FinRL-Meta: A Universe of Near-Real Market Environments for Data-Driven Deep Reinforcement Learning in Quantitative Finance

Xiao-Yang Liu¹, Jingyang Rui², Jiechao Gao³, Liuqing Yang¹, Hongyang Yang¹, Zhaoran Wang⁴, Christina Dan Wang⁵, Jian Guo⁶
¹Columbia University; ²The University of Hong Kong; ³University of Virginia; ⁴Northwestern University; ⁵New York University (Shanghai); ⁶IDEA Research.
XL2427@columbia.edu; zhaoranwang@gmail.com; guojian@idea.edu.cn

Abstract

Deep reinforcement learning (DRL) has shown huge potentials in building financial market simulators recently. However, due to the highly complex and dynamic nature of real-world markets, raw historical financial data often involve large noise and may not reflect the future of markets, degrading the fidelity of DRL-based market simulators. Moreover, the accuracy of DRL-based market simulators heavily relies on numerous and diverse DRL agents, which increases demand for a universe of market environments and imposes a challenge on simulation speed. In this paper, we present a FinRL-Meta framework that builds a universe of market environments for data-driven financial reinforcement learning. First, FinRL-Meta separates financial data processing from the design pipeline of DRL-based strategy and provides open-source data engineering tools for financial big data. Second, FinRL-Meta provides hundreds of market environments for various trading tasks. Third, FinRL-Meta enables multiprocessing simulation and training by exploiting thousands of GPU cores. Our codes are available online at https://github.com/AI4Finance-Foundation/FinRL-Meta.

1 Introduction

In quantitative finance, market simulators play important roles in studying the complex market phenomena and investigating financial regulations [1, 2]. Compared to traditional simulation models, deep reinforcement learning (DRL) has shown huge potentials in building financial market simulators through multi-agent systems [3]. However, due to the high complexity of real-world markets, raw historical financial data involve significant noise and may not reflect the future of markets. This issue usually degrades the fidelity of DRL-based simulation. Moreover, in reality, there are innumerable participators that impact together on the market. To better simulate the market, numerous and diverse DRL agents are needed to represent those participators with different aims and strategies.

Recently, researchers have explored various applications of DRL in quantitative finance [3, 8, 9, 10]. Lussange et al. [8] have proposed a market simulation model using multi-agent reinforcement learning. Although it has shown the feasibility of DRL-based market simulation, only a few DRL agents are used. The potentials of DRL-based market simulators are not fully explored yet. The FinRL framework [8] has proposed a DRL framework as a full pipeline of developing trading strategies and is growing an open-source community AI4Finance that can contribute diverse DRL agents. However, it [8] focuses on developing trading strategies instead of building market simulations.

In this paper, we develop a FinRL-Meta framework that is a universe of near real-market environments for data-driven financial reinforcement learning. First, we apply the DataOps paradigm...
Overview of FinRL-Meta

Data Layer

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Type</th>
<th>Range and Frequency</th>
<th>Request Limits</th>
<th>Raw Data</th>
<th>Preprocessed Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yahoo! Finance</td>
<td>US Securities</td>
<td>Frequency-specific, 1min</td>
<td>2,000/hour</td>
<td>OHLCV Prices &amp; Indicators</td>
<td>Prices &amp; Indicators</td>
</tr>
<tr>
<td>CCXT</td>
<td>Cryptocurrency</td>
<td>API-specific, 1min</td>
<td>API-specific</td>
<td>OHLCV Prices &amp; Indicators</td>
<td>Prices &amp; Indicators</td>
</tr>
<tr>
<td>WRDS.TAQ</td>
<td>US Securities</td>
<td>2003-now, 1ms</td>
<td>5 requests each time</td>
<td>Intraday Trades Prices &amp; Indicators</td>
<td>Prices &amp; Indicators</td>
</tr>
<tr>
<td>Alpaca</td>
<td>US Stocks, ETFs</td>
<td>2015-now, 1min</td>
<td>Account-specific</td>
<td>OHLCV Prices &amp; Indicators</td>
<td>Prices &amp; Indicators</td>
</tr>
<tr>
<td>RiceQuant</td>
<td>CN Securities</td>
<td>2005-now, 1ms</td>
<td>Account-specific</td>
<td>OHLCV Prices &amp; Indicators</td>
<td>Prices &amp; Indicators</td>
</tr>
<tr>
<td>JointQuant</td>
<td>CN Securities</td>
<td>2005-now, 1min</td>
<td>3 requests each time</td>
<td>OHLCV Prices &amp; Indicators</td>
<td>Prices &amp; Indicators</td>
</tr>
<tr>
<td>QuantConnect</td>
<td>US Securities</td>
<td>1998-now, 1s</td>
<td>NA</td>
<td>OHLCV Prices &amp; Indicators</td>
<td>Prices &amp; Indicators</td>
</tr>
</tbody>
</table>

