



# 南京大學

## 研究生畢業論文 (申請碩士專業學位)

論 文 題 目         基于注意力機制和列表排序算法的

        股票推薦排序研究

作 者 姓 名                         張雅楠

學 科、專 業 方 向                         工程碩士 (計算機技術)

研 究 方 向                         人工智慧

指 導 教 師                         申富饒 教授

2022 年 5 月 23 日

学 号：MF1933122

论文答辩日期：2022 年 5 月 17 日

指 导 教 师： (签字)

# **Stock Recommendation Sorting Based on Attention Mechanism and Listwise Sorting Algorithm**

by

**Zhang Yanan**

Supervised by

**Shen Furao**

A dissertation submitted to  
the graduate school of Nanjing University  
in partial fulfilment of the requirements for the degree of

MASTER

in

Computer Technology

Department of Computer Science and Technology  
Nanjing University

May 23, 2022



# 南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于注意力机制和列表排序算法的股票推荐排序研究

工程硕士（计算机技术） 专业 2019 级硕士生姓名：张雅楠  
指导教师（姓名、职称）：申富饶 教授

## 摘 要

随着社会的发展，股票交易成为一种常见的投资策略。股票作为一个极其复杂的非线性时序系统，其变化不仅与历史数据相关，市场状况、社会政策和投资者情绪等因素都会影响股价走势。因此如何使用深度学习模型，综合考虑市场信息对股票进行预测分析，规避投资风险获得更高收益，成为当下最具有挑战性的课题之一。

当前常见的股票投资方法，大多是针对单只股票进行预测分析，这种预测方式没有直接从利润的角度优化投资目标，并且忽略了市场内部不同股票的相互影响。因此本文使用针对候选股票的推荐排序问题解决股票投资问题，综合考虑候选股票间的相互影响，将预测涨幅越大的股票排在最终序列的越前端，作为最终的推荐结果。

针对上述问题，本文完成了以下的工作：

1. 本文提出了基于注意力机制的时序特征处理算法 Attention Feature Convolution and Gated Recurrent Unit (AFC\_GRU)。该算法将特征卷积和 GRU 模块通过注意力机制相结合，缓解了长期历史遗忘问题，更好地对股票时序特征进行融合处理，得到最终的表示结果。
2. 本文提出了基于注意力解码器的 Listwise 排序算法 Attention Encoder-Decoder Ranking Model (AEDRM)。该算法使用 Decoder 打分模块，并行地对候选项进行综合打分，考虑到不同候选项之间的影响；并且使用 Mask 方法修改 loss 计算方法，加强模型对序列前端候选项的约束效果。
3. 本文设计实现了一个股票分析推荐系统。系统采用 Django 框架和 Bootstrap 框架搭建系统，系统内使用提出的基于注意力机制的时序特征处理算法 AFC\_GRU 和注意力解码器的 Listwise 排序算法 AEDRM，完成股票的数据分析和推荐排序功能。

实验表明，本文提出的算法能够有效地对股票特征进行融合表示，并且能够有效地进行候选股票推荐排序，从而达到提高收益率的目的。同时股票分析推荐系统的实现，方便了本文提出算法的实际落地使用，也能够便捷地为投资者们展示分析结果。

**关键词：** 股票分析, 时序处理, 排序算法, 注意力机制

## 南京大学研究生毕业论文英文摘要首页用纸

THESIS: Stock Recommendation Sorting Based on  
Attention Mechanism and Listwise Sorting Algorithm

SPECIALIZATION: Computer Technology

POSTGRADUATE: Zhang Yanan

MENTOR: Shen Furao

### **Abstract**

With the development of society, stock trading has become a common investment strategy. As an extremely complex non-linear time-series system, stock changes are not only related to historical data, but also factors such as market conditions, social policies, and investor sentiment can affect stock price movements. Therefore, how to use deep learning models to predict and analyze stocks considering market information, and avoid investment risks to obtain higher returns has become one of the most challenging topics nowadays.

Most of the current common stock investment methods are prediction analyses for a single stock, which does not optimize the investment objective directly from the perspective of profit and ignores the mutual influence of different stocks within the market. Therefore, this paper solves the stock investment problem using a recommendation ranking problem for candidate stocks, which takes into account the interactions among the candidate stocks and ranks the stocks with higher predicted increases at the front end of the final series as the final recommendation result.

To address the above issues, the following work is done in this paper:

1. We propose AFC\_GRU, a temporal feature processing algorithm based on the attention mechanism. this algorithm combines feature convolution and GRU modules through the attention mechanism to alleviate the long-term history forgetting problem and better fuse the stock temporal features to obtain the final representation results.
2. We propose attention decoder-based Listwise ranking algorithm AEDRM. the algorithm uses the Decoder scoring module to synthetically score the candidates in

parallel, taking into account the influence between different candidates; and uses the Mask method to calculate the loss to strengthen the model's constraint effect on the candidates at the front end of the sequence.

3. A stock analysis recommendation system is designed and implemented. The system uses the Django framework and Bootstrap framework to build the system, and the proposed attention mechanism-based temporal feature processing algorithm AFC\_GRU and attention decoder's Listwise sorting algorithm AEDRM are used within the system to complete the data analysis and recommendation sorting functions of stocks.

Experiments show that the algorithm proposed in this paper can effectively fuse the stock features and can effectively rank the candidate stock recommendations to achieve the goal of improving returns. At the same time, the implementation of the stock analysis recommendation system facilitates the practical implementation of the algorithm proposed in this paper, and can also conveniently present the analysis results for investors.

**keywords:** Stock analysis, Timing processing, Sorting algorithm, Attention mechanism

# 目 次

目 次 .....	v
插图清单 .....	ix
附表清单 .....	xi
<b>1 绪论 .....</b>	<b>1</b>
1.1 研究背景与意义 .....	1
1.2 研究现状与挑战 .....	3
1.3 研究内容 .....	5
1.4 论文纲要 .....	5
<b>2 相关工作 .....</b>	<b>7</b>
2.1 时序预测任务 .....	7
2.2 股票预测 .....	8
2.2.1 传统股票预测模型 .....	9
2.2.2 基于机器学习的股票预测模型 .....	10
2.2.3 基于深度学习的股票预测模型 .....	11
2.3 排序算法 .....	13
2.3.1 Pointwise .....	13
2.3.2 Pairwise .....	14
2.3.3 Listwise .....	14
2.3.4 性能指标 .....	15
2.4 问题表述 .....	16
2.5 本章小结 .....	18
<b>3 基于注意力机制的时序特征处理算法 .....</b>	<b>21</b>
3.1 注意力机制 .....	21
3.1.1 基础注意力机制 .....	21

---

3.1.2 键值对注意力机制 .....	23
3.2 时序循环网络 .....	24
3.3 基于注意力机制的时序处理算法设计 .....	26
3.3.1 ALSTM .....	26
3.3.2 特征卷积结构 .....	28
3.3.3 AFC_GRU 模型算法流程 .....	29
3.4 实验与分析 .....	30
3.4.1 实验设置 .....	31
3.4.2 实验结果 .....	34
3.4.3 赢利能力测试 .....	35
3.4.4 消融实验 .....	37
3.5 本章小结 .....	40
<b>4 基于注意力解码器的列表排序算法 .....</b>	<b>41</b>
4.1 当前排序方法的局限性 .....	41
4.2 当前排序算法损失函数的局限性 .....	43
4.3 基于注意力解码器的排序算法设计 .....	45
4.3.1 Decoder 模块 .....	45
4.3.2 Loss 设计 .....	48
4.3.3 算法流程 .....	49
4.4 实验与分析 .....	50
4.4.1 实验设置 .....	50
4.4.2 实验结果 .....	53
4.4.3 赢利能力测试 .....	54
4.4.4 消融实验 .....	54
4.5 本章小结 .....	57
<b>5 股票分析推荐系统 .....</b>	<b>59</b>
5.1 相关背景 .....	59
5.2 需求分析 .....	60
5.3 系统架构 .....	60
5.4 系统实现 .....	61
5.4.1 数据收集处理 .....	61

目 次	vii
5.4.2 后端算法实现 .....	62
5.4.3 前端交互设计 .....	63
5.5 本章总结 .....	65
<b>6 总结与展望 .....</b>	<b>67</b>
6.1 工作总结 .....	67
6.2 未来展望 .....	68
参考文献 .....	<b>69</b>
致 谢 .....	77
简历与科研成果 .....	79
学位论文出版授权书 .....	<b>81</b>



# 插图清单

1-1	不同策略在股票预测与利润分析示例 .....	4
2-1	时间序列预测任务分析 .....	7
2-2	传统时间序列模型及分析方法 .....	9
2-3	机器学习、深度学习算法在股票预测任务中的应用 .....	12
2-4	股票预测排序任务模型结构 .....	17
3-1	键值对注意力 .....	23
3-2	循环神经网络 RNN 网络结构 .....	24
3-3	Attentive LSTM 网络结构 <sup>[1]</sup> .....	27
3-4	特征卷积计算流程 .....	28
3-5	基于时空卷积注意力机制的时序特征处理 .....	29
3-6	AFC_GRU 算法内注意力分布结果示例 .....	36
3-7	在上证 50 数据集上，不同投资模型在各个回测方法下的结果比较 ·	37
3-8	在 NASDAQ 数据集上，不同投资模型在各个回测方法下的结果 比较 .....	37
4-1	DLCM 模型结构图 <sup>[2]</sup> .....	42
4-2	PRM 模型结构图 <sup>[3]</sup> .....	43
4-3	Decoder 模块的结构 .....	46
4-4	遮蔽自注意力方法中的遮蔽实现 .....	46
4-5	采用 Attention Decoder 的 Listwise 排序算法 .....	47
4-6	mask 方法 .....	48
4-7	不使用 WarmUp 和使用 WarmUp 策略时学习率变化情况 .....	52
4-8	在上证 50 数据集上，不同投资模型在各个回测方法下的结果比较 ·	54
5-1	股票预测排序系统架构 .....	61
5-2	模型训练与使用框架 .....	63
5-3	股票数据查询页面实现 .....	64

---

5-4	股票预测排序页面实现 .....	64
5-5	股票预测排序新增模型功能实现 .....	65
5-6	股票投资策略回测分析页面实现 .....	65

# 附表清单

2-1	不同排序方法的对比 .....	15
2-2	上海证券市场中 sh.600000 浦发银行在 2020 年 12 月 01 日至 2020 年 12 月 31 日内 23 个交易日的原始数据特征 .....	18
3-1	描述股票每日趋势的特征 .....	32
3-2	在上证 50 数据集上, 不同时序特征处理方法的结果对比 .....	34
3-3	在 NASDAQ 数据集上, 不同时序特征处理方法的结果对比 .....	35
3-4	卷积核大小对比实验结果 .....	38
3-5	Timestep 时间步调节对比实验结果 .....	39
3-6	特征连接方式对比实验结果 .....	39
4-1	在上证 50 数据集上, 不同排序方法的结果对比 .....	53
4-2	在 NASDAQ 数据集上, 不同排序方法的结果对比 .....	54
4-3	不同编码器个数设置下, 模型的结果对比 .....	55
4-4	不同损失函数训练情况下, 模型的结果对比 .....	56
4-5	不同验证集使用方法下, 模型的结果对比 .....	57



# 第一章 绪论

本章主要阐述了股票分析问题的研究背景和意义，总结了目前股票分析预测任务的主要方法和技术手段，最后说明了本文的研究内容与创新点，并对文章结构进行了介绍。

## 1.1 研究背景与意义

股票市场是现代金融体系的重要组成部分，在很大程度上反应了社会发展的情况。随着我国经济的快速发展，股票买卖已经成为一种常见的投资策略，受到大批投资者的青睐。投资者们通过在不同的时间点进行买入和卖出操作，利用两次交易时的股价差异，获取相应的收益。为了得到更好的收益情况，人们都希望能够在股票的价格低点处进行买入，在价格最高点处进行卖出。这也意味着如何准确地预测和估计股票的未来走势，如何在投资过程中降低风险值，是投资者们最关注的课题。除此之外，股票预测不仅是个体投资者最重要的投资策略之一，金融、私募行业也一直对成功预测股票涨跌情况有着极大的兴趣。由于股票市场是一个极其复杂而庞大的非线性时序系统，市场内不同股票的价格变化受到诸多因素的影响，例如国内外经济情况、世界政局波动和社会新闻等因素，并且市场中同类股票和竞争性股票的变化趋势之间也存在相关性。因此，如何有效利用市场信息和历史数据对股票进行分析预测，降低投资风险，提高买卖收益，成为当下最具有挑战性的课题之一。

1970年美国经济学家尤金·法玛提出了有效市场理论，该理论的定义为“如果在一个证券市场中，价格完全反映了所有可获得的信息，每一种证券的价格都永远等于其投资价值，那么就称这样的市场为有效市场”<sup>[4]</sup>。在有效市场理论下，股票市场的价格是无法预测的。该理论认为任何有价值的信息已经准确及时地反映在股票的价格走势上了，股价中也包含了企业的当前价值和未来的预期价值，在没有任何市场操作的情况下，投资者是不可能通过分析历史价格来获得超过市场平均水平的额外利润。基于这一假设，学者们提出了随机游走模型和著名的布莱克-斯科尔斯公式等经典理论，认为股价的短期变动不可

预测。尽管对于股票市场是否可预测这一问题仍有争议<sup>[5-7]</sup>。然而，越来越多的实验证明，金融市场的发展规律在一定程度上是可预测的，并且越来越多的股票预测算法被提出，也激发了对股票预测技术的进一步探索<sup>[8-12]</sup>。常见的股票分析和预测方法一般可以分为两大类：

### 1. 基本分析法

基本分析法是使用计量分析或统计方法对过去市场走势进行分析，结合宏观经济、行业发展、企业经营状况，分析导致股票价格波动的一般规律。基本分析法利用不同的因素，如公司当前的经济环境和财务状况来估计其股票价值，不同的关键比率，如单股股利、市盈率、市净率等，也可以用来判断公司的财务状况和股票的价值。这种方法侧重于研究股票的长期投资价值<sup>[13]</sup>。不过这类方法通常需要专业人士对影响股价的各种因素进行分析解读，对普通投资者而言并不友好。

### 2. 技术分析法

技术分析认为只需要关注股票历史数据的变化情况即可以预测未来涨跌趋势，通过对股票历史价格和交易量数据分析，预测未来波动走向。相比于基本分析法，技术分析法则重点分析市场活动产生的数据，如数量和价格。目前的技术分析法建立在“股价遵循趋势运动”、“市场行为反应一切信息”、“历史很可能会重演”三大假设基础上<sup>[14]</sup>。

这两类方法各有所长，基本分析法从宏观的角度进行分析，技术分析法从量化分析的角度进行判断。通常，投资者们会结合两种方法进行分析，利用各自的优势共同为股票投资提供帮助。

随着中国股票市场的不断发展和完善，其在推动国民经济发展中起着非常重要的作用，越来越多的人和机构也参与到股票投资中。合理有效的股票分析预测，可以为投资者提供更全面的投资分析与参考结果，从而降低风险，帮助投资者获得更高收益，对市场而言，投资者们的买入卖出策略有助于增加市场流通性，从而促进市场活跃度。合理的股票投资也能够为国家经济的发展做出贡献。因此，在综合考虑了股票预测任务的意义，并对现有的技术方法分析后，我们开展了本文的研究。

## 1.2 研究现状与挑战

在股票分析问题中，最广泛使用的方法是基于对过去市场走势数据分析的传统统计方法<sup>[15]</sup>。在早期，股票预测的传统方法是通过专家分析等手段，运用简单的线性数学模型对股票数据进行处理，这些方法通常基于“时间序列是由线性过程产生的”这一假设<sup>[16]</sup>。但是由于股票市场的复杂性和不确定性，并且伴随着数据量增加和任务需求提高的影响，这类简单的线性模型的自身局限性越来越明显。之后，学者们开始尝试使用和构建更好的非线性模型来进行股票预测研究，将支持向量机 (Support Vector Machines, SVM)<sup>[17]</sup>、神经网络 (Neural Networks, NNs) 等方法运用到股票预测中。任务的主要目的可以分为预测未来股票价格的回归问题和预测未来趋势的分类问题<sup>[18]</sup>。现在基于机器学习的解决方案已经显现出其提供更准确股票预测结果的能力，并成为现代定量投资系统中不可或缺的组成部分。

随着计算机硬件的更新和存储计算能力的增强，深度学习逐渐被应用在各个领域，成为当前学者们重点关注的解决方法之一。因此越来越多的学者开始使用深度学习解决股票分析预测问题。2012年，Maknickien<sup>[20]</sup>使用长短期记忆模型 (Long Short-Term Memory, LSTM)<sup>[21]</sup>来替代前馈神经网络，提升了预测汇率的性能。Catalin等<sup>[22]</sup>利用LSTM和卷积神经网络 (Convolutional Neural Network, CNN)分别构建了股票预测模型，并根据预测结果进行买卖。Kim<sup>[23]</sup>将LSTM与CNN模型结合，从时间序列和股票数据图像两个角度综合分析股票数据特征，预测未来走势。与传统的机器学习方法相比，深度学习中能够自动地对特征进行筛选，减少了人工选择特征的难度，提升了模型的适应性，但是深度学习这样的“黑盒”运算方法可解释性较差，得到的结果很难找到对应的经济学理论支撑，这也是当前人们使用深度学习进行股票预测的顾虑之一。

这些比较常见的股票预测方式都是针对单个股票来进行分析，根据股票历史时间段内的特征值，例如每天的开盘价、最高价、最低价、收盘价、成交量和成交额等，从中提取出有效的走势信息，并以此为依据进行下一天涨跌幅的预测。这种策略重点关注某只特定股票的历史数据特征，并以此为依据预测它之后一天或一段时间的涨跌情况。一些学者为了能够得到更加准确的预测结果，希望从不同的数据进行股票特征抽取，比如可以依据当前大盘趋势、股票的相关新闻，政府的相关文件和相关网友的评论中进行文本分析，抽取出的信息，并以此作为股票的特征之一<sup>[24,25]</sup>。

虽然在特征选择和处理上存在差异，但是上述方法们的目标都是预测未来涨跌幅值或涨跌情况，并以此为依据进行后续的买卖操作。而这样也导致上述方法们都没有直接从利润的角度优化投资目标。例如在图 1-1 中，我们对股票预测目前的方法在模型表现和利润上进行了对比分析。首先我们假定有 4 只股票，并给出了它们真实的涨跌幅值。在利润计算时，我们采用当天买入预测涨跌幅值最大（或上涨概率最大）的股票，次日卖出该股票的策略，利润值即为真实涨跌幅值的大小。分析数据可以发现，在回归策略中，虽然策略 R1 的均方误差（Mean Squared Error, MSE）值相比于策略 R2 更差，但是在使用 Top1 买入卖出策略时，策略 R1 的利润大于策略 R2。在分类策略中，虽然策略 C1 的预测准确度（Accuracy, ACC）远高于策略 C2，但策略 C1 的利润低于策略 C2。这说明仅仅依据模型预测表现选择模型，并不一定能够保证得到更高的利润。

股票		S1	S2	S3	S4	模型表现	利润 (Top-1)
真实涨跌幅		+50 ↑	+30 ↑	-10 ↓	-35 ↓		
预测涨跌幅 (回归问题)	R1	60	10	10	-45	MSE	250
	R2	25	35	-10	-25		<b>187.5</b>
预测涨跌概率 (分类问题)	C1	0.65 ↓	0.85 ↑	0.5 ↓	0.75 ↑	ACC	70%
	C2	0.8 ↑	0.65 ↓	0.45 ↓	0.7 ↑		50%

图 1-1: 不同策略在股票预测与利润分析示例

并且，上述方法们忽略了同一时间段内不同股票的相互影响，将每只股票视为独立于其他股票。这种独立假设与股票市场中的相互影响作用不一致，这不可避免地限制了它们实现更好的股票预测性能的能力。通常而言，在某段时间内，同一类型的股票可能拥有相同的走势，这说明股票之间复杂的关系中存在丰富的信号。例如在新冠疫情影响下，旅游和交通等行业的公司股价出现下跌，而医疗保健相关股票的股价则出现上涨。如果股票预测模型能够发现这种类似的规律，并且将这种规律应用在股票分析投资策略上，那么投资者就能够更加有依据地进行操作。

基于目前投资分析模型的局限性，在本文我们使用学习排序的方法解决股票投资问题。整个方法由时序特征处理和学习排序两部分组成，使得模型在分析时序特征的基础上，也考虑到了不同股票之间的相互影响。排序得到的序列中包含了预测涨跌幅值的相对大小信息，并以此为后续投资的依据。如何对股票的历史特征进行融合，得到更好的股票特征表示，是本篇论文讨论的重点之

一。另外相比于其他预测任务，我们希望能够预测相对走势时，考虑到不同候选股票的相关影响，并以此为依据进行排序。那么如何更好地捕获相互之间的关系从而进行推荐排序，也是本文要讨论的重点之一。

## 1.3 研究内容

本文围绕着股票投资问题进行研究，利用学习排序的方法对股票数据进行推荐排序，并且结合股票数据的特性，研究如何更好地对股票时序特征进行融合，以及如何设计算法充分提取到候选股票之间的相互影响，进行股票涨跌幅排序，得到最终的推荐结果。在此基础上，本文提出了一个时序特征处理算法和一个排序算法针对性地解决上述问题，并将其应用在实际的股票分析推荐系统中。本文的主要内容和创新点如下：

1. 本文提出了基于注意力机制的时序特征融合算法 AFC\_GRU。该算法将 GRU 与特征卷积网络通过注意力机制连接，有效缓解了传统循环神经网络中的历史遗忘现象，更有利于对股票历史阶段的特征进行融合，捕获到有用的历史信息。
2. 本文提出了一种基于注意力解码器的列表排序算法 AEDRM。该算法在打分阶段采用注意力机制的 Decoder 模块进行综合打分，并根据排名分数得到最终的推荐股票序列，更全面地捕获到不同股票之间的相互影响关系。同时，本文设计了一个 Mask Attention Rank (MAR) 损失函数进行模型训练，增加模型对真实序列前端候选项的关注程度，更加符合真实投资场景。
3. 本文搭建了一个股票分析推荐系统。该系统使用本文提出的时序特征处理和排序算法为用户提供数据获取和分析、股票推荐排序和回测的功能，实现算法的落地使用，方便用户对股票市场进行投资分析。

## 1.4 论文纲要

本文共分为六个章节，其主要内容分别如下：

第一章为绪论，对本论文关注的股票分析投资问题进行了介绍，分析了当前主要股票分析方法存在的问题，并以此为出发点，介绍了本文的创新点和文章结构。

第二章为相关工作，对相关的论文和方法进行了回顾，主要涉及到股票预测问题、时序特征处理和排序算法，并对本文关注的股票预测排序问题进行了符号表述。

第三章详细解释了本论文提出的时序特征处理方法 AFC\_GRU，使用时间卷积提取查询特征，并将其与 GRU 的输出进行注意力机制计算，从而得到股票特征。通过与其他不同的方法进行对比，验证了本文提出的时序特征处理方法的有效性。

第四章介绍了本论文提出的基于注意力解码器的列表排序方法 AEDRM。算法在排序过程中使用 Attention Decoder 的方法，考虑不同股票之间的影响，并且使用 Mask Attention Rank 损失函数，提高对预测序列前端的要求。

