量化策略



# 机器学习系列（2）：强化学习模型轮动框架下的行业配置

**周萧潇 分析员**

SAC 执证编号：S0080521010006

SFC CE Ref：BRA090

xiaoxiao.zhou@cicc.com.cn

**郑文才 分析员**

SAC 执证编号：S0080523110003

SFC CE Ref：BTF578

wencai3.zheng@cicc.com.cn

**刘均伟 分析员**

SAC 执证编号：S0080520120002

SFC CE Ref：BQR365

junwei.liu@cicc.com.cn

我们在机器学习系列第一篇报告《机器学习系列（1）：使用深度强化学习模型探索因子构建范式》中使用强化学习模型生成因子表达式，所挖掘的因子在样本外有效性较为显著。本篇报告作为机器学习系列报告的第二篇，我们将回归强化学习的优势领域：组合优化任务。

我们将深度强化学习模型应用到行业配置（行业轮动） 中，充分发挥强化学习在序列决策任务上的优势，让强化学习模型选出可能具有相对优势的行业。在尝试了多种创新性训练方法以避免模型的过拟合等关键问题后，我们发现使用**强化学习模型轮动的训练框架**可以最终得到**兼顾稳定与收益**的结果。

**强化学习的优势领域：组合优化**

我们认为强化学习相较于其他传统机器学习模型而言，与环境交互更新策略的方式使其更加适用于序列决策任务， 这使得其与多为时序类型的金融数据更加匹配。我们使用强化学习模型在金融领域较为经典的FinRL框架， 全面梳理并测试PPO 、TRPO 、SAC 、A2C 、DDPG和TG3 六大主流强化学习模型在行业配置上的表现。

**单次训练模式：缺乏确定性的收益**

由于机器学习尤其是强化学习的高复杂度，模型稳定性一直是业界较为关注的议题。我们首先使用单次切分的方法把样本数据以 2022 年 1 月为界划分成样本内外。在样本内进行模型训练，在样本外验证其有效性。我们发现周频调仓下相对于行业等权和因子等权来说，两年期**样本外分别获得 11.9%和 6.0%的年化超额收益**，但并不是所有随机数种子的样本外都可以达到这一理想结果。

在对其稳定性进一步测试时，我们使用了不同强化学习模型、超参数、随机数种子以及不同样本区间起点等条件测试其对于训练效果的影响。 **我们发现训练敏感性方面：模型选择>样本区间>随机数种子>关键超参数**。以上不同选择带来的超额收益率极差可达 20%以上。因此单次训练模型面对的样本外不稳定风险是我们接下来主要解决的问题。

**平衡收益与风险：拓展搜参vs模型轮动**

为了缓解模型的不稳定性，我们尝试**拓展训练超参搜索**和**模型滚动**两种方法对模型进行更新训练。两种方式都确保模型能够及时接受最新的市场数据，实验表明模型轮动的思路更好地兼顾了稳定性与高收益。

我们首先使用拓展训练的模型， 固定每一期样本内长度，样本外数据则采用外推半年的方法确定，下一期样本内截止时间依次推后半年。在每一期样本内的训练中我们引入Optuna框架寻找最优参数，将样本内最优的参数组合及模型应用于当期样本外。拓展训练结果显示相对于单次训练的平均效果并没有太明显的超额收益，尤其是在因子等权超额的角度来看，和单次训练差距不大。

而使用滚动模型框架时我们测试得到在因子等权和行业等权的超额收益对比单次训练的样本外平均水平都有了显著提升。**模型滚动框架相对行业等权样本外年化超额收益 16.4%，相对于因子等权年化超额收益 7.7%**。稳定性方面，在多次滚动中模型战胜等权基准的**胜率均达到 100%**，表现也显著优于单次训练的结果，效率上并行训练方案也比搜参方案平均**提升超过 5 倍**。

风险提示：样本内测试结果不代表样本外表现的可持续性， 不同测试框架可能会带来测试结果的差异。

- 量化策略 | 机器学习系列（1）：使用深度强化学习模型探索因子构建范式 (2024.04.07)
- 量化策略 | 量化多因子系列（12）：高频因子手册 (2024.01.15)
- 量化策略 | 另类数据策略（2）：如何优化新闻文本因子 (2023.09.12)
- 量化策略 | 量化多因子系列（7）：价量因子手册 (2022.08.06)
- 量化策略 | 量化多因子系列（5）：基本面因子手册 (2022.04.26)

