观发现各指标间存在较强的相关性.为了消除变量间的冗余信息,提高后续计算效率,利用 PCA 方法对多个股价指标进行降维,结合碎石图和特征值计算结果来看,前 3 个主成分可以有效反映原始数据的本质特征,于是将前 3 个主成分构成的 3 维数据替代原始的 23 维数据作为模型的输入.最后,结合股价指标对选取出来的 3 个主成分进行了解释.

(2) 搭建 BO-CNN-LSTM 股价预测模型.首先,基于 CNN 的卷积层和池化层能够有效地提取数据特征并降低数据维度,以及 LSTM 在缓解梯度消失的基础上具备较强的长期时序特征提取的两大优势,提出了 CNN-LSTM 组合模型,同时对模型的搭建过程进行了详细的说明.其次,对 BO 算法的优化目标函数以及优化算法进行了确定,其中以测试集 RMSE 为优化目标函数, TPE 构建优化算法;同时给出了 CNN-LSTM 模型需要优化的超参数及其范围域.最后,给出了经 BO 算法优化后的模型超参数结果,以及用来衡量模型性能的评估指标,从而完成了 BO-CNN-LSTM 模型的构建.

(3) 实证分析.选取了中国具有代表性的沪深 300 指数,作为 BO-CNN-LSTM 模型的实验对象,并展开了两个方面的实验来验证模型的有效性.实验一:将 CNN 模型、LSTM 模型、未优化的 CNN-LSTM 模型作为基准模型进行对比,并

分析了各模型的预测效果. 实验结果表明随着时间的延长, BO-CNN-LSTM 模型的预测精度仍然能够保持稳定, 对股价波动趋势预判基本正确, 且模型表现最优, 这说明 BO-CNN-LSTM 模型能够有效地处理短期股价预测问题. 实验二: 为了进一步验证 BO-CNN-LSTM 模型在股票价格预测上的优越性, 将当前使用较多的参数优化算法 GA 和 PSO 分别搜索 CNN-LSTM 模型的超参数, 具体介绍了 GA、PSO 算法的参数设置以及 CNN-LSTM 模型超参数优化的范围域. 然后对 BO-CNN-LSTM 模型、GA-CNN-LSTM 模型以及 PSO-CNN-LSTM 模型进行训练, 统计了各模型在测试集上预测结果的 RMSE 和模型训练时长. 实验结果显示, 当训练数据量较小时, 可以用 GA 和 PSO 算法优化模型超参数, 以达到较小的预测误差; 而对于训练数据量较大时, 可以用 BO 算法优化模型超参数, 以节省时间. 通过两个对比实验结果表明, 本文搭建的 BO-CNN-LSTM 模型在股票价格预测精度与耗时上具有一定的优势.

## 6.2 研究展望

本文提出的 BO-CNN-LSTM 组合模型, 在沪深 300 数据集上进行的股票价格预测的效果较好, 但仍存在一些不足和可提升的空间. 在未来的研究中, 可以从以下几个方面改进:

(1) 收集更广泛的数据集, 如上证指数、深证指数等, 探索 BO-CNN-LSTM 模型在其他数据集上的预测可能性.

(2) 探索更多股价预测的可能影响因素, 使模型不断学习进化, 灵活应对市场中各种复杂情况. 下一步, 通过收集股票新闻、交易网站上的网友评论等文本信息, 运用自然语言处理技术将文本信息转换成结构化数据, 再利用统计分析技术对数据进行整理并将其作为模型特征的输入, 以获得更多关于股票市场的有效信息.

## 参考文献

- [1] Ani D, Ionel V C. Effectiveness of various types of analysis in investment decisions on the stock market: fundamental analysis and technical analysis[J]. Ovidius University Annals, Economic Sciences Series, 2012, xii(1): 1114-1119.
- [2] 李刚. 技术分析在股市中的意义[J]. 中小企业管理与科技, 2010(22): 73-74.
- [3] 李金林, 金钰琦. 中国股票 A 股市场随机游走模型的检验[J]. 北京工商大学学报: 自然科学版, 2002, 20(4): 49-52.
- [4] Bollerslev T. Generalized autoregressive conditional heteroskedasticity[J]. Journal of Econometrics, 1986, 31(3): 307-327.
- [5] Zhang G, Zhang X, Feng H. Forecasting financial time series using a methodology based on autoregressive integrated moving average and Taylor expansion[J]. Expert Systems, 2016, 33(5): 501-516.
- [6] 李斌, 邵新月, 李玥阳. 机器学习驱动的基本面量化投资研究[J]. 中国工业经济, 2019, 377(8): 61-79.
- [7] 易丹辉, 王燕. 应用时间序列分析(第 5 版)[M]. 北京: 中国人民大学出版社, 2019: 1-7.
- [8] White H. Economic prediction using neural networks: The case of IBM daily stock returns[C]. Proceedings of International Conference on Neural Networks. IEEE, 1988: 451-158.
- [9] Gencay R. Non-linear prediction of security returns with moving average rules[J]. Journal of Forecasting, 1996, 15(3): 165-174.
- [10] 吴微, 陈维强, 刘波. 用 BP 神经网络预测股票市场涨跌[J]. 大连理工大学学报, 2001, 41(1): 9-15.
- [11] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003, 50(17): 159-175.
- [12] 孙延凤, 梁艳春, 张文力, 等. RBF 神经网络最优分割算法及其在股市预测中的应用[J]. 模式识别与人工智能, 2005, 18(03): 374-379.
- [13] 王波, 张凤玲. 神经网络与时间序列模型在股票预测中的比较[J]. 武汉理工大学学报: 信息与管理工程版, 2005, 27(6): 69-73.
- [14] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
- [15] 赵申剑, 黎戡君, 符天凡, 等. 深度学习[M]. 北京: 人民邮电出版社, 2017: 282-300.
- [16] Nelson D M Q, Pereira A C M, De Oliveira R A. Stock market's price movement prediction with LSTM neural networks[C]. Proceedings of International Joint Conference on Neural Networks. IEEE, 2017: 1419-1426.
- [17] 邓凤欣, 王洪良. LSTM 神经网络在股票价格趋势预测中的应用——基于美港股票市场个股数据的研究[J]. 金融经济: 下半月, 2018(7): 96-98.
- [18] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial

