# 基于强化学习的量化交易采样研究

摘要

深度强化学习在金融投资领域的应用已经早有时日，市面上的许多量化投资企业都使用了深度强化学习。然而，深度强化学习存在训练上的困难，且股票市场是复杂多变的，更加强了模型的训练难度，主要的原因便是强化学习的算法在训练样本的选择上不契合股票市场。本文在综合了三种actor-critic算法：近端优化算法(PPO)、优势actor-critic算法(A2C)以及深度确定性策略梯度算法(DDPG)得到的集成交易模型上，基于牛顿冷却定律针对算法采样方案进行了优化，以提升模型训练的稳定性。

我们选用30只道琼斯股票搭建训练环境，针对不同算法在不同的参数区间中进行了多轮组合测试。通过平均收益、夏普率分别衡量模型在不同参数下的性能及风险调整收益。选中一组表现较好的参数作为应用参数，在中国股市中进行测试。

【关键词】：深度强化学习、actor-critic算法、牛顿冷却定律

研究意义

随着科技水平的提高以及算力的提高，深度强化学习的训练问题得到了较好的解决，因为强化学习训练得到的智能体效果拔群，因此许多研究者尝试将深度强化学习结合到各自的学科从而带来更大的提升，深度强化学习应用较为成功的领域便是金融投资领域。

早期科学家们通过简单的雅达利游戏环境训练和测试深度强化学习模型时便发现，基于深度强化学习的智能体在经过训练后能够习得一些高阶的隐藏特征，从而表现出高级玩家的水平，这种特性十分契合金融投资领域。由于股票市场的复杂多变，基础的股民们需要非常高的代价学习，即使是资深股民也不一定能够掌握其中的规律，而深度强化学习模型经过训练可以掌握高维特征，从而为雇主获得稳定的投资收益。前人(论文插入)提出基于三种actor-critic算法：近端优化算法(Proximal Policy Optimization以下简称PPO)、优势actor-critic算法(Advantage Actor Critic以下简称A2C)以及深度确定性策略梯度算法(Deep Deterministic Policy Gradient以下简称DDPG)算法，构建一个集成模型，该模型以上述三种模型的夏普率为选择标准，在预测时选用夏普率高的模型作为表现模型，从而达到较好的性能。

但是强化学习模型普遍存在训练困难的问题，尤其是在多变的金融领域。早期强化学习模型多用游戏环境进行训练，为满足样本独立同分布的假设，因此科学家们提出的算法多对训练样本进行随机抽取，这种采样方法使得模型难以学习环境中的连续性特征，因此本文基于牛顿冷却定律对采样算法进行重新设计，让模型尽量学习较新的样本，尝试降低模型的训练难度，并略微提升性能表现。

预备知识

马尔可夫决策过程

马尔可夫决策过程(Markov Decision Process以下简称MDP)是强化学习(Reinforcement Learning以下简称RL)的理论基础。它以四元组来表示状态空间、动作空间、状态转移函数以及奖励函数，使用四元组记录决策对环境的影响以及环境的反馈，其中表示时刻环境状态、表示时刻做出的决策或行动、表示环境在作用下在时刻的状态、表示时刻做出的动作带来的反馈或奖励。通过调整策略函数使得模型朝着收益最大化的方向学习，从而得到最优策略。

策略函数可以接收当前环境状态，经过计算输出应该采取的行为，对于不同的训练环境，策略可以分为确定性策略与随机性策略。初始时需要预先设定一个策略，然后智能体会基于该策略与环境交互，得到一系列的状态-动作序列。而后，MDP会将上述动作得到环境反馈序列加权求和记为当前状态的收益通过对当前状态的收益取期望得到当前状态的价值(Value)，用于表示当前状态的未来期望收益，其中为衰减因子，当时，模型会专注于眼下的奖励；当时，模型会着眼于未来的收益。类似的可以定义动作价值函数，用于表示当前状态下，动作a可能带来的收益，不难发现，状态价值函数与动作价值函数存在如下关系