# 目录



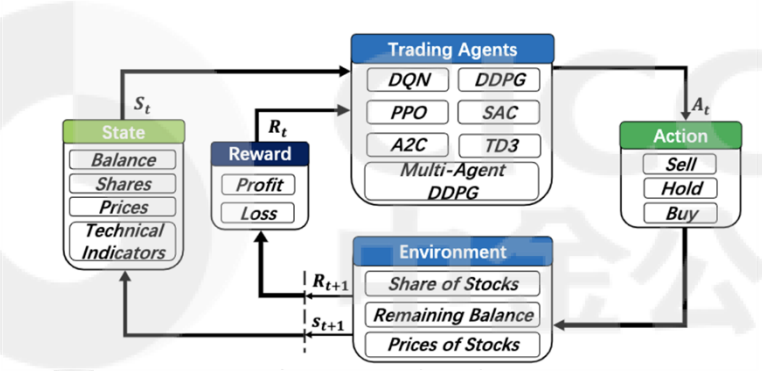
# 图表

# 深度强化学习算法应用到行业配置任务

## 面向时间序列金融场景的深度强化学习模型

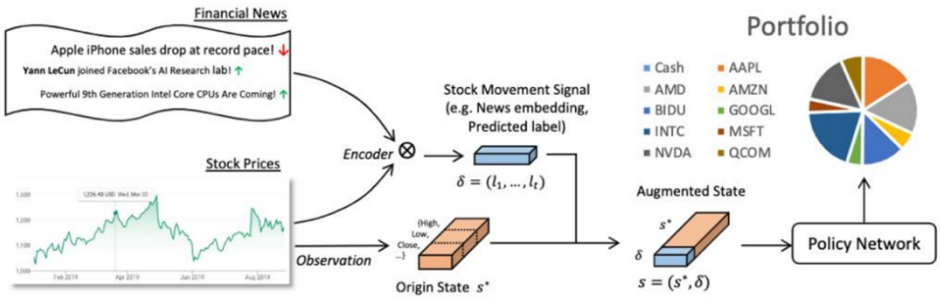
近年来随着人工智能技术的日益兴起，深度强化学习被大量应用到了围棋、游戏和自动驾驶等序列决策场景。在金融领域类似的场景以资产配置/组合优化任务为主。例如， Liu 等人在 2021 年提出的 FinRL 框架[1]，首次将前沿的强化学习算法系统应用到了资产配置任务中， 并开源了对应的算法框架；Ye 等人在 2020 年提出的SARL 模型[2]，增加资产信息与价格变动预测作为额外的状态， 该预测可以仅基于财务数据（例如资产价格）或利用新闻等替代来源的资产数据。也有部分研究在因子挖掘及合成任务上取得了突破性的进展[3]。如我们在机器学习系列第一篇报告《机器学习系列（1）：使用深度强化学习模型探索因子构建范式》中使用强化学习模型生成因子表达式的思路，将原论文改进后所挖掘的因子在样本外有效性较为显著。本篇报告作为机器学习系列报告的第二篇，我们将回归强化学习的优势领域： 组合优化，并将其应用到行业配置任务中。

**图表 1：FinRL 整体框架**




资料来源： FinRL: Deep Reinforcement Learning Framework to Automate Trading in Quantitative Finance. Xiao-Yang Liu. 2021

**图表 2：SARL 整体框架**




资料来源： Reinforcement-Learning based Portfolio Management with Augmented Asset Movement Prediction States. Yunan Ye. 2020

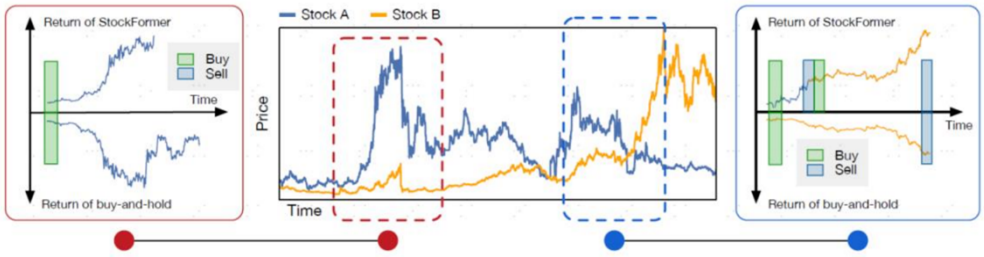
[1] FinRL: Deep reinforcement learning framework to automate trading in quantitative finance.