Table 1: Data platforms. OHLCV means open, high, low, close, volume data.

MDP Model for Trading Tasks: We model a trading task as a Markov Decision Process (MDP) \((S,A,P,r,\gamma)\), where \(S\) and \(A\) denote the state space and action space, respectively, and \(P(s'|s,a)\) denotes the transition probability, \(r(s,a)\) is a reward function, and \(\gamma \in (0,1)\) is a discount factor. Specifically, the state denotes an observation that a DRL agent receives from a market environment; the action space consists of actions that an agent is allowed to take at a state; the reward function \(r(s,a,s')\) is the incentive for agents to learn a better policy. A trading agent aims to learn a policy \(\pi(s_t|a_t)\) that maximizes the expected return \(R = \sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)\).

Overview of FinRL-Meta: We utilize a layered structure, as shown in Fig. 1. FinRL-Meta consists of three layers: data layer, environment layer, and agent layer. Each layer executes its functions and is relatively independent. Meanwhile, layers interact through end-to-end interfaces to implement the complete workflow of algorithm trading.

DataOps for Data-Driven DRL in Finance: We follow the DataOps paradigm in the data layer. First, we establish a standard pipeline for financial data engineering, ensuring data of different formats from different sources can be incorporated in a unified RL framework. Second, we automate this pipeline with a data processor, which can access data, clean data and extract features from various data sources with high quality and efficiency. Our data layer provides agility to model deployment. The data sources are shown in Table 1.

Multiprocessing Training: We utilize thousands of GPU cores to perform multiprocessing training, which significantly accelerates the training process. In each CUDA core, a trading agent interacts with a market environment to produce transitions in the form of (state, action, reward, next state). Then all the transitions are stored in a replay buffer and used to update a learner and evaluator. By adopting this technique, we successfully achieve multiprocessing simulation of hundreds of market environments to improve the performance of DRL trading agents on large datasets.
Table 2: Performance of backtesting (red) and paper trading (blue) for stock trading.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cumul. return</td>
<td>0.968% / -0.652%</td>
<td>1.335% / 0.191%</td>
<td>0.099% / -1.56%</td>
</tr>
<tr>
<td>Annual return</td>
<td>22.425% / -16.746%</td>
<td>32.106% / 5.492%</td>
<td>2.108% / -35.522%</td>
</tr>
<tr>
<td>Annual volatility</td>
<td>15.951% / 14.113%</td>
<td>19.871% / 15.953%</td>
<td>9.196% / 9.989%</td>
</tr>
<tr>
<td>Sharpe ratio</td>
<td>1.457 / -1.399</td>
<td>1.621 / 0.447</td>
<td>0.289 / -4.894</td>
</tr>
<tr>
<td>Max drawdown</td>
<td>-2.657% / -1.871%</td>
<td>-2.932% / -1.404%</td>
<td>-1.438% / -2.220%</td>
</tr>
</tbody>
</table>

**Plug-and-Play**: In the development pipeline, we separate market environments from the data layer and the agent layer. Any DRL agent can be directly plugged into our environments, then trained and tested. Different agents can run on the same benchmark environment for fair comparison.