将收益表达式代入状态价值函数中并进行适当的变形可以得到如下贝尔曼方程



同样的，将收益表达式代入动作价值函数中并适当变形可得如下贝尔曼方程



利用上述的两个贝尔曼方程，我们可以利用迭代的方法对马尔可夫决策过程中的所有状态计算其状态价值及动作价值，然后根据动作价值函数贪心地选择动作执行从而得到最优策略。

强化学习及算法

强化学习是一种基于马尔可夫决策过程的机器学习方法，与其他机器学习方法以减少损失函数为目标不同的是，强化学习的目标是最大化奖励函数。在强化学习中需要构造智能体(Agent)，让智能体依据环境(Environment)的状态决定采取行动与环境交互，通过环境给予的反馈(Reward)调整自身的行动策略(Policy)，从而得到能够在该环境中获取高奖励的策略。与马尔可夫决策过程类似，强化学习的目标是得到状态价值函数和策略函数，进而得到能够使整体收益最大的行动策略。

对与强化学习而言，存在两种较为普遍的算法，一种为价值迭代算法(Value iteration以下简称VI算法)，另一种为策略梯度算法(Policy Gradient以下简称PG算法)。后续为了融合两种算法的优点，科学家们提出了行动者-评论者算法(Actor-Critic以下简称AC算法)，并基于此提出了多种改进算法如优势AC算法(Advantage Actor-Critic以下简称A2C算法)、近段优化算法(Proximal Policy Optimization以下简称PPO算法)以及深度确定性策略梯度算法(Deep Deterministic Policy Gradient)，下面我们分别介绍这几种算法。

VI算法

VI算法是结合了通过贝尔曼方程对价值函数进行多次迭代从而得到稳定的，然后根据贪心的进行策略选择的算法。

算法首先假设动作空间与状态空间是有限的，并对所有状态的价值初始化为0。然后根据如下方程组





对所有的状态进行迭代更新，直到，即所有的状态价值收敛为止。其中表示在状态中采取动作时环境的奖励，表示在在状态中采用动作转移到状态的概率，表示第次迭代时状态的状态价值，表示第次迭代时在状态中采用动作的动作价值，表示衰减因子。

得到收敛的后对于每个状态通过以下方程



即贪心地选择使得下一个状态的收益为最大的动作作为当前状态的动作选择，如此便得到对于MDP过程中所有状态的最优动作选择策略。

PG算法

PG算法是基于蒙特卡洛采样以及梯度上升的策略函数训练方法。其目的是得到一个策略函数，以当前状态作为输入，输出期望收益最大的动作。

算法首先需要定义一个初始策略函数，其中为该函数的参数。接着定义目标函数用于表示某个马尔可夫决策过程的期望收益，其中表示服从策略函数的某个具体策略，表示在状态中采取动作的收益。为了估计目标函数，可以采用蒙特卡洛方法，对策略函数进行m次随机采样，得到m个具体策略，分别对每个具体策略进行T次模拟，得到T个描述环境变化的四元组，并使用如下方程利用收益估计目标函数。



在每次计算得到目标函数后，通过链式法则计算其对于的导数，即策略梯度，并对进行更新，其中为学习率。当或达到最大迭代次数时停止迭代并得到局部最优策略。

AC算法

AC算法融合了VI算法的价值函数以及PG算法的策略函数，使用Actor网络表示策略函输出当前状态的最佳动作和Critic网络表示价值函数输出当前状态价值，并通过两者的互相监督训练，从而使得两个网络都能取得较好的效果。

首先定义Actor网络与Critic网络，其中和分别为Actor网络和Critic网络的参数。设t时刻的状态为，将作为参数输入Actor网络得到t时刻的动作，带入Critic网络得到动作价值，并通过与环境交互得到奖励和下一时刻的状态，并重复上述过程得到、、以及。通过公式



得到时序差分误差(Temporal-Difference error)，其中可视为状态的价值，为衰减因子，因此可视为下一时刻的状态价值与上一时刻的估计值的差距。分别求出两个网络相对自身参数的导数和，利用TD-error对参数和进行更新。





重复上述步骤直到且或达到最大迭代次数为止，即可得到最优的策略网络和评价网络。

A2C算法

A2C算法是一种改进了的AC算法，它引入了如下优势函数(Advantage Function)





