



Giao dịch chứng khoán tự động với mô hình DDPG trong thư viện FinRL

TRÍ TUỆ NHÂN TẠO - CS106.M11.KHCL

Nguyễn Dương Hải, 19521464

Email: 19521464@gm.uit.edu.vn

Github: <https://github.com/iamthedh7/Automated-stock-trading-with-DDPG-models-in-FinRL-library>

Mở đầu - Hiện nay, thị trường chứng khoán ngày càng trở nên sôi động và cởi mở, thu hút nhiều đối tượng tham gia. Những người tham gia này cần có một chiến lược phù hợp để tối đa hóa lợi nhuận cá nhân. Vì vậy, chiến lược kinh doanh cổ phiếu đóng một vai trò rất quan trọng trong đầu tư. Tuy nhiên, rất khó để thiết kế được một chiến lược trong một thị trường chứng khoán phức tạp và năng động như vậy. Trong bài báo này, tôi đề xuất một chiến lược sử dụng học sâu tăng cường Deep Reinforcement Learning (DRL) để cụ thể hóa chiến lược giao dịch chứng khoán bằng cách tối đa hóa lợi tức đầu tư.

Chúng tôi đào tạo một tác nhân DRL và có được một chiến lược giao dịch tổng hợp bằng cách sử dụng thuật toán Actor-Critic: **Deterministic Policy Gradient (DDPG)**. Mọi thứ sẽ được thực hiện thông qua một thư viện mạng học sâu tăng cường có tên là FinRL (<https://github.com/AI4Finance-Foundation/FinRL>).

I/ GIỚI THIỆU

Giao dịch cổ phiếu là một phần quan trọng của lĩnh vực tài chính, công việc của nó về cơ bản là nhà đầu tư sẽ phải quyết định đầu tư vào sản giao dịch chứng khoán nào và đầu tư bao nhiêu cổ phiếu để tối đa hóa lợi nhuận. Tuy nhiên, những rủi ro liên quan đến lạm phát, lãi suất,... cùng với những biến động của thị trường như đại dịch Covid-19 đã tác động tiêu cực đến thị trường tài chính, khiến cho việc giao dịch chứng khoán của các nhà đầu tư trở nên khó khăn và một số trường hợp dẫn đến thua lỗ nặng khi tham gia. Vì vậy điều quan trọng là phải đưa ra các chiến lược đầu tư hợp lý để đối phó với những thay đổi phức tạp của thị trường chứng khoán, điều này đang rất được giới đầu tư quan tâm trong thời điểm hiện tại.

Hiện nay, DRL nổi lên như một cách tiếp cận hiệu quả và được nhiều nhà nghiên cứu áp dụng để

tìm ra giải pháp cho vấn đề tự động hóa các giao dịch tài chính. Điều này là do mô hình Deep Neural Network (DNN) được áp dụng trong kỹ thuật DRL có tác động mạnh mẽ đến việc ước tính lợi nhuận kỳ vọng ở mỗi bước giao dịch và DRL kết hợp việc quan sát lợi nhuận hoặc rủi ro thực hiện trong giao dịch tiếp theo ở tại thời điểm hiện tại giao dịch (thăm dò) và các hành động giao dịch trong quá khứ (khai thác) để tìm ra chiến lược tối ưu.

Trong bài báo này, tôi sẽ trình bày mô hình DDPG trong DRL để mô hình hóa các bài toán giao dịch chứng khoán tự động dưới dạng bài toán quy trình quyết định Markov - (Markov Decision Process - MDP).

Quá trình đào tạo sẽ liên quan đến việc quan sát sự thay đổi giá cổ phiếu, thực hiện giao dịch vào thời điểm tiếp theo và tính toán lợi nhuận để điều chỉnh chiến lược đầu tư phù hợp nhằm tối ưu hóa lợi nhuận kỳ vọng.



Tiềm năng của kỹ thuật DRL trong các giao dịch tài chính là rất lớn, tuy nhiên việc triển khai và phát triển thuật toán phải trải qua công việc phức tạp của các bước tiêu chuẩn như quản lý trạng thái giao dịch trung gian, chuẩn hóa đánh giá dữ liệu, trích xuất dữ liệu, thông số, tỷ lệ tính năng,... để xảy ra sai sót trong quá trình thực hiện và mất nhiều thời gian xây dựng. Vì vậy, tôi đã quyết định sử dụng thư viện FinRL trong bài báo này để giải quyết tình hình trên. Thư viện FinRL chứa các thuật toán DRL tiêu chuẩn đã được tinh chỉnh để phù hợp với thị trường chứng khoán thay đổi liên tục, bao gồm cả thuật toán DDPG. Với FinRL, chúng ta có thể cấu hình từ bộ dữ liệu thị trường chứng khoán trên môi trường ảo như các sản giao dịch chứng khoán Dow Jones 30, DJIA, NASDAQ-100,... đào tạo các tác nhân giao dịch như mạng DNN, và đánh giá hoạt động giao dịch thông qua quá trình “backtesting”.