[2] Reinforcement-learning based portfolio management with augmented asset movement prediction states.

[3] Generating Synergistic Formulaic Alpha Collections via Reinforcement Learning.

强化学习在资产配置任务取得如此广泛应用的一个主要原因在于其在序列决策任务上的灵活性。以股票交易场景为例，相较于固定策略在给定窗口的起始和结束时间分别执行买和卖的交易行为，强化学习策略能够更灵活的选择卖点以获取更高的收益；此外， 当存在多只股票时，相比固定策略只会挑选一只股票进行投资，强化学习模型可以更灵活的进行调仓，即图示中先买入股票 A，而后调仓买入股票 B，使整体收益最大化。

**图表 3：强化学习交易策略相比固定策略在资产配置任务上的优势**




资料来源： StockFormer: Learning Hybrid Trading Machines with Predictive Coding. Siyu Gao. 2023

## 序列决策任务建模流程与主流算法介绍

我们在《机器学习系列（1）：使用深度强化学习模型探索因子构建范式》提出强化学习具有四个特点： 1.适合处理序列决策任务；2.输入数据无需遵从独立同分布的假设；3.通过与环境交互探索来不断优化当前策略； 4.数据无需具备标签。 针对行业序列决策任务，我们训练强化学习智能体与市场环境进行交互，这个交互过程可以通过马尔可夫决策过程进一步展开说明。当下主流的强化学习算法框架为演员-评论家算法（Actor-Critic），代表性的模型包括 SAC 和 A2C 与 PPO，它们也是本篇报告行业配置预测框架所使用的主要方法。

### 马尔可夫决策过程

强化学习模型的数学基本框架为马尔可夫决策过程（MDP），它共包含五个核心元素： (S, A, T, R, γ)，其目标为最大化每一个回合累计奖励值，公式如下，其中 P 表示当前回合的总步长。

$$G_t = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=1}^{P}\gamma^{t-1}r_t\right]$$

对标行业配置任务，每个元素的定义如下：

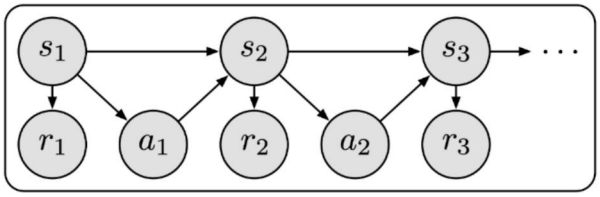
- S 表示可观测的数据集合（强化学习示意图中的状态$s_t$集合）， 在行业配置任务中，它可以定义为行业因子的集合，行业量价数据，交易账户余额， 每只行业持仓信息等。
- A 表示动作集合(强化学习示意图中的动作$a_t$集合)，在行业配置任务中，它被定义为一个长度为 N （行业数量），元素值域为[0, 1]的向量， 表示模型对 N 个行业未来收益的打

分。在后续利用该动作向量与环境时，根据打分排序取前六名行业进行等权买入操作。

- T 表示状态转移概率方程$T(s_{t+1}|s_t, a_t)$。
- R 表示奖励函数（强化学习示意图中的$r_t$），可以根据预期的策略进行针对性的设计，在行业配置任务中，它可以被定义为净值变化率、相较于行业等权基准的超额收益夏普比率等。

- γ 表示折扣因子，值域为(0,1)，表示对未来奖励值打折扣。有些时候，可以把折扣因子设为 0，智能体只关注当前的奖励；也可以把折扣因子设为 1，对未来的奖励并没有打折扣，未来获得的奖励与当前的奖励是一样的。折扣因子可以作为超参数进行调整。