其中为状态价值函数，即状态的平均价值，通过该状态的所有动作价值的均值求得。优势函数表示，在状态下采取动作相比于平均水平的优劣程度，如此可以使得仅对高于平均水平的动作具有正向价值，即，而低于平均水平的动作具有负向价值，即。

原始的策略梯度为



其中为目标函数，为某一个具体的马尔科夫决策过程，表示在马尔可夫决策过程中的策略，表示在马尔可夫决策过程中得到的奖励。

利用蒙特卡洛方法进行随机采样对策略梯度进行估计，则策略梯度变为如下形式



其中和分别表示在马尔可夫决策过程中t时刻的状态与动作，表示在状态时，策略采取动作的概率。

利用优势函数替代奖励函数用于估计策略梯度，则策略梯度变为如下形式



为了同一时间只更新一个网络的参数，为此，将动作价值函数通过公式



转化为状态价值函数并带入梯度策略公式中，得到



此时Critic网络变为估计状态价值的网络，Actor网络仍为策略输出网络。通过该梯度分别更新两个网络直到参数收敛或者达到最大迭代次数为止，即可得到局部最优的行动策略。

A2C方法通过引入基线(Base line)构造优势函数，并用于估计策略梯度从而达到提高学习效率和提升模型稳定性的目的。

PPO算法

PPO算法是一种在A2C的基础上引入了重要性采样(Importance Sampling)以及策略梯度限制机制改进的强化学习算法，它的目标是在保证策略更新的安全性和有效性的同时，提高数据利用率和学习速度。

因为A2C算法在训练时会存在因为一个动作选择较差，从而得到较差的策略梯度，使用较差的策略梯度去更新网络参数会使得策略变差，从而在下一时刻得到更差动作的选择，并难以跳出这个循环的问题。究其原因在于，A2C算法在更新参数时并没有考虑到模型参数更新前后变化的大小，只是一味依据梯度更新，从而可能出现上述情况。而PPO算法则将参数变化加入目标函数中考虑，从而限制模型参数更新前后的变化幅度，不至于一步错步步错。

算法首先引入了重要性采样。定义两个策略和，前者是需要更新的策略，后者是与环境交互进行采样的策略。为了让两个策略的期望一致，因此需要对目标函数进行适当的变化。



但是即使两个策略的期望值是一致的，方差也并不一致，因此为了让两个策略的方差相差不多，我们需要一个指标用于衡量两个策略的相似程度。我们可以使用KL散度作为衡量指标，KL散度的计算公式如下。



不难发现，当时，；当与非常不同时，，因此可以通过限制两个策略的KL散度从而维持其相似度，同时也限制了模型参数的更新幅度不会过大以至于陷入A2C的负优化循环之中。

算法首先初始化策略参数、KL散度的最大值以及KL散度的处罚因子。根据策略在环境中采样得到轨迹，依据轨迹估计优势函数，利用优势函数替代奖励函数得到目标函数



求其相对于的梯度并通过如下公式更新



其中当时，说明两个策略的相似度较小，因此需要引导模型进行较小的参数调整，令；当时，说明两个策略相似度较大，因此可以让模型进行较大程度的参数调整，令。重复若干次上述步骤，直到参数收敛或达到最大迭代次数，即可获得局部最优策略。

DDPG算法

与A2C和PPO边训练边更新不同的是，DDPG是训练模型与采样模型完全分离的算法，通过采样模型与环境进行交互收集数据并存放在回放缓存中，然后在回访缓存中采样，对训练模型进行训练，若干次训练后再将采样模型与训练模型的参数同步，再进行上述循环。且DDPG算法输出的并非动作分布，而是确定的连续动作，通过对给定动作添加噪声再采样从而得到一定随机性的动作。基于这两个特点，DDPG不需要假设特定分布的动作空间也可以在连续动作空间的环境进行训练，同时可以利用离线数据提高数据的利用效率。

算法首先初始化两个模型，一个是采样模型，一个是训练模型。每个模型中都有两个网络，即Actor网络与Critic网络，不妨设采样模型的Actor网络为，Critic网络为，训练模型的Actor网络为，Critic网络为，并初始化回放缓存。首先在环境中初始化随机噪声，并得到时刻的初始状态。然后对于每个时刻根据当前采样策略加上噪声得到动作，然后在环境中执行动作并得到奖励与下一时刻的状态，并将四元组存入回放缓存，然后令继续重复上述步骤直到缓存中有足够的样本。

