初稿版本1-基于Transformer-DoubleDQN深度强化学习的跨式期权量化交易

摘要

随着金融市场的不断发展，投资者的对系统性风险管理需求也在不断增加，金融期货期权市场得以蓬勃发展。一些相关的人工智能技术得以应用来辅助交易员进行交易决策，例如基于深度学习的股价预测模型，基于深度强化学习的期货量化交易模型。但市场上的大部分模型是基于标的价格的运动方向进行交易，一般以订单的浮盈浮亏作为激励函数，虽在波动较小，趋势流畅的市场效果良好，但在高波动的金融市场中走势充满了随机性，归其原因在于市场噪音较多，走势难以预测。这会导致难以让模型学习到有效信息。一旦发生黑天鹅及尾部风险事件，往往会对市场产生较大冲击，造成较大亏损。期权的买方理论上具有亏损有限，盈利上限极高的特点，可以通过构造跨式期权交易策略以承受时间损耗为代价来规避价格方向的风险，依靠价格的波动获得盈利。本文提出了Transformer-DQN深度强化学习模型来学习跨式期权交易策略。相比于其他模型，创新性地加入了阻力位，波动率特征信息，并且改进了激励函数，旨在鼓励模型自己探索，学习更多有用的交易信息。本文开发了一个与智能体进行交互的环境来模拟真实的期权交易市场，期望模型能够尽可能捕捉到大的市场波动。实验结果表明，所提算法在总体收益率，夏普比率，最大回撤率方面均优于基准模型。

关键词：期权交易，深度强化学习，Transformer-DQN，激励函数，阻力位，波动率

1、引言

在金融市场中，量化交易凭借数理统计学作为基石，利用数学模型进行交易决策，成为了一种引人注目的交易方式。这种方法能够通过交易程序有效地剔除非理性决策，从而规避情绪驱动的交易行为，展现出了其严谨的交易纪律性，有效地抵御了人性弱点，如贪婪和恐惧。正因如此，量化交易备受投资者青睐。

最早的量化交易是传统的技术分析。 查尔斯·亨利·道是是技术分析的开山鼻祖，他所提出的道氏理论是最古老、最著名的技术分析理论之一。该理论仅仅通过图表来分析预测价格走势，并揭示了价格的运动存在趋势。罗伯特·D·爱德华兹、迈吉等在《股市趋势技术分析》一书中归纳出了K线图中所出现的各种各样的形态来分析预测后续的行情走势。威科夫的量价关系理论则提供了一种把价格和成交量综合起来分析市场多空双方博弈的方法。但传统技术分析语言描述较为主观，同一幅图表可能被解读为不同的行情，买卖的规则无法用精确的量化语言描述。

为了改进传统技术分析无法量化的缺点，人们开始寻找各种公式来量化的描述市场的状态，并依据这些技术指标来形成固定的规则来进行交易。威尔斯.威尔德在《技术交易系统的新概念》一书中提出了RSI，CSI等著名指标，并依据这些指标构建交易系统，Richard Krivo提出的Vegas隧道交易法，依据EMA均线进行趋势跟踪交易。机械的交易规则使得买卖的规则有了较为精确的描述。但缺陷也很明显，每一种交易系统只能适用于特定的行情，不能灵活变化。

机器学习方法的应用已经成为了提高交易策略泛化能力的有效途径，因为它们能够有效地学习市场在不同时间段内所呈现的趋势特征。Rudra Kalyan Nayak等使用改进的支持向量机模型SVM-KNN对股价进行预测，其预测走势和真实走势差别不大，在趋势流畅的投资品种中获得了较好的回报。Andersen 和 Mikelsen (2012) 提出了一个结合进化算法和机器学习的新型交易框架，用于优化投资组合。此框架通过自动调整和学习市场行为，显著提高了交易的性能。目前的交易策略的研究主要分为两大类：一是利用深度学习技术进行投资标的价格预测，另一是使用深度强化学习模型在金融市场环境中培训智能体来代替人类进行投资标的交易。这两种方法的核心都是通过机器学习模型来进行交易决策，即在价格上涨时采取做多策略，在价格下跌时采取做空策略。

然而，训练模型来预测价格方向目前仍面临着一些挑战：

1. 金融市场的价格走势受到多方面因素的影响，导致存在大量噪音样本数据，这些噪音数据可能会干扰模型的训练过程。
2. 当价格运动到阻力位时，多空双方的激烈交战使得走势面临着极大的不确定性，这增加了预测价格走势的难度。
3. 当出现重大新闻事件时，市场价格可能会突然向与预期相反的方向急速波动，从而给投资者带来巨大的损失。这种情况下，模型很难有效地适应新情况，可能会产生较大的损失。

因此，尽管机器学习方法在金融交易中具有潜力，但要克服这些挑战，仍需要进一步的研究和改进，以提高模型的稳健性和泛化能力。

随着金融风险管理的需求日益凸显。期权作为一种有效的风险转移工具逐渐受到投资者的关注与重视。在期权市场中，投资者不仅可以对价格方向进行交易，还可以对标的资产的波动率进行交易。此外，他们还可以通过同时买入或卖出不同类型的期权来构建投资组合，以对冲不想要的风险敞口。跨式期权组合则通过同时购买一定数量的看涨期权和看跌期权来对冲价格运动方向的风险。当投资标的资产价格出现较大波动时，跨式期权组合将实现盈利；而若价格长时间处于横盘震荡状态，则可能导致亏损。这种投资策略的盈利与价格运动方向无关，而取决于价格是否发生较大波动。

跨式期权的特性给予我们了一个新的研究想法：让机器去学习如何交易资产价格的波动而不是资产价格的运动方向。跨式期权的量化交易策略涉及诸多因素，包括波动率的预测、阻力位的识别以及适当的激励函数的选择等。Transformer-DQN作为一种结合了深度学习和强化学习的模型，为研究跨式期权交易策略提供了新的思路和方法。然而，如何有效地利用Transformer-DQN模型来优化期权交易策略，以应对金融市场的不确定性和复杂性，仍然是当前研究的焦点和挑战。

在这样的背景下，本文旨在探讨如何利用Transformer-DQN模型来优化期权的量化交易策略，并通过实证研究验证其有效性和可行性。我们将从建立期权交易环境模型开始，逐步探讨如何设计合适的奖励函数和优化策略，以及如何应对期权交易中的各种挑战和限制。通过这样的研究，我们希望为期权交易领域的进一步发展和应用提供新的思路和方法。

本文贡献如下：

1、设计了Transformer-DQN的模型来学习跨式期权交易策略。

2、使用延迟奖励，并引进了止损机制的奖励函数使得训练得以有效进行。

3、提供了新的特征信息：当资产价格运动到阻力位区域会向模型发出信号，辅佐模型更好的做出交易决策。

2 文献综述

2.1 机器学习在交易中的应用

基于机器学习的股价预测通常将其视为分类问题或回归问题。在分类问题背景下，一般以下一个交易日的涨跌作为标签进行分类器训练。若为视为回归问题，则将下一个交易日的价格作为标签进行训练。Htun等人（2024）使用随机森林、支持向量机和长短期记忆网络预测S&P 500的相对回报，显示了机器学习在处理非线性和非平稳时间序列数据中的有效性。深度学习特别是卷积神经网络 (CNN) 和长短期记忆网络 (LSTM) 在金融时间序列分析中常被应用。但是在金融市场中资产价格的走势是不确定的，其利润是来源于一串盈亏期望为正的交易行为，而股价预测模型往往存在只在乎预测是否正确，而没有考虑赔率的现象。Mhlongo等人（2024）指出股价预测模型过度依赖历史数据和量化分析，而对市场动态和行为因素的解释不足，这限制了其在复杂市场条件下的适应性和实用性。因此，未来发展股价预测模型时，研究者需要探索如何整合赔率等关键金融指标，以增强模型的市场感知能力和决策支持效果。这不仅能够提升预测模型的实用性，也有助于在多变的市场环境中提供更为准确的风险评估和投资指导。

强化学习通过模拟投资者在市场环境中进行交易，能较好的学习如何平衡在交易中胜率与赔率之间的关系。如

Mahdi Massahi等针对大宗商品期货市场提出了基于深度Q网络（DQN）及其改进版本模型（DDQN）的自动化交易系统。该系统利用带有门控循环单元（GRU）网络的多代理架构来近似Q值函数，以降低维度并提高性能。Liu Yang 等提出了iRDPG的自适应交易模型，该模型结合了深度强化学习和模仿学习技术，在高频数据中进行连续的自动化交易。Yue Deng等 提出了一种直接强化交易的方法（DDR），该方法不依赖于学习价值函数，而是直接从参数化策略族中学习连续动作。这允许系统从连续的感官数据（市场特征）中直接学习交易策略。Qing Yang Eddy Lim等使用LSTM模型预测股价来向Q网络提供辅助决策信息，以此来动态调整投资组合，有效的提高了整体收益率。尽管机器学习在股价预测和自动交易方面中取得了一定的成效，但它们在应对市场中的黑天鹅事件方面仍显示出局限性，可能导致在极端市场事件中承受较大亏损。Gravdal 和 Vollset (2023) 的研究表明，即使是高级预测模型也难以准确预测金融危机期间的市场表现 。