第五章主要介绍了股票分析推荐系统的搭建过程，分别从系统需求分析、系统架构、以及最终的实现展示进行了介绍。

第六章对整个论文进行了总结和回顾。

## 第二章 相关工作

股票数据作为代表性的时间序列特征数据，其预测问题可以被看做为时序预测问题。因此本章介绍了相关的时序预测任务，之后对当前的股票预测任务和排序算法进行介绍与回顾，最后在问题表述中对本文讨论的任务进行了说明与定义。

### 2.1 时序预测任务

时序预测任务一直是学术界关注的重点，序列预测任务的目的是希望在已知历史特征变量和历史相关变量的情况下，预测未来的特征变量值。如图 2-1 所示，时间序列预测任务可以被形式化表示为，设当前时刻为  $t$ ，我们利用  $t$  之前的时间序列  $X_{1:t}$  内的特征变量  $F_{1:t}^{back}$  和相关变量值  $C_{1:t}^{back}$  来预测未来  $T$  个时间点的特征值  $F_{t+1:t+T}^{fore}$ ，具体的公式表示如公式 (2-1) 所示。

$$F_{t+1:t+T}^{fore} = f(F_{1:t}^{back}, C_{1:t}^{back}) \quad (2-1)$$

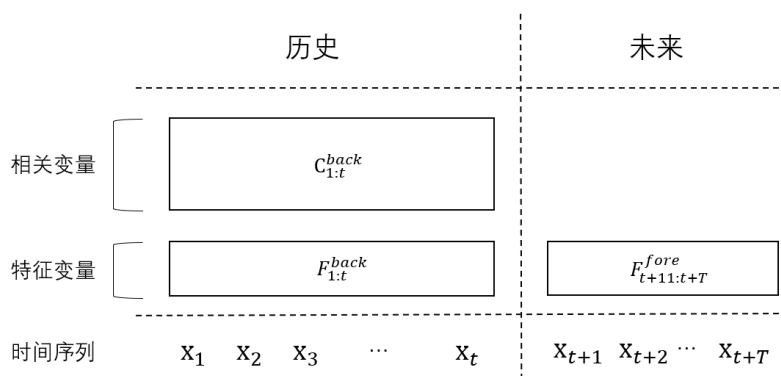


图 2-1: 时间序列预测任务分析

时间序列预测有着非常广泛的使用，例如在文本翻译时，我们可以根据之前已输入的文本信息，预测之后的单词，并以此为翻译的依据。另外，在天气预测时，之前的天气状态和相关信息，都可以作为后续预测的依据。这类问题

涉及到的特征值具有时间维度上的连续性或周期性，因此在处理这类数据进行预测时，我们希望模型能够较好地捕获到其在时间维度上的变化规律。

常见的时序特征处理算法主要分为三大类。第一类是基于统计学的传统方法，包括随机游走法（Random Walk）、移动平均法（Moving Average, MA）、整合自回归移动平均法（Autoregressive Integrated Moving Average, ARIMA）<sup>[26]</sup>和指数平滑法（Exponential Smoothing, ETS）<sup>[27]</sup>等。第二类是使用传统机器学习模型，例如条件随机场（conditional random field, CRF）、梯度提升决策树（Gradient Boosting Decision Tree, GBDT）等。第三类是基于深度学习的方法，模型结构相比于机器学习模型网络更深更复杂。其中的循环神经网络（Recurrent Neural Network, RNN）能够将历史信息在时间序列中进行传递，具有更强的时序特性，所以常被用来解决时序问题。长短期记忆网络（Long Short-Term Memory, LSTM）<sup>[28]</sup>、门控循环单元（Gated Recurrent Unit, GRU）<sup>[29]</sup>作为RNN的变形结构，很好得解决了RNN中存在的历史遗忘和梯度消失等问题，也得到了广泛的使用。另外随着Transformer<sup>[30]</sup>的广泛使用，Attention和Transformer技术也逐渐被应用在时序预测问题上，另外也有一些学者尝试利用CNN结构提取特征，例如时序卷积网络（Temporal Convolutional Network, TCN）<sup>[31]</sup>采用空洞卷积结合因果卷积的网络结构，使用因果卷积对 $t$ 时刻及其之前的输入进行处理得到 $t$ 时刻的输出。空洞卷积的使用在时间复杂度不变的情况下，引入了更长历史信息，提升预测结果。

时序特征处理与时序预测任务息息相关。时序预测可以认为是从时序数据中进行特征处理和提取，再利用处理后得到的特征进行最终的预测，所以我们可以将时序预测模型前半部分的工作抽象为是在对时序特征进行处理，模型后半部分则是在完成预测任务。

## 2.2 股票预测

股票预测，是指通过对股票市场的历史和现状进行分析，得到股市发展的规律性，使用各类方法预测未来的发展前景。股票预测作为特定类别的时序预测任务，一直是金融行业中备受关注的的问题之一。众所周知，股票价格受到多种因素的影响，确切价格是很难预测的，所以大部分学者的研究重点是预测股价的走势，例如价格是否为上涨/下跌，或者价格的变化是否会超过一定的阈值，这类问题比预测准确的股价更容易实现。因此，股票预测任务的目的是可以

分为两类，一类是预测指定未来时间区间内股票准确涨跌幅的回归问题；另一类是预测未来股票涨跌情况的分类问题。科学有效的股票预测方法不仅能够为投资者们提供获利和投资策略的参考性指导，也可以为证券监管等机构制定稳定金融市场的政策提供基本信息。根据预测模型的构建理论，股票价格预测模型主要分为三类<sup>[32]</sup>，一类是传统预测模型，主要基于统计学和概率论进行研究，主要使用线性模型解决问题；另一类是基于机器学习，尝试使用非线性模型解决股票预测任务；第三类是基于深度学习的预测模型，使用更复杂、更深层的模型解决问题。

### 2.2.1 传统股票预测模型

传统的股票预测模型是通过统计学与概率进行预测研究。由于股票数据是一种具有强烈时间序列性质的数据，其特点表现为值之间具有强烈的时间顺序相关性，并且在一定时间内数值呈现周期性变化。因此传统时间序列模型和隐马尔科夫模型等都被用来捕捉股票数据之间的连续性特征。主要的预测模型及分析方法如图 2-2 所示。

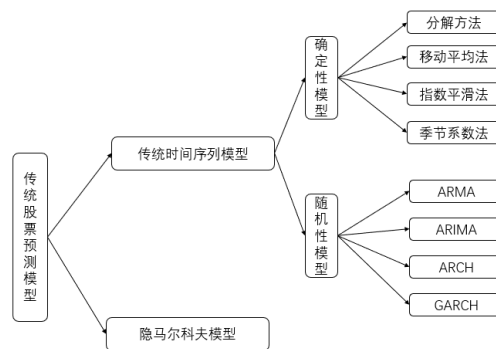


图 2-2: 传统时间序列模型及分析方法

传统时间序列模型可以分为确定性模型和随机模型。确定性模型是指时序数据会依据某种规律进行变化，数据本身具有一定的发展变化规律，模型的任务则是捕获到这类变化规律进行分析与预测。这类模型的选择过程通常需要先对数据变化规律进行分析后，获取变化规律与特点，再选择合适的模型进行构建，例如移动平均法<sup>[34]</sup>、指数平滑法<sup>[35]</sup>等。随机性模型则主要被用于时序数据会受到较多随机因子影响的情况，这类情况会使得数据的变化规律较难被预测。该类模型主要有自回归滑动平均模型（Auto Regressive Moving Average, ARMA）<sup>[35]</sup>、差分整合移动平均自回归模型 ARIMA<sup>[26]</sup>、自回归条件异方差模

型 (Autoregressive Conditional Heteroskedasticity, ARCH) [36] 等。Ariyo 等人 [37] 使用 ARIMA 模型解决股票价格预测问题, 结果表明 ARIMA 模型具有很强的短期预测能力。Babu 等人 [39] 提出了一种基于 ARIMA 和 GARCH 的混合模型, 先使用两个模型分别进行预测, 再将两个结果融合得到最终预测值, 实验证明, 对于多步提前预测, 该模型在精度和保持数据趋势方面优于传统模型。

随着隐马尔科夫模型 (Hidden Markov Model, HMM) 在分析和预测与时间相关任务上取得了成功, HMM 模型也开始被应用于股票预测任务 [40]。Gupta 等人 [41] 提出了最大后验 HMM 方法来预测第二天的股价, 最终结果表明, HMM 模型相比于其他传统模型, 简单且有效。

传统方法通常使用线性模型解决股票预测问题, 而股价受到市场、社会等因素影响, 其变化规律更加复杂, 通常为非线性的。这一数据特性限制了传统预测方法的应用。另外, 传统方法也容易受到样本数量不足、数据分布不均匀等问题影响, 进而导致无法获得较准确的预测结果。

## 2.2.2 基于机器学习的股票预测模型

股票价格在社会、市场、新闻等因素的影响下呈现非线性运动, 变化规律更加复杂。因此, 相比于传统预测模型, 基于机器学习的预测模型能够更有效地提取到数据中的发展规律, 模型的预测效果也明显提升。

Madge 等人 [42] 使用支持向量机 (Support Vector Machines, SVM) 预测股票价格, 使用 34 只科技股票的日收盘价来计算个股和整个行业的价格波动, 并以此作为 SVM 模型参数, 最终结果证明该模型在长期预测能力上表现较好。Meesad 等人 [43] 使用支持向量回归 (Support Vector Regression, SVR) 分析预测股票市场价格以及预测股市趋势。Henrique 等人 [44] 也使用 SVR 技术预测股票价格, 结果表明 SVR 具有较强的预测能力。虽然上述方法都证明了 SVM 和 SVR 模型在股票预测任务上的优势, 但是其核函数的选择难度和在大规模训练中的时间与内存空间消耗问题, 限制了支持向量机在之后预测任务的发展空间。

人工神经网络 (Artificial Neural Network, ANN) 是指由大量神经元相互连接组成的网络模型, 是一个模仿人脑神经结构的数学信息处理模型。网络中每层的神经元通过带权重的边建立连接, 从而实现数据的传递与处理。Hammad [45] 采用前馈多层神经网络搭建模型, 实现对股票价格预测的功能, 实验证明 ANN 的使用提供了快速收敛、高精度的股票价格预测能力。Varghese

等人<sup>[46]</sup>使用 ARMA，移动平均和一个 ANN 模型预测股票的未来价格。但是由于股票数据存在波动大、维度复杂等特性，浅层的 ANN 结构并不足以捕获到数据内的复杂特征与关系，从而导致模型容易出现过拟合现象。不过，随着计算机运算能力和存储能力的提升，具有更强的特征提取和分析能力的深度结构得到了广泛的关注和应用，为多个领域都带来了较大的提升。

### 2.2.3 基于深度学习的股票预测模型

随着计算机计算能力和存储能力的提升，具有更深层、更复杂结构的卷积神经网络（CNN）、循环神经网络（RNN）和前馈神经网络（FNN）等深度学习模型，在图像处理和自然语言处理等领域被广泛应用。目前，越来越多的学者也希望通过深度学习技术，来解决股票预测问题。

Majumder 等<sup>[47]</sup>分别使用 FNN 和传统线性模型预测分析了多种股票，结果表明相比于传统模型，深度学习模型中的前馈神经网络是预测股票指数的最佳算法。Sharma 等人<sup>[48]</sup>使用 LSTM 模型预测未来股票价格的随机性质，最终模型的均方误差值显著降低。Selvin 等人<sup>[49]</sup>提出了一种基于深度学习的股票价格预测模型，在非线性模型 RNN、LSTM、CNN 上使用滑动窗口方法来进行股价预测，最终证明在误差计算方面，非线性模型优于线性模型，并且 CNN 结构在处理市场突发变化情况时，具有更好的趋势预测效果。

另外部分学者也针对当前模型的缺点，提出了改进和优化方案。Feng 提出的 ALSTM 模型<sup>[1]</sup>在 LSTM 上添加了 Attention 机制，来模拟不同时间步的数据对整个序列的表现具有不同的贡献。同时该论文还利用了对抗训练来解释股票市场的随机性来学习股票走势预测，通过构造对抗样本参与训练，从而提升模型训练的稳定性。Wu 提出的 TCT 模型<sup>[50]</sup>将一种可学习的调度器引入序列学习，它可以根据模型状态和当前的训练数据自适应地选择训练辅助任务，通过利用辅助任务来提高模型在主要任务上的性能。Lin 提出的 TRA 模型<sup>[51]</sup>在网络中构建了多个预测器，通过一个路由器在不同的时间段将当前的预测任务分配到不同的预测器上得到预测的结果，从而提升了模型的泛化能力。为了解决模型缺乏处理股价随机特性能力的问题，StockNet<sup>[52]</sup>提出了一种基于高市场随机性、混沌市场信息和时间相关的股票走势预测模型。该模型利用推特内容和股票历史价格进行编码，然后利用变分运动解码器从市场信息中推断和解码潜在变量，来更好地处理市场随机性，同时该模型也使用注意力时间辅助方法，建立预测任务上的依赖相关性。深度学习模型的特征提取能力，能够有效地从

影响股价的多个因子中挑选出比较重要的因子作为后续预测的依据，但是与此同时，这样的“黑盒”选择方式，增加了模型可解释性的难度。

由于股票价格预测任务的复杂性，也有一些学者将集成学习技术使用到股票预测任务中，通过多种特征选择的方式，提升任务的精度。Khaidem 等人<sup>[53]</sup>利用随机森林分类器对股价的未来涨跌方向进行了预测，实验证明，随机森林模型在预测股票未来走势方面非常稳健。谢琪等人<sup>[54]</sup>提出了一个基于 LSTM 的集成学习分类模型，分类结果表示下一个交易日的涨跌情况，该方法采用集成策略，并行训练多个 LSTM 网络共同完成分类任务，再将多个结果汇总后得到最终的预测涨跌情况，实验显示，其预测精确率较单一的 LSTM 模型有了较大的提升。Zhang 等人<sup>[55]</sup>利用 Adaboost 建立多类别变量组合预测股票的年收益率，实验结果显示，虽然 Adaboost 算法与随机森林和决策树算法在平均错误率上效果相差不大，但是 Adaboost 算法的错误率在类别之间更加平衡。

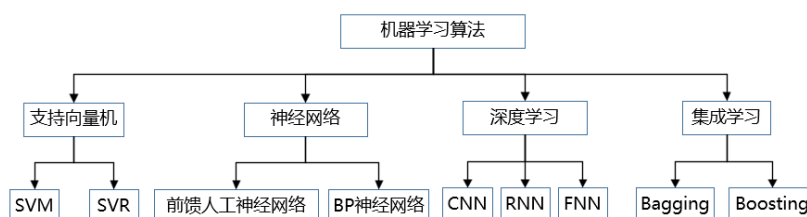


图 2-3: 机器学习、深度学习算法在股票预测任务中的应用

现有大多数的股票预测方法都是针对单只股票进行分析，而忽略了市场对股票的影响，以及市场中不同股票之间的影响。为了增加对市场的了解，一些学者也开始从互联网的相关文本数据中抽取到相关的信息作为股票的特征进行训练，这种方法会使股票的特征数据更加丰富，从而得到更加准确地结果。例如 Bollen 等人<sup>[56]</sup>认为金融决策在很大程度上是由情绪驱动的，所以他们分析社交应用中的内容，提取出包含的有积极或消极情绪的关键内容作为特征，预测股票价格的变化趋势。此外 Ding 等人<sup>[57,58]</sup>通过考虑与股票或关联公司相关的新闻事件，进而对股票走势进行预测，Li 等人<sup>[59]</sup>使用隔夜新闻数据建立股票之间的联系，提出了一个 LSTM-RGCN 模型来预测相关股票的隔夜股价走势。另外在同一时间段内，相同行业的股票或处于上下游行业的股票之间都会存在相互之间的波动联系。为了更综合考虑到股票市场内股票的相互影响，Feng 等人<sup>[60]</sup>使用学习排名的方式来解决股票预测问题，并在网络中设计了一个时态图卷积（Temporal Convolution）组件捕获股票关系的领域知识，Sawhney 等人<sup>[61]</sup>

也使用学习排序方法解决股票预测问题，并提出了一种新的时空超图注意网络，将不同类型和程度的股票关系建模为股票排名的超图。

## 2.3 排序算法

学习排序（Learning to Rank, LTR）是指希望机器能够区分出候选数据的重要性，从而达到数据排序的目的。排序算法广泛使用在推荐搜索任务中，重点关注于如何更合理地显示排列从数据库中召回的数据，从而提升用户的点击率。不同查询关键词或者不同用户的个性属性，都会对排序的内容和序列造成不同的影响。一个好的排序算法除了要有高标准的排序结果，也对实时性要求较高。LTR 解决方法可以根据模型结构和损失函数的计算方法分为三类，分别是单文档方法（Pointwise）、文档对方法（Pairwise）和文档列表方法（Listwise）。由于方法及网络复杂程度的不同，各类 LTR 方法的模型训练和预测时间也不尽相同。所以在搜索排序时，不同任务阶段会根据数据量和任务需求，采用不同的模型对排序结果进行预测和调整，从而得到更好的序列情况。

### 2.3.1 Pointwise

Pointwise 排序方法是针对单个候选项进行处理，将排序问题转化成为查询词和候选项之间的分类或回归问题。模型每次输入一个查询词（query）-候选项（item）对，会通过模型内的打分函数  $f$  返回它们的相关性数值，利用数值即可对候选序列进行排序。计算如公式 (2-2) 所示。

$$y = f(query, item) \quad (2-2)$$

在用户点击率 CTR 预测中，target 可以用 0/1 分别表示用户不点击或点击，模型预测的结果  $y$  则表示用户点击的概率。通过比较 target 与  $y$  之间的差异，训练网络，提升模型的预测效果。常见的 Pointwise 排序方法有 PRank<sup>[62]</sup> 等。

Pointwise 具有模型复杂度低、训练时间快等优点，目前在工业界中的使用比较广泛。但是这类方法完全从单个候选项的分类角度进行计算，没有考虑到不同候选项之间的相互影响关系，在一些实际应用场景中，比如搜索推荐，不

同候选项之间的相对顺序也是评判模型效果的重要指标之一。

### 2.3.2 Pairwise

Pairwise 排序方法把问题转换为候选项对的相对顺序任务，每次成对的选择一组候选项对  $[item_a, item_b]$  输入到网络中，预测这两个候选项之间的相对顺序关系，计算如公式 (2-3) 所示：

$$y = f(query, item_a, item_b) \quad (2-3)$$

其中输出空间可以定义为  $\{-1, 1\}$ ，其中 -1 则表示候选项  $item_a$  应当在候选项  $item_b$  之后，1 则表示候选项  $item_a$  应当在候选项  $item_b$  之前。常见的 Pairwise 模型有 SVMrank<sup>[63]</sup>、RankNet<sup>[64]</sup> 等。Pairwise 方法虽然能够得到不同候选项对之间的相对关系，但是并没有直接对最终的排序结果进行预测，这样模型结果容易受到噪声影响。如果其中某对候选项的相对顺序预测错误，整个序列都会被打乱。并且对于不同的排序任务而言，候选项对个数的差异会非常大，这会影响模型最终的效果。

### 2.3.3 Listwise

Listwise 与上述两种方法的不同主要在于，它直接从优化整体排序结果的角度出发，每次将整个查询的候选项  $[item_1, item_2, \dots, item_n]$  全部输入到网络中，通过模型预测最终的排序结果情况，计算如公式 (2-4) 所示：

$$y = f(query, item_1, item_2, \dots, item_n) \quad (2-4)$$

其输出空间为长度为  $n$  的序列，表示每个候选项应当排序的位置 index。为了能够得到更准确的排序结果，Listwise 方法通常会在 loss 中，添加一个评估对整体排序结果的值，并以此为依据训练模型，使得模型能够得到更好的完整序列排序结果。例如在 ListNet<sup>[65]</sup> 中，模型利用交叉熵损失计算预测排序和真实排序的概率分布差异，通过不断调整网络参数以最小化损失值，得到最优模型。

在 Pairwise 方法中，模型整体的目的是为了减少序列中的逆序对，这样会更加关注排在后面的候选项。而在一般情况下，搜索排序系统应当更加关注靠前位置的排列情况，所以 Listwise 能够更合理地预测序列。不过，虽然 Listwise 方法在序列整体约束上的效果会比较好，但是由于模型精度的需求，

模型的训练时间会比较长，不适合过大的数据集任务，所以一般使用在搜索排序模型最后的重排阶段。

上述三类方法的对比如表 2-1 所示。在实际应用中，由于数据量大小的问题，通常会在排序任务中联合使用多种排序方法。在排序最开始，首先使用 Pointwise 方法从大量的数据库中召回可能相关的候选项，之后使用 Pairwise 的方法对其进行粗排后得到较相关的候选项，最后在使用 Listwise 方法对较相关候选项进行精排，从而得到最终的排序结果。这样的方法在保证排序准确程度的基础上，同时保证了对时间消耗的控制。

表 2-1: 不同排序方法的对比

	Pointwise	Pairwise	Listwise
输入	Query, <i>item</i>	Query, ( <i>item<sub>a</sub></i> , <i>item<sub>b</sub></i> )	Query, ( <i>item<sub>1</sub></i> , <i>item<sub>2</sub></i> , ..., <i>item<sub>n</sub></i> )
输出	(0, 1) 区间内的 float 值 表示相关程度值	{-1, 1} 表示相对顺序结果	[ <i>Index<sub>1</sub></i> , <i>Index<sub>2</sub></i> , ..., <i>Index<sub>n</sub></i> ] 长度为 n 的 index 序列 表示整个序列情况
loss 计算方式	均方误差	交叉熵	交叉熵

### 2.3.4 性能指标

为了评估排序结果的优劣程度，常使用的评价指标包括 Precision、Mean Average Precision (MAP)、Normalized Discounted Cumulative Gain (NDCG) 等。

精确率 (Precision, P) 表示预测结果的前 n 个中，正确被预测项所占的比例，计算方法如公式 (2-5)。其中  $P(n)$  表示预测的前 n 个结果， $T(n)$  表示真实的前 n 个结果。

$$P = \frac{|P(n) \cap T(n)|}{|P(n)|} \quad (2-5)$$

与其相关的还有召回率 (Recall, R)，表示真实结果的前 n 个中，被正确预测的概率，计算方法如公式 (2-6)。

$$R = \frac{|P(n) \cap T(n)|}{|T(n)|} \quad (2-6)$$

平均精度均值 MAP 是在 AP 值上进行不同排序结果之间的平均，衡量整个系统的性能。AP 是将精确率看做是召回率的函数，计算方法则是将精确率沿着召回率的增加进行积分求得，表示平均精确率。MAP 计算过程如公式 (2-7)

所示。

$$\begin{aligned} AP &= \frac{1}{R} \times \sum_{r=1}^R \frac{r}{\text{position}(r)} \\ MAP &= \frac{\sum_{q=1}^Q AP(q)}{Q} \end{aligned} \quad (2-7)$$

其中  $R$  表示相关结果的总个数， $\text{position}(i)$  表示相关结果  $i$  在列表中的位置， $Q$  表示所有的排序任务， $q \in Q$  表示当前的排序任务。

归一化折扣累计收益 NDCG 表示对序列中的累计增益进行归一化，计算过程如公式 (2-8) 所示。

$$\begin{aligned} CG_p &= \sum_{i=1}^p \text{rel}_i \\ DCG_p &= \sum_{i=1}^p \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \\ IDCG_p &= \sum_{i=1}^{|\text{REL}|} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \\ NDCG_p &= \frac{DCG_p}{IDCG_p} \end{aligned} \quad (2-8)$$