Bài báo này sẽ bao gồm:

- * phần I: Giới thiệu (hiện tại),
- * phần II: Tổng quan về các nghiên cứu liên quan,
- * phần III: Trình bày thư viện FinRL và cách thuật toán DDPG được sử dụng,
- * phần IV: Trình bày các thực nghiệm đánh giá,
- * phần V: Kết luận.

II/ TỔNG QUAN VỀ CÁC NGHIÊN CỨU LIÊN QUAN

Trong phần này, tôi sẽ giới thiệu tổng quan về một số thư viện lập trình DRL đóng gói sẵn (DRL framework) tương tự như thư viện FinRL và các thuật toán hiện đại của kỹ thuật DRL.

A. Thư viện lập trình DRL thông dụng:

Hiện tại, các nhà nghiên cứu đã phát triển các thư viện hỗ trợ cho việc học máy trong python, điều này đã làm cho python trở thành một công cụ mạnh mẽ trong việc giải quyết các vấn đề phức tạp mà con người không thể giải quyết một mình và chính xác, cũng như việc dự đoán và đào tạo các mô hình. Có rất nhiều thư viện học tăng cường sâu có các thuộc tính tương tự như thư viện FinRL và đây là một vài ví dụ:

• **OpenAI Gym** là một bộ công cụ phổ biến được xây dựng dựa trên các thuật toán học sâu tăng cường, nó cung cấp các môi trường nhiệm vụ được tiêu chuẩn hóa, còn được gọi là môi trường gym và thực hiện đào tạo đối tượng trên các môi trường này.

• **DeeR** cũng là một thư viện python để học sâu tăng cường, được xây dựng theo mô-đun để có thể dễ dàng đáp ứng các vấn đề khác nhau.

• **Tensorforce** là một thư viện lập trình mã nguồn mở để học sâu tăng cường, được sử dụng rộng rãi trong các lĩnh vực nghiên cứu. Nó có thiết kế dựa trên thành phần mô-đun, tách các thuật toán học sâu khỏi các ứng dụng và có đầy đủ các mô hình TensorFlow.

B. Những thuật toán thông dụng trong kỹ thuật DRL:

Trong kỹ thuật DRL, chúng ta thường có hai cách tiếp cận chính là:

- (i) học tập dựa trên giá trị hành động
- (ii) học tập dựa trên chiến lược

• Đối với **học dựa trên giá trị hành động**, chiến lược tối ưu sẽ phụ thuộc vào việc lựa chọn thực hiện hành động có giá trị lớn nhất ở mỗi trạng thái. Giá trị hành động ở đây được hiểu là điểm thưởng mong đợi nhận được khi từ một trạng thái hành động tạo ra trạng thái tiếp theo. Thuật toán tiêu biểu cho cách tiếp cận này là mạng Deep Q Network (DQN), bên cạnh đó còn có các thuật toán như A2C, PPO,... là những thuật toán hoạt động trên không gian hành động rời rạc.

• **Học dựa trên chiến lược** là cách agent ở mỗi trạng thái sẽ chọn hành động có xác suất cao nhất để thực hiện vì đặc điểm của quá trình học này là dựa trên không gian hành động rời rạc tức là không gian hành động rất lớn, chúng ta có thể sử dụng hàm mật độ xác suất để ước tính xác suất cho mỗi hành động. Một số thuật toán áp dụng cách tiếp cận này có thể kể đến như SAC, đạo hàm chiến lược đơn định (DDPG), đạo hàm chiến lược đơn định đa tác nhân.

Như đã giới thiệu ở trên, để giải quyết vấn đề giao dịch chứng khoán tự động, chúng ta phải đào tạo tác nhân giao dịch trên nhiều sàn giao dịch chứng khoán, và ở mỗi sàn giao dịch chứng khoán ta phải ra quyết định nên mua hay nên bán bao nhiêu cổ phiếu nên làm cho không gian hành động rất lớn. Do đó, các thuật toán tuân theo phương pháp học dựa trên chiến lược sẽ hợp lý hơn trong trường hợp này. Và chúng tôi chọn thuật toán **DDPG** vì tính phổ biến cũng như đơn giản giúp nhà đầu tư dễ dàng tập trung hơn vào chiến lược giao dịch chứng khoán.