**Training-Testing-Trading Pipeline**: We employ a training-testing-trading pipeline. The DRL agent first learns from the training environment and is then validated in the validation environment for further adjustment. Then the validated agent is tested on historical datasets. Finally, the tested agent will be deployed in paper trading or live trading markets. First, this pipeline solves the information leakage problem because the trading data are generated yet when adjusting agents. Second, a unified pipeline allows fair comparisons among different trading strategies.

**Supported Trading Tasks**: We have supported and achieved satisfactory trading performance for trading tasks such as stock trading, cryptocurrency trading, and portfolio allocation. Derivatives such as futures and forex are also supported. Besides, we have supported multi-agent simulation and execution optimizing tasks by reproducing the experiment in [9].

3 Performance Evaluation

To provide benchmarks for researchers, we will continuously add typical trading tasks with corresponding environments. Here, we show results of stock trading and cryptocurrency trading.

3.1 Experiment Settings

**Stock trading task**: We select the 30 constituent stocks in Dow Jones Industrial Average (DJIA), accessed at the beginning our testing period. We use the Proximal Policy Optimization (PPO) algorithm [12] of ElegantRL [10], Stable-baselines3 [11] and RLlib [13], respectively, to train agents and use the DJIA index as the baseline. We use 1-minute data from 06/01/2021 to 08/14/2021 for training and data from 08/15/2021 to 08/31/2021 for validation (backtesting). Then we retrain the agent using data from 06/01/2021 to 08/31/2021 and conduct paper trading from 09/03/2021 to 09/16/2021. The historical data and real-time data are accessed from the Alpaca’s database and paper trading APIs.

**Cryptocurrency trading task**: We select top 10 market cap cryptocurrencies [1]. We use the PPO algorithm [12] of ElegantRL [10] to train an agent and use the Bitcoin (BTC) price as the baseline. We use 5-minute data from 06/01/2021 to 08/14/2021 for training and data from 08/15/2021 to 08/31/2021 for validation (backtesting). Then we retrain the agent using data from 06/01/2021 to 08/31/2021 and conduct paper trading from 09/03/2021 to 09/16/2021.

---

1The top 10 market cap cryptocurrencies as of Oct 2021 are: Bitcoin (BTC), Ethereum (ETH), Cardano (ADA), Binance Coin (BNB), Ripple (XRP), Solana (SOL), Polkadot (DOT), Dogecoin (DOGE), Avalanche (AVAX), Uniswap (UNI). Tether (USDT) and USD Coin (USDC) are excluded.
3.2 Trading Performance

**Stock trading:** In the backtesting stage, both ElegantRL \[10\] agent and Stable-baselines3 \[11\] agent outperform DJIA in annual return and Sharpe ratio, as shown in Fig. 2 and Table 2. The ElegantRL agent achieves an annual return of 22.425% and a Sharpe ratio of 1.457. The Stable-baselines3 agent achieves an annual return of 32.106% and a Sharpe ratio of 1.621. In the paper trading stage, the results are consistent with the backtesting results. Both the ElegantRL agent and the Stable-baselines3 agent outperform the baseline.

**Cryptocurrency trading:** In the backtesting stage, the ElegantRL agent outperforms the benchmark (BTC price) in most performance metrics, as shown in Fig. 3 and Table 3. It achieves an annual return of 360.823% and a Sharpe ratio of 2.992. The ElegantRL agent also outperforms the benchmark (BTC price) in the paper trading stage, which is consistent with the backtesting results.

4 Conclusion

In this paper, we followed the DataOps paradigm and developed a FinRL-Meta framework. FinRL-Meta provides open-source data engineering tools and hundreds of market environments with multi-processing simulation.

For future work, we are building a multi-agent based market simulator that consists of over ten thousands of agents, namely, a FinRL-Metaverse. First, FinRL-Metaverse aims to build a universe of market environments, like the Xland environment \[14\] and planet-scale climate forecast \[15\] by DeepMind. To improve the performance for large-scale markets, we will employ GPU-based massive parallel simulation as Isaac Gym \[16\]. Moreover, it will be interesting to explore the evolutionary perspective \[17\] \[18\] \[19\] \[20\] to simulate the markets. We believe that FinRL-Metaverse will provide insights into complex market phenomena and offer guidance for financial regulations.

References