**图表 4：马尔可夫决策过程示意图**

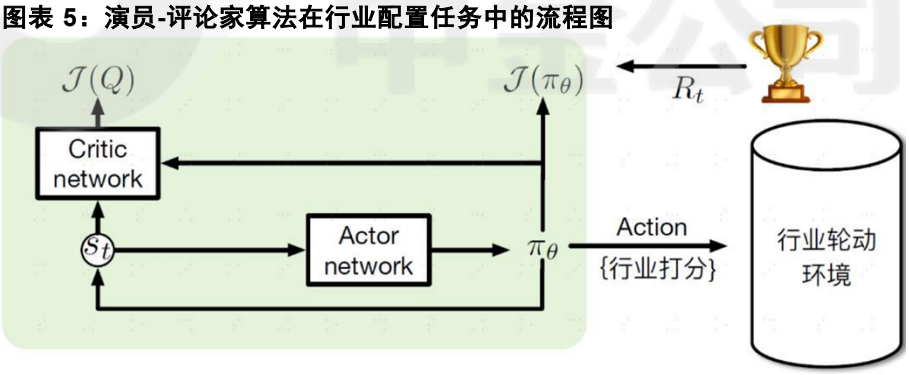



资料来源： StockFormer: Learning Hybrid Trading Machines with Predictive Coding. Siyu Gao. 2023

## 演员-评论家（Actor-Critic）结构

当下强化学习模型中一大主流的算法框架为演员-评论家（Actor-Critic）结构，它同时考虑了策略梯度和时序差分学习的思想。该算法包含两个主要组成模块：演员和评论家网络。其中，演员指代的是策略函数$\pi_\theta(a_t|s_t)$，即学习一个策略以得到尽可能高的回报，评论家是指价值函数$Q_\pi(s_t, a_t)$，对当前策略的价值函数进行估计，即评估演员的好坏。借助于价值函数，演员-评论家算法可以进行单步参数更新，不需要等到回合结束才进行更新，因此具有采样效率高，采样方差有所下降的优势。在演员-评论家中，代表性的算法为 SAC（Soft-Actor-Critic）和 A2C（Advantage Actor-Critic）算法。

**图表 5：演员-评论家算法在行业配置任务中的流程图**




资料来源： StockFormer: Learning Hybrid Trading Machines with Predictive Coding. Siyu Gao. 2023

## SAC 算法

SAC 算法是一种采用离线策略[4]的算法，它通过行为策略与环境进行交互采集样本，其中样本可以表示为包含五个变量的元组：$(s_t, a_t, s_{t+1}, r_t, d_t)$，元组中$d_t$代表当前回合是否结束。SAC 算法具有良好的鲁棒性和抗干扰能力，以及较好的探索能力；但相对应的它的稳定性和收敛能力偏弱，策略的表现依赖奖励函数的定义。SAC 采用了三个网络： 一个策略（Actor）网络和两个 Q 值（Critic）网络， 同时引入了熵正则项来鼓励策略的随机性。

[4] 离线策略的方法将收集数据作为强化学习算法中单独的一个任务，它准备两个策略：行为策略与目标策略。行为策略是专门负责学习数据的获取，具有一定的随机性，总是有一定的概率选出潜在的最优动作。目标策略借助行为策略收集到的样本以及策略提升方法提升自身性能，并最终成为最优策略。

目标值使用目标 Q 值网络和策略网络来计算：

$$y_i = r_i + \gamma * [\min(Q_{\phi}'_1(s_{i+1}, a_{i+1}), Q_{\phi}'_2(s_{i+1}, a_{i+1})) - \alpha \log \pi_\theta(a_{i+1}|s_{i+1})]$$

**更新 Q 值网络**：最小化损失函数。

$$L(\phi_j) = \frac{1}{N}\sum\left(Q_{\phi_j}(s_i, a_i) - y_i\right)^2$$

**策略更新中的熵优化：**使用策略梯度方法来进行优化，通过最大化$\hat{Q}_\pi(s_t, a_t)$的目标函数， SAC 算法能够有效地在连续动作空间中进行采样， 从而提高采样效率，实现性能优化。

$\log \pi_\theta(a_i|s_i)$表示策略的熵。 通过最大化策略的熵来优化策略。熵是一个度量策略不确定性的指标，有助于策略更加均衡和多样化，进而提高算法对于不同环境和任务的适应性。

$$J(\theta) = \frac{1}{N}\sum\left(\alpha \log \pi_\theta(a_i|s_i) - \min\left(Q_{\phi_1}(s_i, a_i), Q_{\phi_2}(s_i, a_i)\right)\right)$$