III/ THƯ VIỆN FINRL

Thư viện FinRL sẽ được trình bày theo mô hình ba lớp:

- (i) lớp môi trường
- (ii) lớp tác nhân
- (iii) lớp ứng dụng

Trong đó, trọng tâm của tôi sẽ là lớp tác nhân, nơi tôi sẽ trình bày cách áp dụng thuật toán DDPG để đào tạo tác nhân thực hiện các giao dịch.

A. Môi trường đào tạo tác nhân:

Khi nói đến giao dịch cổ phiếu, giá trị cổ phiếu là điều quan trọng nhất đối với các nhà đầu tư vì nó có ảnh hưởng đến sự thành bại trong chiến lược đầu tư của họ. Tuy nhiên, trong thị trường chứng khoán, rất nhiều yếu tố ảnh hưởng đến giá trị của cổ phiếu như mối tương quan nghịch giữa thị trường chứng khoán và lạm phát, giá trị lãi suất tăng, các quy định của chính phủ về chính sách hàng hóa, ... Do đó, môi trường đào tạo của chúng ta sẽ mang tính ngẫu nhiên (còn được gọi là môi trường stochastic).

Để kỹ thuật DRL hoạt động tốt trong môi trường này, tôi đề xuất giải pháp mô hình hóa vấn đề về quy trình quyết định Markov, bao gồm các thành phần sau:

• **Không gian trạng thái S:** mô tả các quan sát mà tác nhân nhận được từ môi trường như giá trị cổ phiếu, tổng tài sản lưu động, lợi nhuận kỳ vọng, số lượng cổ phiếu được phép giao dịch trên

chứng khoán, ... Được mô phỏng giống như một nhà giao dịch cần nắm bắt thông tin trước khi đưa ra quyết định giao dịch.

• **Không gian hành động A:** mô tả các hành động đối với tác nhân giống như hành vi của nhà kinh doanh để giữ hoặc mua và bán bao nhiêu cổ phiếu. Chúng tôi sử dụng biến k để biểu diễn các trường hợp:

$k \in (0; +\infty)$: bán k cổ phiếu

$k = 0$: giữ nguyên số lượng cổ phiếu hiện có

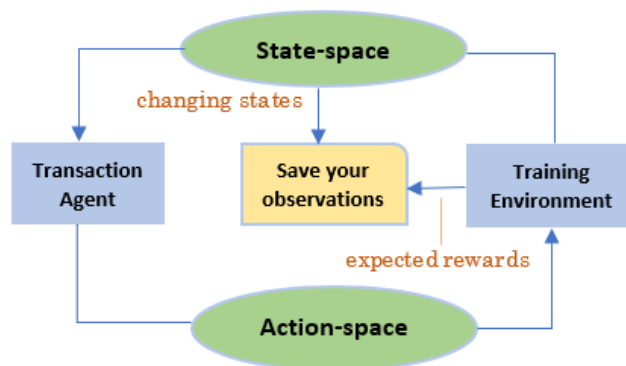
$k \in (-\infty, 0)$: mua k cổ phiếu

• **Điểm thưởng R (s, a, s'):** điểm thưởng nhận được khi thực hiện hành động a tại trạng thái s để mở một trạng thái mới s' .

• **Mô hình chuyển đổi T (s, a, s', r):** mô tả lại quá trình khi một tác nhân thực hiện hành động, nó sẽ mở một trạng thái mới và nhận thưởng.

• **Hàm giá trị hành động Q (s, a):** mô tả giá trị của hành động a ở trạng thái s . Giá trị của hành động ở đây được hiểu là điểm thưởng mong đợi nhận được ở trạng thái tiếp theo khi thực hiện hành động a .

• **Chiến lược π :** mô tả một chiến lược giao dịch ở trạng thái s . Thông thường, nó sẽ chứa một hàm phân phối xác suất biểu thị xác suất xảy ra của tất cả các hành động có thể xảy ra ở trạng thái s và quyết định hành động nào nên thực hiện.



Hình 1: Các hành động của tác nhân giao dịch chứng khoán

Nhiệm vụ của DRL là đào tạo tác nhân giao dịch thực hiện các hành động tương tác với môi trường để khám phá các trạng thái mới và điểm thưởng mong đợi ở trạng thái đó, kết hợp khai thác thông tin học được từ các trạng thái cũ để có được chiến lược tối ưu. Hoạt động của DRL được thể hiện như trong **Hình 1**.

Trong quá trình thử nghiệm, chúng tôi sẽ dựa vào thư viện FinRL để mô phỏng môi trường giao dịch cho tác nhân vì nó chứa dữ liệu thông tin tiêu chuẩn của thị trường chứng khoán thực như NASDAQ-100, DJIA, CSI 300, S&P 500.