其中  $CG$  表示前  $p$  个位置累计得到的增益，其中  $\text{rel}_i$  表示第  $i$  个候选项的相关程度； $DCG$  在  $CG$  基础上引入了位置信息的度量，使越前端的位置具有更大的价值； $IDCG$  为按照最优的方式对文档排序下的序列增益。 $NDCG$  则为对不同排序结果使用对应  $IDCG$  归一化之后的累计折损增益值。

根据  $P$ 、 $MAP$  和  $NDCG$  值，我们能够将当前的排序结果和真实排序情况进行比较，有效衡量排序算法的有效性。

## 2.4 问题表述

本文将股票预测问题作为一个学习排序 (LTR) 的问题。假设具有  $N$  只股票的集合  $S = \{s_1, s_2, \dots, s_n\}$ 。在交易日  $t$ ，每只股票  $s_i \in S$  都具有历史特征数据  $F_i^{t-T:t}$  和当天的回报率  $r_i^t = \frac{c_i^t - c_i^{t-1}}{c_i^{t-1}} \times 100\%$ ，其中  $T$  表示历史特征的时间窗口大小， $c_i^t$  表示股票  $i$  在第  $t$  天的收盘价。在任何给定的交易日  $t$ ，股票集合  $S$  存在一个最佳排名序列  $Y^t = \{y_1^t > y_2^t > \dots > y_n^t\}$ ，在这个序列中，获得较高排名分数的股票预计会在交易日  $t$  获得较高的投资收益 (回报率)。我们的目标是，在输入时间窗口大小为  $T$  的股票序列特征之后，学习一个排名分数函数  $f$ ，输出按

照预测回报率排序的股票排名列表  $y^t$ 。在学习到的排名列表中，排名越高的股票在第  $t$  天时预期获得的投资收益利润越高。具体计算如公式 (2-9) 所示，我们希望预测的排名列表  $y^t$  越接近真实排名序列  $Y^t$  越好。

$$y^t = f(F_1^{t-T:t}, F_2^{t-T:t}, \dots, F_n^{t-T:t}) \quad (2-9)$$

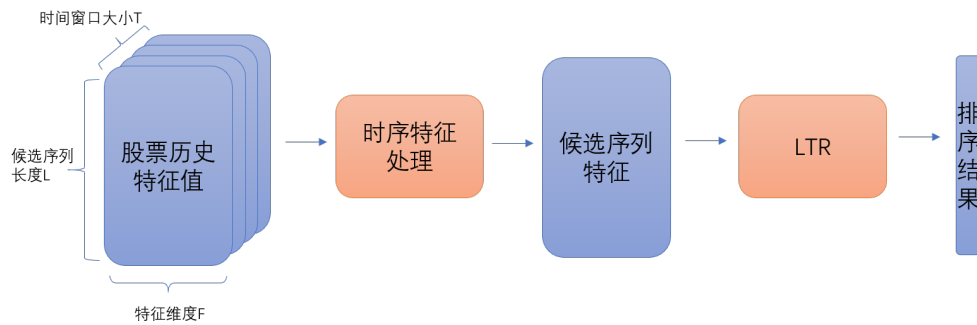


图 2-4: 股票预测排序任务模型结构

整个任务的流程结构如图 2-4 所示，模型的输入数据为  $x \in \mathbb{R}^{L \times T \times F}$ ，其中  $L$  表示候选序列中股票的个数， $T$  表示回看的股票历史特征的时间窗口大小， $F$  表示每只股票每天的特征维度。经过时序特征处理后，数据压缩成为  $x_{new} \in \mathbb{R}^{L \times F}$ ，之后经过排序阶段后，得到候选序列的排序结果  $y \in \mathbb{R}^{L \times 1}$ ，序列  $y$  中的每一个值表示对应的股票在序列中的排序结果。

下面我们举例对模型流程进行说明。在 2021 年 01 月 01 日，针对 10 只候选股票的推荐排序任务中，我们首先需要收集当前 10 只股票在 2020 年 12 月 01 日至 2020 年 12 月 31 日内 23 个交易日的数据特征，得到特征  $X \in \mathbb{R}^{10 \times 23 \times 7}$ ，其中 7 表示每个交易日的单只股票的特征维度，分别表示其在该交易日的开盘价、收盘价、最高价、最低价、交易量、交易额和涨跌幅。表 2-2 以上海证券市场内的股票 sh.600000 浦发银行为例，给出了该只股票的原始特征值。在对原始特征值进行处理、归一化之后，股票推荐排序模型首先通过时序特征处理阶段处理股票特征，得到新的候选股票特征表示  $X_{new} \in \mathbb{R}^{10 \times 7}$ ，其中 10 表示候选股票个数，7 则为股票特征维度。之后模型再通过排序阶段得到每只股票的排名分数  $y \in \mathbb{R}^{10 \times 1}$ ，分别给出候选股票的分值，分值越大则表示对应的股票应排在序列越前端，为越推荐的股票候选。股票推荐排序模型通过排序结果完成推荐，可以帮助投资者们从排名靠前的股票中选择投资目标，达到提高收益的目的。

表 2-2: 上海证券交易所中 sh.600000 浦发银行在 2020 年 12 月 01 日至 2020 年 12 月 31 日内 23 个交易日的原始数据特征

股票号	交易日	开盘价	最高价	最低价	收盘价	交易量	交易额	涨跌幅
sh.600000	2020/12/1	10.08	10.35	10.01	10.3	107812159	1099269636	2.3857
sh.600000	2020/12/2	10.26	10.39	10.19	10.25	78702779	808812295.6	-0.4854
sh.600000	2020/12/3	10.23	10.28	10.16	10.24	57195735	584153273.4	-0.0976
sh.600000	2020/12/4	10.24	10.25	10.05	10.17	47336450	479183315	-0.6836
sh.600000	2020/12/7	10.11	10.17	9.9	9.99	57810545	577584100.5	-1.7699
sh.600000	2020/12/8	9.97	10.06	9.93	9.98	31522398	315079837.7	-0.1001
sh.600000	2020/12/9	9.98	10.09	9.91	9.91	44090061	440752941.2	-0.7014
sh.600000	2020/12/10	9.91	9.96	9.76	9.82	39400878	388252087.6	-0.9082
sh.600000	2020/12/11	9.84	9.88	9.69	9.7	48386569	471620087.1	-1.222
sh.600000	2020/12/14	9.75	9.8	9.68	9.75	38743836	376928042.5	0.5155
sh.600000	2020/12/15	9.73	9.74	9.6	9.71	39323070	379468277.5	-0.4103
sh.600000	2020/12/16	9.72	9.78	9.64	9.65	47998904	465187094.4	-0.6179
sh.600000	2020/12/17	9.64	9.79	9.59	9.77	42901555	415257095.5	1.2435
sh.600000	2020/12/18	9.74	9.79	9.69	9.76	31884462	310411756.3	-0.1024
sh.600000	2020/12/21	9.7	9.77	9.65	9.75	31965504	310813655.4	-0.1025
sh.600000	2020/12/22	9.75	9.75	9.55	9.61	50418001	485974239.5	-1.4359
sh.600000	2020/12/23	9.6	9.64	9.48	9.55	38656338	368816679.1	-0.6243
sh.600000	2020/12/24	9.55	9.66	9.51	9.55	24871912	238279375.6	0
sh.600000	2020/12/25	9.56	9.61	9.51	9.58	22420976	214427367.1	0.3141
sh.600000	2020/12/28	9.55	9.63	9.5	9.57	33765935	322826382.3	-0.1044
sh.600000	2020/12/29	9.59	9.62	9.52	9.53	44150648	421945426	-0.418
sh.600000	2020/12/30	9.52	9.54	9.44	9.5	54885564	519975315.2	-0.3148
sh.600000	2020/12/31	9.51	9.69	9.49	9.68	65992362	635666178.5	1.8947

## 2.5 本章小结

本章总结与回顾了时序预测任务、股票分析预测方法和排序方法，并对本论文关注的问题进行了解释与定义。

在时序特征处理中，我们首先对时序特征处理的应用场景进行了介绍，之后总结回顾了基于统计学的传统方法、机器学习方法和深度学习的方法。

在股票分析预测方法里，我们首先说明了股票预测问题的定义，之后从传统预测方法、基于机器学习的方法和基于深度学习的方法进行了简单介绍和总结，最后介绍了目前常见的改进思路。

针对排序方法，本文详细介绍了常见的三种排序方法，分别是 Pointwise、Pairwise 和 Listwise，对比了三者之间的差异，并且对排序算法的性能指标进行了简单的介绍。

问题表述模块明确了本文要解决的问题，使用推荐排序方法对股票数据进

---

行分析，解决股票投资问题，更有效地关注不同股票之间的相互影响，贴合股票投资的实际场景需求。



# 第三章 基于注意力机制的时序特征处理算法

前文分析了当前股票预测问题面临的挑战与问题，并说明本文将股票投资问题使用学习排序方法进行解决，整个模型结构分为时序特征处理与排序两部分。本章重点关注特征表示任务，分析回顾当前时序特征表示方法中存在的问题与挑战。最后结合股票数据特性，我们设计了一个基于注意力机制的特征卷积与 GRU 时序处理模型，此网络结构利用卷积操作更合理地得到时序特征，同时使用注意力机制缓解对长距离特征的遗忘。通过对股票时序特征进行更好地提取，从而达到提升股票预测排序任务的效果。

## 3.1 注意力机制

注意力是指对某项事物的集中关注能力。注意力机制意在模仿人类在平时会主动忽略身边的无用信息，而集中于获取重要信息的现象。当大量信息输入神经网络时，其中包含了许多噪声与无用数据，我们便可以利用注意力机制，使得模型只需重点关注于有用信息，降低噪声数据的影响。注意力机制最初是由 Mnih 等人<sup>[66]</sup>提出的，后逐渐应用于视觉领域<sup>[67]</sup>和自然语言处理领域<sup>[30]</sup>，这些应用将注意力机制模块与传统网络结构相结合，从而起到对关键信息捕捉的作用。下面我们将分别对基础注意力和键值对注意力机制进行介绍。

### 3.1.1 基础注意力机制

基础注意力机制的使用场景为，在任务 Query 下，通过对任务建模得到任务特征，即查询向量（Query Vector），注意力则需完成能够从大量信息  $X$  中提取出与任务相关特性的工作。相关性使用打分函数进行计算，来评估不同信息与当前任务的相关程度。打分函数常见的为点积模型和缩放点积模型。

点积模型通过将 Query 特征和 x 特征进行点积计算，得到相似程度：

$$s(x_i, q) = x_i^T q, \quad (3-1)$$

点积模型计算便捷，但是容易受到特征维度的影响，例如当维度  $d$  较高时，最终的相关性值会较大，而数量级大的输入会将 softmax 推向梯度平缓区域，使得网络收敛困难。而缩放点积模型在相似度计算时，对结果使用  $\sqrt{d}$  进行归一化，从而降低了维度大小对最终值的影响：

$$s(x_i, q) = \frac{x_i^T q}{\sqrt{d}}, \quad (3-2)$$

在使用打分函数计算出不同输入特征  $X$  的相似性后，模型对所有的相似性进行归一化处理，将其转换为注意力在所有输入特征  $X$  上的注意力分布情况，该分布中不同的值则表示在所有输入特征中， $x_i$  与任务  $q$  的相对相关程度。即在任务  $q$  下，输入特征  $x_i$  的相对相关程度  $\alpha_i$  为：

$$\begin{aligned} \alpha_i &= p(z = i | X, q) \\ &= \text{Softmax}(s(x_i, q)) \\ &= \frac{\exp(s(x_i, q))}{\sum_{j=1}^N \exp(s(x_j, q))}. \end{aligned} \quad (3-3)$$

得到了注意力分布情况之后，注意力机制还需要依据此对信息进行整合处理。整合的方法可以分为两种，一种是通过注意力分布情况对所有输入特征进行加权求和计算，得到综合处理后的特征表示，即为软性注意力机制（Soft Attention Mechanism），计算如公式 (3-4) 所示；另一种是只选择最受关注的输入特征表示最终结果，即为硬性注意力机制（Hard Attention Mechanism），如公式 (3-5) 所示。

$$\text{att}(X, q) = \sum_{i=1}^N \alpha_i x_i \quad (3-4)$$

$$\text{att}(X, q) = \arg \max_{x_i \in X} \alpha_i \quad (3-5)$$

相比于硬性注意力机制，软性注意力机制综合考虑到了更多、更全面的信息，所以实际应用效果会更好，也被更多地采用。

### 3.1.2 键值对注意力机制

键值对注意力机制则是将输入信息  $x$  转换为对应的键值对 (key-value pair) 表示。每个信息对应的键向量  $k$  和值向量  $v$  数值不相等，但计算过程存在一定的关联。他们被使用在计算中的不同步骤，以提取到更多特征空间内的相关性。

如图 3-1，在任务  $Query$  下，输入特征首先被映射成为键向量  $k$  和值向量  $v$ ，之后使用任务向量  $Query$  与所有的键向量  $k$  进行相似度计算，并得到注意力分布情况。在得到注意力分布情况后，将其与所有的值向量  $V$  进行加权计算整合，得到最终的输出表示  $AttentionValue$ 。通过这样的键值对注意力计算方法，能够捕获到更多特征空间的特性，提高注意力计算的准确性与最终特征表示的准确性。

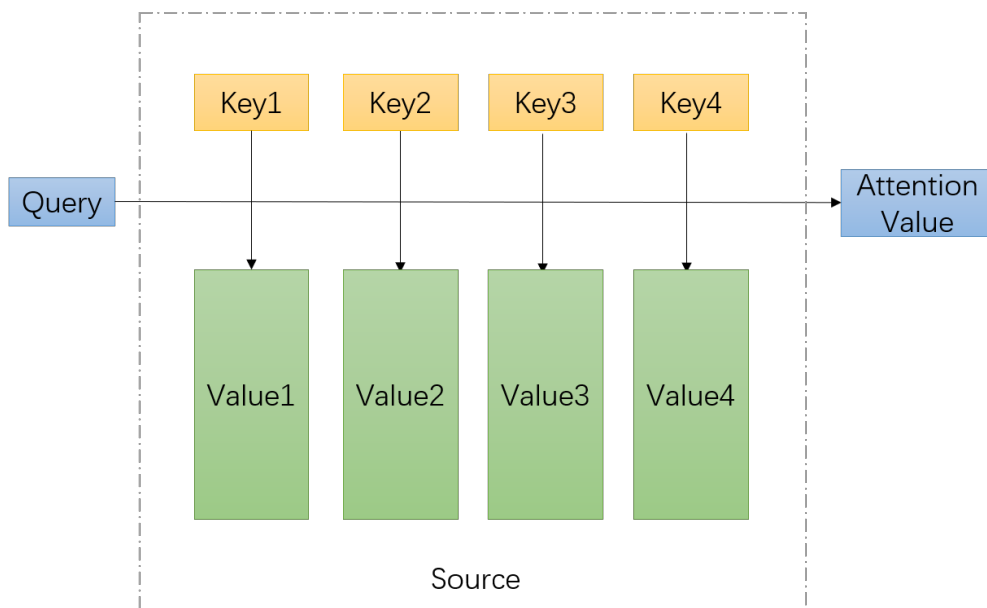


图 3-1: 键值对注意力

$$\begin{aligned}
 \text{att}((K, V), q) &= \sum_{i=1}^N \alpha_i v_i \\
 &= \sum_{i=1}^N \frac{\exp(s(k_i, q))}{\sum_{j=1}^N \exp(s(k_j, q))} v_i
 \end{aligned} \tag{3-6}$$

在实际应用，为了使网络具备更强大的拟合能力，Transformer 中采用了多头注意力方法，使用不同的映射方法得到不同特征空间内的键值对，再分别对

其进行注意力计算，最后将结果整合输出。多头注意力机制计算如公式 (3-7) 所示，通过捕获到不同空间域内的特征注意力情况，提高网络的表达能力。

$$\begin{aligned} \text{multihead}(Q, K, V) &= \text{concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) W^o \\ \text{head}_i &= \text{attention}(QW^Q, KW^K, VW^V) \end{aligned} \quad (3-7)$$

## 3.2 时序循环网络

时序处理由于其数据之间的时间相关性，最常使用到的模型结构为循环神经网络（Recurrent Neural Network, RNN）。RNN 主要由三部分组成，分别为输入层、隐藏层和输出层。其中隐藏层中的隐藏状态，不仅仅取决于当前的输入，还取决于上一次的隐藏层值，因此对历史特征值进行了保留，搭建起序列之间的信息传递通道。RNN 的网络结构如图 3-2 所示，计算公式为公式 (3-8)。

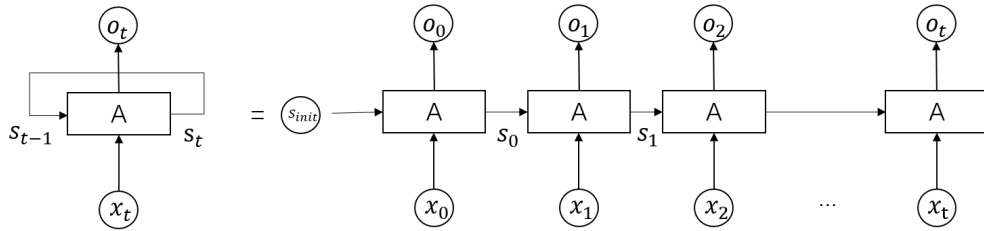


图 3-2: 循环神经网络 RNN 网络结构

$$\begin{aligned} o_t &= g(V \cdot s_t) \\ s_t &= f(U \cdot x_t + W \cdot s_{t-1}) \end{aligned} \quad (3-8)$$

其中  $x_t$  表示当前时刻的输入， $s_{t-1}$  表示上一个时刻的隐藏状态， $U$  是输入数据的权重矩阵， $W$  是上一时刻隐藏状态的权重矩阵，控制历史信息的传递， $s_t$  即为当前时刻的隐藏状态。 $V$  是隐藏层到输出层的权重矩阵， $o_t$  表示当前时刻的输出。

循环网络中隐向量的使用，使得历史数据能够得到长久的保留，并在时序计算过程中很好地进行了信息的传递。但是这种叠加计算的方法，为循环网络带来了以下问题：（1）循环网络针对多个时间步进行叠加计算，这会造成计算过程中同一个权重矩阵的累乘，容易出现梯度消失或梯度爆炸的问题；（2）循环神经网络的输入需要严格遵守其时间关系，逐个地按顺序地输入网络中进行计算，这种串行的计算方法增加了计算时间的消耗，降低了模型

效率；（3）研究表明，在循环网络中，模型最终输出容易受到邻近历史信息的影响，而对较远历史的信息保留较少，而在实际场景中，不同时间步信息的重要性不单单由时间邻近程度而决定，与数据值的大小等都存在一定的关系。

从循环网络的缺陷出发，研究者们不断对 RNN 进行修改调整，得到更加稳定高效的循环神经网络结构。其中长短期记忆网络（Long Short-Term Memory, LSTM）<sup>[28]</sup> 则为代表性的模型。LSTM 仍通过网络中的隐藏状态保留历史信息。但不同于 RNN，LSTM 增加了单元状态来存储长期记忆，并在结构中通过三个门控制信息的传递。在提供 LSTM 的结构和计算公式之前，我们首先描述与 LSTM 相关的术语。在每个时间步长  $t$  上， $x_t \in \mathbb{R}^D$  表示一个输入向量，其中  $D$  是输入维数。向量  $C_t$  和  $h_t \in \mathbb{R}^U$  分别表示单元状态和隐状态，其中  $U$  表示隐向量维度。向量  $\tilde{C}_t \in \mathbb{R}^U$  是一个信息转换模块，向量  $i_t, o_t, f_t$  分别表示输入门、输出门和遗忘门。转换模块、状态向量和控制门通过以下方程定义：

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
 C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
 o_t &= \sigma(W_o [h_{t-1}, x_t] + b_o) \\
 h_t &= o_t * \tanh(C_t)
 \end{aligned} \tag{3-9}$$

其中  $W_f, W_i, W_C, W_o \in \mathbb{R}^{U \times D}$  为映射矩阵， $b_f, b_i, b_C, b_o \in \mathbb{R}^U$  为偏置向量。

门控循环单元（Gated Recurrent Unit, GRU）<sup>[29]</sup> 是 LSTM 网络的变体。GRU 模型中只存在隐藏状态向量  $h_t$ ，并且只采用两个门控，分别是更新门  $z_t$  和重置门  $r_t$ 。更新门  $z_t$  用于控制历史信息的传入，其值越小则表明前一时刻的状态信息会越多地传入当前隐藏状态的计算过程。重置门  $r_t$  控制前一状态信息的传入，其值越大，前一状态的状态信息会越多地写入到当前隐藏状态中。状态向量和控制门通过以下方程定义：

$$\begin{aligned}
 z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
 r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
 \tilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\
 h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
 \end{aligned} \tag{3-10}$$

其中  $W_z, W_r \in \mathbb{R}^{U \times D}$  为映射矩阵。更新公式可以理解为执行以下步骤：（1）利用前一时刻隐藏状态  $h_{t-1}$  和当前时刻输入  $x_t$ ，计算更新门  $z_t$  的值。（2）利用前一时刻隐藏状态  $h_{t-1}$  和当前时刻输入  $x_t$ ，计算重置门  $r_t$  的值。（3）利用重置门  $r_t$  对前一时刻隐藏状态  $h_{t-1}$  进行选择，并将其与当前时刻输入  $x_t$  进行信息融合，得到候选隐藏状态  $\tilde{h}_t$ 。（4）利用更新门  $z_t$  融合来自前一时刻隐藏状态  $h_{t-1}$  和候选隐藏状态  $\tilde{h}_t$  的信息，更新当前隐藏状态  $h_t$ 。

通过上述分析可以发现，LSTM 和 GRU 模型中的单元状态  $C_t$  和隐藏状态  $h_t$  控制了时序中信息的传递，增强了循环网络的对长期历史的记忆，降低了网络的训练难度。不过虽然 LSTM、GRU 的变体循环网络结构促进了循环网络的广泛应用，但是并没有从根本上解决循环网络第 3 点的问题。根据之前的分析，近几年学者提出的注意力机制能够在不受到距离限制的情况下进行注意力关注，所以将循环神经网络与注意力机制相结合，能够解决上述提到的网络缺陷。

### 3.3 基于注意力机制的时序处理算法设计

通过上节的分析，我们发现目前的时序循环网络仍不能够很好地解决针对久远历史遗忘现象，而股票数据中，历史时间段内的最高价和最低价都会对未来的预测产生重要影响。因此在本文提出的算法中，我们使用注意力机制，将其与特征卷积网络和循环神经网络结合得到股票时序特征处理的方法。下面我们将分别介绍基于注意力机制的 LSTM 模型，特征网络结构和整体特征处理算法流程。

#### 3.3.1 ALSTM

注意力机制已广泛用于基于 LSTM 的序列学习问题解决方案中，是将不同时间步长的隐藏表示压缩成具有自适应权重的整体表示。注意力机制旨在模拟这样一个现象，即不同时间步长的数据可能对整个序列的表示存在不同的贡献。对于股票表示而言，不同时间步长的状态也可能有不同的贡献。例如，在时间窗口内具有最高和最低价格的天数可能对总体表示有着更高的贡献。因此，ALSTM 中使用注意力机制将历史特征进行整合，结构如图 3-3 所示，具体计算如公式 (3-11) 所示：

$$\begin{aligned}
 \tilde{\alpha}_t^s &= u_a^T \tanh(W_a h_t^s + b_a) \\
 \alpha_t^s &= \frac{\exp(\tilde{\alpha}_t^s)}{\sum_{t=1}^T \exp(\tilde{\alpha}_t^s)} \\
 a^s &= \sum_{t=1}^T \alpha_t^s h_t^s
 \end{aligned} \tag{3-11}$$