**价值函数学习：** 引入了价值函数的学习，通过学习价值函数，可以更准确地估计状态-动作对的价值。价值函数的学习可以通过最小化贝尔曼（Bellman）误差来实现，进一步提高算法的性能。

**自适应温度参数：** 引入了自适应温度参数 α， 通过优化温度参数的选择，可以在最大化预期累计奖励和最小化策略熵之间取得平衡。自适应温度参数能够更好地适应不同任务和环境，提高算法的性能。

$$L(\alpha) = -\alpha(\log \pi_\theta(a_i|s_i) + \mathcal{H})$$

**软更新目标 Q 值网络**：

$$\phi_{\boldsymbol{j}}' \leftarrow \tau\phi_j + (1-\tau)\phi_{\boldsymbol{j}}'$$

## A2C 算法

A2C 算法是一种采用在线策略[5]的算法，具有训练速度快和提升整体策略稳定性的优势， 在小型环境中表现良好。与之对应的，模型不一定能够找到最优策略；且由于每次更新都需要进行新的采样，模型对样本的利用效率较低。除了与 SAC 相似的策略优化和价值函数学习的核心思想，A2C 算法的独特贡献在于提出了优势函数（A），它的数学表达式为：A=Q-V。优势函数表示的是在当前状态下执行某个动作的价值（Q）与该状态平均价值（V）的差异，有助于提高模型的效率和稳定性，减少策略网络的数值差异。

$$A_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$$

$$\nabla_\theta J(\theta) = \nabla_\theta \log \pi_\theta(a_t|s_t) A_t$$

$$L(\phi) = (1/2)[r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)]^2$$

A2C 算法的核心思想是使用两个网络： 一个是策略网络（Actor），另一个是价值网络

（Critic）。Actor：生成动作的概率分布，根据当前状态选择动作。 Critic：评估当前状态的价值，用来衡量动作选择的好坏。

[5] on-policy 的算法只能使用当前正在优化的 policy 生成的数据来进行训练。

## DDPG 算法

DDPG（Deep Deterministic Policy Gradient）是一种用于处理连续动作空间的强化学习算法。DDPG 结合了深度学习和策略梯度方法， 适用于高维度和连续动作的复杂任务。 DDPG 是基于 Actor-Critic 架构的算法，结合了确定性策略梯度和深度 Q 学习（DQN）的思想。其核心包括两个主要的网络：Actor 网络和 Critic 网络。

**Actor 网络**：负责生成给定状态下的动作。它是一个确定性策略函数，表示为$\mu(s \mid \theta^{\mu})$，其中$\theta^{\mu}$是 Actor 网络的参数。

**更新 Actor 网络**：使用确定性策略梯度更新 Actor 网络。

$$\nabla_{\theta}^{\mu} J \approx (1/N) * \Sigma(\nabla_{a} Q(s, a|\theta^{Q})|_{s=si, a=\mu(si)} \nabla_{\theta}^{\mu} \mu(s|\theta^{\mu})|_{s=si})$$

**Critic 网络**：负责评估给定状态-动作对的价值。它是一个 Q 值函数，表示为$Q(s, a|\theta^{Q})$，其中$\theta^{Q}$是 Critic 网络的参数。

**更新 Critic 网络**：计算目标 Q 值$y_i$以及最小化损失函数：

$$y_i = r_i + \gamma * Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$$

$$L(\theta^{Q}) = (1/N) * \Sigma(y_i - Q(s_i, a_i|\theta^{Q}))^2$$

软更新策略网络：

$$\theta^{Q'} \leftarrow \tau * \theta^{Q} + (1-\tau) * \theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau * \theta^{\mu} + (1-\tau) * \theta^{\mu'}$$

其中$\tau$是软更新的速率。

TD3 （Twin Delayed Deep Deterministic Policy Gradient）是对 DDPG（Deep Deterministic Policy Gradient）算法的改进版本，用于处理连续动作空间的强化学习问题。TD3 通过引入几种关键改进来解决 DDPG 中的一些问题，如过估计偏差和不稳定性。 TD3 的核心思想在于通过使用双重 Critic 网络、目标策略平滑和策略延迟更新等技术，来提高学习的稳定性和性能。

**双重 Critic 网络：**

TD3 使用两个独立的 Critic 网络$Q_1$和$Q_2$来估计状态-动作对的价值。

目标 Q 值通过这两个 Critic 网络的最小值计算，从而减少过估计偏差：

$$y = r + \gamma * \min\left(Q_1\left(s', \mu'\left(s'|\theta_{\mu}'\right)\right), Q_2\left(s', \mu'\left(s'|\theta_{\mu}'\right)\right)\right)$$