B. Áp dụng thuật toán DDPG để đào tạo tác nhân giao dịch:

Như đã đề cập ở trên, tiếp theo chúng ta sẽ đến lớp tác nhân của thư viện FinRL, nơi chúng ta chọn thuật toán DDPG để huấn luyện tác nhân tương tác với môi trường và chúng ta sẽ gọi tác nhân này là "**DDPG agent**".

a) Yếu tố cần thiết: Đầu tiên, chúng ta cần xác định một số yếu tố cần thiết để xây dựng thuật toán DDPG:

- **Các thời điểm giao dịch:** tôi xác định s_0 là trạng thái bắt đầu, tương ứng với ngày nhà đầu tư bắt đầu tham gia giao dịch; s_t là trạng thái hiện tại, tương ứng với phiên giao dịch hiện tại của nhà đầu tư; s_{t+1} là trạng thái tiếp theo tương ứng với phiên tiếp theo; s_f là trạng thái đóng cửa tương ứng với ngày nhà đầu tư muốn kết thúc giao dịch.

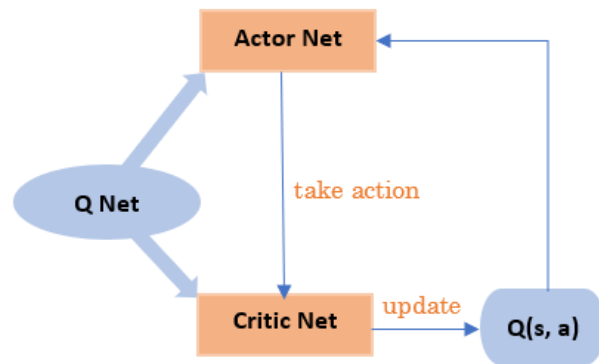
- **Mạng Q:** mục đích là cập nhật hàm giá trị hành động $Q(s, a)$. Trong mạng này, chúng tôi thiết kế thêm hai mạng con có tên là mạng actor và mạng critic.

- **Mạng actor:** chứa thông tin trạng thái và giá trị của các hành động có thể xảy ra tại trạng thái đó. Chúng tôi đặt θ làm tham số cho mạng này để tạo điều kiện cập nhật.

- **Mạng critic:** được sử dụng để cập nhật giá trị $Q(s, a)$ sau khi hành động tối ưu đã được thực hiện. Chúng ta đặt ϕ làm tham số cho mạng này để tạo điều kiện cập nhật.

- **Mạng mục tiêu:** là một bản sao của mạng Q, mục đích là khi mạng tác nhân trải qua đủ các hành động ở một trạng thái và chọn được hành động tối ưu, chúng ta sẽ cập nhật mạng này để lưu trữ thông tin ở trạng thái đó. Chúng tôi cũng sẽ đặt θ' và ϕ' làm các tham số của mạng này để thuận tiện cho việc cập nhật.

- **Bộ lưu trữ \mathcal{R} :** bộ phát lại trải nghiệm được sử dụng để lưu trữ các chuyển đổi của "DDPG agent" từ trạng thái này sang trạng thái khác.



Hình 2: Mô hình mạng Q

b) Các bước hoạt động: Giống như các thuật toán DRL khác, tác nhân DDPG cũng phải trải qua các bước:

- (i) lựa chọn thực hiện hành động ở mỗi trạng thái dựa trên những gì được biết
- (ii) quan sát điểm thưởng r và trạng thái tiếp theo
- (iii) dựa vào các quan sát để cải thiện hiểu biết và tìm chiến lược tối ưu.

Đây là một vấn đề tương đối khó, vì ban đầu agent không có thông tin gì về các trạng thái hay những thuận lợi hay rủi ro trước mắt mà phải tự tìm hiểu thông qua tương tác với môi trường.

Để huấn luyện DDPG agent giải quyết vấn đề khó khăn này, trước tiên chúng ta cần khởi tạo các yếu tố cần thiết đã nêu ở trên bao gồm: thông tin trạng thái bắt đầu (s_0) và thông tin trạng thái kết thúc (s_f), 2 mạng Actor và Critic: μ_θ và Q_ϕ với trọng số ngẫu nhiên, các mạng mục tiêu: $\mu_{\theta'}$ và $Q_{\phi'}$ và bộ lưu trữ \mathcal{R} rỗng.



Sau khi khởi tạo các phần tử cần thiết, chúng tôi tiến hành yêu cầu tác nhân DDPG lặp lại công việc sau:

- Tại mỗi trạng thái, chọn hành động ngẫu nhiên để khám phá các trạng thái mới. Mỗi lần tôi thực hiện một hành động mới, tôi lưu trải nghiệm của hành động đó bao gồm thông tin $(st, at, st+1, rt+1)$ vào bộ nhớ \mathcal{R} .