当缓存中有足够的样本后，从其中随机采样一部分样本用于训练，计算训练网络的目标值。



利用目标值，通过均方误差计算Critic网络的损失值。



最小化该损失值从而更新Critic网络的参数。对于Actor网络首先估计期望奖励的梯度。



利用该梯度更新参数。

对训练网络进行更新后，将训练网络与采样网络的参数部分同步。





不是完全同步是为了使采样网络与训练网络保持一定的差异，以避免过拟合或震荡现象，其中表示更新比例。

经过若干次训练后，即可得到局部最优的Critic网络，通过对输入的状态贪心的选择Q值最大的动作即可得到最佳策略。

前置模型

随着时代发展以及技术革新，强化学习在量化投资领域的研究和应用已有许多成果，本文选取了一种前人研究的一个集成模型作为基础，接下来就介绍一下这个集成模型。

对于复杂多变的股票交易市场而言，简单的VI算法和PG算法都是不太合适的。对于VI算法而言，由于股票交易的动作空间是连续的，因此VI算法需要求解的Q表格将会变得维度非常大，从而难以求解。PG算法则是可能存在因市场波动导致的梯度剧烈变化而导致训练困难的问题。因此前人使用结合了VI算法和PG算法优点的三个算法，即A2C、PPO和DDPG作为基础，构建集成模型。

文中首先介绍了股票环境的搭建方法。作者使用了OpenAI Gym框架来搭建股票训练环境，该环境可以模拟一个投资者在股票市场上进行买卖操作的过程。作者使用了美国道琼斯指数（DJIA）中的30只股票作为交易对象，这些股票的数据来源于雅虎财经（Yahoo Finance），时间范围为2009年1月1日到2019年12月31日，数据频率为每日收盘价。作者对每只股票的价格进行了对数差分处理，以消除非平稳性和趋势性，并使用了技术分析指标（如MACD、RSI等）来增强特征向量。

为了描述股票交易环境，作者定义了一个马尔可夫决策过程，该马尔可夫决策过程包含以下五个要素：

·状态空间S：每个状态表示时刻股票投资者所面临的市场状态以及自身的投资状态。其中包括30只股票在当前时刻的价格，投资者持有的30只股票的份额以及剩余流动资金。

·动作空间A：每个动作表示在时刻状态时采取的动作，对于每只股票而言，可以有买入、卖出以及持有三种选择，这三种操作将会导致对应股票的份额增加、减少以及不变。

·奖励函数：在时刻执行动作后，会由状态变化为状态，使用记录状态变化时获得的奖励。

·策略函数：表示状态时的交易策略。输入为某一个时刻状态，输出为在该状态下采取的动作的概率分布。

·Q值：表示在状态情况下采取动作的评价，在文中使用采取动作后的期望回报作为该动作的评价。

因为投资者的流动资金不能为负，因此在执行股票买入操作时需要限制买入的数量，以满足流动资金大于等于零的条件。同样的，投资者持有的股票数量不能为负，因此在执行股票卖出操作时需要限制卖出的数量，以满足股票持有大于等于零的条件。

考虑到实际情况中，股票交易需要雇佣经纪人，而经纪人的佣金各不相同，为了尽可能简单，在模型中作者假设交易成本是每笔交易价值的0.1%，这里的交易包括买入或卖出行为。

同时为了规避现实中可能导致股市崩盘的情况，例如战争、金融泡沫破裂、经济危机等，需要对模型的风控能力有一定要求，因此作者采用金融振荡指数对其进行衡量，其计算公式如下



其中表示时刻的股票收益，表示股票的历史平均收益，表示股票历史收益的协方差矩阵。当振荡值高于设定的阈值时，表明市场正处于高风险时期，此时为了保证投资人的利益，设定Agent会自动停止购入股票并将所持有的所有股票抛售。当振荡值低于阈值时，表明市场回归到正常环境，此时Agent恢复交易。

参考文献：

[1]黄志刚,刘全,张立华等.深度分层强化学习研究与发展[J].软件学报,2023,34(02):733-760.DOI:10.13328/j.cnki.jos.006706.