其中  $h_t^s$  表示 LSTM 中每层的隐藏状态， $W_a \in \mathbb{R}^{E \times U}$ 、 $b_a$  和  $u_a \in \mathbb{R}^E$  是需要学习的参数， $\tilde{\alpha}_t^s$  表示计算得到的注意力值， $\alpha_t^s$  表示归一化后的注意力值， $a^s$  即为对序列中的整体特征进行编码的聚合表示。在预测阶段，ALSTM 中并不是直接使用  $a^s$ ，而是将  $a^s$  与最后的隐层状态  $h_t^s$  结合，作为股票  $s$  的最终表示  $e^s \in \mathbb{R}^{2U}$ ，最后使用全连接层，构建股票特征与预测结果  $\hat{y}^s$  之间的预测函数，具体计算如公式 (3-12) 所示。这样在避免久远时间步遗忘的同时，进一步强调了较近的时间步特征。

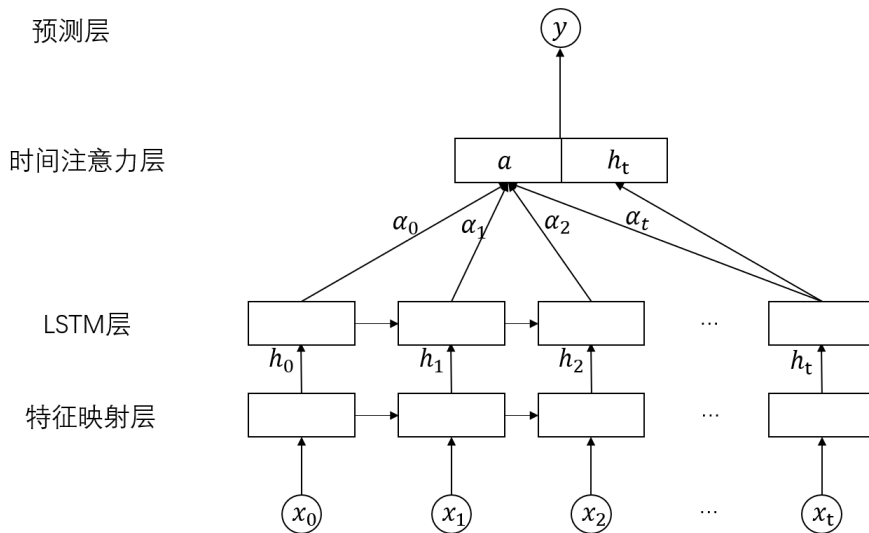


图 3-3: Attentive LSTM 网络结构<sup>[1]</sup>

$$\begin{aligned}
 e^s &= [a^{sT}, h_T^{sT}]^T \\
 \hat{y}^s &= w_p^T e^s + b_p
 \end{aligned} \tag{3-12}$$

### 3.3.2 特征卷积结构

在之前分析的 LSTM 和 GRU 模型的优缺点中，我们可以得知 GRU 模型内参数更少，模型更小，计算更加便捷，而在股票投资分析任务对模型时效性的要求较高。所以本算法使用 GRU 模型替换 ALSTM 中的 LSTM 结构。同时由于 ALSTM 是通过学习参数  $u_a$  来计算注意力值，而不像基础注意力机制，通过使用查询向量  $q$  进行注意力值计算。这种通过学习参数的方法增加了网络的不确定性。为了更好地计算注意力分布权重，受到时序卷积网络（Temporal Convolutional Network, TCN）<sup>[31]</sup> 的启发，本模型在网络中添加了特征卷积结构，如图 3-4 所示，通过卷积结构整合出不同特征在历史时间步的信息，将所有特征信息拼接后作为查询向量与 GRU 中的隐状态进行注意力计算。

在特征卷积结构中，我们使用一维卷积结构提取不同特征的变化趋势。每个一维卷积核在得到当前特征的变化结果后，沿着股票特征维度滑动，对不同特征进行分析处理。在我们的任务中，为了能够得到历史时间步内股票的变化情况，我们设定卷积核沿着特征方向上移动，通过控制卷积核的大小，表示每次处理特征的时间窗口大小。每次的卷积操作，对时间段内的指定特征进行卷积操作，得到特征历史变化趋势。

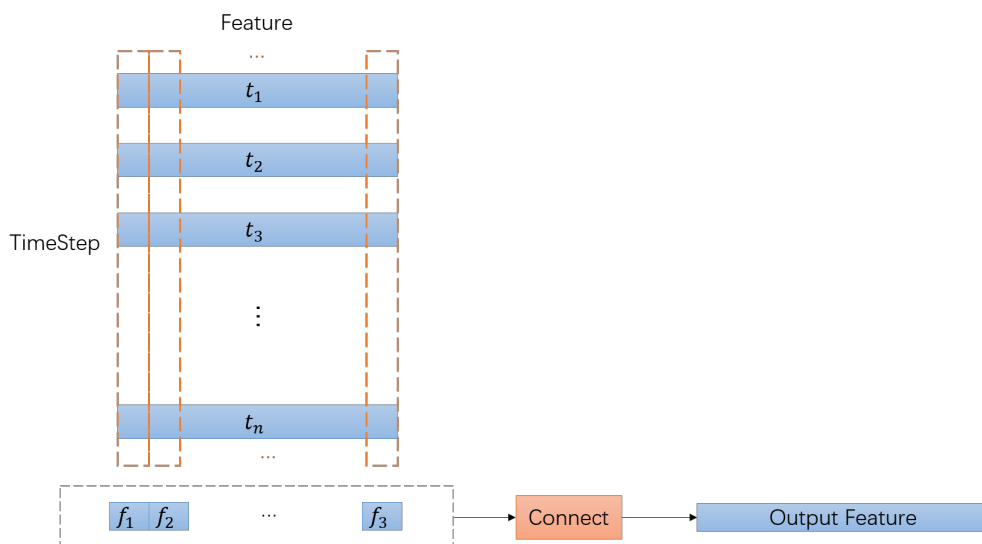


图 3-4: 特征卷积计算流程

### 3.3.3 AFC\_GRU 模型算法流程

在使用特征卷积得到历史特征的变化趋势后，我们便需要使用注意力机制将特征卷积结果与 GRU 进行结合，达到有效捕获股票历史特征及变换趋势的目的。本节将详细介绍结合方法，描述整体算法流程，得到基于注意力机制的特征卷积与 GRU 时序处理模型 AFC\_GRU（Attention Feature Convolution and Gated Recurrent Unit），模型结构如图 3-5 所示。

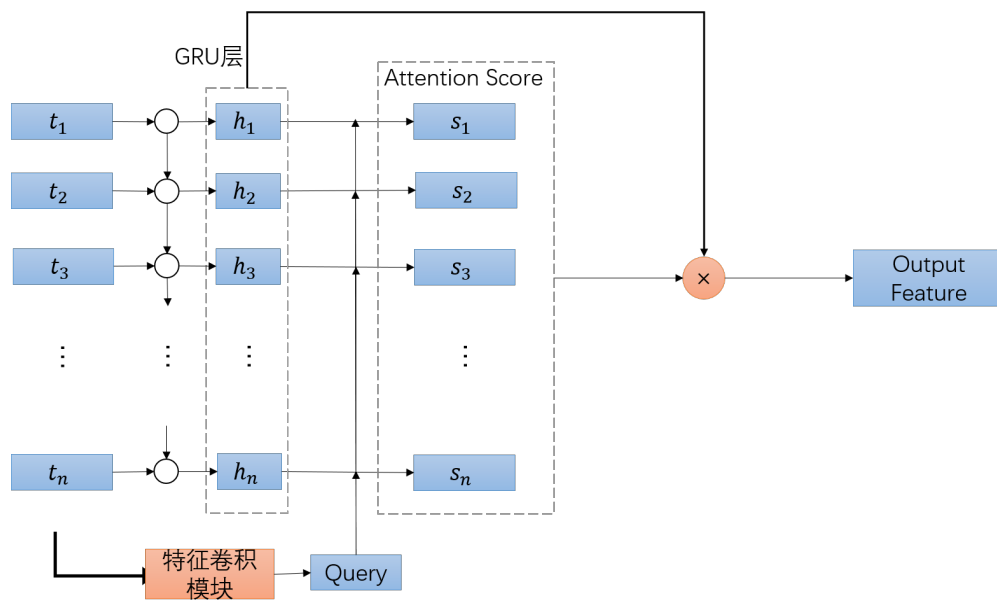


图 3-5: 基于时空卷积注意力机制的时序特征处理

在特征卷积过程中，我们计算得到了向量  $q$ ，表示所有特征在历史上的变化趋势特征。另外我们将股票特征按照时间顺序输入 GRU 结构中，并将每个隐藏的输出作为键向量  $K$  和值向量  $V$ ，使用查询向量  $q$  和键值对  $K, V$  进行键值注意力计算，得到最终的序列特征表示。

我们的算法流程如算法 3.1 所示，主要通过一维卷积层和注意力结构组成。算法的输入向量  $x \in \mathbb{R}^{Batch \times ListSize \times Timestep \times FeatureSize}$ ，其中  $Batch$  表示批处理大小， $ListSize$  表示候选序列长度， $Timestep$  表示历史时间窗口大小， $FeatureSize$  表示股票每天的特征维度大小。最初为了方便计算，我们先将四维输入压缩为三维，因为当前步骤主要是对单个股票的特征进行处理。之后我们在将一维卷积核  $kernel \in \mathbb{R}^{1 \times Timestep}$  在  $FeatureSize$  上移动，捕捉到在历史时间窗口  $Timestep$  内，不同特征的变化情况，得到  $Query \in \mathbb{R}^{(Batch \times ListSize) \times 1 \times FeatureSize}$ 。另外我们将股票特征序列输入 GRU 模块中，得到

隐层输出  $Totalouts \in \mathbb{R}^{(Batch \times ListSize) \times Timestep \times FeatureSize}$ 。之后我们再通过一个注意力结构，将特征变化趋势  $Query$  与 GRU 的隐层输出  $Totalouts$  进行注意力计算，即可得到每只候选股票在固定时间窗口内，注意力的分布情况  $AttentionScore \in \mathbb{R}^{(Batch \times ListSize) \times 1 \times Timestep}$ 。最后根据注意力值，更新得到股票特征的表示  $out \in \mathbb{R}^{Batch \times ListSize \times FeatureSize}$ 。

---

### 算法 3.1 时序特征处理算法

---

输入：股票历史时间步特征

输出：股票表示

- 1:  $x = x.reshape(batch*list, time,dim)$ ,  
对股票特征数据进行维度转换
  - 2:  $Query = tcn(x)$ ,  
使用 1 维卷积核在  $x$  的  $dim$  维度上移动，每次对指定特征  $dim$  下的  $time$  维度进行卷积。最终的输出  $(batch*list, 1,dim)$
  - 3:  $Totalouts = gru(x)$ ,  
将每只股票的特征按照时间顺序输入  $gru$  模型中，将每层的状态输出保存为  $Totalouts$ 。最终的输出  $(batch*list, time,dim)$
  - 4:  $q = Query$
  - 5:  $K = Linear(Totalouts)$ ,  
使用线性层对  $Totalouts$  进行处理
  - 6:  $V = Totalouts$ ,  
将  $Totalouts$  看作 Value 值
  - 7:  $Attention\ Score = q*K*(K.size(-1)**-0.5)$ ,  
利用  $q$  和  $K$  计算注意力分值
  - 8:  $Attention\ Score = softmax(Attention\ Score)$ ,  
对注意力分值进行归一化处理
  - 9:  $out = Attention\ Score * V$ ,  
将注意力值和 Value 值相乘, 得到最终的特征
  - 10: **return** out
- 

## 3.4 实验与分析

为了验证本章提出的基于注意力机制和 GRU 的时序特征处理方法效果，我们将其应用在股票预测排序任务中进行检验，对排序结果进行了分析。另外我们还对模型中的超参设置进行了对比试验，以获得更好的模型结构。

### 3.4.1 实验设置

#### 3.4.1.1 数据集

**上证 50。**上证 50 是从上海证券市场中挑选出规模大、流通性好的最具代表性的 50 只股票组成样本股，以此作为代表，综合反应上海证券市场状况。我们使用 baostock 接口获取到上证 50 只股票的历史数据，并根据时间段将不同交易日划分为训练集和测试集。

**NASDAQ<sup>[60]</sup>。**纳斯达克是一个相当不稳定的美国交易所，数据集收集了在 2013 年 1 月 2 日至 2017 年 12 月 8 日之间有交易记录的 NASDAQ 市场股票，这些股票涵盖了标准普尔 500 指数和纳斯达克综合指数。

#### 3.4.1.2 股票 T+1 策略

T+1 策略是指交易日  $t$  买进的股票，需等到下一个交易日，即  $t+1$  后才能够卖出。该策略被应用于我国上海证券交易所和深圳证券交易所。T+1 策略能够更好地保障股票市场的稳定性，避免过度投机，同时也避免了频繁买卖操作产生的大量交易费用。

根据 T+1 策略，我们将预测频率设置为日水平，任务的目标就是在交易日  $t$ ，我们根据最近  $S$  个交易日的每日历史数据，预测下一个交易日  $t+1$  的股票排名。由于股票的收益率表示股票的预期收益，我们将股票  $i$  的真实排序得分设为其一日收益率  $r_i^{t+1} = (c_i^{t+1} - c_i^t) / c_i^t$ ，其中  $c_i^t$  表示股票  $i$  在第  $t$  天的收盘价。

#### 3.4.1.3 特征选择与预处理

每只股票的特征值包括它们在每一天内的开盘价（open）、收盘价（close）、前一天收盘价（preclose）、最高价（high）、最低价（low）、成交量（volume）和成交总额（amount）。

除了股票每天的原始价格特征以外，我们还参考 ALSTM，构造了 9 个时间特征来表述股票在交易日  $t$  的趋势。表 3-1 中阐述了与预测相关的特征。定义这些特征的目的是：（a）归一化不同股票的价格；（b）明确捕获到不同价格之间的相互影响，（c）得到每只股票在历史上的波动情况。

由于原始数据中存在数据丢失等情况，直接使用会降低模型的训练速度和预测精度。同时由于股票数据随机性较大，插补法并不能有效地对缺失值进行

表 3-1: 描述股票每日趋势的特征

特征	计算示例
$c\_open, c\_high, c\_low$	$c\_open = open_t / close_t - 1$
$n\_open, n\_close$	$n\_close = close_t / close_{t-1} - 1$
$5 - day, 10 - day, 15 - day, 20 - day$	$5 - day = \frac{\sum_{i=0}^4 close_{t-i} / 5}{close_t} - 1$

精确填补。因此本文在数据处理阶段删除了存在缺失值的股票数据。

同时由于股票不同特征的数据单位不一致，数量级差异较大，例如成交额与收盘价之间存在较大差异，因此本实验采用标准差标准化方法对所有候选股票的同一特征进行归一化处理。经过处理后的特征数据平均值为 0，标准差为 1。转化函数如公式 (3-13) 所示：

$$\tilde{x} = \frac{x - \mu}{\sigma} \quad (3-13)$$

其中  $\mu$  为样本的均值， $\delta$  为样本的标准差。经过归一化处理，各个指标被处理成为同一量级，提升了可比性，同时避免了噪声点的影响，加快了网络的收敛速度。

我们将 2018.01.01-2021.04.01 的数据作为训练集，将 2021.04.01-2021.06.30 的数据作为测试集。我们使用一个长度为 T 的时间窗口沿着相应的交易日移动，以构建每只候选股票的特征，同时将候选股票后一天的涨跌幅值百分比记作标签，以此为标准进行候选股票集的排序。

### 3.4.1.4 基准

为了检验模型性能，我们将以下的时序处理方法和股票预测方法作为基准，进行比较：

- **RNN**：传统的时序处理方法，通过隐藏状态控制股票历史信息在信息流中的传播，最终输出股票的特征值。
- **LSTM**：为 RNN 的变形。模型使用输入门、输出门和遗忘门，控制历史信息的传播。该方法将股票序列特征按照时间顺序依次输入 LSTM 结构，最终输出股票的特征值。
- **GRU**：为 LSTM 模型的变形。使用更新门和重置门，保存历史信息的传递，该方法按照时间顺序输入历史特征，得到最终的股票特征值。

- **ALSTM(Attentive LSTM)**<sup>[1]</sup>: 该方法是一个股票趋势预测的机器学习解决方案。模型在 LSTM 后添加了一个 Attention 层对隐层表示进行聚合, 模拟不同时间步的股票数据对整个序列的表示具有不同贡献的事实, 使得模型对整个股票序列中的状态有区分地进行选择, 从而得到股票的特征向量。在这里, 我们仅使用该模型中的特征处理阶段结构, 省略最后的预测层。
- **AGRU(Attentive GRU)**: 该方法与 ALSTM 类似, 使用 GRU 替代结构中的 LSTM。该模型在 GRU 层后添加 Attention 层对隐层表示进行聚合, 有区分地选择股票序列中的特征, 最终得到股票的特征向量。
- **TCN(Temporal Convolutional Network)**<sup>[31]</sup>: 该方法使用空洞卷积和因果卷积的方法对股票时序特征进行处理, 捕获历史变化特征, 最后对扩充维度进行压缩, 得到股票特征向量。
- **MFNN**<sup>[68]</sup>: 该方法的目的是预测股票价格变动。该方法使用卷积神经网络和循环神经网络共同组成多滤波器神经网络, 执行特征学习任务, 从而获取不同特征空间和市场层面的信息。该模型利用单层卷积层、双层卷积层和循环网络提取特征映射并进行组合, 主要用于金融时间序列样本的特征提取。
- **LSTM-CNN**: 这是一个预测股票涨跌情况的深度学习解决方法。该模型将长短期记忆网络和卷积网络进行连接, 捕获时序特征之间的依赖关系, 提取有效特征, 最终得到股票的特征表示。

### 3.4.1.5 参数设定

为了实验的公平性, 我们使用深度学习框架 Pytorch<sup>[69]</sup> 实现所有模型, 并使用 Adam 优化器<sup>[70]</sup> 对其进行优化。上证 50 数据集的隐藏向量维度设置为 52。训练批量大小设置为 30。学习率参考 Transformer<sup>[30]</sup> 使用 WarmUp 方式进行变化。为了缓解过拟合问题, 我们使用 Dropout 技术, 将神经元丢弃概率设置为 0.2。

### 3.4.1.6 评估指标

由于不方便直接评估特征表示的优劣, 在这里我们借助排序的结果来间接衡量特征表示的优劣程度。我们通过三个指标评估模型预测结果, 分别是准确率 P 值、MAP 值和 NDCG 值。P@N 值的变化范围是 [0, 1], 其值越大表示 TopN 中预测准确程度越好, MAP@N 值的变化范围是 [0, 1], 其值越大模型在

TopN 上准确和召回的更加准确。NDCG@N 值的变化范围是 [0, 1]，其值越大模型在 TopN 上排序更正确。

### 3.4.2 实验结果

在上证 50 数据集上的实验结果如表 3-2 所示。通过多种时序处理方法的实验结果对比可以看出，GRU 结构相比与传统的 RNN 结构在 P 指标、MAP 指标和 NDCG 指标上都有所提升，这说明 GRU 结构能够更好地表示时序特征，从而提升召回与排序结果。GRU 结构与 LSTM 结构相比虽然 NDCG@10 指标相差不大，但是 MAP@10 和 P@10 提升了 1.5% 左右，P@5 提升了近 5%。另外，ALSTM 相比于 LSTM，指标上均有近 2% 左右的提升；添加注意力机制的 AGRU 模型，相比于传统 GRU 而言，预测的效果更好，在 MAP@10、P@10 和 NDCG@10 指标上都有 2% 以上的提升，说明模型中的注意力机制有效的缓解了传统循环网络的缺陷，能够更好地表示特征，得到更优的排序结果。TCN 模型虽然在 NDCG@10 上效果略好，但是其他指标均相比其他模型有所下降。最后本文提出 AFC\_GRU，相比于 AGRU 采用不同的注意力方式，在大部分指标上都达到了最好的预测结果。这意味着通过时间卷积的方法得到历史变化趋势，能够更好地完成注意力操作。将本文提出的 AFC\_GRU 与股票分析任务中的 ALSTM、MFNN 和 LSTM-CNN 结构对比分析可以发现，我们提出的算法在排序指标上都取得最好结果，并且有大幅度提升，这表明本文提出的算法能够有效提取股票数据特征，在股票市场具有分析有效性。

表 3-2: 在上证 50 数据集上，不同时序特征处理方法的结果对比

	MAP@5	MAP@10	P@5	P@10	NDCG@10
RNN	0.11389	0.18315	0.22000	0.34167	0.57823
LSTM	0.11222	0.17655	0.20667	0.33833	0.57487
ALSTM	0.14133	0.20552	0.22667	0.35500	0.59536
GRU	0.14239	0.19242	<b>0.25333</b>	0.35167	0.57928
AGRU	0.14322	0.22097	0.22000	0.37167	0.61303
TCN	0.13261	0.17829	0.22667	0.32000	0.58193
MFNN	0.11700	0.17211	0.21000	0.32833	0.57638
LSTM-CNN	0.11872	0.18903	0.22667	0.36000	0.58124
<b>AFC_GRU(ours)</b>	<b>0.15294</b>	<b>0.23474</b>	0.24333	<b>0.38333</b>	<b>0.61598</b>

在 NASDAQ 数据集上的实验结果如表 3-3 所示，最优结果采用加粗方法标出。通过对比可以发现，本文提出的 AFC\_GRU 模型在 MAP10、P10 和 NDCG 排序指标上都达到了最优，MAP@10、P@10 和 NDCG@10 相比于 AGRU 模型均有近 1% 的提升。这进一步验证了 AFC\_GRU 模型中的特征卷积方法在时序特征处理任务上的有效性。AFC\_GRU 模型相比于股票分析任务中的 ALSTM、MFNN 和 LSTM-CNN 模型，在所有排序指标上都存在提升，这表明本文提出的算法在股票市场的分析有效性和适用性，在不同市场的股票数据上均有较好性能。

表 3-3: 在 NASDAQ 数据集上，不同时序特征处理方法的结果对比

	MAP@5	MAP@10	P@5	P@10	NDCG@10
RNN	0.14592	0.15577	<b>0.24719</b>	0.30000	0.54328
LSTM	0.14022	0.16375	0.24270	0.30562	0.55463
ALSTM	0.12779	0.14743	0.22022	0.28539	0.55077
GRU	0.14408	0.15909	0.22921	0.30000	0.54814
AGRU	0.13273	0.15870	0.22472	0.30787	0.55343
TCN	0.14322	0.15462	0.22697	0.29213	0.54778
MFNN	0.13191	0.14029	0.21124	0.28090	0.54300
LSTM-CNN	0.13240	0.15980	0.21348	0.30787	0.55523
<b>AFC_GRU(ours)</b>	<b>0.14801</b>	<b>0.16813</b>	0.23596	<b>0.31685</b>	<b>0.56174</b>

图 3-6 中给出了本文提出的 AFC\_GRU 算法内注意力分布情况的结果示例，其中纵轴表示不同的候选股票，横轴表示每只股票在其历史时间步上的注意力分布情况，横坐标值越小表示其距离当前时刻越久远。从图中可以看出，虽然由于股票数据特性的影响，较近历史的注意力值会较大，但是本文提出的 AFC\_GRU 算法在一定程度上缓解了长期历史的遗忘问题。