- Khi thời gian thực hiện hành động ngẫu nhiên đạt đến ngưỡng ϵ nhất định, chúng ta tiến hành tính giá trị hành động cho hàm mục tiêu theo công thức:

$$y_k = r_k + \gamma Q_{\varphi'}(s'_k, \mu_{\theta'}(s'_k)).$$

Trong đó:

γ là hệ số chiết khấu lợi nhuận $\gamma \in (0, 1)$

$Q_{\varphi'}(s'_k, \mu_{\theta'}(s'_k))$ là giá trị trung bình của mạng đích.

- Tiếp theo, chúng tôi cập nhật mạng Critic bằng cách sử dụng đạo hàm:

$$\mathcal{L}(\varphi) = \frac{1}{N} \sum_k (y_k - Q_{\varphi}(s_k, a_k))^2$$

trong đó N là số hành động được phát hiện bởi tác nhân ở trạng thái s trong \mathcal{R} .

Mục đích của việc cập nhật mạng Critic là để giảm sự khác biệt kỳ vọng giữa mạng Q và mạng mục tiêu.

- Cập nhật Actor để tìm chiến lược tối ưu bằng cách sử dụng đạo hàm:

$$\nabla_{\theta} J(\theta) = \frac{1}{N} \sum_k \nabla_{\theta} \mu_{\theta}(s_k) \times \nabla_a Q_{\varphi}(s_k, a)|_{a=\mu_{\theta}(s_k)}$$

Thông thường, DDPG agent sẽ tính điểm thưởng ở trạng thái st theo phương trình Bellman.

- Sau khi cập nhật mạng Critic và mạng Actor, chúng tôi cập nhật mạng mục tiêu dựa trên kiến thức hiện có của tác nhân được lưu trữ trong bộ nhớ. Mạng mục tiêu được cập nhật theo công thức dưới đây, với $\tau \ll 1$:

$$\theta' \leftarrow \tau \theta + (1 - \tau) \theta'$$

$$\varphi' \leftarrow \tau \varphi + (1 - \tau) \varphi'$$

Tác nhân DDPG sẽ lặp lại các thao tác trên cho đến khi gặp trạng thái kết thúc và sẽ trả lại tổng phần thưởng $\sum_{t=f-1}^{t=f-1} r(st, at, st+1)$ tối ưu cùng với các chiến lược giao dịch tại từng thời điểm.

Tuy nhiên, như đã nói ở trên, việc thực hiện thuật toán này rất khó và dễ xảy ra sai sót trong quá trình xây dựng, vì vậy chúng tôi sẽ sử dụng mô hình DDPG có sẵn trong thư viện FinRL để mô phỏng công việc đào tạo tác nhân.

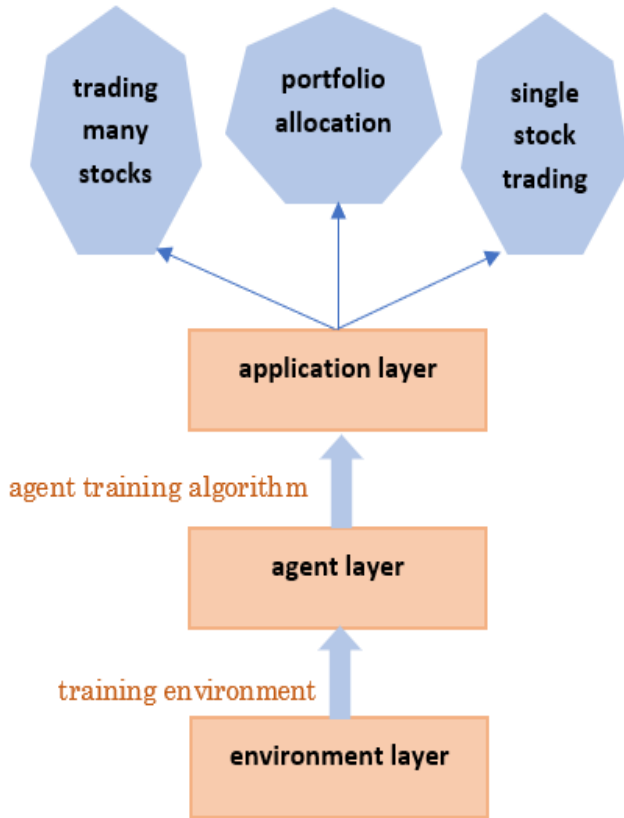
C. Ứng dụng trong thực tế:

Như chúng ta đã biết, đầu tư chứng khoán là một lĩnh vực rất phức tạp và tiềm ẩn nhiều rủi ro nhưng nếu gặp thời kỳ thuận lợi và có chiến lược đầu tư hợp lý thì nhà đầu tư cũng có thể thu được lợi nhuận lớn mà không cần nhiều lao động. Trước những thách thức và cơ hội như vậy, chủ đề trí tuệ nhân tạo áp dụng cho giao dịch chứng khoán tự động, nơi mà các nhà nghiên cứu xây dựng và đào tạo các tác nhân tham gia giao dịch với mục tiêu giảm thiểu rủi ro thua lỗ và tối ưu hóa lợi nhuận để đầu tư đang rất được quan tâm hiện nay.