TD3 中的 Actor 网络和目标网络更新的频率低于 Critic 网络。这意味着 Critic 网络在 Actor 网络更新前会多次更新，从而提高 Critic 网络的准确性。通常， Critic 网络在每个时间步更新，

而 Actor 网络每隔固定的步数（例如每 2 个时间步）才更新一次。TD3 能够处理高维度的连续动作空间，适用于机器人控制、自动驾驶等领域。TD3 的这些改进使其在需要高精度和稳定性的连续控制任务中表现优异。

## TRPO 算法

TRPO（Trust Region Policy Optimization，信赖域策略优化） 是一种用于强化学习的策略梯度方法，由 Open AI 的 John Schulman 等人于 2015 年提出[6]。TRPO 的主要目标是通过引入信赖域约束，确保策略更新时不会出现过大的变化，从而提高训练过程的稳定性和可靠性。但实现和调整 TRPO 算法可能需要更多的计算资源和调参经验， 因为需要解决一个带约束的优化问题。

**策略梯度：**

$$g = E_t[\nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_{\theta old}}(s_t, a_t)]$$

**策略优化目标：** TRPO 优化目标是在给定旧策略$\pi_{\theta_{old}}$的情况下， 找到一个新策略$\pi_\theta$，使得策略更新后性能得到提升。 TRPO 通过解决以下优化问题来更新策略：

$$max_\theta E_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A^{\pi_{\theta_{old}}}(s_t, a_t) - \beta D_{KL}\left(\pi_{\theta_{old}}(\cdot|s_t)|\pi_\theta(\cdot|s_t)\right)\right]$$

其中：

$\pi_\theta(a_t|s_t)$是当前策略的动作选择概率。

$\pi_{\theta_{old}}(a_t|s_t)$是旧策略的动作选择概率。

$A^{\pi_{\theta old}}(s_t, a_t)$是在状态$s_t$选择动作$a_t$的优势函数。

$D_{KL}$是 KL 散度，衡量两个概率分布之间的距离。

$\beta$是用来调整 KL 散度约束的超参数。

由于直接优化上述目标函数并满足约束条件是一个复杂的优化问题， TRPO 通过二阶泰勒展开近似 KL 散度，并将优化问题转化为一个二次规划问题（Quadratic Programming, QP），从而使问题更加易于求解。

**信赖域约束：** TRPO 通过引入信赖域约束，确保新旧策略之间的变化不会过大。具体来说，这个约束可以用 KL 散度（Kullback-Leibler Divergence）来表示。其中$\delta$是事先设定的信任区域半径。

$$\theta_{new} = \arg\max_\theta g \quad \text{subject to} \quad D_{KL}(\pi_{\theta_{old}}(\cdot|s_t)|\pi_\theta(\cdot|s_t)) \leq \delta$$

**迭代优化：**在每次迭代中，TRPO 通过采样当前策略与环境的交互数据，计算优势函数， 并利用二次规划方法进行策略更新。整个过程不断迭代，逐步提升策略性能。

## PPO 算法

PPO（Proximal Policy Optimization，近端策略优化）是一种强化学习算法， 仍由 Open AI 的 John Schulman 在 2017 年提出[7]。PPO 是 TRPO 的改进版，它保留了 TRPO 的核心思想， 但

通过简化约束优化问题，使算法更易于实现和计算效率更高。PPO 使用了“剪辑”方法来限制策略更新的大小，从而确保策略在更新过程中不会发生过大的变化。以下是 PPO 算法的原理和

[6] Trust Region Policy Optimization. John Schulman. 2015

[7] Proximal Policy Optimization Algorithms. John Schulman. 2017

核心内容：

**策略优化目标：**

$$L^{CLIP}(\theta) = E_t[\min(r_t(\theta) * A_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon) * A_t)]$$