### 3.4.3 赢利能力测试

参考 Matthew 等<sup>[71]</sup>提出的方法，我们采用每日买入-持有-卖出的交易策略来评估股票预测方法的收益表现，在测试期间的每个交易日  $t+1$ ，我们用股票预测方法模拟一个交易员进行如下的交易：

1. 当市场在交易日  $t$  收盘时，交易员利用该方法得到预测，即每只股票的预

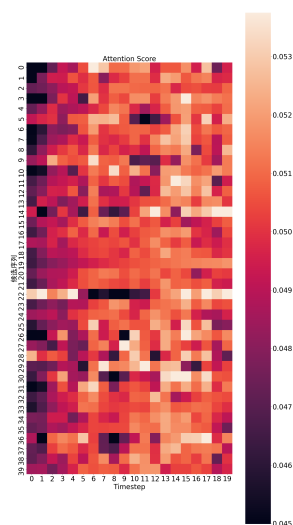


图 3-6: AFC\_GRU 算法内注意力分布结果示例

期收益率排名列表。交易员购买预期收益最高的股票（即排名最高的股票）。

2. 当市场在交易日  $t+1$  收盘时，交易员卖出在交易日  $t$  买入的股票。

在计算累计投资回报率时，我们遵循以下简单的假设：（1）交易员在每个交易日花费相同的金额，以便进行公平的比较。（2）市场具有足够的流动性，买入操作以  $t$  日收盘价成交，卖出价格为  $t+1$  日收盘价。（3）为了方便计算，我们在买卖操作中，忽略了交易成本。

本节在 Top1、Top5、Top10 三种不同的回测策略下研究了我們提出的方法的表现，三种策略分别采取购买前 1、5、10 个预期收益最高股票的投资策略。例如，在前 10 大股票的回测策略中，我们平均分配预算，在每个测试日交易排名前 10 的股票，计算内部收益率的方法是将 10 只选定股票在当前测试日的平均收益率平均。图 3-7 中显示了在上证 50 数据集上，不同的特征表示策略策略在不同回测方法上的性能比较情况，可以看出虽然在 Top1 和 Top5 回测实验时，AFC\_GRU 与 AGRU 的累计回报率相差不大，不过仍旧优于其他模型；而在 Top10 策略回测时，AFC\_GRU 的累计回报率优于其它模型，其优势在第 30 个交易日附近开始显现。这说明我们本章提出的基于注意力机制的时序特征处理算法能够有效对股票特征的数据进行处理，帮助完成后续的预测任务。

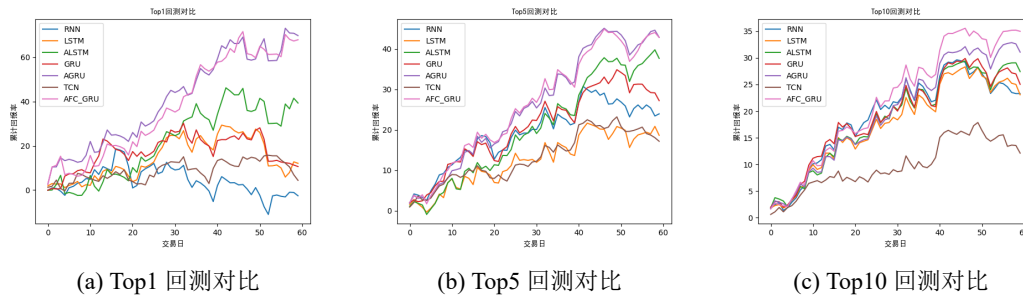


图 3-7: 在上证 50 数据集上, 不同投资模型在各个回测方法下的结果比较

图 3-8则显示了在 NASDAQ 数据集上的回测对比结果。从图中可以看出, 在 Top1 回测时, AFC\_GRU 累计回报率高于其它模型, 在第 40 个交易日左右达到最高收益效果, 之后由于大盘整体走势的影响, 累计回报率虽然下降, 但是最终的累计回报率仍是几个模型中表现最好的。在 Top5、Top10 回测对比中, 我们可以发现在前 20 个交易日, AFC\_GRU 模型的累计回报率走势都是比较的, 但是在后续预测中, 该模型效果相比于 TCN 模型有所下降。这主要是由于随着时间移动, 股票市场的波动变化较大, 这使得之前训练学习到的模型策略并不一定适应当前的市场环境, 所以出现了在后续的较近未来时 (例如此例中的后 20 个交易日), 模型效果表现更优的情况。未来我们计划在实际应用过程中, 增加在线学习模块, 通过利用每个交易日最新的数据, 对模型进行微调, 使得模型不断随着市场的变化而变化。

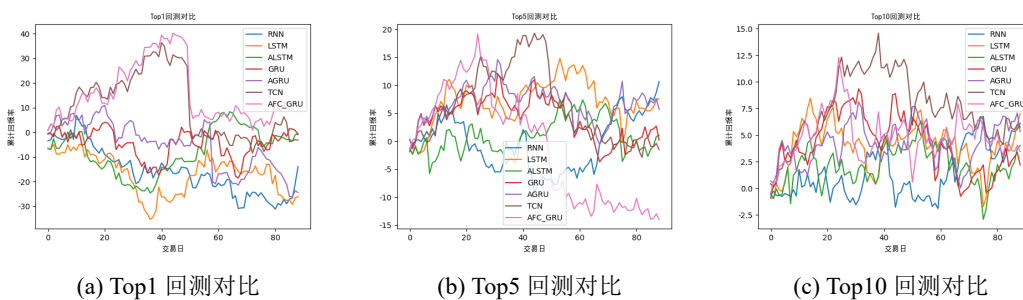


图 3-8: 在 NASDAQ 数据集上, 不同投资模型在各个回测方法下的结果比较

### 3.4.4 消融实验

为了验证模型中注意力层和时序卷积的效果, 以及调节各个超参, 本节分别针对卷积核大小设置、特征结合方式和时间步大小设计了实验, 确定更好的模型结构。

### 3.4.4.1 卷积核大小的设置

在对特征序列进行卷积操作时，卷积核的大小意味着对特征维度综合分析窗口的大小。在本小节，我们对比分析了不同卷积核大小情况下，模型最终排序情况的准确程度，卷积核的大小设置范围为 {1, 2, 3}。表 3-4 中列出了对比卷积核大小的实验结果。其中 Kernel\_Size 为 x 则表示模型中选择的卷积核大小为 x。可以看出，当卷积核大小为 1 时，评价指标值相比其他情况都更好，MAP@5 与 P@5 指标高出近 1.5%。这说明选择卷积核为 1 进行特征卷积操作，能够更好地捕获到特征的变化规律，得到更好的时序特征表示结果。

表 3-4: 卷积核大小对比实验结果

Kernel Size	MAP@5	MAP@10	P@5	P@10	NDCG@10
<b>1</b>	<b>0.15294</b>	<b>0.23474</b>	<b>0.24333</b>	<b>0.38333</b>	<b>0.61598</b>
2	0.13939	0.22517	0.22667	0.38000	0.61071
3	0.13778	0.19609	0.22000	0.35000	0.58750

### 3.4.4.2 Timestep 时间步调节

在时间序列问题中，序列长度会影响到模型的效果，序列过长容易导致模型数据中噪声点过多，序列过短容易出现数据信息不全的问题。在本文讨论的问题中，序列长度即为股票特征的历史时间步大小。时间步是候选股票表示时的关键参数，这意味着每只股票受到历史多长时间特征的影响，例如当我们设 timestep=5，每只股票表示时需要关注到历史前 5 天的特征，当我们设 timestep=10，则需要关注到历史前 10 天交易日的特征。不同的 timestep 使网络拥有不同的信息量，并且也会改变网络的训练复杂度。为了能够找到更好的 timestep 参数，我们分别设置 timestep 的变化范围为 {5, 10, 15, 20, 30} 进行对比实验，实验结果如表 3-5 所示。

对实验结果分析可以发现，当时间步取值过小时，每只股票的历史特征参考太少，导致模型在 Topn 召回和排序效果上都不太好；当时间步超过 20 时，MAP@5 指标呈现上升趋势，但是 MAP@10、P@10 和 NDCG@10 值都有所下降，这表明随着时间步长的增加，模型虽然能够更好的召回前 5 个候选股票，但是整体排序的能力不断下降。同时随着时间步的增加，模型数据量也在不断增加，训练时间也在增长。通过对模型效果和任务需求平衡之后，最终选择时

表 3-5: Timestep 时间步调节对比实验结果

Timestep	MAP@5	MAP@10	P@5	P@10	NDCG@10
5	0.13178	0.17895	0.23333	0.31833	0.57534
10	0.13561	0.19536	0.21667	0.33500	0.58708
15	0.11533	0.18743	0.20000	0.34333	0.58109
20	0.15294	<b>0.23474</b>	0.24333	<b>0.38333</b>	<b>0.61598</b>
30	0.16983	0.22769	<b>0.27000</b>	0.38167	0.61313
40	<b>0.17861</b>	0.23349	0.26667	0.37500	0.61012

间步长为 20。

### 3.4.4.3 特征连接方式

在本模型结构中，我们希望能够通过注意力捕获到更加全面的历史信息特征，而捕获到该特征之后，如何将其与之前全局处理得到的特征进行更好地融合，是本节实验的目的。我们尝试了三种方法，第一种方法是直接取代 (replace)，第二种方法是将两者拼接 (connect)，第三种方法是两个特征拼接后使用线性层降维 (reduction after connect)。表 3-6 中列出了不同特征连接方式的实验结果，可以发现当使用特征直接取代方法时，虽然 P@5 指标有所下降，表示序列前 5 只股票的精确程度略有降低，但是在 MAP@10、P@10 以及 NDCG@10 指标上都有近 1% 的提升，MAP@5 上也有 0.3% 提升，这说明模型在 Top 排序上有更好的结果。所以使用直接取代的方式进行特征表示，特征卷积与注意力结构能够更好地学习到时序特征，进而提升排序任务中的召回和排序效果。

表 3-6: 特征连接方式对比实验结果

Connection Method	MAP@5	MAP@10	P@5	P@10	NDCG@10
<b>replace</b>	<b>0.15294</b>	<b>0.23474</b>	0.24333	<b>0.38333</b>	<b>0.61598</b>
connect	0.14944	0.21526	<b>0.26333</b>	0.37667	0.60403
reduction after connect	0.14628	0.19635	0.23667	0.35167	0.59358

### 3.5 本章小结

候选股票的表示可以被看做是一个时序问题，但是由于股票的特殊属性，这里的时序问题与常见的时序预测任务有所不同。更好的股票表示能够捕获到更详细的股票历史走势情况，也能够帮助我们更好地完成后续的预测排序任务。

在本章，我们首先简单介绍了本论文提出的基于注意力机制的时序特征处理算法，利用特征卷积网络计算特征变化趋势，利用 GRU 结构关注历史时间步的相互影响，得到股票每天新的特征表示，之后再通过注意力层，将特征卷积网络与 GRU 结构相结合，避免历史时间步的特征遗忘问题。我们将本文提出 AFC\_GRU 方法与传统时序模型在不同数据集上进行了对比，发现该方法能够有效提升整体模型的准确性。通过 AFC\_GRU 与股票预测网络 ALSTM 的对比，我们也证明了该算法在股票市场的有效性和适用性。另外在本章中，我们还对卷积核大小、时间步大小等超参进行了对比实验，以确定更精确的模型结构。对实验结果分析得到，当卷积核大小为 1、时间步长为 20 以及采用注意力特征直接取代方法时，模型能够得到更好的效果。

# 第四章 基于注意力解码器的列表 排序算法

统计学习方法的组成分为模型、策略和算法三个要素<sup>[72]</sup>。模型定义好了假设的空间，而策略指明了模型收敛的方向。对于股票预测排序任务而言，我们不仅需要从时序特征处理的角度上对股票数据进行分析，还需要在排序阶段，结合股票特有的属性设计模型和策略。通过结合股票特有属性设计模型，使得任务在更合理的假设空间内，利用学习策略（即损失函数）找到最优模型。本章首先分析了当前排序算法和损失函数的缺点，并以此为出发点，对模型进行调整优化，使其更适用于股票投资分析任务，得到本文提出的排序模型和损失函数，最后设计对比实验和消融实验验证模型效果。

## 4.1 当前排序方法的局限性

常见深度学习排序方法可以根据约束条件分为 Pointwise、Pairwise 和 Listwise 三大类。Pairwise 方法相比于 Pointwise 方法，对序列中的候选项成对地进行了比较，这样在排序打分阶段更注意的是相对顺序。虽然相比于 Pointwise 方法，Pairwise 能够从相对顺序的角度进行偏序预测，但是其 loss 函数中并没有精准地预测排序后的位置信息，这会使得最终的排序结果容易受到噪声的影响，一个错误标注则会引起多个错误，使得整个序列发生巨大变化。同时，Pairwise 方法没有充分利用到同一个查询下，候选项间的内部依赖性。Listwise 方法相比于 Pairwise，考虑到了候选项之间的依赖关系，直接对排序的结果进行优化，更直接地关注排序目标，因此往往效果也是最好的。不过虽然从模型目标上，深度学习排序方法可以分为以上三类，但是从结构上而言，它们都可以简单概括为，首先使用编码器对特征进行处理，之后通过打分函数预测候选项目的排名信息。

Listwise 排序方法比较常见的有 DLCM 模型<sup>[2]</sup>和 PRM 模型<sup>[3]</sup>。DLCM 模型结构如图 4-1 所示，其使用单向 GRU 从候选列表中提出序列中的上下文信

息，再利用上下文信息计算不同候选项在序列中的相关性并进行排序。PRM 模型如图 4-2 所示，其使用类似 Transformer 的编码器结构，利用 Attention 机制并行地计算候选项的相互作用，更新候选项特征，最后通过打分函数得到每个候选项的相关性并排序。同时 PRM 模型为了能够更有针对性的进行个性化排序，在候选项原始特征上添加了本次排序任务的个性化向量，以此达到个性化的结果。可以发现，由于 GRU 结构的串行特性，DLCM 的计算时间复杂度为  $O(n) \times RT$ ，而 PRM 的时间复杂度为  $O(1) \times RT$ ，其中  $n$  表示候选序列长度， $RT$  表示排序任务个数。相比于 DLCM 模型，PRM 能够更有效、更高效地对候选项相互之间的影响进行建模。

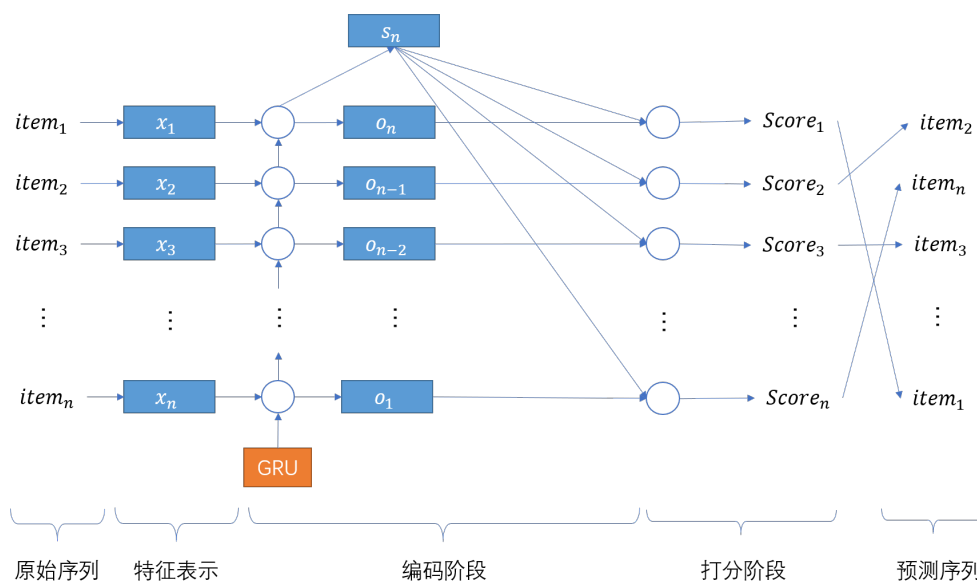
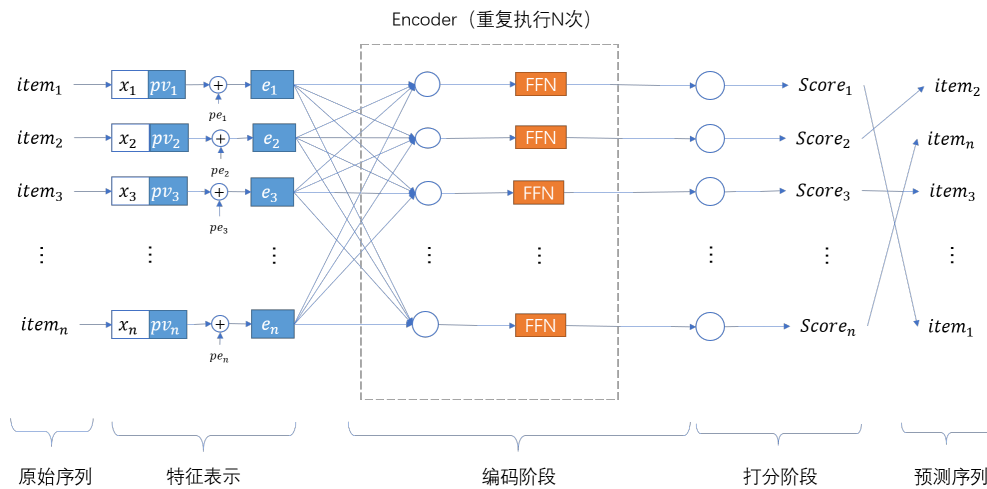


图 4-1: DLCM 模型结构图<sup>[2]</sup>

虽然目前的排序模型都在尝试考虑候选项的相互影响来调整编码阶段结构，从而提升排序结果，但是现阶段排序模型的打分阶段，主要使用简单的线性层来实现，是对整个序列中的候选项独立打分后再统一进行排序。这样的打分过程能够很快地、并行地得到整体排序结果。但是事实上，在打分环节时，候选项之间也存在着相互的影响，这种相互作用会影响整个序列的结果。另外也有一些模型使用 Decoder 串行生成排名列表，例如 Seq2Slate<sup>[73]</sup> 使用指针网络顺序生成排序列表。但是这样串行结构会增加计算复杂度，对预测速度造成影响。在实际应用过程中，模型排序的时效性对系统的使用体验有着至关重要的影响。所以在模型设计过程中，我们既需要达到更好的模型效果，需要考虑到模型效率。

图 4-2: PRM 模型结构图<sup>[3]</sup>

## 4.2 当前排序算法损失函数的局限性

排序算法被广泛应用于实际场景中，为推荐和检索任务生成结果序列，序列中的顺序即代表排序的结果。排序算法通过最小化真实排序与预测排序之间的误差来学习，损失函数计算方法如公式 (4-1) 所示：

$$\mathcal{L} = \sum_{r \in \mathcal{R}} \ell(\{y_i, P(y_i | x_i; \theta) | i \in I_r\}), \quad (4-1)$$

其中  $\mathcal{R}$  表示所有的排序任务集合， $I_r$  表示对于任务  $r$  而言的候选项目集合， $x_i$  表示项目  $i$  的特征向量， $y_i$  表示项目  $i$  的标签，例如点击或不点击， $P$  是由参数为  $\theta$  的排序模型预测的点击率概率， $\ell$  表示  $y_i$  与  $P$  之间的误差计算方法。这是常见的学习排序损失函数，只对单个候选项的预测准确度添加了限制。

作为 Pairwise 排序方法，RankNet<sup>[64]</sup> 更加关注于各个候选项对之间的相对关系，确保网络预测结果中的错误相对关系越少越好。所谓错误的相对关系是指预测顺序与真实顺序不相符，如果模型预测  $U_i$  排在  $U_j$  前面，但真实序列中  $U_i$  排在  $U_j$  之后，那么就记该预测结果为一个错误 pair，RankNet 通过最小化模型预测结果中的错误 pair 更新训练模型。在训练网络时，RankNet 使用概率损失函数计算损失值，模型通过计算得到  $U_i$  以一定的概率  $P$  排在  $U_j$  前面，通过最小化预测概率  $P$  与真实概率的差距更新模型参数。其中，RankNet 使用交叉熵损失函数来衡量两个概率的差距程度，计算如公式 (4-2) 所示：

$$C = -\bar{P}_{ij} \log P_{ij} - (1 - \bar{P}_{ij}) \log (1 - P_{ij}), \quad (4-2)$$

其中  $P_{ij}$  表示模型预测  $U_i$  比  $U_j$  应当排在更前端的概率，如公式 (4-3) 所示：

$$P_{ij} = P(U_i > U_j) = \frac{1}{1 + e^{-\sigma(s_i - s_j)}}, \quad (4-3)$$

$\bar{P}_{ij}$  表示  $U_i$  比  $U_j$  应当排在更前端的真实概率，如果  $U_i$  比  $U_j$  更相关，那么  $S_{ij} = 1$ ；如果  $U_i$  不如  $U_j$  相关，那么  $S_{ij} = -1$ ；如果  $U_i$ 、 $U_j$  与 Query 的相关程度相同，那么  $S_{ij} = 0$ ，具体计算如公式 (4-4) 所示：

$$\bar{P}_{ij} = \frac{1}{2}(1 + S_{ij}). \quad (4-4)$$

然而，学习到一个好的打分函数或相对顺序分类函数是不足够的，我们还应当在排序时注意到项目对之间的相互影响。DLCM 中提出了一个新的 Listwise 损失函数计算方法，名为 **Attention Rank**，更好地利用项目对相互信息。它将对排名列表的评估制定为注意力分配的过程。假设候选项目中包含的信息互斥，排名列表的总信息增益是每个项目增益的累计，之后进一步假设项目的相关性判断分数直接反映了其信息增益，那么在有限时间内最大化总信息增益的最佳策略是将更多的注意力分配给最佳结果，较少关注一般结果，而不关注无关结果。**Attention Rank** 使用模型预测的排名分数计算注意分布，并将其与用相关性判断计算的注意策略进行比较。设相关标签  $y$  表示项目  $i$  的信息增益，那么序列  $R$  最佳的注意力分布策略则如公式 (4-5) 所示：

$$a_i^y = \frac{\psi(y_{(q,d_i)})}{\sum_{d_k \in R_q^n} \psi(y_{(q,d_k)})}, \quad (4-5)$$

其中  $d_i$  表示排序序列  $R_q^n$  的一个候选项， $y_{(q,d_i)}$  表示在查询  $q$  下，候选项  $d_i$  的真实注意力权重， $\psi(x)$  是一个修正函数，当  $x > 0$  时等于  $e^x$ ，否则等于 0。相似的，我们可以通过模型预测得分，计算模型的注意力分布情况，然后将模型的注意分布与最佳注意分布之间的交叉熵作为序列  $R_q^n$  的损失，计算如公式 (4-6) 所示：

$$\ell(R_q^n) = - \sum_{d_i \in R_q^n} (a_i^y \log(a_i^s) + (1 - a_i^y) \log(1 - a_i^s)) \quad (4-6)$$

其中  $d_i$  表示序列  $R_q^n$  的一个候选项， $\alpha_i^y$  表示候选项的真实注意力权重值， $S$  表示模型估计的结果，则  $\alpha_i^s$  为候选项的预测注意力权重值。