Nhà đầu tư có thể dựa vào kết quả đào tạo các tác nhân giao dịch trên thị trường chứng khoán thực tế để nắm được thông tin về các bước giao dịch trong quá khứ, cũng như theo dõi diễn biến thị trường chứng khoán. Để từ đó hình thành ý tưởng đầu tư, đương đầu với những thăng trầm của thị trường.

Có ba lĩnh vực mà nhà đầu tư có thể tham khảo tác nhân giao dịch bao gồm:

- giao dịch cổ phiếu đơn lẻ
- giao dịch nhiều cổ phiếu
- phân bổ danh mục đầu tư



Hình 3: Mô hình sơ đồ 3 lớp của FinRL.

IV/ THỰC NGHIỆM

Các thử nghiệm được tiến hành dựa trên môi trường thị trường chứng khoán thực do thư viện FinRL cung cấp bao gồm NASDAQ-100; DJIA; S&P 500; SSE 50; CSI 300; HSI.

Môi trường ban đầu chứa các thông tin như: ngày giao dịch (date), tên các sản giao dịch chứng khoán (tic), giá mở cửa của phiên giao dịch (mở cửa), giá đóng cửa của phiên giao dịch (đóng cửa), giá cổ phiếu cao nhất (cao), giá cổ phiếu thấp nhất (thấp), số lượng cổ phiếu được phép giao dịch (khối lượng) và các tỷ lệ tài chính liên quan.

Có rất nhiều cột dữ liệu, ở đây tôi chỉ hiển thị **5 trường dữ liệu đầu tiên**.

	date	tic	open	high	low	close	volume
0	2009-01-02	AAPL	3.067143	3.251429	3.041429	2.778781	746015200.0
1	2009-01-02	AMGN	58.590000	59.080002	57.750000	45.615871	6547900.0
2	2009-01-02	AXP	18.570000	19.520000	18.400000	15.618538	10955700.0
3	2009-01-02	BA	42.799999	45.560001	42.779999	33.941105	7010200.0
4	2009-01-02	CAT	44.910000	46.980000	44.709999	32.475796	7117200.0

Hình 4: Các thông tin ban đầu.

A. Các tỉ lệ đánh giá:

Các tỷ lệ đánh giá cơ bản được cung cấp để giúp chúng tôi đánh giá hiệu suất của chiến lược giao dịch của mình bao gồm: lợi nhuận hàng năm, lợi nhuận tích lũy, biến động hàng năm, tỷ lệ Sharpe (tỷ lệ Sharpe), tỷ lệ sụt giảm (Max drawdown), ...

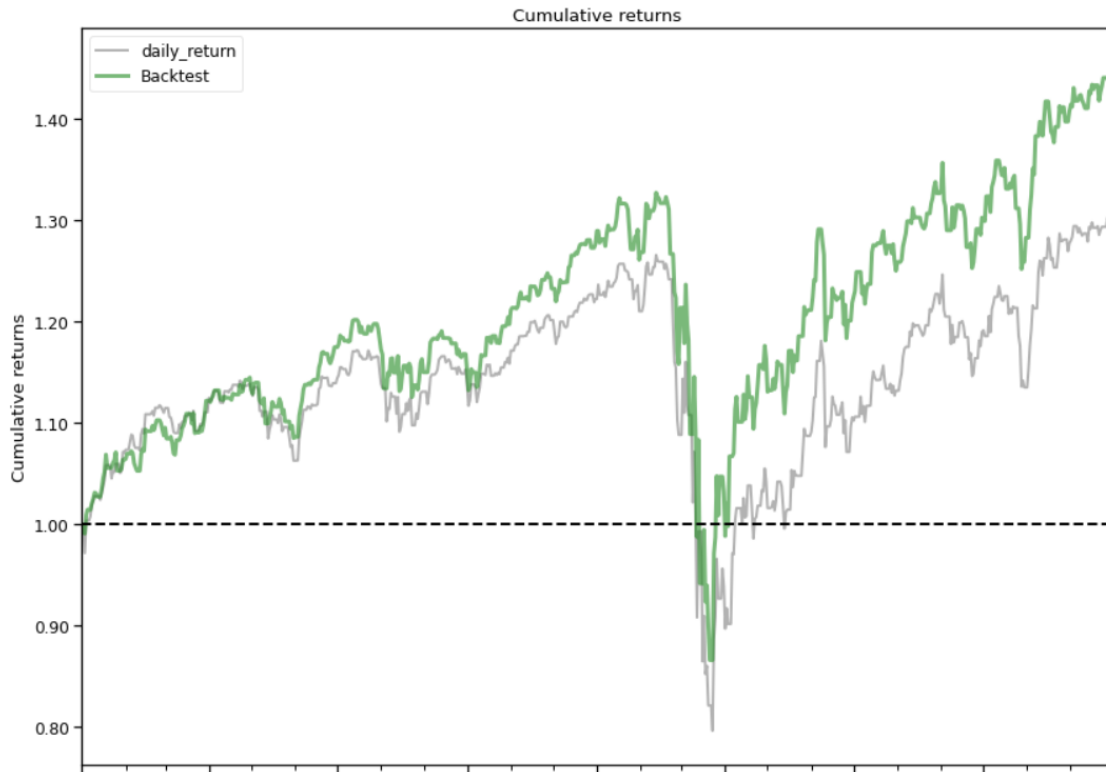
Hình 5 cho thấy các tỷ lệ để các nhà đầu tư đánh giá hiệu suất giao dịch tổng thể:

	Backtest
Annual return	13.839%
Cumulative returns	45.642%
Annual volatility	22.892%
Sharpe ratio	0.68
Calmar ratio	0.40
Stability	0.52
Max drawdown	-34.752%
Omega ratio	1.20
Sortino ratio	0.95
Skew	NaN
Kurtosis	NaN
Tail ratio	0.91
Daily value at risk	-2.822%
Alpha	0.08
Beta	0.68

Hình 5: Các số liệu đánh giá chiến lược

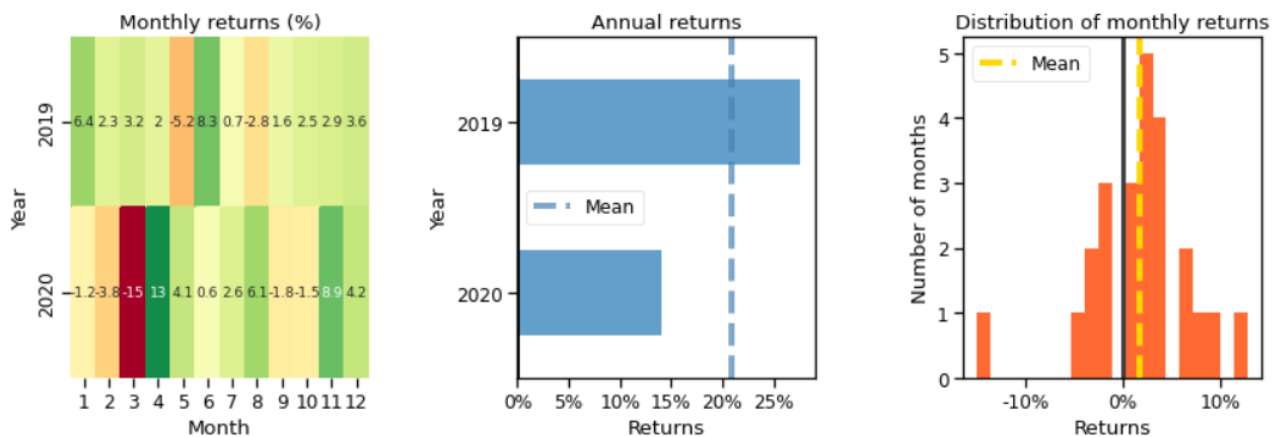
B. Kết quả thực nghiệm:

Sau khi tiến hành thực nghiệm với dữ liệu cụ thể dưới sự hỗ trợ của thư viện FinRL, tôi thu được những kết quả như sau:

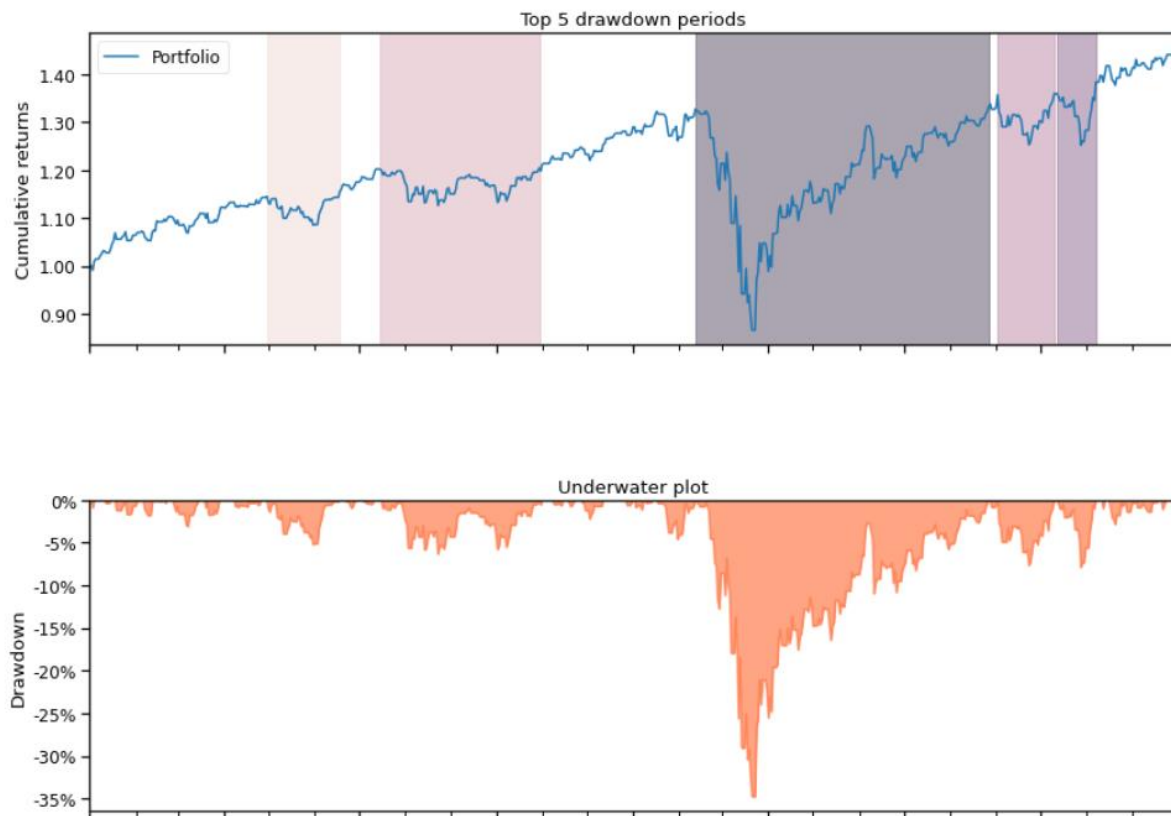


Hình 6: Lợi nhuận hằng năm

Để mô phỏng tốt hơn giao dịch thực tế, chúng tôi sử dụng công cụ để đánh giá hoạt động giao dịch. Trong phần mô tả thực nghiệm này, chúng tôi sẽ chọn thị trường chứng khoán **Dow Jones Index** để làm ví dụ minh họa, với thời gian giao dịch là từ ngày 1/1/2019 đến 31/12/2020.



Hình 7: Biểu đồ lợi nhuận hằng năm và hằng tháng



Hình 8: Mức sụt giảm của các tài khoản qua các giai đoạn



Hình 9: Tỷ lệ Sharpe



V/ KẾT LUẬN

Trong bài báo này, chúng tôi đã trình bày về mô hình ba lớp của thư viện FinRL và tận dụng những tài nguyên sẵn có trong thư viện để tiến hành thực nghiệm. Bên cạnh đó, tôi còn trình bày về thuật toán DDPG trong mô hình DDPG của thư viện, cách huấn luyện một DDPG agent thực hiện giao dịch chứng khoán tự động. Chúng tôi cũng đã mô phỏng việc huấn luyện tác nhân thực hiện giao dịch đa cổ phiếu trên thị trường chứng khoán **Dow Jones Index**. Kết quả chiến lược mô phỏng thực nghiệm có thể tốt trên thị trường này nhưng cũng có thể lại kém hiệu quả trên thị trường khác.

Trong tương lai, tôi sẽ cố gắng cập nhật lại ngày kết thúc giao dịch cho mô hình để cho phù hợp với xu thế của thị trường, đồng thời nghiên cứu thêm một số thuật toán của kỹ thuật DRL để cải thiện mô hình trở nên hiệu quả hơn.

TÀI LIỆU THAM KHẢO

<https://github.com/AI4Finance-Foundation/FinRL>

Deep Reinforcement Learning for Trading:

<https://arxiv.org/pdf/1911.10107.pdf>

Deep Deterministic Policy Gradient (DDPG) :

<https://spinningup.openai.com/en/latest/algorithms/ddpg.html>

Backtest trading Policy :

<https://github.com/quantopian/pyfolio>

=====
--- Thank you for watching! ---