2.2 跨式期权应用于算法交易

跨式期权作为一种灵活的交易策略，不仅在风险管理中发挥重要作用，同时也是提高投资组合收益的有效工具。Shivaprasad 和 Geetha（2022年）详细讨论了长跨式和短跨式策略在不同市场条件下的表现，指出在高波动性市场中，长跨式期权能有效对冲风险同时带来潜在的高回报。此外，Kownatzki、Putnam 和 Yu（2022年）通过市场事件的案例研究，展示了跨式期权在应对突发市场事件时的策略灵活性和风险对冲能力。在交易策略方面，Brenner（2006）指出跨式期权可以作为一种主动的交易策略，通过预测市场的大幅波动来获取收益。这种策略特别适用于那些预期市场会出现重大价格波动，但方向不明确的情况。相比于其他金融衍生品，跨式期权提供了更大的灵活性和较低的初始投资成本。Lapan 和 Moschini（1991年）的研究强调了跨式期权在多种市场条件下提供有效对冲的同时，还能够保持资本流动性和较低的交易成本。

跨式期权结合算法交易策略已成为金融工程领域的一个研究热点。这些策略利用高级算法来优化期权交易，尤其是在高波动性市场中控制风险和提高交易效率。Jadeja, Patel, 和 Patel (2023) 研究了一个基于Delta值的算法，用于控制波动性指数期权卖出策略中的损失。该算法通过调整跨式策略的对冲参数来降低风险，展示了在高频交易系统中的应用潜力 。另一个方向是通过深度强化学习方法预测波动率来向跨式期权交易提供决策信息。（波动率上升是跨式期权盈利的充分条件）。J Zhang 和 Y Lei (2022) 的研究探讨了深度强化学习在股票价格波动预测中的应用，特别是通过结合策略梯度方法提高预测模型的精确度和反应速度。A Millea (2021) 文章提出，通过使用分层DRL和模型基础的RL技术，可以有效地模拟和预测不同市场环境下的资产波动。这些技术通过更精细的环境建模和策略调整，使DRL模型能够更好地理解和适应市场波动性。Y Yu 等人 (2023) 开发了一个结合VMD方法和深度神经网络的DRL模型，用于更有效地预测股价指数的真实波动率。H Yang 等人 (2020) 探索了DRL在自动化股票交易系统中的应用，重点是如何通过管理年化波动率和最大回撤来最大化回报。

3、问题规范化（定义各个变量字母）

3.1数据预处理

原始的股票数据为15分钟的k线数据，每一条数据包含该周期的开盘价，收盘价，最高价，最低价，成交量。成交额这六条属性，数据来源于上海证券交易所和深证证券交易所。数据的前4项属性和后两项属性的量纲不同，波动程度也不同，这会使得模型在刚开始学习时受到较强的干扰。绝对的价格变化意义不大，我们更应该让模型关注价格和成交量的相对波动，于是我们做出了如下对数化处理：

$\bar{p}_t = ln(\frac{p_t}{p_{t-1}})$ 其中 p_t 是在t时刻的价格，转化后描述的是t时刻价格的对数变化率

$\bar{vol}_t = ln(\frac{vol_t}{\sum_{i=1}^{n} vol_{t-mi}/n})$ 其中 vol_t 是t时刻的成交量，m是一个交易日产生的K线数量。转化后描述的是成交量与过去n天的同一时刻均值的对数变化率，这么处理可以减少刚开盘时成交活跃带来的影响

3.2特征构建

3.2.1阻力位标识

当价格运动到阻力位时，多空双方交战激烈，价格走势面临着更大的不确定性。在交易心理上，人们存在锚定效应（相关文献：行为金融学），对价格的相对高点或低点会更加关注。在这里本文将其定义为阻力位。在K线上寻找近期的历史阻力位可以为模型提供更多的参考信息，对应算法如下：

算法：阻力位标注

1、定义一个固定的时间长度d

2、定义在时刻t时，指数点位是 p_t ，在时间长度d下，指数的动量为 $M_t = p_t - p_{t-d}$

3、若出现 $M_t * M_{t-1} < 0$,则说明在时间区间d之内出现了一个反转点，若 $M_t > 0$ 则代表指数探底回升，是一个相对低点，是一个可能的支撑位， $M_t < 0$ 则代表指数见顶回落，是一个相对高点，是一个可能的压力位。

4、若相对高点比前一个支撑位涨幅大于d%，或突破前一个压力位e%，记为一个压力点位。若相对低点比前一个压力位跌幅超过d%，或跌破前一个支撑位e%，记为一个支撑点位。

5、将这些支撑点位和压力点位统筹记作阻力位。

获得阻力位后，本文将阻力位附近正负%0.3记作阻力位区域，当价格运动到阻力位区域时，向模型发出阻力位信号，标识为1，否则为0。

示意如图3.1所示：

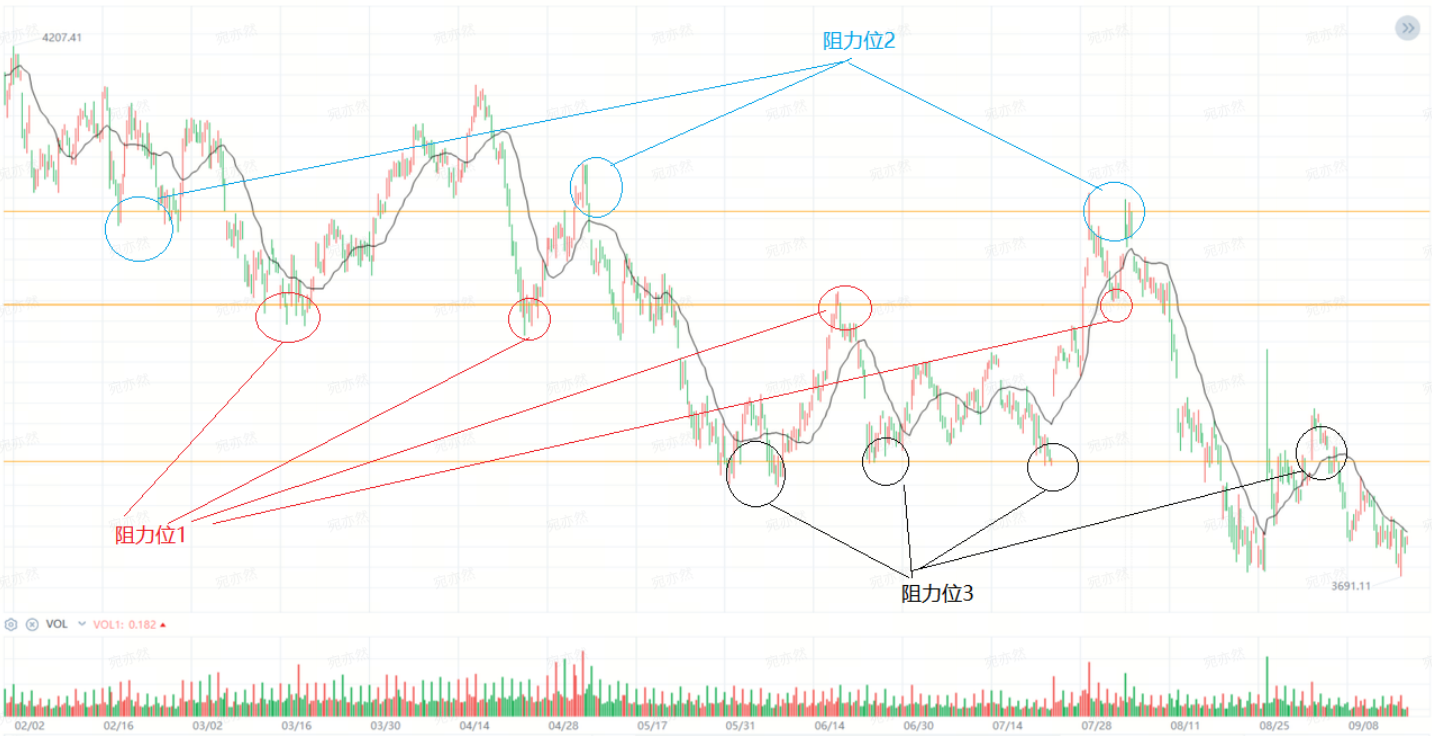


图3.1：本图展示了沪深300指数2023年2月至8月的一段k线周期为30分钟的历史行情，图中标记了3个较为显著的阻力位。他们由若干个相对高(低)点组成。

3.2.2历史波动率（参考文献：波动率交易）

波动率的标准定义是标准差，其公式为：

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(r_i - \bar{r})^2}{N-1}}$$

其中ri为该周期的对数收益率,即3.1中的 $\bar{p}_t$ ， $\bar{r}$ 为样本序列的平均收益率，N为样本数量。为了统一为年化波动率，还需要乘上年化因子 $\sqrt{F}$ ，其中F为一年中该周期K线数量。