**Attention Rank** 并不直接预测单个项目的相关性程度，而是更关注排序序列中每个结果的相对重要性，例如，一个不相关结果列表中的一般项目可能比一个完美结果列表中的优秀项目得到更多的关注。**Attention Rank** 的主要优点在于能够对整体预测结果进行简单高效的评估，希望序列中所有候选项的相关性程度估计值都越接近真实值。但在实际股票排序任务中，序列前端的候选项对整体预测结果影响更大，才是后续买入卖出操作的主要依据。例如之前在图 1-1 的分析中，最终的模型利润主要受到序列前端股票的影响，在真实涨跌幅值越大的股票上进行买入卖出操作，才能够保证模型得到更大的利润。因此我们希望模型更关注序列前端的结果，预测的序列头部排列情况越准确越好，而对序列后半部分的要求并不高，即使他们之间的相互顺序存在逆序情况。

## 4.3 基于注意力解码器的排序算法设计

根据之前对排序模型分析，Listwise 排序方法相比于其他方法，直接对排序结果进行了约束，更符合我们股票预测任务的初衷。另外在 loss 上，采用注意力分配的排序损失相比于其他损失函数，将约束限制从单个候选项上扩展到整个候选队列，也更加符合股票预测任务。同时，结合之前指出的目前模型和损失函数所具有的缺点，在本节我们分别从模型角度和 loss 角度上设计了股票排序模型。

### 4.3.1 Decoder 模块

传统 Transformer<sup>[30]</sup> 中的 Decoder 部分与 Encoder 部分结构类似，由若干个层堆砌而成，上层接收下层的输出作为输入，具体如图 4-3 所示。Decoder 每一层由两个注意力结构拼接一个前馈网络结构组成，其中两个注意力结构分别是多头自注意力（Masked Multi-Head Self-Attention）和编码器-解码器多头注意力（Encoder-Decoder Multi-Head Attention）。多头自注意力是针对目标序列中不同输出的注意力计算，并添加了 mask 掩码，主要是为了防止 Decoder 在训练时接触到未来信息。编码器-解码器多头注意力是在目标序列和 Encoder 输出的隐藏状态之间计算注意力权重，其目的是为了得到原输入的隐藏状态对当前输出的重要性，然后输出最终的预测结果。

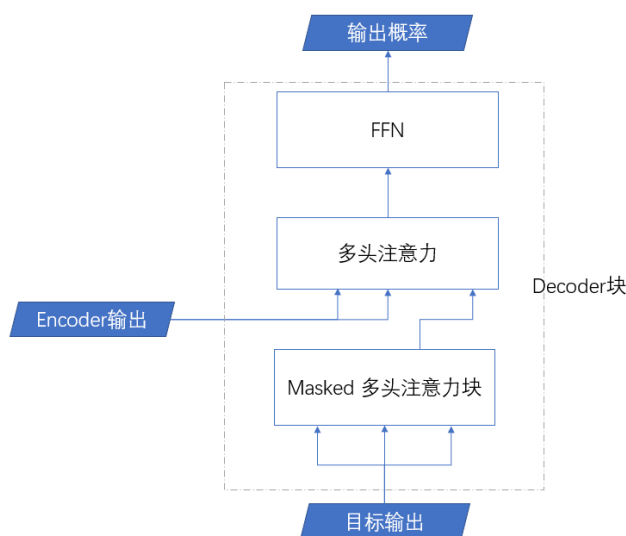


图 4-3: Decoder 模块的结构

整个 Decoder 设计，通过 Multi-Head Self-Attention 模块让翻译过程中每个词的表示都有侧重点地包含了其他已知目标词汇的信息，并通过 Encoder-Decoder Attention 模块，让模型凭借序列前几个词的信息来预测下一个词汇的同时，会有目的地与源语言进行对比，得到当前词汇的生成更依赖源语言中哪些词汇，这样更符合人类在翻译时的习惯。

另外，Transformer 采用了带有遮蔽的自注意力方法，具体实现的方法如图 4-4所示。这样的遮蔽操作除了能避免还未预测到的后续信息遭到泄露，还能够实现模型并行化训练，每个时刻的输入不再依赖上一时刻的输出，而是依赖正确的样本，而正确的样本在训练集中已经全量提供了。但是在测试阶段，这样的 Decoder 结构需要串行执行，因为序列  $i$  位置的输出依赖  $i + 1$  位置的输出。近期提出的 Informer 模型<sup>[74]</sup> 解决了串行测试这一问题。

	原始分数矩阵		遮蔽矩阵		遮蔽后分数矩阵																																																				
	<table border="1" style="border-collapse: collapse;"> <tr><td>&lt;start&gt;</td><td>0.7</td><td>0.1</td><td>0.1</td><td>0.1</td></tr> <tr><td>I</td><td>0.1</td><td>0.6</td><td>0.2</td><td>0.1</td></tr> <tr><td>am</td><td>0.1</td><td>0.1</td><td>0.7</td><td>0.1</td></tr> <tr><td>fine</td><td>0.1</td><td>0.2</td><td>0.2</td><td>0.5</td></tr> </table>	<start>	0.7	0.1	0.1	0.1	I	0.1	0.6	0.2	0.1	am	0.1	0.1	0.7	0.1	fine	0.1	0.2	0.2	0.5	+	<table border="1" style="border-collapse: collapse;"> <tr><td>0</td><td>-inf</td><td>-inf</td><td>-inf</td></tr> <tr><td>0</td><td>0</td><td>-inf</td><td>-inf</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>-inf</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	-inf	-inf	-inf	0	0	-inf	-inf	0	0	0	-inf	0	0	0	0	=	<table border="1" style="border-collapse: collapse;"> <tr><td>0.7</td><td>-inf</td><td>-inf</td><td>-inf</td></tr> <tr><td>0.1</td><td>0.6</td><td>-inf</td><td>-inf</td></tr> <tr><td>0.1</td><td>0.1</td><td>0.7</td><td>-inf</td></tr> <tr><td>0.1</td><td>0.2</td><td>0.2</td><td>0.5</td></tr> </table>	0.7	-inf	-inf	-inf	0.1	0.6	-inf	-inf	0.1	0.1	0.7	-inf	0.1	0.2	0.2	0.5
<start>	0.7	0.1	0.1	0.1																																																					
I	0.1	0.6	0.2	0.1																																																					
am	0.1	0.1	0.7	0.1																																																					
fine	0.1	0.2	0.2	0.5																																																					
0	-inf	-inf	-inf																																																						
0	0	-inf	-inf																																																						
0	0	0	-inf																																																						
0	0	0	0																																																						
0.7	-inf	-inf	-inf																																																						
0.1	0.6	-inf	-inf																																																						
0.1	0.1	0.7	-inf																																																						
0.1	0.2	0.2	0.5																																																						

图 4-4: 遮蔽自注意力方法中的遮蔽实现

在股票预测排序任务中，我们首先采用第 3 章提出的基于注意力机制的时序特征处理算法对股票特征进行融合表示，之后在编码阶段使用 Encoder 模块

计算股票序列中的相互影响，使得每只股票特征都有侧重地包含了其他股票的信息，最后在打分阶段，我们采用 Decoder 模块对候选股票序列进行并行打分，在打分时也从整体序列的角度进行考量。由于对于候选股票数据而言，其并不像翻译任务一样具有强烈的先后顺序，不需要 mask 操作即可并行化计算。因此在本文提出的算法中，Decoder 模块使用两层简单的 Attention 模块，计算不同分数之间的注意力影响，并对最后的分数进行更新。通过这样的计算，我们希望在股票预测的打分阶段，模型能够得到股票  $i$  在整个候选序列中综合分析后的分数值。最终我们的模型结构如图 4-5 所示。

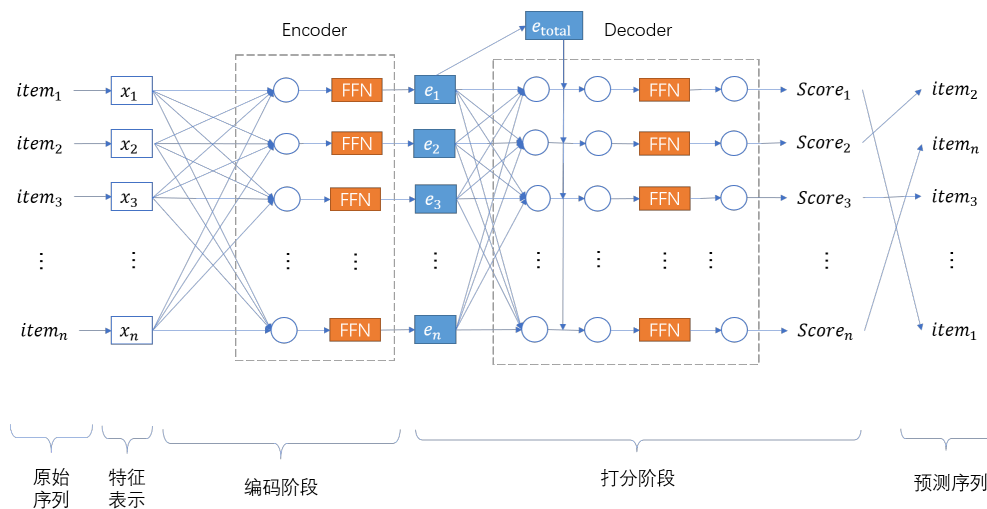


图 4-5: 采用 Attention Decoder 的 Listwise 排序算法

排序模型的输入是候选序列中的股票特征  $X \in \mathbb{R}^{Batch \times Listsize \times FeatureSize}$ ，每个候选股票由在经过第 3 章中提出的时序特征处理算法得到的一维特征表示，其中  $Batch$  表示批处理大小， $Listsize$  表示候选序列的长度， $FeatureSize$  表示股票的特征维度。在 Encoder 结构中，模型通过 Attention 模块，对候选序列中的股票特征进行注意力计算，并更新股票特征值，得到新的股票特征表示  $E \in \mathbb{R}^{Batch \times Listsize \times FeatureSize}$ 。在 Decoder 结构中，模型首先通过一个 Attention 模块更新特征表示  $X \in \mathbb{R}^{Batch \times Listsize \times FeatureSize}$ ，之后再通过一个 Attention 模块，计算当前特征  $X$  和 Encoder 输出  $E$  之间的相互影响  $Attn \in \mathbb{R}^{Batch \times Listsize \times Listsize}$ ，并更新股票特征，最后打分得到相关性结果  $S \in \mathbb{R}^{batch \times listsize \times 1}$ ，得到最终的序列。相比于其他排序模型，本文提出的模型在考虑到候选项相互影响的同时，并没有增加时间复杂度，其值仍为  $O(1) \times RT$ ，其中  $RT$  表示排序次数。

### 4.3.2 Loss 设计

在股票预测排序任务中，我们更希望预测的序列结果中，前端结果越准确越好，对于序列尾部的候选项之间的顺序要求并不高。这主要是因为我们后续的买入操作，主要是在序列前端的候选股票上进行，只要排序结果能够将真实的前端候选股票进行召回，即可达到提高回报率的目的。

根据 4.2 节中分析的当前损失函数的优缺点，我们在 Attention Rank 的基础上添加 mask，得到 Mask Attention Rank (MAR) 损失函数，让模型更加关注于回报率更大的候选股票。Mask 的窗口大小可以根据项目需求中的关注比例进行调节，例如当序列长度为 10，关注比例为前 33% 时，mask 的情况如图 4-6 所示。图中的蓝色部分表示 loss 计算过程中考虑 Attention Rank 的候选项，白色部分则为被 mask 掉的候选项，他们的真实值直接被标记为 0，而不是在整个序列中的注意力分布值。这意味着原始的 loss 计算，是作用在序列中所有候选项上的，但是 mask 之后，loss 计算则重点关注更相关的候选项排序结果，以及相关候选项与不相关候选项的划分。具体的 loss 计算如公式 (4-7) 所示。

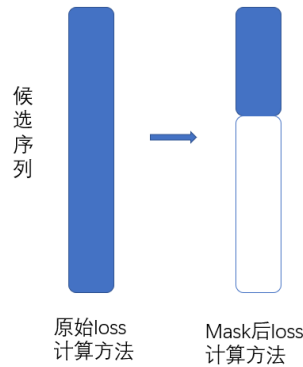


图 4-6: mask 方法

$$\begin{aligned}
 y^{mask} &= y \times mask \\
 a_i^{y^{mask}} &= \frac{\psi(y_{(q,d_i)}^{mask})}{\sum_{d_k \in R_q^n} \psi(y_{(q,d_k)}^{mask})} \\
 \ell(R_q^n) &= - \sum_{d_i \in R_q^n} \left( a_i^{y^{mask}} \log(a_i^S) + (1 - mask) \left( 1 - a_i^{y^{mask}} \right) \log(1 - a_i^S) \right)
 \end{aligned} \tag{4-7}$$

其中,  $mask$  为二值矩阵, 用来区分 Topn 序列与后续序列。y 表示真实的候选股票涨跌幅值,  $y^{mask}$  表示按照一定 Topn 比例对 y 进行 mask 操作后的真实值。 $\psi(x)$  是一个修正的指数函数, 当  $x>0$  时等于  $e^x$ , 否则等于 0, 我们使用  $\psi(x)$  计算得到  $a_i^{y^{mask}}$ , 表示在这次排序任务  $R_q^n$  中, Topn 候选项上的注意力分布情况。最后  $\ell(R_q^n)$  表示 loss 计算方法, 我们使用交叉熵损失函数, 对预测分布  $a_i^S$  和真实分布  $a_i^{y^{mask}}$  进行约束, 前半部分  $a_i^{y^{mask}} \log(a_i^S)$  主要是对 Topn 项的注意力分布情况进行约束, 后半部分  $(1 - mask) \left(1 - a_i^{y^{mask}}\right) \log(1 - a_i^S)$  主要是对序列尾部进行约束。

### 4.3.3 算法流程

基于前两节的介绍, 本节重点讨论我们提出的基于注意力解码器的 Listwise 排序算法 (Attention Encoder-Decoder Ranking Model, AEDRM)。在排序时, 我们首先使用 Encoder 部分进行候选股票编码, 之后在打分阶段采用并行的 Decoder 打分结构, 对候选股票进行打分排序。这部分的排序模型和之前的特征表示层共同组成了整个的股票预测任务。之后在网络训练时, 我们采用本文提出的 MAR 损失函数, 决定模型的优化方向。通过添加 Decoder 部分的排序算法, 我们解决了排序模型在打分阶段时, 没有综合考虑候选序列的问题, 另外通过 MAR 损失函数, 我们使得模型结果更加关注于真实序列前端的候选项, 避免序列尾部排序的影响。具体的排序算法流程如算法 4.1 所示。

基于注意力解码器的 Listwise 排序算法是在股票时序特征处理之后, 算法的输入为  $X \in \mathbb{R}^{Batch \times ListSize \times FeatureSize}$ , 表示候选股票序列的特征。之后首先通过 Encoder 部分对其进行相互之间注意力的计算, 更新每只股票的特征表示  $X_{new} \in \mathbb{R}^{Batch \times ListSize \times FeatureSize}$ , 再通过 Decoder 中的两个 Attention 模块, 在打分时也注意到候选股票相互之间的影响, 最终得到打分的结果  $L \in \mathbb{R}^{Batch \times ListSize \times 1}$ 。

整个股票预测任务模型的训练流程如算法 4.2 所示, 股票首先通过 embedding 层, 利用时序特征处理模型对股票历史时间步内的特征进行融合, 之后通过排序模块得到最终的排序结果。将预测排序结果与真实排序结果进行比较, 根据 MAR 计算 loss 值, 通过将 loss 反向传播, 再利用优化器对模型参数进行更新, 使得模型不断收敛, 达到准确预测排序的效果。

**算法 4.1** 基于注意力解码器的 Listwise 排序模型算法 (AEDRM) 流程

输入: 序列股票特征  $X$

输出: 排序结果  $L$

Encoder 处理部分

1:  $out, attn = attn\_net(X)$

使用注意力计算方法, 得到候选股票序列两两之间的注意力程度, 并进行特征的更新

2:  $out = FFN(out)$

前馈网络计算

Decoder 处理部分

3:  $memory = out$

4:  $q = Linear(out), k = Linear(out), v = Linear(out)$

计算得到  $q, k, v$

5:  $attn\_score = q \times k$

6:  $out = attn\_score \times v$

7:  $out = FFN(out)$

更新股票的特征表示

8:  $s_q = Linear(out), s_k = Linear(memory), s_v = Linear(memory)$

9:  $attn\_score = s_q \times s_k$

10:  $out = attn\_score \times s_v$

11:  $out = FFN(out)$

使用 Encoder 输出与当前特征计算 Attn, 再次更新股票表示

12:  $L = score(out)$

打分排序

13: **return**  $L$

## 4.4 实验与分析

为了验证本文提出的排序算法的有效性, 本节中设计了不同的对比实验和消融实验, 从排序结果和股票赢利能力两个角度进行分析。

### 4.4.1 实验设置

#### 4.4.1.1 数据集

与第 3.4.1.1 小节一致, 在本节实验中, 我们使用到的数据集为上证 50 股票数据和 NASDAQ 数据集, 分别作为国内市场和美国市场的代表性数据集。

股票交易策略采用与第 3.4.1.2 小节一致的 T+1 策略。我们任务的目标是在交易日  $t$  时, 根据最近  $S$  个交易日的每日历史数据, 预测下一个交易日  $t+1$  的股票排名。下一天预测收益率越大的股票的排名应当位于越前端。由于股票的

**算法 4.2** 基于 AFC\_GRU 和 AEDRM 的股票预测排序任务训练流程

输入：候选股票特征  $X$

真实涨跌幅值  $y$

输出：预测模型  $Model$

```

1: for  $epoch = 1$  to  $EPOCHS$  do
2:   for  $x, y$  in  $Dataloader$  do
3:      $x = input\_embedding(x)$ 
       对股票时序特征进行处理
4:      $out = Rank(x)$ 
       使用排序模型对股票候选序列进行排序
5:      $loss = MAR(out, y)$ 
       计算 loss 值
6:      $loss\ backward$ 
       loss 反向传播
7:      $optimizer\ step$ 
       利用优化器对参数进行更新
8:   end for
9: end for

```

收益率表示股票的预期收益，我们将股票  $i$  的真实排序得分设为其一日收益率  $r_i^t = (c_i^{t+1} - c_i^t) / c_i^t \times 100\%$ ，其中  $c_i^t$  表示股票  $i$  在第  $t$  天的收盘价。

股票的特征选择和预处理方法与第 3.4.1.3 小节一致。每个股票的特征除了它们在每一天内的开盘价（open）、收盘价（close）、最高价（high）等原始市场特征值以外，还有构造的每日趋势特征等。在预处理阶段，我们使用归一化方法不同特征进行处理。经过归一化处理，各个指标被处理成为同一量级，提升了可比性，同时降低了噪声数据影响，加快了网络的收敛速度。

#### 4.4.1.2 基准

为了验证本章提出模型的性能，我们分别选择了一个 Pointwise 的排序网络和两个 Listwise 的排序网络作为基准，进行对比实验。

- **PointNet**. PointNet 是一个利用线性层实现的简单 Pointwise 排序模型。它使用全连接层，直接根据单只股票的输入特征预测排名概率。
- **DLCM**<sup>[2]</sup>. DLCM 是一个用于信息检索的排序模型。它使用 GRU 学习上下文信息，表示任务向量，再计算任务向量在每个候选项特征上的注意力分布情况，并据此得到排序结果，得到一个更强的排序模型。
- **PRM**<sup>[3]</sup>. PRM 是一个用于推荐和信息检索的个性化重排模型。它使用

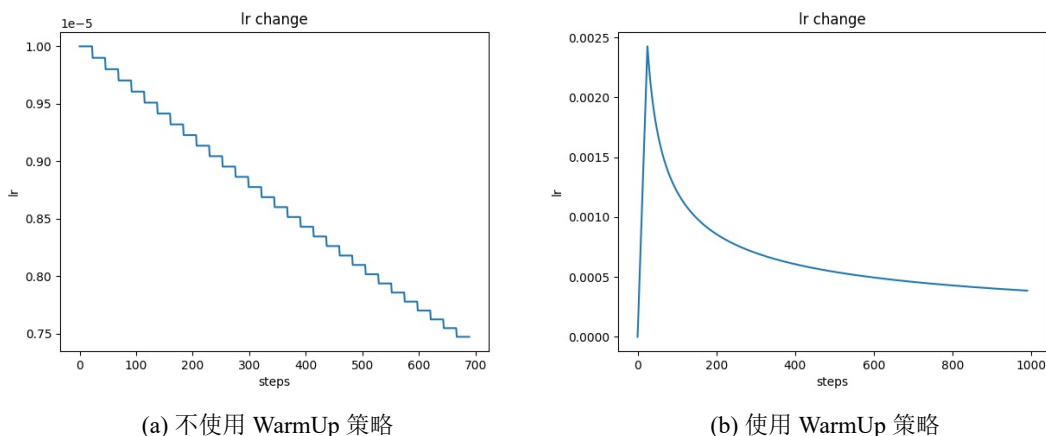


图 4-7: 不使用 WarmUp 和使用 WarmUp 策略时学习率变化情况

Transformer 结构有效地编码候选列表中所有项目的信息，从而优化整个推荐列表，同时在模型中添加个性化嵌入向量得到个性化排序结果。

#### 4.4.1.3 参数设置

为了实验的公平性，我们使用深度学习框架 Pytorch<sup>[69]</sup> 实现所有模型，并使用 Adam 优化器<sup>[70]</sup> 对其进行优化。上证 50 数据集的隐藏向量维度设置为 52。训练批量大小设置为 60。学习率参考 Transformer<sup>[30]</sup> 使用 WarmUp 方式进行变化，具体的计算如公式 (4-8) 所示，最初学习率会随着训练步数线性增加，在到达最大值后，学习率则随着步数的增加成比例减少，这里的最大值与特征维度  $d_{model}$  成反比，并且我们使用  $warmup\_steps = 25$ ，具体的变化区别如图 4-7 所示。这样的学习率变化策略增加了训练的时间，并且避免在训练开始阶段，由于 loss 过大导致的梯度变化过大问题，帮助模型在后续有效地收敛。为了缓解过拟合问题，我们使用了 Dropout 技术，将神经元丢弃概率设置为 0.2。本节提出的算法主要是在排序阶段进行模型改进，因此在时序特征处理部分，不同模型均采用章中提出的基于注意力机制的时序特征处理方法。

$$lrate = d_{model}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5}) \quad (4-8)$$

#### 4.4.1.4 评估指标

我们通过三个指标评估模型排序结果，分别是准确率 P 值、MAP 值和 NDCG 值。P@N 值的变化范围是 [0, 1]，其值越大表示 TopN 中预测准确程度越好，MAP@N 值的变化范围是 [0, 1]，其值越大模型在 TopN 上准确和召回的更加准确。NDCG@N 值的变化范围是 [0, 1]，其值越大模型在 TopN 上排序更正确。另外除了对模型的排序结果进行评估以外，我们对不同模型的赢利能力进行了测试对比，以此来分析不同模型对于股票投资的指导性意见是否有效。