- market predictions[J]. *European Journal of Operational Research*, 2018, 270(2): 654-669.
- [19] 彭燕, 刘宇红, 张荣芬. 基于 LSTM 的股票价格预测建模与分析[J]. *计算机工程与应用*, 2019, 55(11): 209-212.
- [20] Lanbouri Z, Achhab S. Stock market prediction on high frequency data using long-short term memory[J]. *Procedia Computer Science*, 2020, 175: 603-608.
- [21] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: A convolutional neural-network approach[J]. *IEEE Transactions on Neural Networks*, 1997, 8(1): 98-113.
- [22] Persio D 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(2016): 403-413.
- [23] Gudelek M U, Boluk S A, Ozbayoglu A M. A deep learning based stock trading model with 2-D CNN trend detection[C]. *Proceedings of Symposium Series on Computational Intelligence*. IEEE, 2017: 1-8.
- [24] 陈祥一. 基于卷积神经网络的沪深 300 指数预测[D]. 北京: 北京邮电大学, 2018: 1-70.
- [25] Tang W, Long G, Liu L, et al. Rethinking 1D-CNN for time series classification: A stronger baseline[J]. *ArXiv preprint arXiv: 2002.10061*, 2020: 1-7.
- [26] Chen W, Jiang M, Zhang W G, et al. A novel graph convolutional feature based convolutional neural network for stock trend prediction[J]. *Information Sciences*, 2021, 556(1): 67-94.
- [27] Rather A M, Agarwal A, Sastry V N. Recurrent neural network and a hybrid model for prediction of stock returns[J]. *Expert Systems with Applications*, 2015, 42(6): 3234-3241.
- [28] Liu S, Zhang C, Ma J. CNN-LSTM neural network model for quantitative strategy analysis in stock markets[C]. *Proceedings of Neural Information Processing: 24th International Conference*. Springer International Publishing, 2017: 198-206.
- [29] 罗向龙, 李丹阳, 杨戡, 等. 基于 KNN-LSTM 的短时交通流预测[J]. *北京工业大学学报*, 2018, 44(12): 1521-1527.
- [30] Kim T, Kim H Y, Hernandez Montoya AR. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data[J]. *PloS One*, 2019, 14(2): e0212320.
- [31] 刘恒, 侯越. 贝叶斯神经网络在股票时间序列预测中的应用[J]. *计算机工程与应用*, 2019, 55(12): 225-229.
- [32] 乔若羽. 基于神经网络的股票预测模型[J]. *运筹与管理*, 2019, 28(10): 132-140.
- [33] 包振山, 郭俊南, 谢源, 等. 基于 LSTM-GA 的股票价格涨跌预测模型[J]. *计算机科学*, 2020, 47(6A): 467-473.
- [34] 胡聿文. 基于优化 LSTM 模型的股票预测[J]. *计算机科学*, 2021, 48(6A): 151-157.
- [35] Shahriari B, Swersky K, Wang Z, et al. Taking the human out of the loop: A review of bayesian optimization[J]. *Proceedings of the IEEE*, 2015, 104(1): 148-175.
- [36] Abdi H, Williams L J. Principal component analysis[J]. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2010, 2(4): 433-459.
- [37] Mirjalili S, Mirjalili S. Genetic algorithm[J]. *Evolutionary Algorithms and Neural Networks: Theory and Applications*, 2019, 780: 43-55.
- [38] Kennedy J, Eberhart R. Particle swarm optimization[C]. *Proceedings of ICNN'95-International Conference on Neural Networks*. IEEE, 1995, 4: 1942-1948.

- [39] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.
- [40] Nabipour M, Nayyeri P, Jabani H, et al. Deep learning for stock market prediction[J]. Entropy, 2020, 22(8): 840-862.
- [41] Lu W, Li J, Li Y, et al. A CNN-LSTM-based model to forecast stock prices[J]. Complexity, 2020, 2020(1): 1-10.
- [42] 崔佳旭, 杨博. 贝叶斯优化方法和应用综述[J]. 软件学报, 2018, 29(10): 3068-3090.
- [43] 李亚茹, 张宇来, 王佳晨. 面向超参数估计的贝叶斯优化方法综述[J]. 计算机科学, 2022, 49(6A): 86-92.
- [44] 王星. 统计学习导论: 基于 R 应用[M]. 北京: 机械工业出版社, 2015: 92-94.