其中：

$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$是策略概率比率。

$\pi_\theta(a_t|s_t)$是当前策略的动作选择概率。

$\pi_{\theta_{old}}(a_t|s_t)$是旧策略的动作选择概率。

$A_t$是在状态$s_t$选择动作$a_t$的优势函数，衡量在状态$s_t$选择动作$a_t$相对于平均水平的优越性。

$\varepsilon$是一个超参数，用于控制策略更新的范围。

**剪辑目标函数**（Clipped Objective Function）：PPO 算法会对策略更新的目标函数进行剪辑，以确保策略在更新过程中不会发生过大的变化。具体来说，PPO 通过“剪辑”策略概率比率$r_t(\theta)$来限制策略更新的范围。具体来说，如果$r_t(\theta)$超过$1+\varepsilon$或低于$1-\varepsilon$，则将其剪辑到这个范围内。

**Trust Region 方法：** PPO 受到了 Trust Region Policy Optimization（TRPO）算法的启发，但与 TRPO 不同的是，PPO 通过一种更简单的方式来实现类似的效果。TRPO 通过解决一个复杂的优化问题来限制策略更新的范围，而 PPO 则采用了更简单且易于实现的剪辑方法。相比于 TRPO，PPO 更易于实现和调试。适用于大规模和高维度的强化学习问题。

# 单次训练模式：缺乏确定性的收益

强化学习模型以序列决策任务见长，但同时模型收益性和稳定性的权衡也是备受讨论的话题之一。尤其是在金融领域，如果收益的稳定性无法保障，那么回测出来的高收益曲线也只是空中楼阁。本文将在这一章首先提出强化学习模型在时序数据预测或者组合优化任务中的一般框架：从数据的处理到模型的构建再到模型表现的测试。

在对其稳定性进一步测试时，我们使用了不同超参数、随机数种子以及不同样本区间起点等数据测试其对于模型的影响。 **我们发现模型敏感性方面：模型选择>样本区间>随机数种子>关键超参数**。以上不同选择带来的超额收益率极差可达 20%以上。因此单次训练模型面对的样本外不稳定风险是我们主要面对的问题。

## 强化学习预测时序数据基础框架

### 数据集构造

本文采用中信一级行业作为行业配置任务的数据来源。考虑到模型收敛对样本量的需求，删除历史数据长度较短的综合金融（CI005030.WI）行业，保留其他 29 个行业。根据下图所展示的有效行业价量及基本面因子， 将数据集的长度范围定义为 2013/07/09-2024/03/31，根据所采用的具体训练模式决定训练集和测试集的划分节点。其中，训练集的样本数据用于训练模型参数，而测试集数据则用于检验模型样本外表现及策略稳定性。

我们使用中金量化及 ESG 团队开发的因子库中的因子作为模型的输入数据。中金量化及 ESG 因子库现存有数百个较低相关的核心因子，包含价量、基本面、另类数据和高频数据因子共 11 类。具体因子构建方式及表现见《量化多因子系列（12）：高频因子手册》 、《另类数据策略（2）：如何优化新闻文本因子》、《量化多因子系列（7）：价量因子手册》和《量化多因子系列（5）：基本面因子手册》。

我们沿用了《行业轮动系列（4）：轮动节奏自适应行业轮动 2.0 模型》的方法，将中金价量和基本面因子从个股层面以下述方式合成至行业层面，对因子全时点有效性进行了测试。对于每种因子，我们测试了因子原值以及基于原值的 3 种变化构建方式，分别为年同比（差分）、月环比（差分）和 3 年时序标准分。我们在每一类因子中选取行业层面 ICIR 最高的因子一共 11 个，作为深度强化学习模型的输入数据。

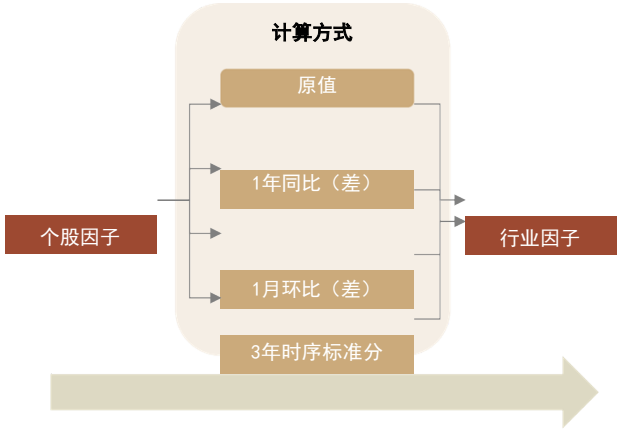
**图表 6：行业因子合成方法参数**

| 设置 | 参数 |
| --- | --- |
| **行业指数标的** | 中信一级行业 |

| | |
|---|---|
| **测试区间** | 2013-07-09至2024-03-31 |
| **调仓频率** | 月频 |
| **个股因子处理** | 去极值、标准化 |
| **合成至行业加权方式** | 流通市值加权 |
| **成分股缺失值处理** | 若非缺失成分股市值占该行业80%，则较具代表性，使用已有成分股数据合成行业数据；反之，则视为该行业缺失 |
| **行业因子处理** | 去极值、标准化、市值中性 |