但在真实金融市场中，将收益率的均值和波动项区分开是一件很困难的事情。本文假设市场在短时间内上涨或下跌的概率都是50%，那么这段时间内市场平均收益率为0。本文选定以每根K线的收盘价进行估算历史波动率，那么其最终公式为：

$$HV = \sqrt{\frac{F}{N-1}\sum_{i=1}^{N}[ln(\frac{C_i}{C_{i-1}})]^2}$$

标准方差波动率计算便捷，容易理解，目前是比较常见的波动率度量方法。本文采用的是15分钟K线数据，若全天为横盘震荡行情也能捕捉到盘中波动信息。

3.3期权交易环境模拟

3.3.1盈亏结算方法(bsm文献)

期权的价格变化主要受到标的的三个方面的影响，分别是当前价格S，期权剩余到期年化时间T，波动程度σ。无风险利率短期内一般视为常数。布莱克-斯科尔斯模型是1973年由美国金融学家Fisher Black和Myron Scholes推导出来的期权定价模型，经过推导并利用欧式期权平价原理，可以得出无红利支付的欧式期权定价公式：

$$C(S,t) = SN(d_1) - Ke^{-r(T-t)}N(d_2)$$

$$P(S,t) = -SN(-d_1) + Ke^{-r(T-t)}N(-d_2)$$

其中d1为

$$d_1 = \frac{ln(S/K) + (r+\sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

其中C(S,t)表示欧式期权看涨期权价格，C(S,t)表示欧式期权看跌期权价格，T-t表示距到期日的年华时间（一般定义为剩余天数除以一年360），r表示无风险利率，K表示执行价格，N(d)表示标准正态分布变量的累计概率分布函数，即小于变量d的概率。布莱克-斯科尔斯定价模型目前被期权市场广泛接受，成为广大投资者重要的参考指标。本文主要研究月期权的短线交易，且ETF若分红交易所也会调整合约的执行价及合约乘数，因此忽略分红对期权价格产生的影响。隐含波动率一般围绕历史波动率附近上下波动，由于该高频数据难以获得，且容易受到市场情绪影响，本文假设市场是风险中性的，并使用近n日历史波动率来替代隐含波动率进行盈亏结算。

3.3.2跨式期权头寸建立的规则

设定执行价的间隔为S，那么价格P必然落在[X,X+S]区间内（其中X=P - (P mod S)），将该区间均匀分为三个小段：S1，S2，S3。若价格落在S1范围内，则选择执行价为X的看涨期权和看跌期权。若价格落在S2范围内，则选择选择执行价为X+S的看涨期权和选择选择执行价为X的看跌期权。若价格落在S3内，则选择执行价为X+S的看涨期权和看跌期权。选定好合约后，同时购买一定数量的看涨期权和看跌期权来构建跨式期权组使得组合整体的delta近似为0。

在确定合约到期日方面，本文选择最当月到期的期权合约，若当月期权剩余时间小于15天，则选择次月合约。此举旨在避免期权末日论效应。

4、方法

4.1、状态与动作空间定义

市场目前所处于的状态特征是模型最核心的输入部分。设置一个长度为d的滑动窗口，将t-d到t这段时间的数据作为时间序列 Seq_t 。 Seq_t 里面包含有n个时间步 k_i :

$$Seq_t = [k_{t-n+1}, k_{t-n+2}, ...k_t]^T$$

Seq_t的每一个时间步k_i包含有如下信息：

1、经过处理的K线数据(高开低收量额)，让模型能够观察过去一段时间市场多空博弈的信息。

2、作为一名投资者，时时刻刻关注自身账户信息也是很重要的，因此单笔交易头寸的浮盈浮亏也要作为时间序列St的一部分。

3、近n日的历史波动率，波动率是影响期权定价的重要因素。

4、距离下一个交易日的间隔天数，因为周末及节假日市场不交易，但期权的时间损耗依旧照常计算。

综上， $Seq_t \in R^{d*10}$

另外有两个信息单独给出，不包含在时间序列数据里：

1、若市场价格运动到了阻力位区域，阻力位标识程序还会发出阻力位信号,记作ResistanceFlag。

2、还有头寸的持有时间，以此提醒模型持有期权要承受时间损耗,记作HoldTime。

以上所有部分构成一个状态 S_t :

$$S_t = [Seq_t, ResistanceFlag, HoldTime]$$

动作是模型的输出结果，t时刻的动作为 $a_t \in A_t$ 。动作空间只有两个：1代表持仓状态，0代表空仓状态。由0到1代表开仓动作，由1到0为平仓动作。

模型必须完成一个空仓等待，开仓，持仓，平仓这一流程，才能视为一次完整的交易

4.3、奖励函数设计

激励函数的设计是影响基于DRL模型性能的关键因素之一。常见的激励函数包括利润最大化、损失最小化和风险调整返回最大化。在不同市场波动性下，激励函数需要优化以适应市场条件变化，如通过调整风险偏好参数来平衡收益与风险。A Millea (2021)指出利用风险度量（如夏普比率和最大回撤）来设计激励函数，促进了在风险控制和收益最大化之间的平衡。但是这种直接采用业绩考核方式的奖励函数并不适用于跨式期权的交易，因为在持仓过程中随着行情的波动会频繁的发出奖惩信号，这会严重干扰模型的训练，使其变得不再稳定。本文使用延迟奖励机制来训练模型，并设置止损制度来控制回撤风险。

定义在t时刻的持仓市值为 $MarketValue_t$ ，定义开仓成本为Cost，则这笔交易在t时刻的对数收益率为 $return_t = ln(\frac{MarketValue_t}{Cost})$ ，设定止损线stop(stop<0)。

若 $a_{t-1} \to a_t$ 为0->1，是开仓动作， $reward_t = 0$ 。

若 $a_{t-1} \to a_t$ 为1->1，是持仓动作，分有两种情况：

1、若return_t>stop， $reward_t = 0$

2、若return_t<stop， $reward_t = e^{return_t} - 1$ （转化为普通收益率）

该设计意味着允许出现一定亏损，可以让模型忽略掉短时间内的噪音，让其更关注一段时间内行情的波动情况。

若 $a_{t-1} \to a_t$ 为1->0，是平仓动作，分两种情况：

1、在止损线下平仓， $reward_t = a$ (a>0)，因为止损是正确行为

2、在非止损情况下， $reward_t = e^{return_t} - 1$ 。若平仓时止盈时的点位相比于开仓点位偏离幅度达到d%以上，给予双倍奖励,旨在鼓励模型在单边极端行情中坚定持仓。

若 $a_{t-1} \to a_t$ 为0->0，是空仓动作， $reward_t = 0$

4.4、模型结构

4.4.1深度强化学习框架Double-DQN算法

Double-DQN是一种深度强化学习框架，由基础的Q-Learning算法改进而来。克里斯·沃特金斯在1989年发明了Q-learing，并证明了它的收敛性（(Watkins & Dayan, 1992)）。它是一种基于价值的无模型的强化学习算法，旨在在没有先验知识的情况下，通过迭代优化Q（st，at）来找到最优决策。这相当于训练了一个Q表来记录各个状态-动作空间对应的估值，当模型在面对不同环境状态就会选择Q值最大的动作，适用于离散动作的场景。但由于市场状态复杂多变，这会使得Q表变得无比庞大。为了解决这个问题， (Mnih et al., 2013; Mnih et al., 2015)提出了使用深度神经网络来估算最优动作值函数 $Q(S_t, A_t, W)$ 。DQN是一种非策略的算法，它在与环境交互获得经验数据 $e = (s_t, a_t, s_{t+1}, done)$ ，并将其存放在在经验回放池中，训练时从经验回放池中抽取若干个样本数据进行训练。Q学习算法从随机初始化开始，遵循ε-greed的策略来选择动作。模型的动作选择是基于学习状态-动作值函数的近似，该函数由神经网络进行训练得到。

