

MASA: Developing A Multi-Agent and Self-Adaptive Framework with Deep Reinforcement Learning for Dynamic Portfolio Risk Management

Abstract: Deep or reinforcement learning (RL) approaches have been adapted as reactive agents to quickly learn and respond with new investment strategies for portfolio management under the highly turbulent financial market environments in recent years. In many cases, due to the very complex correlations among various financial sectors, and the fluctuating trends in different financial markets, a deep or reinforcement learning based agent can be biased in maximising the total returns of the newly formulated investment portfolio while neglecting its potential risks under the turmoil of various market conditions in the global or regional sectors. Accordingly, a multi-agent and self-adaptive framework namely the MASA is proposed in which a sophisticated multi-agent reinforcement learning (RL) approach is adopted through two cooperating and reactive agents to carefully and dynamically balance the trade-off between the overall portfolio returns and their potential risks. Besides, a very flexible and proactive agent as the market observer is integrated into the MASA framework to provide some additional information on the estimated market trends as valuable feedbacks for multi-agent RL approach to quickly adapt to the ever-changing market conditions. The obtained empirical results clearly reveal the potential strengths of our proposed MASA framework based on the multi-agent RL approach against many well-known RL-based approaches on the challenging data sets of the CSI 300, Dow Jones Industrial Average and S&P 500 indexes over the past 10 years. More importantly, our proposed MASA framework shed lights on many possible directions for future investigation.

This work is accepted by the 23rd International Conference on Autonomous Agents and Multi-Agent Systems ([AAMAS-24](#)). This repository is the official implementation of the MASA framework. For more details, please refer to the paper: [arXiv:2402.00515](#).

Keywords: Portfolio Optimisation; Risk Management; Multi-agent System; Deep Reinforcement Learning.

Dataset

Due to the data policies of the data platforms, the used data sets in this work can not be provided here. Please download the stock data and market data from the platforms such as [Yahoo Finance](#), [RiceQuant](#),

and [JoinQuant](#). For more data possible sources, please refer to [AI4Finance-FinRL](#).

To facilitate the use of the customised data set in the MASA framework, an example of the data structure is provided as follows.

File Name	Fields	Description
\\data\\DJIA_10_1d.csv	date, stock, open/high/low/close price, volume	The 'stock' field is the ranking of stocks in terms of company captial. For the file name "DJIA_10_1d.csv", "DJIA" is the market symbol, "10" is the scale of a portfolio (i.e., the number of assets) while "1d" denotes that the trading frequency is 1-Day (i.e., daily trading).
\\data\\DJIA_1d_index.csv	date, tic, open/high/low/close price, volume	The 'tic' field is the symbol of the market index (e.g., DJIA, CSI 300, and S&P 500).
\\data\\DJIA_10_60m.csv (Optional)	date, stock, open/high/low/close price, volume	The 'stock' field is the ranking of stocks in terms of company captial. The data may provide the latest information to the market observer. "60m" in the file name is the data frequeuncy is 60 minutes.
\\data\\DJIA_60m_index.py (Optional)	date, tic, open/high/low/close price, volume	The 'tic' field is the symbol of the market index (e.g., DJIA, CSI 300, and S&P 500). The data may provide the latest information to the market observer.

Requirements

Please run on Python 3.x, and install the libraries by running the command:

```
python -m pip install -r requirements.txt
```

- The experiments of the MASA framework and baseline approaches are run on a GPU server machine installed with the AMD Ryzen 9 3900X 12-Core processor running at 3.8 GHz and two Nvidia RTX 3090 GPU cards.

Entrance Script for Training

You may configure the algorithm and trading settings in `config.py` . After that, run the below command to start training.

```
python entrance.py
```

Details of Building Running Environment

If encountering any problems when building the running environment, you may follow the below steps by using the docker container:

1. Download the pytorch image.

```
docker pull pytorch/pytorch:1.11.0-cuda11.3-cudnn8-devel
```

2. Build the docker container

```
docker run --gpus all -itd -p {host_port}:{container_port} --name masa -v {filepath_in_desktop}:{filepath_in_container} pytorch/pytorch:1.11.0-cuda11.3-cudnn8-devel
```

- `host_port` is the host communication port between the host desktop and the container, you may specify a free port in the desktop. (e.g., 55555)
- `container_port` is the container communication port. (e.g., 22)
- `filepath_in_desktop`: the absolute path of masa files in the host desktop.
- `filepath_in_container`: the absolute mapping path of masa files in the container.

3. Enter the container

```
docker exec -it masa bash
```

- Then locate to the mapping masa file path.

4. Install relevant python libraries

```
python -m pip install -r requirements.txt
```

5. Manually install more libraries

Except for the listed libraries in the requirements.txt that can be installed via **pip install**, there are a few other libraries that have to be manually installed.

- **TA-LIB** (Installation reference source: <https://github.com/TA-Lib/ta-lib-python>)
 - Download the source file via <https://sourceforge.net/projects/ta-lib/>, and move the .tar.gz file to the masa folder in the container.
 - Execute the commands:
 - `tar -xzf ta-lib-0.4.0-src.tar.gz` (change the file name when using other versions of TA-LIB)
 - `cd ta-lib/`
 - `./configure --prefix=/usr`
 - `make`
 - `make install`
 - `pip install TA-LIB`

6.Others:

- If having the error: *"ImportError: libGL.so.1: cannot open shared object file: No such file or directory"*, run the command **apt-get update**, and then **apt-get install libgl1-mesa-glx**.
- If having the error: *"ImportError: libgthread-2.0.so.0: cannot open shared object file: No such file or directory"*, run the command **apt-get update**, and then **apt-get install libglib2.0-0**.

Acknowledgement

- Compared Algorithm Implementation: [PGPortfolio](#)
- Trading Environment: [FinRL](#)
- TD3 Implementation: [Baselines3](#)
- Financial Indicator Implementation: [TA-Lib](#)
- Second-order Cone Programming Solver: [CVXOPT](#) and [CVXPY](#)

We appreciate that they share their amazing works and implementations. This project would not have been finished without their works.

Others

- The implementation is for research purpose only. Please be noted that **the market always has risks**.
- Should you have any questions, please do not hesitate to contact me: lzlong@hku.hk

Reference

Please consider citing our work if you find it helpful to yours.

Pending