资料来源： 中金公司研究部

**图表 7：因子合成变化构建方式**




资料来源： 中金公司研究部

**图表 8：入选行业因子测试结果**

| 因子代码 | 含义 | 类型 | IC均值 | IC_IR | 多空收益 | 超额收益 | 超额回撤 | 多头盈亏比 | 转换类型 |
|---|---|---|---|---|---|---|---|---|---|
| mmt_sec_rank_A | 1年横截面rank动量 | 动量及反转 | 0.0682 | 0.25 | 14% | 8% | -13% | 111% | yoy |
| cs_weighted_relevance_0.6_imp_2 | 高信噪比新闻情感因子 | 另类 | 0.0427 | 0.24 | 8% | 5% | -26% | 119% | origin |
| liq_shortcut_avg_6M | 6个月最短路径非流动因子 | 流动性 | 0.049 | 0.21 | 8% | 3% | -18% | 112% | zscore |
| doc_vol10_ratio | 高频筹码分布因子 | 高频因子 | -0.0492 | 0.19 | 7% | 3% | -20% | 123% | zscore |
| corr_ret_turnd_prior_1M | 换手率变动与收益率相关性因子（量价同步） | 相关性 | 0.0377 | 0.18 | 5% | 4% | -21% | 132% | zscore |
| vol_std_1M | 一个月波动率因子 | 波动率 | 0.0354 | 0.16 | 7% | 4% | -10% | 127% | zscore |
| OP_Q_YOY | 营业利润增速（单季度同比） | 成长 | 0.0556 | 0.29 | 9% | 2% | -23% | 118% | zscore |
| BP_LR | 市净率倒数 | 估值 | -0.0661 | 0.25 | 11% | 6% | -13% | 111% | yoy |
| BUY_SHIFT_DIST_XL | 超大单买入的位移路程比因子 | 资金流 | 0.0483 | 0.19 | 4% | [illegible] | -22% | 111% | |
| HN_z | 股东数目时序标准分数 | 公司治理 | 0.0394 | 0.18 | 8% | | -24% | 127% | |
| RPP_75D | 75日内预测报告数量 | 情绪 | 0.033 | 0.17 | 6% | | -18% | 103% | |

注：1）统计时间为 2013-07-09 至 2024-03-31；2）调仓频率为月频； 3）超额收益的比较基准为中信一级行业等权基准

资料来源： Wind，中金公司研究部

## 预测框架基础设定

结合任务特性，强化学习模型有如下设定：状态空间为 $S \in R^{N\times M}$（M代表输入的因子数量，本文有 $N = 29,\ M = 11$）；行为空间为 $A \in [0,1]^{N\times 1}$，表示智能体对行业未来收益预测的打分；奖励函数 R 定义为强化学习策略相较于行业等权基准的超额收益。具体而言，对于每个交易日 t，智能体可以获得截至该日期的行业因子数据$s_t$，并输出对下期行业收益预测的打分

$a_t$；在$t+1$交易日开盘价等权买入预测得分最高的六个行业，定义换仓频率为周频以及0.03%比例的手续费，随后环境依据调仓前后所获收益计算奖励值反馈给智能体，完成一次交互闭环。

训练过程中，每个回合包含的步数$p=1000$；模型训练步数设定为$steps=200,000$；折扣因子$Y=0.99$；强化学习模型的底层网络结构采用全连接网络，激活函数采用 ReLU 函数；优化器采用 Adam 优化器。本文所有实验均可采用一张 GeForce RTX 3080Ti 完成，强化学习模型代码主要基于开源三方库 Stable-Baselines3。

本文采用如下两个对照基准与强化学习模型的策略进行比较，并采用三个指标对策略性能进行评价：年化收益率、夏普比率和最大回撤。

- 行业等权基准：采用与强化学习模型相同的交易策略等权买入 29 个行业。