DQN的训练目标是寻找到最优动作值函数，该函数应该服从一个重要的恒等式：贝尔曼方程：

$Q^*(s_t, a_t) = E_{s_{t+1} \in S}[r_t + \gamma Q^*(s_{t+1, a^*_{t+1}})]$

其中 $a^*_{t+1} = argmax_a \ Q^*(s_{t+1}, a)$ 。DQN的基本思想就是借助贝尔曼方程来迭代值函数 $Q(S_t, A_t, W)$ 。它的标签y即为：

$y = r_t + \gamma Q(s_{t+1}, a^*_{t+1}, w)$

它的的损失函数即为：

$$Loss = \frac{1}{2}(Q(s_t, a_t) - y)^2$$

当模型训练好后，模型在做出动作决策时遵循

$a^*_t = argmax_{a_t \in A} Q(s_t, a_t, w^*)$

但原始的DQN算法又存在高估Q值和训练不稳定的问题，因为它使用了最大算子来确定下一个状态的Q值且训练时Loss震荡剧烈。于是Hasselt（2016）提出了DoubleDQN的改进算法。DoubleDQN算法使用了使用了两个网络W1和W2分别用来选择最优动作和估计Q值。相比于原DQN算法，在估计 $Q(s_{t+1}, a^*_{t+1})$ 时，用W1选择 a^*_{t+1} 后再用W2去计算该Q值。W1更新若干次后再将参数复制到W2上。整体算法如下：

算法DoubleDQN:

1、初始化两个深度学习神经网络W1，W2，分别为选择最优动作网络和计算估值评分（Q值）网络，初始化经验回放池DataBuffer，大小为N，里面存储的数据量为M，开始训练时所需样本量X，

For episode=1,M do:

初始化时间序列数据，获得第一个状态s1

For t=1,T do:

If DataBuffer里的数据量M小于X：

模型随机选择一个动作at

else

以epsilon的概率在动作空间A中随机选择一个动作at，否则选择 $a_t = argmax_a\ Q(s_t, a，\ W1)$

将at输入市场环境，环境将会返回奖励 r_t 和下一个状态 s_{t+1} ,若该动作为平仓，则设置done=1

生成一条新的经验数据 $e = (s_t, a_t, r_t, s_{t+1}, done)$,并将这个训练样本放入经验回放池（DataBuffer）的末尾

If M>N：则删除经验回放池最早的一条数据

从经验回放池中随机采样 minibatch条数据E

用W1计算 $Q(s_t, a_t, W1)$

用W1获得下一状态的最佳动作 $a^*_{t+1} = argmax\ Q(s_{t+1}, a, W1)$

用W2来计算 $Q(s_{t+1}, a^*_{t+1}, W2)$

根据奖励函数获得rt

计算标签 $y = r_t + \gamma\ Q(s_{t+1}, a^*_{t+1}, W2)$

获得损失 $Loss = \frac{1}{2}(Q(s_t, a_t) - y)^2$

获得训练参数梯度，反向更新传播

If W1更新超过n次，将W1参数复制到W2

训练算法如图所示：

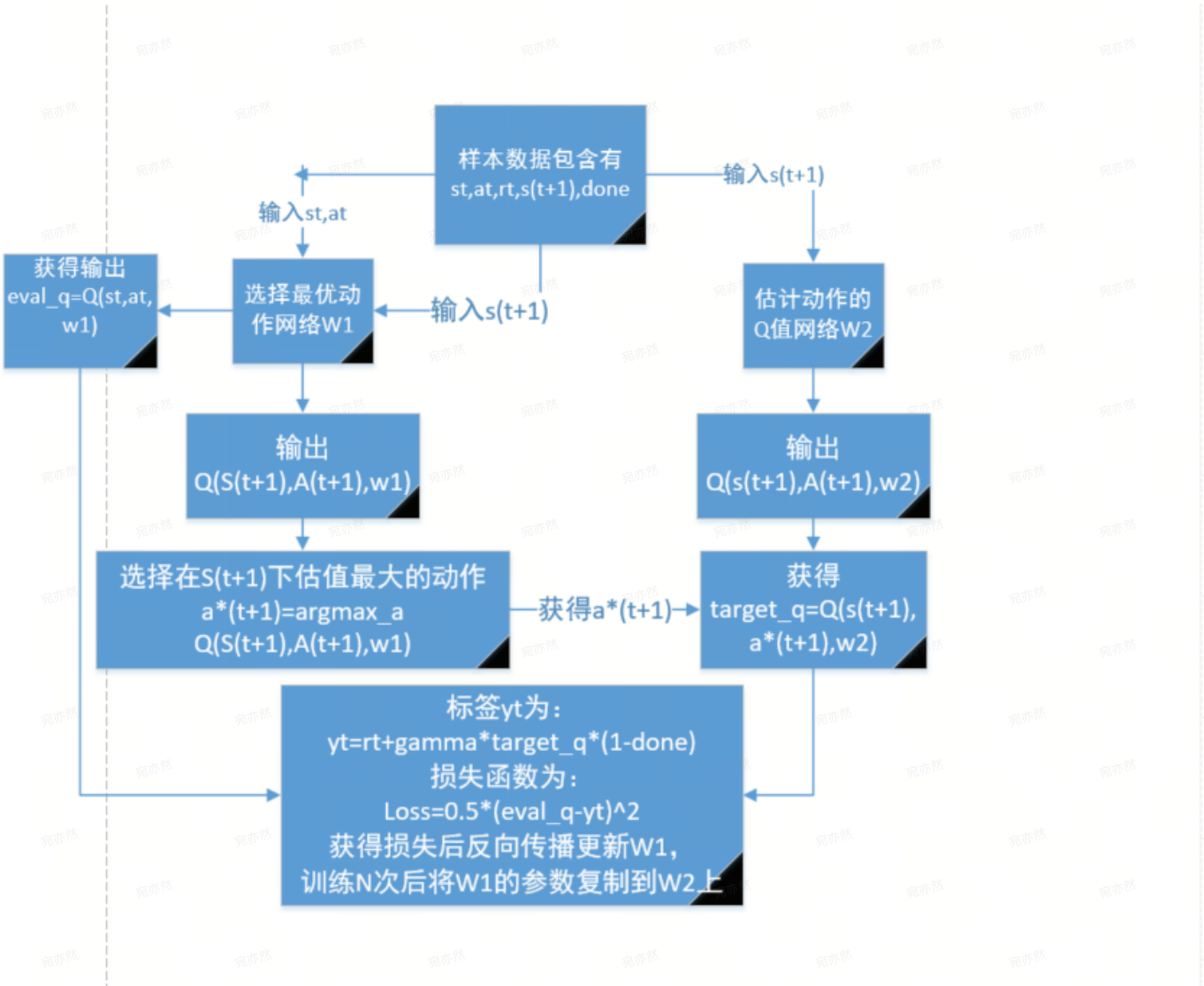


图4.1：训练流程细节

在本研究的第4.1节中，仅定义了两个交易动作：开仓和平仓，构成了一个离散的动作空间。因此，Q学习算法与本文提出的交易系统框架高度契合。整体训练流程如图4.2所示

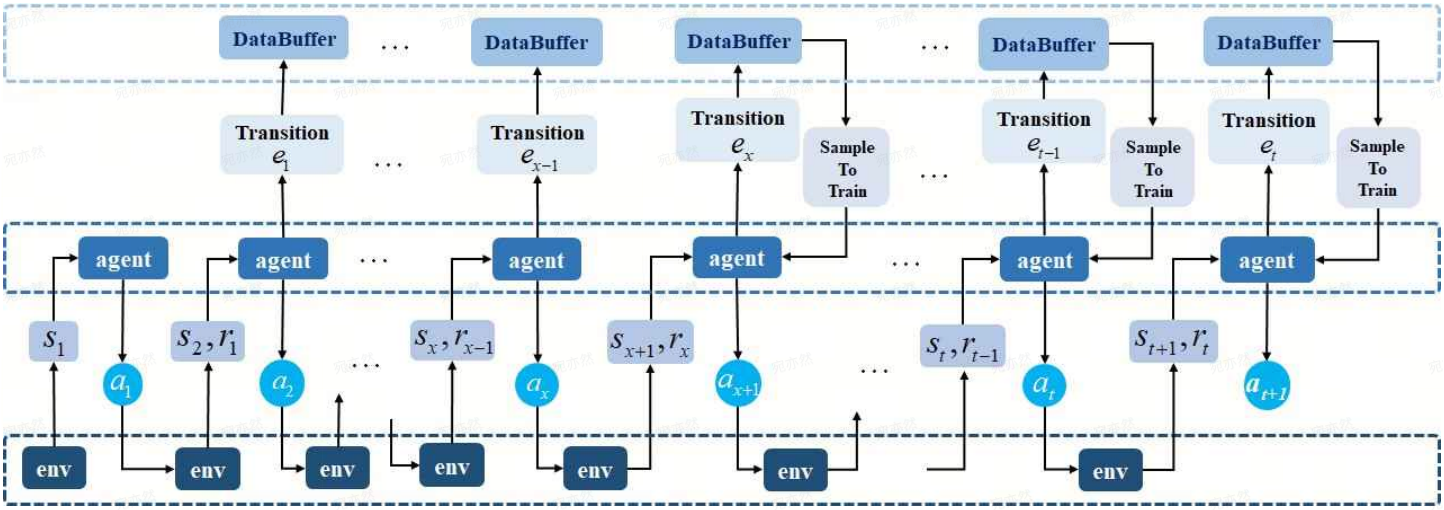


图4.2 Transformer-DQN的整体训练流程

4.4.2估计Q值的深度神经网络Q-Transformer

Transformer是一种用于自然语言处理和其他序列学习任务的深度学习模型。它由Vaswani等人在2017年的一篇论文中首次提出，由于其出色的性能在自然语言处理领域获得了广泛应用。Transformer自注意力机制使得模型在处理长序列数据时仍然能够捕捉到远距离的依赖关系，这在处理序列数据任务中非常重要。现实中有许多数据均类似于文本数据，均具有序列性质。很明显，K线数据也是时间序列数据模型，受此启发可以使用Transformer-Encoder模块基于滑动时间窗口来学习K线数据中所包含的信息。但传统的Transformer的任务是机器翻译，输入输出均为序列数据，但本文是期望拟合一个Q值函数，无法用到Masked Muti-Head层。因此其Decoder模块应该改成全连接层来输出最后结果。

为了适应本文的任务，我们需要对传统的Transformer模型进行改造。在4.1节中输入端分的状态St有两部分：时间序列部分(Seq)和单独给出的阻力位标识(ResistanceFlag)和持有时间(HoldTime)。

首先，时间序列部分的数据Seq直接输入进入Transformer-Encoder模块，以此来学习市场状态信息并提取出相应特征：

$$H_t^1 = Encoder(Seq_t, W)\ H_t^1 \in R^{10*d}$$

之后用一层Flatten层把矩阵转换为向量，再用一层全连接层压缩信息：

$$H_t^2 = \sigma(Flatten(H_t^1)W_{10d*n})(10d > n)$$

把阻力位标识(ResistanceFlag)和持有时间(HoldTime)添加进去：

$$H_t^3 = concat[H_t^2, ResistanceFlag, HoldTime]\ H_t^3 \in R^{n+2}$$

再用一层全连接层学习阻力位标识(ResistanceFlag)和持有时间(HoldTime)相关信息特征

$$H_t^4 = \sigma(H_t^3 W_{(n+2)*n})\ \ H_t^4 \in R^n$$

最后用输出层获得 $Q(S_t, a_t)$ ，即对在该状态下对持仓动作和空仓动作计算一个估值Q

$$Q(S_t, a_t) = \sigma(H_t^4 W_{n*2})\ Q(S_t, a_t) \in R^2$$

这样我们就完成了使用Q-Transformer模型来估算 $Q(S_t, a_t)$ 的工作。估算出 $Q(S_t, a_t)$ 后将其带入到4.4.1节的Double-DQN算法框架中对模型参数进行训练更新。

Tramsfromer相比于LSTM处理时间序列数据在长程依赖性性能会更为出色。有一些研究表明，不论是何种改进的RNN（如LSTM、GRU），它们捕捉长依赖的能力有限，而Transformer很好缓解了这个问题。结构示意图如图4.3所示：