#### 4.4.2 实验结果

本节通过对比本章所提出的排序模型 AEDRM 与其他排序模型在股票预测任务上的性能，验证模型的有效性。

首先我们在表 4-1 中给出不同模型在上证 50 数据集上的对比实验结果，可以看出：（1）本章提出的 AEDRM 模型在 MAP@5、MAP@10、P@5 和 P@10 指标上都远优于其他模型，其中 MAP@5 相比于 PRM 模型提升近 3%，MAP@10 提升近 2%，这意味着该排序模型能够更好地召回候选序列中前端的序列情况，（2）本章提出的 AEDRM 模型在整体 NDCG 值只略高于其他模型，但在 NDCG@5、NDCG@10 上相比 PRM 模型提升 0.5% 左右，说明该模型不仅能够得到更好的整体排序效果，并且也加强了对股票前端序列的关注效果。

表 4-1: 在上证 50 数据集上，不同排序方法的结果对比

	MAP@5	MAP@10	P@5	P@10	NDCG@5	NDCG@10	NDCG
PointNet	0.04689	0.08460	0.09667	0.20167	0.51545	0.55002	0.82415
DLCM	0.11133	0.17544	0.19333	0.31333	0.56719	0.58600	0.83883
PRM	0.12744	0.18760	0.18667	0.30833	0.58768	0.59288	0.84481
<b>AEDRM(ours)</b>	<b>0.15894</b>	<b>0.20874</b>	<b>0.25000</b>	<b>0.35500</b>	<b>0.59313</b>	<b>0.60420</b>	<b>0.84570</b>

在 NASDAQ 数据集上的实验结果如表 4-2 所示。可以看出，NASDAQ 数据集作为其他国家股票市场的代表，本章提出的模型依旧能够在各个指标上都优于其他模型结果，相比于 PRM 模型，AEDRM 在 MAP@5、P@5 和 P@10 提升近 2%，NDCG@5 和 NDCG@10 提升近 2%，MAP@10 与 NDCG 提升近 1%。这说明 AEDRM 模型能够很好地召回候选序列中位于前端的股票，并对其进行排序。

表 4-2: 在 NASDAQ 数据集上, 不同排序方法的结果对比

	MAP@5	MAP@10	P@5	P@10	NDCG@5	NDCG@10	NDCG
PointNet	0.11000	0.12963	0.20449	0.26180	0.51405	0.54507	0.82107
DLCM	0.10880	0.14021	0.19326	0.28764	0.52891	0.55831	0.82937
PRM	0.12375	0.14638	0.21798	0.29326	0.51545	0.54845	0.82406
<b>AEDRM(ours)</b>	<b>0.14801</b>	<b>0.15515</b>	<b>0.23146</b>	<b>0.31685</b>	<b>0.53533</b>	<b>0.56884</b>	<b>0.83303</b>

### 4.4.3 赢利能力测试

与第 3.4.3 小节类似, 我们基于 T+1 策略, 在某个时间段内, 根据模型预测结果进行仿真交易, 通过对比模型的累计回报率进行性能的评估。

本节在 Top1、Top5、Top10 三种不同的回测策略下研究了我們提出的方法的表现。三种策略分别是购买前 1、5、10 个预期收益最高的股票。例如, 在 Top5 回测策略中, 我们平均分配预算, 在每个测试日买入排名前 5 的股票, 并在下一个测试日卖出。每天的收益率值通过将 5 只选定股票在下一个测试日的真实收益率进行平均计算求得。

上证 50 数据集上的回测结果如图 4-8 所示。可以发现, 在 Top1 回测任务中, 本章提出的模型结果与 PRM 结果不相上下; 但是在 Top5 和 Top10 回测任务中, 本章提出的 AEDRM 模型优于其他模型, 能够得到更高的回报率。这说明本章提出的 AEDRM 排序模型能够很好地对回报率更大的股票进行召回排序, 从而达到提高收益的结果。

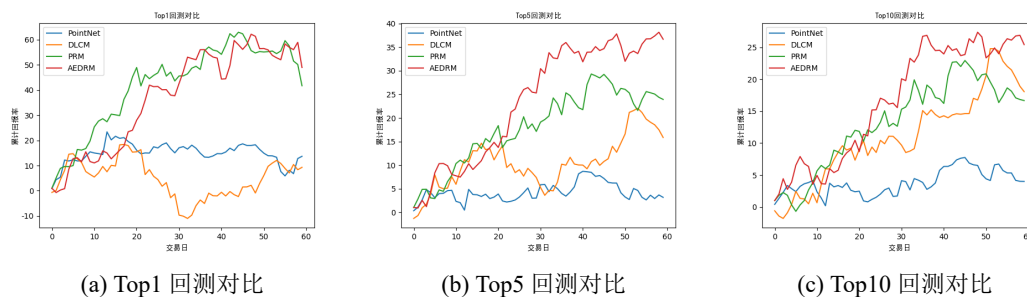


图 4-8: 在上证 50 数据集上, 不同投资模型在各个回测方法下的结果比较

### 4.4.4 消融实验

为了得到更好的模型结构, 本节我们针对编码器个数、loss 设计和验证集使用方法进行了实验, 对模型进行调节。

#### 4.4.4.1 编码器个数对比实验

在 Transformer 中，编码器利用注意力机制去优化每个候选项的不同特征部分，从而进行特征提取过程。编码器个数对网络捕获特征相关的能力存在影响。因此在本文中，我们针对编码器个数进行了对比实验，编码器个数的选择范围为 {1, 2, 3}，最终的实验结果如表 4-3 所示。对比可以发现，随着编码器个数的增加，P@5 指标值呈现递增变化，MAP@10 和 P@10 指标在  $block = 2$  时取到最大值，而整体 NDCG 指标在  $block = 1$  时效果更好。增加编码器个数并没有对整体排序结果带来明显提升，但是会增加网络中的参数量，因此最终我们设置排序模型中编码器个数为 1。

表 4-3: 不同编码器个数设置下，模型的结果对比

Block	MAP@5	MAP@10	P@5	P@10	NDCG@5	NDCG@10	NDCG
1	<b>0.12744</b>	0.18760	0.18667	0.30833	<b>0.58768</b>	<b>0.59288</b>	<b>0.84481</b>
2	0.12694	<b>0.18955</b>	0.20667	<b>0.32000</b>	0.58333	0.59212	0.84274
3	0.12617	0.18543	<b>0.21000</b>	0.31000	0.58217	0.59178	0.84191

#### 4.4.4.2 Loss 设计对比

在股票预测任务中，不同的 loss 设计会使模型在训练过程中拥有不同的优化方向，最终导致模型的效果有所差别，为了验证本章提出的 loss 的有效性，我们对比分析了几种不同 loss 情况下，模型的效果和收敛速度。几种 loss 设计分别为：

##### (1) loss\_point.

这种 loss 使用 pointwise 方法进行模型训练，模型输出为每只股票的涨跌幅预测值。loss 通过计算预测值与真实涨跌幅之间的交叉熵损失值，作为网络训练约束。

##### (2) loss\_attn.

这种 loss 将预测值和真实值都转换为对应的注意力分布情况，并根据两个分布之间的不一致性进行计算，得到 loss 值。

##### (3) loss\_01.

这种 loss 不关注与具体的排序情况，更着重于模型对涨跌幅更大的股票的召回，通过将真实值按照 topn 进行 1、0 划分，值为 1 则表示该候选股票

的真实排序结果在 `topn` 内部，值为 0 则表示不在。该 `loss` 方法以此为目标值，进行网络的训练。

#### (4) `loss_mask_attn`.

该 `loss` 即为本文提出的 Mask Attention Rank 损失值计算方法，通过对真实值进行 `mask`，让模型更关注 `topn` 的排序情况，之后在依据注意力分布情况，对两分布之间的差异值计算，得到 `loss` 值。

最终的实验结果如表 4-4 所示，经过分析可以发现本文提出的 Mask Attention Rank 计算方法虽然在 `NDCG@10` 指标上，比 `loss_attn` 方法略差一点，`P@5` 指标比 `loss_01` 方法也略低一些，但是 `MAP` 值、`NDCG@5` 和 `NDCG` 值上，相比于其他方法都是最优的，其中 `MAP@10` 和 `P@10` 相比于 `loss_01` 提升超过 1%。这说明本文提出的 Mask Attention Rank Loss 计算方法能够很好地对 TopN 的候选股票进行召回排序，并且整体性能也更加稳定，符合我们的设计预期。

表 4-4: 不同损失函数训练情况下，模型的结果对比

	MAP@5	MAP@10	P@5	P@10	NDCG@5	NDCG@10	NDCG
<code>loss_point</code>	0.06039	0.09549	0.11000	0.22333	0.50888	0.54391	0.82165
<code>loss_attn</code>	0.14822	0.22886	0.25333	0.37667	0.59875	<b>0.60941</b>	0.84717
<code>loss_01</code>	0.15800	0.21839	<b>0.25667</b>	0.36000	0.59026	0.60167	0.84415
<b><code>loss_mask_attn</code></b>	<b>0.15894</b>	<b>0.23009</b>	0.25333	<b>0.37833</b>	<b>0.60003</b>	0.60915	<b>0.84778</b>

#### 4.4.4.3 验证集使用方法实验

对于股票数据而言，不同时间段由于市场波动，数据集的分布可能不同，但呈现一定的规律性，并且股票的未来走势受到较近历史的数据影响更大。如若按照常用深度学习实验数据划分方法，将整体数据按照 8:1:1 进行训练集、验证集和测试集划分，则可能会忽略掉股票特征中强烈的时序相关性。更邻近测试集的时间段内的数据对测试集内数据的走势更有影响，并且两者的分布也更加接近，所以我们不能简单地根据时间节点进行数据集划分。如何有效利用数据集便是本次实验的目的，为了让模型对有限的数据集上进行合理使用，更好地预测未来走势，我们对比设计了不同的验证集使用方法，分别是：

##### (1) 传统验证集 (OriDev) .

该方法按照时间节点划分训练集、验证集和测试集。我们使用该方法作为基准。

##### (2) 不设置验证集 (NoDev) .

该方法省略掉验证集，只需要将数据集划分为训练集和测试集，在所有训练数据上进行模型的训练。

### (3) 随机抽取验证集 (RdmDev) .

为了避免训练集、验证集和测试集中，数据分布差异过大的情况，我们首先根据时间节点划分出 train-dev 集和测试集，之后从 train-dev 集中按照随机等比例划分的方法得到训练集和验证集，降低了两者之间的数据分布差异，之后再行模型的训练。

### (4) 微调 (Fine-tune) .

该方法是针对方法 1 的改进，在方法 1 训练完成之后，我们再次使用验证集对模型进行微调，这样使得模型受到更多的邻近历史数据影响，能够更有效地关注到与测试集中数据更相关的历史数据，得到更好的预测结果。

最终的实验结果如表 4-5 所示。可以发现，使用随机抽取验证集的方法时，能够有效缓解由于时间导致的数据分布不一致问题，从而使得网络能够更好地学习出时序特征表示和排序的规律，提升模型的效果。不设置验证集和微调的方法，虽然相比于传统验证集思路，都提升了模型的各个指标，但是效果仍旧不如随机抽取验证集的方法。因此最终我们在模型训练时，选择随机抽取验证集的方法，选择模型。

表 4-5: 不同验证集使用方法下，模型的结果对比

	MAP@5	MAP@10	P@5	P@10	NDCG@5	NDCG@10	NDCG
OriDev	0.15178	0.20588	0.23000	0.35000	0.56657	0.59186	0.84104
NoDev	0.15317	0.22764	0.25000	0.37167	0.59361	0.60651	0.84529
<b>RdmDev</b>	<b>0.15894</b>	<b>0.23009</b>	<b>0.25333</b>	<b>0.37833</b>	<b>0.60003</b>	<b>0.60915</b>	<b>0.84778</b>
Fine-tune	0.14489	0.21669	0.23333	0.36000	0.58308	0.59524	0.84257

## 4.5 本章小结

在本章中，我们首先分析了当前排序模型和排序损失函数存在的问题。以解决这些问题为目的，并结合股票预测任务需求，本文使用 Attention Decoder 的排序模型和 Mask Attention Rank 损失函数计算方法，提出了基于注意力解码器的列表排序模型 AEDRM。通过实验验证，我们发现 AEDRM 在保证网络并行性和运行效率的同时，使得模型在打分阶段也注意到了候选序列中项目的相互作用。而其带有 mask 的 loss 计算方法 MAR，能够使模型更加关注强相关股票，满足实际使用需求。



# 第五章 股票分析推荐系统

量化交易和投资决策是非常复杂的金融任务，不同投资者们和金融机构都希望能够对股票数据全面分析后，再进行买入和卖出操作。那么一个合理准确的股票分析与推荐系统能够为投资者们和金融机构提供更全面的指向性建议。所以在本章我们出于此目的，设计了一个股票分析推荐系统，将本文提出的基于注意力机制的时序特征处理方法和 listwise 股票排序算法应用在股票预测排序系统中，从而提供准确的预测排序结果，以便投资者们参考分析。本章我们将从相关背景进行介绍，之后对系统的需求进行分析，介绍系统的架构，以及展示最终的系统效果。

## 5.1 相关背景

随着我国经济的快速发展，股票投资成为越来越受欢迎的投资手段，但股票市场受到诸多因素的影响，走势波动较大。这样的风险投资在给投资者们带来巨大回报的同时，也容易带来相当大的损失。为了减少损失和风险，股票分析与推荐逐渐成为热点。

股票数据作为强烈的时序相关数据，连续时间段内的数据变化具有一定的连贯性。投资者们会对股票历史数据和当前的市场环境进行分析，试图发现股票市场中潜在的交易模式，以预测股票未来走势。但由于股票市场的波动受到诸多因素的影响，人为的分析很可能不够全面。所以为了更好地帮助投资者们进行数据分析，并对其后续的投资计划提出指导性建议，一个有效的股票数据分析和推荐系统能够提供诸多便捷。

通过上述分析，我们设计并实现了一个股票分析推荐系统，提供更便捷的股票分析功能。系统中的数据分析模块，能够向用户展示指定时间段内的价格走势变化情况。系统中的推荐排序模型，能够根据股票特征进行  $t+1$  天涨跌情况的预测排序，对后续的投资计划进行指导性建议。系统还能够回顾股票市场的历史数据，模拟不同的投资方案在历史市场中的应用效果，并展示投资累计回报率变化情况。良好方便的股票数据收集和推荐系统，对投资者们而言能够

更方便地对比不同股票，也能够借助深度学习模型，综合考量多种影响因子进行投资判断。

## 5.2 需求分析

我们根据实际场景需求，搭建了一个股票分析和推荐排序系统，并将本文第三、四章中提出的时序特征处理方法 AFC\_GRU 和 listwise 排序算法 AEDRM 应用其中，实现了从股票历史特征处理到候选股票排序的全流程，形成一个完整的体系。

股票分析推荐系统主要的使用用户为金融投资者们，其设计目标主要包括三方面内容，第一是提供股票数据获取功能，方便用户对股票的历史数据特征进行查看，也能够方便地对比不同股票之间的变化区别；第二是提供股票推荐排序功能，对用户感兴趣的股票序列进行预测排序，为用户的后续投资操作提出指导性建议；第三是模拟投资回测，通过模拟不同投资策略在历史时间段上的回测使用，观察对比不同策略的有效性。为了实现上述目标，我们设计的股票分析推荐系统存在以下需求：

- 准确及时的数据收集。股票数据的实时性要求较高，准确及时的数据收集功能，能够为后续的分析 and 预测提供数据支撑。
- 精确的股票推荐排序。更精准的股票预测排序功能，能够结合股票历史数据预测之后的走势排序信息，为投资者们的投资操作提出指导性建议。
- 可视的回测功能。回测功能使用户能够对投资策略进行模拟投资分析，进一步评估投资策略的有效性。
- 简单美观的系统展示。更友好便捷的系统交互页面，能够降低用户的使用难度，提升用户的使用体验。

## 5.3 系统架构

根据上节的需求分析，我们设计了如图 5-1 所示的系统架构。整个系统主要分为以下三部分：（1）数据模块：负责数据的存储，数据库中保存着一些股票数据和已训练好的股票分析推荐模型；（2）后端处理模块：主要负责数据使用与处理，以及系统功能逻辑的实现等，具体包括数据的获取与预处理、时序特征处理、股票推荐排序等功能；（3）前端交互模块：负责与用户进行

交互，对用户的行为进行响应，并将后端的结果展示给用户。数据模块和后端处理模块通过数据库语言的使用进行连接，传递数据查询请求和结果。后端处理模块和前端交互模块通过前后端的调用接口进行连接，传递用户的操作和后端处理的结果。这三个模块共同合作，实现整个系统的正常运行。

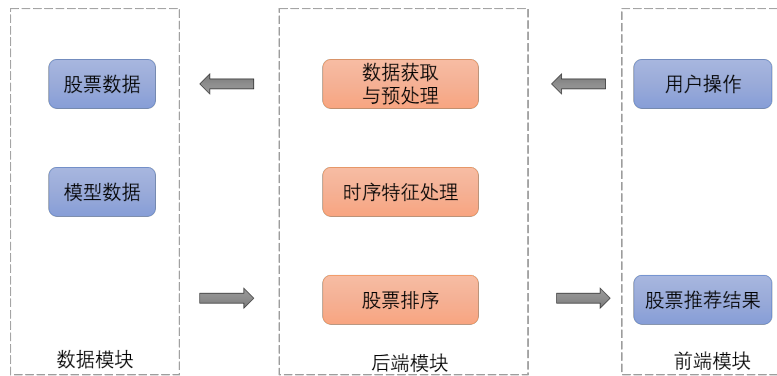


图 5-1: 股票预测排序系统架构

## 5.4 系统实现

基于第 5.3 节设计的系统架构，我们基于开源 Web 框架 Django 和前端框架 Bootstrap 对股票分析预测系统进行具体实现，数据模块使用 SQL 语言进行操作，后端使用 Python 语言进行搭建，前端使用 HTML 和 Javascript 语言进行实现。本节将从数据收集处理、后端算法实现和前端交互设计三个方面进行介绍。

### 5.4.1 数据收集处理

股票分析推荐主要涉及到股票特征数据和股票模型数据，数据的采集方式如下：

- 股票特征数据：通过使用公开股票接口（baostock 接口）获取股票数据，包括股票的交易日期、开盘价、收盘价、最高价、最低价、成交量、成交额、涨跌幅等信息。系统将获取到的数据保存到对应表中，以便后续的查询使用。股票数据表中则记录着不同股票在不同交易日内的具体信息。
- 股票模型：模型来源主要分为两种，首先系统中预加载一些已有模型，方便用户使用。另外用户也可以通过自己选定指定时间段内的数据进行模

型训练，得到自定义模型。模型表则记录着模型名称，对应的模型存储地址，对应的候选股票和训练时间段等信息。

所有数据使用 MySQL 数据库系统进行存储与管理，接口在调用时使用 SQL 语言完成数据的读、取、存操作。

在股票特征数据的收集过程中，系统还需要对股票数据进行特征构造和预处理。具体的，对于数据中的缺失值，在综合考虑股票数据的波动性大、难以差值填充的特点之后，系统采用直接丢弃的方法处理；之后系统根据原始特征，利用第 3.4.1.3 小节中的方法构建一些描述股票历史变化规律的特征，例如过去 5 天、过去 10 天的股价变化情况；另外由于不同股票特征的数据量级不一致，为避免数据大小导致训练偏差，系统中还对不同特征进行了归一化处理，具体的归一化方法如公式 (5-1) 所示：

$$\tilde{x} = \frac{x - \mu}{\sigma}, \quad (5-1)$$

其中  $\mu$  表示均值， $\sigma$  表示标准差。

## 5.4.2 后端算法实现

在得到股票数据特征之后，我们基于第 3 章和第 4 章所提出的算法，对股票历史特征进行处理，实现股票预测排序功能，得到结果序列，将后续走势越好的股票排在序列越前端，作为最终的推荐结果。

股票推荐排序模型可以选择系统中预训练好的模型，也可以根据用户指定的特定时间段内的股票数据进行模型训练，得到针对这些股票在某时间段内的自定义模型，并根据该模型进行后续预测分析。模型的训练和使用过程如图 5-2 所示。在训练阶段，用户首先需要选定候选股票序列和训练的时间段。在选定之后，系统会收集对应股票的特征数据，进行模型训练，本系统会将时间段内的股票数据等比例地随机划分成为训练集和测试集，之后使用本文提出的 AFC\_GRU 和 AEDRM 模型分别进行时序特征的处理和排序学习过程。训练结束后，本系统会将模型保存到相应的地址中，并对模型数据表进行更新。

在用户选择好模型之后，用户可以选择使用模型进行特定时间点的预测排序，或者是在时间段内对模型进行回测分析。模型内部的算法包括使用 AFC\_GRU 对股票时序特征进行处理，使用 AEDRM 对股票数据进行排序，最后将排序结果返回给前端进行显示。在回测时，系统按照 T+1 投资策略，每天

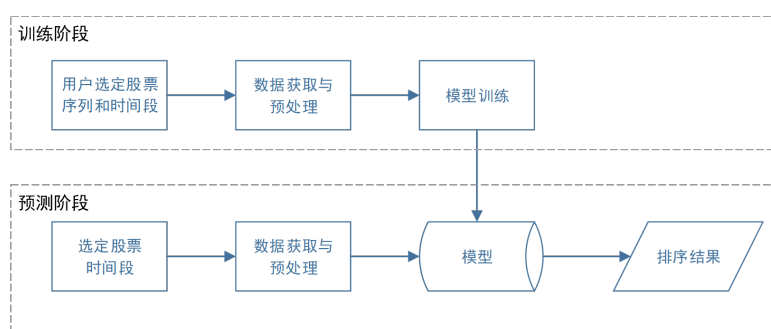


图 5-2: 模型训练与使用框架

根据排序结果进行模型的买入卖出操作，最终将指定时间段内的投资回报率变化情况以图片的形式返回到前端进行显示。

为了实现上述功能，后端模块需要与数据库和前端分别建立交互接口。在后端-数据库交互时，首先建立后端与数据库之间的连接，之后使用 SQL 语句，从数据库中获取对应的内容。在后端-前端交互时，我们设计并实现了不同功能对应的接口，每个接口首先从前端向后端发送请求，后端根据请求进行操作，之后将读取或计算得到的数据使用 JSON 格式进行封装，传递到前端页面中进行显示。

### 5.4.3 前端交互设计

前端是用户最直观的使用模块，前端页面会将用户的操作请求发送给后端，并接收后端返回的股票数据或推荐排序结果，将其展示在相应页面上。简洁良好的前端设计能够帮助用户便捷地进行操作和获取有用信息。下面我们将对系统中的三个主要页面进行展示与介绍。

#### 1. 股票数据查询页面

用户可以在当前页面中选择感兴趣的股票编号，和查询时间段，页面中将会显示出对应股票在指定时间段内的开盘价、最高价、最低价和收盘价等信息，如图 5-3 所示。

#### 2. 股票预测排序页面

在该页面，用户可以选择模型，并选择感兴趣的日期，使用该模型对相关的股票进行预测排序，页面设计如图 5-4 所示。在推荐排序时，该系统使用本文提出的 AFC\_GRU 和 AEDRM 模型分别进行时序特征的处理和排序操作，最终将排序结果从后端传递到前端，作为推荐结果进行显示。同时系统也会对算法得到的推荐结果进行分析，得到其在不同指标上的表现情况，便于用户对

交易时间	开盘价	最高价	最低价	收盘价
2017-01-24	18.59	18.88	18.5	18.87
2017-01-25	18.81	18.95	18.68	18.88
2017-01-26	18.9	19.25	18.82	19.0
2017-02-03	19.08	19.08	18.66	18.7
2017-02-06	18.88	18.9	18.54	18.62
2017-02-07	18.63	18.68	18.55	18.67
2017-02-08	18.55	18.6	18.45	18.6
2017-02-09	18.59	18.93	18.59	18.73
2017-02-10	18.86	18.97	18.73	18.9
2017-02-13	18.9	19.09	18.89	18.97
2017-02-14	19.0	19.03	18.81	18.87
2017-02-15	18.97	19.23	18.9	19.13
2017-02-16	19.19	19.34	19.04	19.14