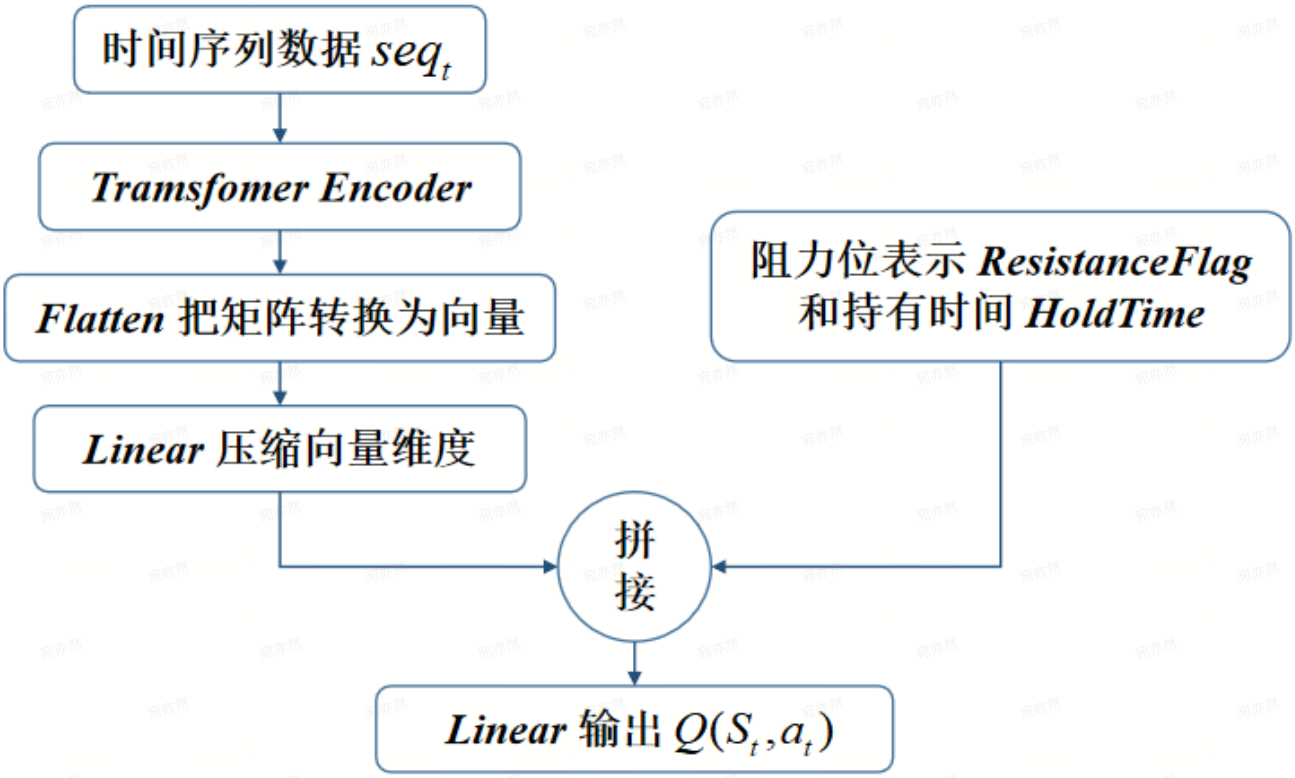


图4.3 Q-Transformer的结构

训练完成后，即可让Transformer-DQN自行与市场互动，独立做出交易决策。（如图4.4）：

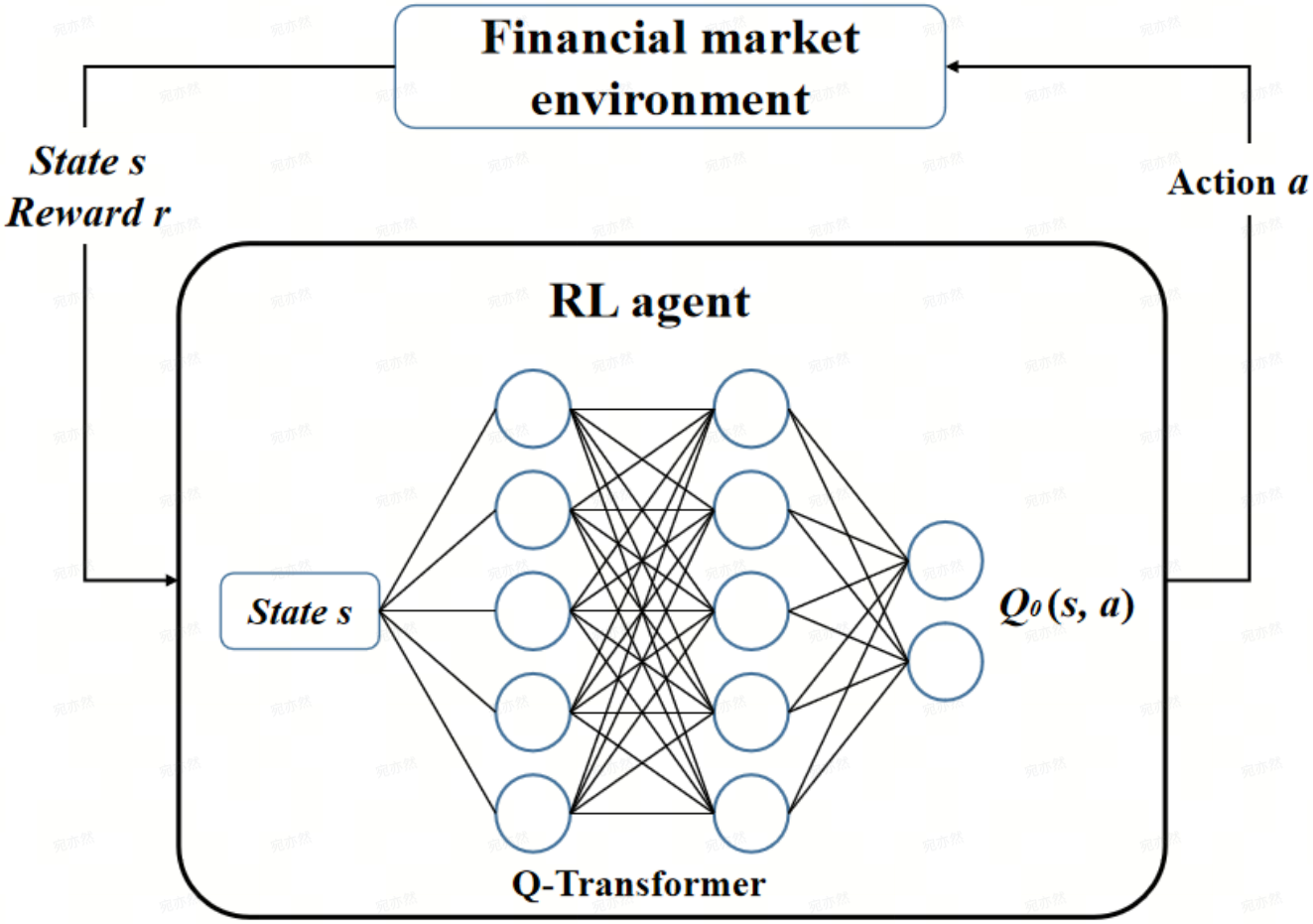


图4.4 完成训练后，Transformer-DQN直接模拟投资者在市场中自主做出交易决策

5、实验

5.1数据集介绍

本文使用上海证券交易所主要的宽基指作为实验的训练和测试数据，分别为上证50，沪深300，中证500。我们收集到了以上数据的15分钟K线数据，时间为2018年1月4日至2022年12月30日，其数据来源为锐思金融数据库（www.resset.com）。每条数据包含了该周期的最高价，最低价，开盘价，收盘价，成交量，成交额。在A股市场中，跟踪指数期权产品有上交所的ETF期权和中金所的股指期权。在这里为了方便统一视为股指期权。

5.2实验环境设置

K线数据经过3.1，3.2节处理后，将2018年1月4日至2021年12月31日作为训练集，2022年1月4日至2022年12月30日作为测试集。模型能回看的历史数据长度为20天，一个交易日有16根15m级别的k线，所以Seq_t的长度为320。历史波动率选择近5日波动率。为了更好的模拟真实的市场环境，本文还加入了以下约束条件：在手续费方面，不同券商不同客户手续费收取标准不一样，本文参考中金所股指期权手续费为一张15元。但中金所的股指期权合约乘数为100，折算下来若合约乘数为一则对应的

手续费为0.15元。交易所为了防范风险，遏制过度投机，还设置了限仓制度，即开仓时合约市值不能超过总资金20%。实验设定初始资金为100万元。在baseline实验中，设定ETF交易手续费为成交总额的万分之一。由于主要是研究短线交易，因此还设置了最长持仓时间不得超过5天，以此防止承担过高的时间损耗。在4.3节的止损线设置为15%。

实验的评价指标选择如下：

到期总对数收益率：$R_t = ln(p_{end}/p_{begin})$，即期末总资产除以期初总资产后取对数。

夏普比率：$Sr = \frac{E(R_p) - R_f}{\sigma_P}$，其中E(Rp)为平均年化收益率，Rf：年化无风险利率，σp：年化收益率的标准差。夏普比率衡量的是投资者每承受一单位总风险，会产生多少的超额回报

最大回撤率：$Mdd = -\frac{\max(P_i - P_j)}{P_i}, j > i$，该指标衡量了历史上最大的亏损程度，显示了可能最糟糕的情况。

5.3与baseline比较的实验

5.3.1baseline选择

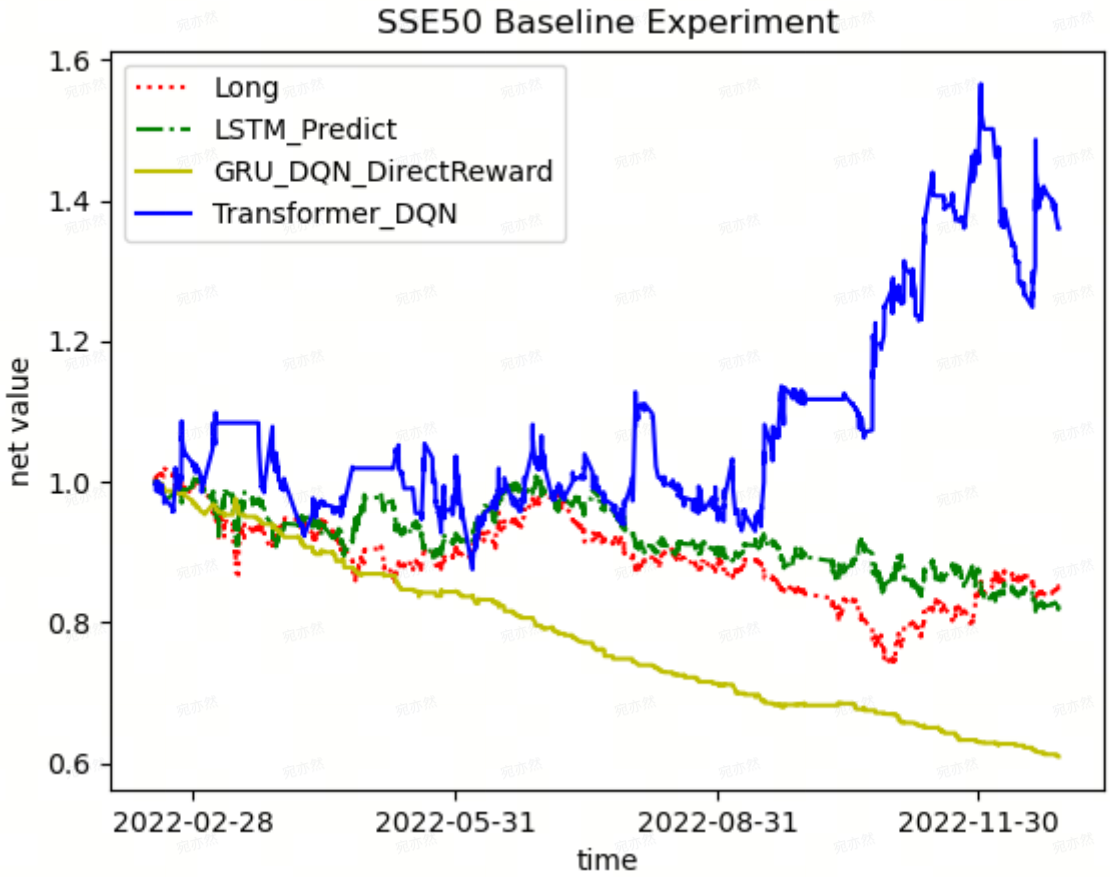
本文选择了一下方法作为baseline：

1、满仓买入并持有指数ETF，可以近似视为指数本身的收益率

2、使用使用LSTM网络每天进行股价预测。在使用股价预测模型时，预测下一天股价上涨则满仓买入ETF做多，预测下一天股价下跌则满仓融券卖空ETF做空。

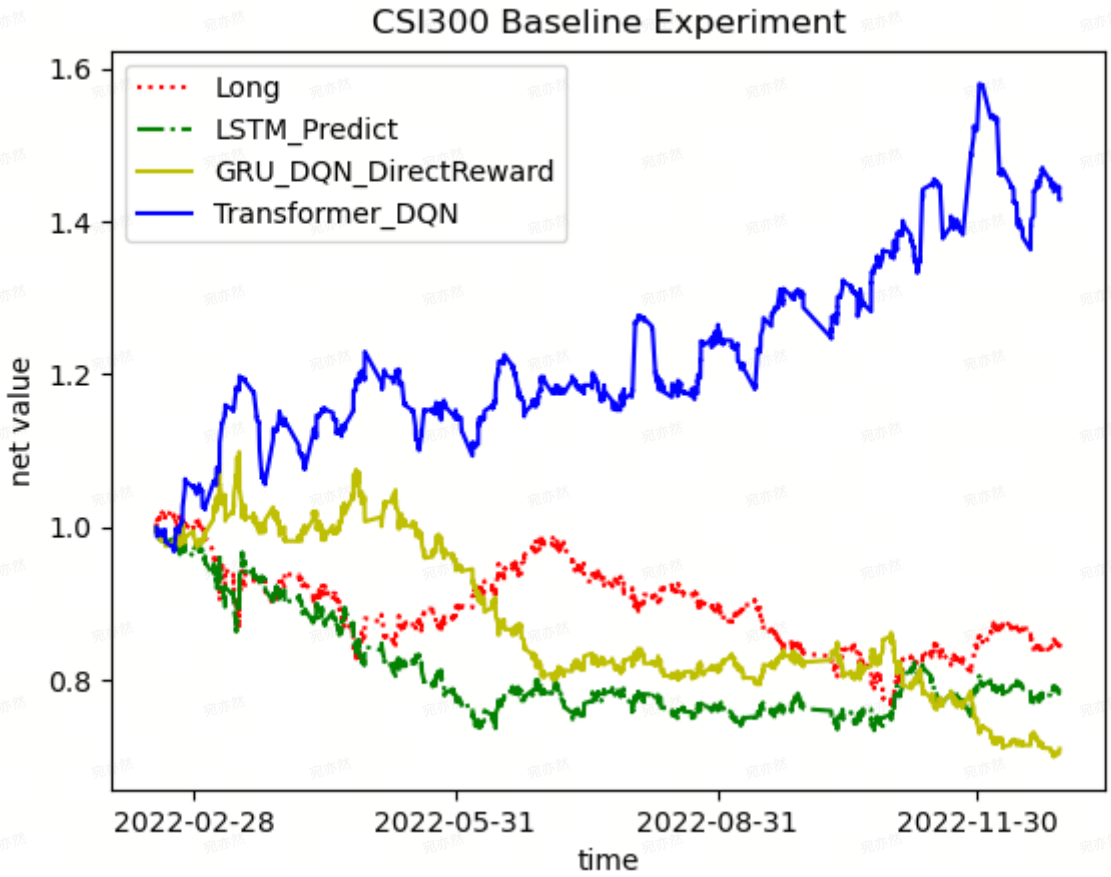
3、使用以浮盈浮亏作为激励函数的GRU-DQN的深度强化学习模型，它是非策略无模型的强化学习算法，在动作空间为离散形式表现良好，被运用在期货交易上，它交易的是价格运动方向。

上证50结果如图：



模型\指标	对数收益率	夏普率	最大回撤率
Long	-0.1628	-0.8377	-0.3222
LSTM_Predict	-0.1982	-1.0133	-0.2122
GRU_DQN_DirectReward	-0.4959	-8.3131	-0.4959
Transformer_DQN	0.3077	0.68433	-0.2271

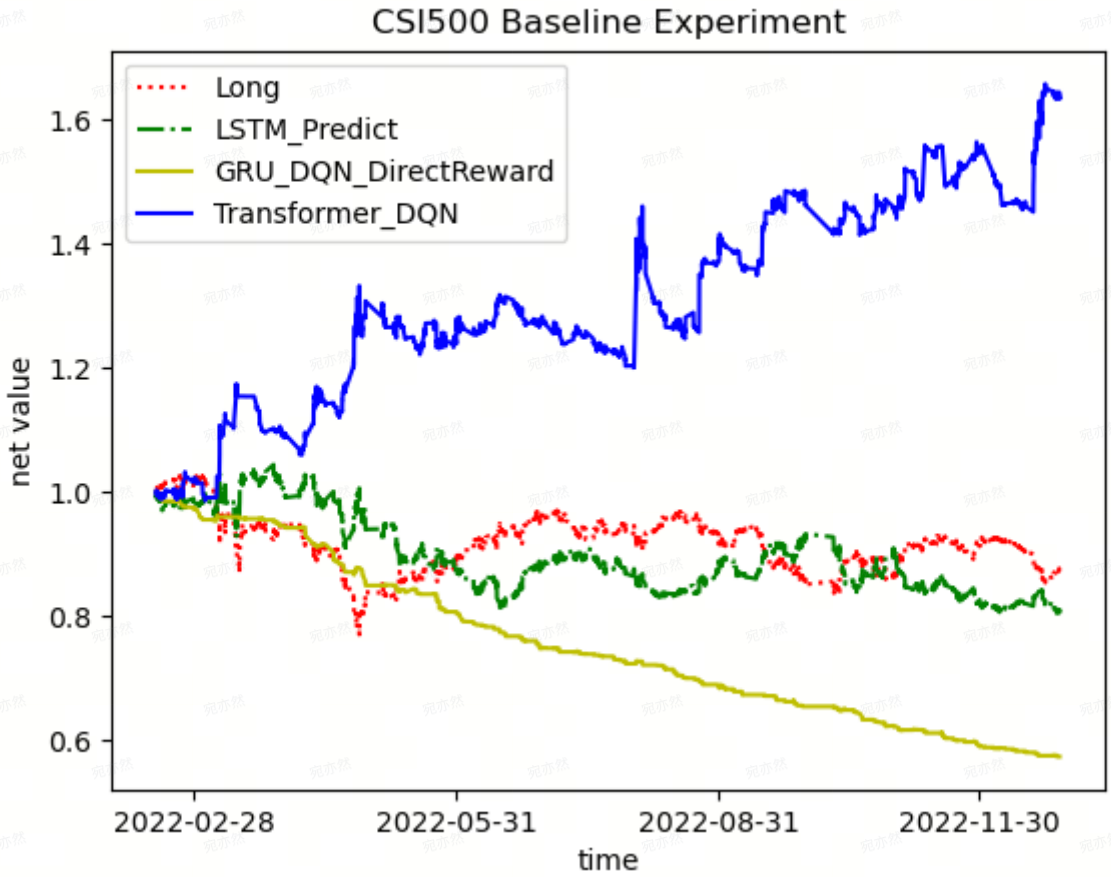
沪深300结果图如下：



评价指标：

模型\指标	对数收益率	夏普率	最大回撤率
Long	-0.1667	-0.8725	-0.2847
LSTM_Predict	-0.2406	-1.2588	-0.3054
GRU_DQN_DirectReward	-0.3422	-1.8369	-0.4505
Transformer_DQN	0.3565	1.42582	-0.1475

中证500结果如图：



模型\指标	对数收益率	夏普率	最大回撤率
Long	-0.1348	-0.6753	-0.2868
LSTM_Predict	-0.2113	-1.0599	-0.2600
GRU_DQN_DirectReward	-0.5545	-9.6939	-0.5563
Transformer_DQN	0.4912	1.9104	-0.1578

5.3.2结果展示与分析

结果如图所示，本文提出的Transformer_DQN跨式期权量化交易模型性能显著优于基于股价预测进行交易的模型和以盈利为奖励来交易价格方向的深度强化学习模型。由于A股市场波动大，使得在在预测价格运动方向较为困难，一旦交易方向出现错误，就会面临较大亏损，而Transformer-DQN转向专注于学习价格波动相关信息。通过承担时间损耗的风险来对冲价格运动方向的风险，只要价格偏离开仓点位一定程度即可产生盈利，以此在高度不确定的市场中寻找潜在的盈利可能性。LSTM-Predict基于股价预测模型进行交易表现不佳的是由于仅仅追求胜率而忽略了赔率因素，没有让模型学习在获胜时能盈利多少时应该止盈，而失败时在亏损多少时进行止损。而交易本身是不可能完全避免亏损的，能持续产生盈利一定是胜率和赔率的综合结果。GRU-DQN-DirectReward则表现训练几乎是失败

的，由于市场的高波动性使得一会儿产生盈利，一会儿又会面临亏损，这在模型训练中由于频繁的奖惩切换产生了巨大的干扰，从而使得模型难以学习有价值的信息。

在不同的数据集上，Transformer-DQN的性能表现也有一定差异，其盈利表现从SSE50，SCI300，SCI500依次变强。这是由于跨式期权本身的特点所导致的，它是依靠价格的波动来产生盈利。很明显，由于SSE50的主要成分股均为大盘蓝筹股，其波动较小，自然其盈利表现较差，而CSI500主要是中小盘股，个股表现更为活跃，波动更大，所以其盈利表现最强，CSI300处于这两者之间。