图 5-3: 股票数据查询页面实现

当前模型进行评估。用户也可以自生成相关的模型，输入对应的模型名称，选择相应的股票序列和训练数据时间段，即可开始模型训练流程。页面设计如图 5-5 所示。

p@5	p@10	MAP@5	MAP@10	NDCG@5	NDCG@10	NDCG@1
0.23146	0.31685	0.14801	0.15515	0.53533	0.56884	0.83303

序号	预测顺序-股票ID	真实顺序-股票ID
0	sh.600588	sh.600276
1	sh.600309	sh.600588
2	sh.600050	sh.600031
3	sh.600585	sh.601336
4	sh.603986	sh.600893
5	sh.600690	sh.601919
6	sh.601888	sh.600690
7	sh.601857	sh.600196
8	sh.600031	sh.601211
9	sh.601166	sh.601899
10	sh.601801	sh.600309
11	sh.601398	sh.600030

图 5-4: 股票预测排序页面实现

### 3. 策略回测分析页面

在当前页面，用户可以选择相关模型和时间区段进行回测，最终页面会显示回测累计回报率变化情况，方便用户观察在一段时间内使用该投资策略的投资回报情况，进行投资模型的分析，具体页面设计如图 5-6 所示。在进行回测分析时，本系统模拟执行 T+1 投资策略，即每天等价买入当天预测序列前 1 只、5 只或 10 只的股票，并于第二天卖出。每天的回报率即为买入股票涨跌幅的平均值，通过在时间段内重复执行该策略，得到最终的投资回报情况。

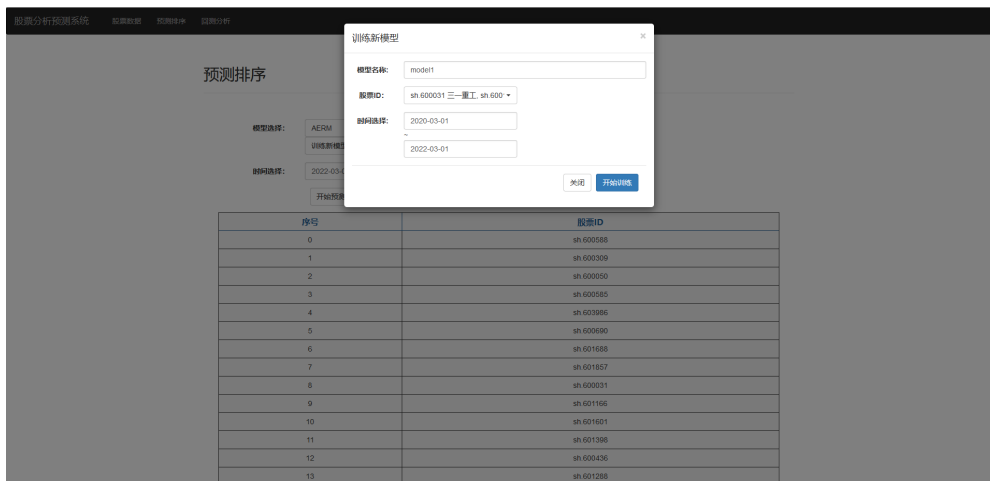


图 5-5: 股票预测排序新增模型功能实现

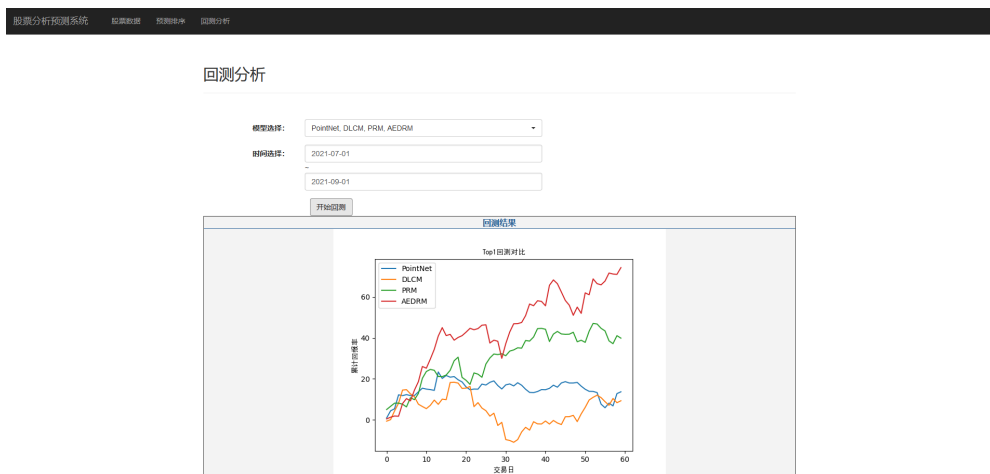


图 5-6: 股票投资策略回测分析页面实现

## 5.5 本章总结

本章介绍了我们搭建的股票分析推荐系统。该系统的主要功能是获取用户感兴趣的股票数据，使用排序算法对候选股票序列进行推荐排序，利用回测方法对比不同投资策略，从而实现对股票数据分析和处理的能力。我们将本文提出的基于注意力机制的时序特征处理方法 AFC\_GRU 和基于注意力解码器的 listwise 排序算法 AEDRM 应用于该系统中，实现了更加精确的股票预测排序模型。实践表明，我们所提出的算法能够方便有效地对股票进行推荐排序，充分证明了本文所提出算法的实际应用价值。



## 第六章 总结与展望

本章首先对本文的工作进行了总结，之后对股票预测任务的未来研究方向进行了展望。

### 6.1 工作总结

股票投资是常见的理财方式。每只股票的走势都会受到来自市场环境变量的影响。在不同的时间段内，由于市场环境的变化，股票的涨跌趋势也会有所不同。为了得到更高的收益，我们希望能够利用股票的历史特征，预测股票未来的涨跌情况作为投资依据。同时由于市场中不同股票之间也会彼此影响，同类型或竞争类型的股票走势都会存在一定的相关性。为了能够捕获到股票之间的相互影响，在现有的候选股票中找到未来走势更好的子股票集，我们设计了一种考虑到股票相互影响的推荐排序算法来解决股票投资问题。

本文中提出的两部分算法，分别关注时序特征表示和排序算法实现。主要工作点如下所示：

1. 在第三章中，本论文提出了一种特征卷积结构与注意力权重结合的时序特征表示算法 AFC\_GRU，利用卷积提取到历史上特征的变化趋势信息，再通过注意力方法计算历史时间段内的股票特征权重，加权后得到最终的股票表示特征，从而缓解长期历史遗忘问题，得到更好的股票特征表示。
2. 在第四章，本论文提出了一种基于注意力解码器的列表排序算法 AEDRM，在打分阶段综合考虑候选股票集，在保证时效性基础的同时提升排序效果，另外本论文也提出了一个新的 loss 计算方法 Mask Attention Rank，让排序的结果更加关注于序列中 Topn 的召回与 Topn 内部的顺序，保证每次召回更关注的前端序列排序情况更加准确。
3. 在第五章中，本文简单实现了一个股票分析推荐系统。用户可以在系统内便捷地查看股票历史数据，还可以使用推荐模型得到股票的未來涨跌幅顺序，同时也能够使用赢利能力分析对比不同预测模型。

经过实验证明，本文提出的时序特征表示算法 AFC\_GRU 和基于 Listwise

的股票排序算法 AEDRM 能够对股票数据进行更好的处理，得到更精确的排序结果。通过赢利能力测试模块，进一步说明本文提出的股票推荐排序算法能够在投资阶段带来更大的回报率，为投资者们提供更可靠的指导性分析结果。

## 6.2 未来展望

在未来的研究中，我们希望从以下两方面继续优化本文提出的股票预测模型：

(1) 由于市场的波动情况，不同时间段内的投资倾向会有所不同。为了能够得到更好的收益，了解市场倾向，我们希望能够使用自然语言处理技术对互联网中的相关新闻以及网友讨论内容进行分析，从中提取出有用的信息，并将其作为市场特征加入模型，帮助股票预测任务得到更好的结果。

(2) 股票走势会根据市场因子的变化而进行波动，股票预测任务需要能够敏感地获取到市场的变化信息，并及时地对模型进行调整。所以未来我们可以尝试使用在线学习来不断调整模型，保证模型获得最新的股票数据和市场数据，从而提升模型的时效性。

## 参考文献

- [1] FENG F, CHEN H, HE X, et al. Enhancing stock movement prediction with adversarial training[J]. arXiv preprint arXiv:1810.09936, 2018.
- [2] AI Q, BI K, GUO J, et al. Learning a deep listwise context model for ranking refinement[C] // The 41st international ACM SIGIR conference on research & development in information retrieval. 2018 : 135 – 144.
- [3] PEI C, ZHANG Y, ZHANG Y, et al. Personalized re-ranking for recommendation[C] // Proceedings of the 13th ACM conference on recommender systems. 2019 : 3 – 11.
- [4] JENSEN M C. Some anomalous evidence regarding market efficiency[J]. Journal of financial economics, 1978, 6(2/3) : 95 – 101.
- [5] LEVY R A. The predictive significance of five-point chart patterns[J]. Journal of Business, 1971 : 316 – 323.
- [6] BESSEMBINDER H, CHAN K. Market efficiency and the returns to technical analysis[J]. Financial management, 1998 : 5 – 17.
- [7] OLSON D. Have trading rule profits in the currency markets declined over time?[J]. Journal of banking & Finance, 2004, 28(1) : 85 – 105.
- [8] CHONG E, HAN C, PARK F C. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies[J]. Expert Systems with Applications, 2017, 83 : 187 – 205.
- [9] WANG S, JIANG Z-Q, LI S-P, et al. Testing the performance of technical trading rules in the Chinese markets based on superior predictive test[J]. Physica A: Statistical Mechanics and its Applications, 2015, 439 : 114 – 123.

- [10] ZHU H, JIANG Z-Q, LI S-P, et al. Profitability of simple technical trading rules of Chinese stock exchange indexes[J]. *Physica A: Statistical Mechanics and its Applications*, 2015, 439 : 75 – 84.
- [11] HU Z, LIU W, BIAN J, et al. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction[C] // *Proceedings of the eleventh ACM international conference on web search and data mining*. 2018 : 261 – 269.
- [12] ZHANG L, AGGARWAL C, QI G-J. Stock price prediction via discovering multi-frequency trading patterns[C] // *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017 : 2141 – 2149.
- [13] WANG B, HUANG H, WANG X. A novel text mining approach to financial time series forecasting[J]. *Neurocomputing*, 2012, 83 : 136 – 145.
- [14] MURPHY J J. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*[M]. [S.l.] : Penguin, 1999.
- [15] AGRAWAL J, CHOURASIA V, MITTRA A. State-of-the-art in stock prediction techniques[J]. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2013, 2(4) : 1360 – 1366.
- [16] KUMAR D A, MURUGAN S. Performance analysis of Indian stock market index using neural network time series model[C] // *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*. 2013 : 72 – 78.
- [17] PLATT J. Sequential minimal optimization: A fast algorithm for training support vector machines[J], 1998.
- [18] ZHANG X-D, LI A, PAN R. Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine[J]. *Applied Soft Computing*, 2016, 49 : 385 – 398.
- [19] CANZIANI A, PASZKE A, CULURCIELLO E. An analysis of deep neural network models for practical applications[J]. *arXiv preprint arXiv:1605.07678*, 2016.

- 
- [20] MAKNICKIENĖ N, MAKNICKAS A. Application of neural network for forecasting of exchange rates and forex trading[C] // The 7th international scientific conference” Business and Management. 2012 : 10–11.
- [21] 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.
- [22] STOEAN C, PAJA W, STOEAN R, et al. Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations[J]. *PloS one*, 2019, 14(10): e0223593.
- [23] KIM T, KIM H Y. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data[J]. *PloS one*, 2019, 14(2): e0212320.
- [24] AKITA R, YOSHIHARA A, MATSUBARA T, et al. Deep learning for stock prediction using numerical and textual information[C] // 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). 2016 : 1–6.
- [25] SHYNKEVICH Y, MCGINNITY T M, COLEMAN S A, et al. Forecasting movements of health-care stock prices based on different categories of news articles using multiple kernel learning[J]. *Decision Support Systems*, 2016, 85 : 74–83.
- [26] ENGLE R F. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation[J]. *Econometrica: Journal of the econometric society*, 1982 : 987–1007.
- [27] HOLT C C. Forecasting seasonals and trends by exponentially weighted moving averages[J]. *International Journal of Forecasting*, 2004, 20(1): 5–10.
- [28] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory[J]. *Neural Computation*, 1997, 9(8): 1735–1780.
- [29] CHUNG J, GULCEHRE C, CHO K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. *arXiv preprint arXiv:1412.3555*, 2014.

- [30] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. *Advances in neural information processing systems*, 2017, 30.
- [31] BAI S, KOLTER J Z, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. *arXiv preprint arXiv:1803.01271*, 2018.
- [32] LUO B, CHEN Y, JIANG W. Stock market forecasting algorithm based on improved neural network[C] // *2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*. 2016 : 628–631.
- [33] CLEVELAND R B, CLEVELAND W S, MCRAE J E, et al. STL: A seasonal-trend decomposition[J]. *J. Off. Stat*, 1990, 6(1) : 3–73.
- [34] WEI W W. Time series analysis[G] // *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*. 2006.
- [35] MONTGOMERY D C, JOHNSON L A, GARDINER J S. Forecasting and time series analysis[M]. [S.l.] : McGraw-Hill Companies, 1990.
- [36] BOLLERSLEV T. Generalized autoregressive conditional heteroskedasticity[J]. *Journal of econometrics*, 1986, 31(3) : 307–327.
- [37] ARIYO A A, ADEWUMI A O, AYO C K. Stock price prediction using the ARIMA model[C] // *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. 2014 : 106–112.
- [38] DEVI B U, SUNDAR D, ALLI P. An effective time series analysis for stock trend prediction using ARIMA model for nifty midcap-50[J]. *International Journal of Data Mining & Knowledge Management Process*, 2013, 3(1) : 65.
- [39] BABU C N, REDDY B E. Selected Indian stock predictions using a hybrid ARIMA-GARCH model[C] // *2014 International Conference on Advances in Electronics Computers and Communications*. 2014 : 1–6.
- [40] HAMILTON J D. A new approach to the economic analysis of nonstationary time series and the business cycle[J]. *Econometrica: Journal of the econometric society*, 1989 : 357–384.

- [41] GUPTA A, DHINGRA B. Stock market prediction using hidden markov models[C] // 2012 Students Conference on Engineering and Systems. 2012 : 1–4.
- [42] MADGE S, BHATT S. Predicting stock price direction using support vector machines[J]. Independent work report spring, 2015, 45.
- [43] MEESAD P, RASEL R I. Predicting stock market price using support vector regression[C] // 2013 International Conference on Informatics, Electronics and Vision (ICIEV). 2013 : 1–6.
- [44] HENRIQUE B M, SOBREIRO V A, KIMURA H. Stock price prediction using support vector regression on daily and up to the minute prices[J]. The Journal of finance and data science, 2018, 4(3) : 183–201.
- [45] HAMMAD A A A, ALI S M A, HALL E L. Forecasting the Jordanian stock price using artificial neural network[J]. Intelligent engineering systems through artificial neural networks, 2007, 17 : 1–6.
- [46] VARGHESE A, TARHEN H, SHAIKH A, et al. Stock Market Prediction Using Time Series[J]. International Journal on Recent and Innovation Trends in Computing and Communication, 2016, 4(5) : 427–430.
- [47] MAJUMDER M M R, HOSSAIN M I, HASAN M K. Indices prediction of Bangladeshi stock by using time series forecasting and performance analysis[C] // 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). 2019 :1–5.
- [48] SHARMA A, MODAK S, SRIDHAR E. Data visualization and stock market and prediction[J]. International Research Journal of Engineering and Technolog, 2019, 6(9) : 2037–2040.
- [49] SELVIN S, VINAYAKUMAR R, GOPALAKRISHNAN E, et al. Stock price prediction using LSTM, RNN and CNN-sliding window model[C] // 2017 international conference on advances in computing, communications and informatics (icacci). 2017 : 1643–1647.

- [50] WU X, WANG L, XIA Y, et al. Temporally correlated task scheduling for sequence learning[C] // International Conference on Machine Learning. 2021 : 11274–11284.
- [51] LIN H, ZHOU D, LIU W, et al. Learning multiple stock trading patterns with temporal routing adaptor and optimal transport[C] // Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021 : 1017–1026.
- [52] XU Y, COHEN S B. Stock movement prediction from tweets and historical prices[C] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018 : 1970–1979.
- [53] KHAIDEM L, SAHA S, DEY S R. Predicting the direction of stock market prices using random forest[J]. arXiv preprint arXiv:1605.00003, 2016.
- [54] 谢琪, 程耕国, 徐旭. 基于神经网络集成学习股票预测模型的研究 [J]. 计算机工程与应用, 2019, 55(8): 238–243.
- [55] GUOYING Z, PING C. Forecast of yearly stock returns based on AdaBoost integration algorithm[C] // 2017 IEEE International Conference on Smart Cloud (SmartCloud). 2017 : 263–267.
- [56] BOLLEN J, MAO H, ZENG X. Twitter mood predicts the stock market[J]. Journal of computational science, 2011, 2(1): 1–8.
- [57] DING X, ZHANG Y, LIU T, et al. Deep learning for event-driven stock prediction[C] // Twenty-fourth international joint conference on artificial intelligence. 2015.
- [58] DING X, ZHANG Y, LIU T, et al. Knowledge-driven event embedding for stock prediction[C] // Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers. 2016 : 2133–2142.
- [59] LI W, BAO R, HARIMOTO K, et al. Modeling the stock relation with graph network for overnight stock movement prediction[C] // Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. 2021 : 4541–4547.

- 
- [60] FENG F, HE X, WANG X, et al. Temporal relational ranking for stock prediction[J]. *ACM Transactions on Information Systems (TOIS)*, 2019, 37(2): 27.
- [61] SAWHNEY R, AGARWAL S, WADHWA A, et al. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach[J]. *Proceeding of AAAI*, 2021 : 497 – 504.
- [62] CRAMMER K, SINGER Y. Pranking with Ranking[C] // *NIPS'01 : Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Cambridge, MA, USA : MIT Press, 2001 : 641 – 647.
- [63] JOACHIMS T. Training linear SVMs in linear time[C] // *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006 : 217 – 226.
- [64] BURGESS C J. From ranknet to lambdarank to lambdamart: An overview[J]. *Learning*, 2010, 11(23-581): 81.
- [65] CAO Z, QIN T, LIU T-Y, et al. Learning to rank: from pairwise approach to listwise approach[C] // *Proceedings of the 24th international conference on Machine learning*. 2007 : 129 – 136.
- [66] MNIH V, HEESS N, GRAVES A, et al. Recurrent models of visual attention[J]. *Advances in neural information processing systems*, 2014, 27.
- [67] XU K, BA J, KIROUS R, et al. Show, attend and tell: Neural image caption generation with visual attention[C] // *International conference on machine learning*. 2015 : 2048 – 2057.
- [68] LONG W, LU Z, CUI L. Deep learning-based feature engineering for stock price movement prediction[J]. *Knowledge-Based Systems*, 2019, 164 : 163 – 173.
- [69] PASZKE A, GROSS S, MASSA F, et al. Pytorch: An imperative style, high-performance deep learning library[J]. *Advances in neural information processing systems*, 2019, 32.
- [70] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. *arXiv preprint arXiv:1412.6980*, 2014.

- 
- [71] DIXON M, KLABJAN D, BANG J H. Classification-based financial markets prediction using deep neural networks[J]. *Algorithmic Finance*, 2017, 6(3-4): 67–77.
- [72] 李航, OTHERS. 统计学习方法 [M]. [S.l.]: Qing hua da xue chu ban she, 2012.
- [73] BELLO I, KULKARNI S, JAIN S, et al. Seq2slate: Re-ranking and slate optimization with rnns[J]. *arXiv preprint arXiv:1810.02019*, 2018.
- [74] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C] // *Proceedings of AAAI*. 2021.

# 致 谢

转眼研究生三年即将结束，自己的学生生涯也即将画上句号。离别在即，可开学时的一切仿佛还历历在目。仔细回顾，这三年中已经发生了很多快乐的、难过的、惊喜的、难忘的事情，这些都会是我今后珍贵的回忆。

首先我要感谢申富饶老师，每周都与我们讨论探讨，帮助我们明确研究方向与思路，在思考问题的角度和解决方法上，都让我有了更清晰的科研认识。其次要感谢赵健老师，在组会上与我们讨论，并指导我们的论文写作。

感谢 RINC 组的同学们，三年里大家一起研究、一起做项目，都让我从中学习到了许多，很幸运能够与大家结识。另外，也特别感谢严骅同学和赖碧兰同学在我论文完成过程中提出的建议。

感谢我的舍友颜洁，和我一起分享聊天、一起寻找美食，让我度过了十分舒适的宿舍生活。感谢从本科陪伴我到现在的同学们，虽然大家相隔很远，但却始终彼此关心。感谢实习期间认识的许多朋友，给予我指导与帮助。感谢 Locking 队的小伙伴们，和我一起运动跳舞，每周都拥有快乐时光！

感谢我的父母和家人，谢谢你们对我的照顾和包容，以及给我的一切支持。

最后感谢我自己。



# 简历与科研成果

## 基本信息

张雅楠，女，汉族，1997年8月出生，山西省运城人。

## 教育背景

2019年9月—2022年6月 南京大学计算机科学与技术系 硕士

2015年9月—2019年6月 南京理工大学计算机科学与工程学院 本科

## 攻读硕士学位期间完成的学术成果

1. 吴宜真，张雅楠，严骅，刘晓涛，申富饶。“基于LDA和Bert的特定类别文本标题二分类方法”（202011177997.X）。

## 攻读硕士学位期间参与的科研课题

1. 国家自然科学基金项目，“基于深度感知增量式联想记忆神经网络的信息融合系统研究”（课题年限：2019年01月—2022年12月）。



# 《学位论文出版授权书》

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》(以下简称“章程”),愿意将本人的学位论文提交“中国学术期刊(光盘版)电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版,并同意编入《中国知识资源总库》,在《中国博硕士学位论文评价数据库》中使用和在互联网上传播,同意按“章程”规定享受相关权益。

作者签名: \_\_\_\_\_  
\_\_\_\_\_年\_\_\_\_月\_\_\_\_日

论文题名	基于注意力机制和列表排序算法的股票推荐排序研究				
研究生学号	MF1933122	所在院系	计算机科学与技术系	学位年度	2022
论文级别	<input type="checkbox"/> 学术学位硕士 <input checked="" type="checkbox"/> 专业学位硕士 <input type="checkbox"/> 学术学位博士 <input type="checkbox"/> 专业学位博士 (请在方框内画钩)				
作者 Email	zhangyn@smail.nju.edu.cn				
导师姓名	申富饶				

论文涉密情况:

不保密

保密, 保密期(\_\_\_\_年\_\_\_\_月\_\_\_\_日至\_\_\_\_年\_\_\_\_月\_\_\_\_日)

注: 请将该授权书填写后装订在学位论文最后一页(南大封面)。