5.4消融实验

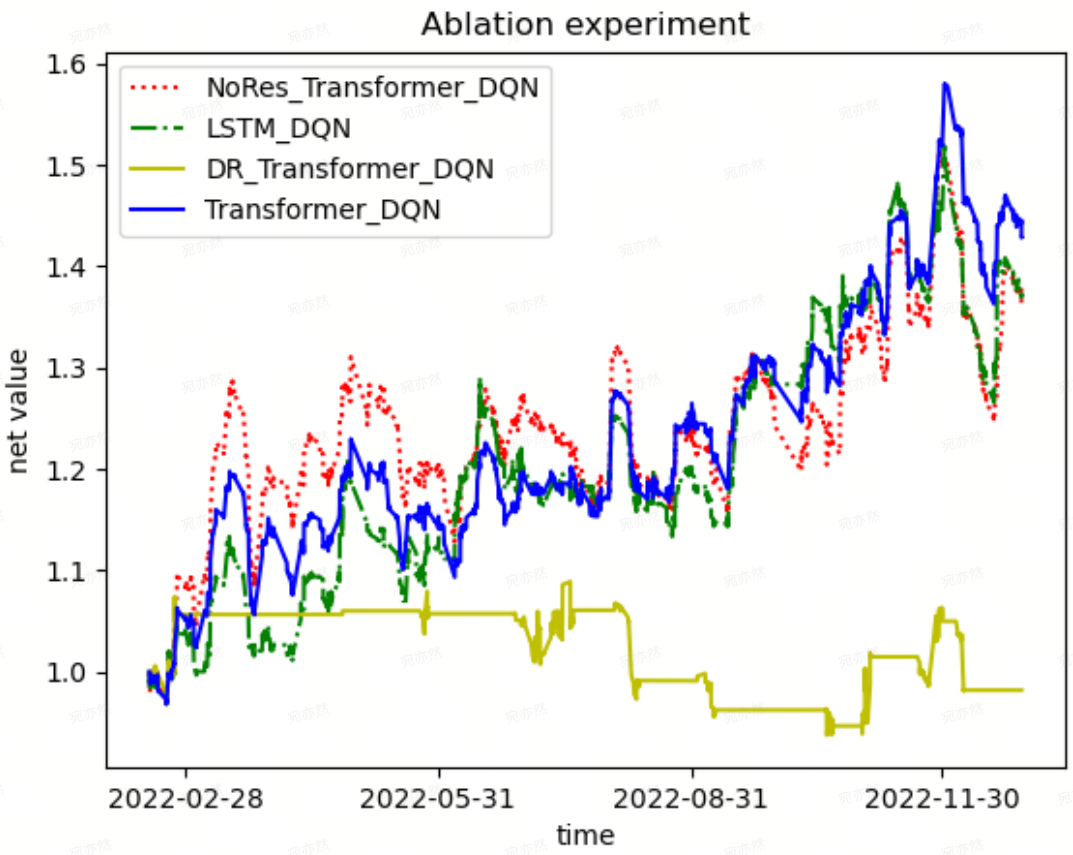
本文在沪深300指数上进行了消融实验，以显示本文模型的各个部分是如何影响实验最终结果。本文选用了三个模型变体，每个变体更改了模型的一个组件：

1、模型变体NoRes-Transformer-DQN：掩盖阻力位信息，其余不变

2、模型变体DR-Transformer-DQN：采用常见的业绩考核指标：收益率，将其作为激励函数，其余不变

3、模型变体LSTM-DQN：用于估计Q(S,a)的神经网络TransformerEncoder部分使用LSTM来学习 Seq_t 的信息，其余不变。

结果如图所示：



（去掉波动率，阻力位等信息，吧激励函数改回来,transfmoer换lstm，和原本的模型做比较）

模型\指标	对数收益率	夏普率	最大回撤率
NoRes_Transformer_DQN	0.3063	0.9861	-0.1925
DR_Transformer_DQN	-0.0183	-0.1070	-0.1498
LSTM_DQN	0.3176	1.1918	-0.1818
Transformer_DQN	0.3565	1.4242	-0.1475

依据图*所示，三种模型的变体性能均不如完整的模型Transformer_DQN。对于NoRes_Transformer_DQN而言，其波动过大。由于去掉了阻力位信息，使得模型仅仅关注于当下的波动而忽略了历史上重要的多空博弈区域，及其容易发生在高位出现接盘波动率的情况。对于DR_Transformer_DQN而言，训练可以认为是完全失败的。由于构建的是跨式期权头寸，因此指数在横盘小幅震荡时，其头寸的盈亏会发生摇摆（即一会盈利一会亏损），这对模型产生了极大的干扰，使其无法学习指数波动特征的信息。对于LSTM_DQN而言，其性能略逊于Transformer-DQN。通过查看交易信息，我们发现发现其交易次数高达600多次，而Transformer-DQN只有300多次，交易频率过高，导致交易手续费侵蚀了一部分利润。说明LSTM-DQN对于短期行情波动过于敏感而忽略了历史上的波动信息。由于Transformer相比于LSTM能更好的捕捉长依赖性，可以有效过滤市场上的噪音波动，从而专注于抓取大行情的波动信息。

6、结论及未来工作

本文提出了Transformer-DoubleDQN模型来学习跨式期权量化交易策略，其重点是研究交易资产的波动性。相比于交易资产价格运动方向的模型，其主要风险来源是时间的损耗而不是资产价格的波动，因此可以更好的管理风险敞口，不会因黑天鹅事件而产生较大亏损，反而可以因此受益。隐含波动率是影响期权价格的重要因素之一，正常情况下隐含波动率往往围绕着历史波动率上下震荡。由于受到实验条件限制，本文假设隐含波动率近似等于历史波动率，但在真实市场中，期权市场的隐含波动率代表了一种市场情绪，难以被预测。当市场出现大幅度下跌时，隐含波动率会大幅度飙升，在后续行情中期权买方也要注意期权隐含波动率降低的风险。因此，如何将期权隐含波动率信息融入深度强化学习模型将会是我们未来的研究方向。

7、参考文献

1. Andersen, A. C., & Mikelsen, S. (2012). A novel algorithmic trading framework applying evolution and machine learning for portfolio optimization. Department of Industrial Management and Technology.
2. Brenner, M., Ou, E. Y., & Zhang, J. E. (2006). Hedging volatility risk. *Journal of Banking & Finance*.
3. Gravdal, A., & Vollset, K.B. (2023). Optimization and Machine Learning Methods for The Stochastic Joint Replenishment Problem under Seasonal Demand.

4. Henry, J., & Jace, R. (2024). Machine Learning in Financial Markets: Predictive Modeling for Trading Strategies.
5. Htun, H.H., Biehl, M., & Petkov, N. (2024). Forecasting relative returns for S&P 500 stocks using machine learning. *Financial Innovation*.
6. Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E. W. T., & Liu, M. (2015). Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing, 35*, 530-541.
7. Jadeja, S. C., Patel, S., & Patel, S. (2023). Delta value-based algorithm to control loss in option selling strategies for volatile indexes. ICSET
8. Kownatzki, C., Putnam, B., & Yu, A. (2022). Case study of event risk management with options strangles and straddles. *Review of Financial Economics*.
9. Lapan, H., & Moschini, G. (1991). Production, hedging, and speculative decisions with options and futures markets. *American Journal of Agricultural Economics*.
10. Mhlongo, N.Z., Falaiye, T., Daraojimba, A.I., & Olubusola, O. (2024). Artificial Intelligence in stock broking: A systematic review of strategies and outcomes. World Journal of Advanced Research and Reviews.
11. Millea, A. (2021). Deep reinforcement learning for trading—A critical survey. *Applied Sciences*.
12. Shivaprasad, S. P., & Geetha, E. (2022). Choosing the right options trading strategy: Risk-return trade-off and performance in different market conditions. *Investment Management and Financial Innovations*.
13. Vinitnantharat, N., Inchan, N., & Worasucheep, C. (2019). Quantitative trading machine learning using differential evolution algorithm. Proceedings of the 2019 Joint Conference on AI.
14. Wang, Q., Zhang, L., Zeng, Y., Wu, S., Yu, C., & Li, J. (2024). Enhancing Stock Trading Strategies: Integrating Discrete Wavelet Transformation with Deep Q-Network. *Journal of Circuits, Systems and Computers*.
15. Yang, H., Liu, X. Y., Zhong, S., & Walid, A. (2020). Deep reinforcement learning for automated stock trading: An ensemble strategy.
16. Yu, Y., Lin, Y., Hou, X., & Zhang, X. (2023). Novel optimization approach for realized volatility forecast of stock price index based on deep reinforcement learning model.*Expert Systems with Applications.*
17. Zhang, J., & Lei, Y. (2022). Deep reinforcement learning for stock prediction.
18. Lim Q Y E, Cao Q, Quek C. Dynamic portfolio rebalancing through reinforcement learning[J]. Neural Computing and Applications, 2022, 34(9): 7125-7139.