Deep Direct Reinforcement Learning for Financial Signal Representation and Trading

Đặng Quốc Toàn - 20051051, Nguyễn Trường Chinh - 20045391, Đỗ Thu Đông- 20043131

Khoa Công nghệ Thông tin, Đại học Công nghiệp Thành phố Hồ Chí Minh

TÓM TẮT

- *Deep Direct Reinforcement Learning for Financial Signal Representation and Trading* (Học tăng cường trực tiếp sâu cho biểu diễn tín hiệu tài chính và giao dịch) là một phương pháp sử dụng học tăng cường (reinforcement learning) và mạng nơ-ron sâu (deep neural networks) để biểu diễn tín hiệu tài chính và thực hiện giao dịch tài chính một cách tự động.
- Trong phạm vi bài báo này, nhóm cố gắng giải quyết thách thức về việc huấn luyện máy tính để đánh bại các nhà giao dịch kinh nghiệm trong giao dịch tài sản tài chính. Nhóm giới thiệu một mạng nơ-ron sâu đa trạng thái (recurrent deep neural network) để biểu diễn tín hiệu tài chính và thực hiện giao dịch trong thời gian thực. Mô hình của chúng tôi được lấy cảm hứng từ hai khái niệm học liên quan đến sinh học trong deep learning (DL) và reinforcement learning (RL).
- Từ khóa liên quan: Deep Learning (DL), xử lý tín hiệu tài chính, mạng nơ ron (NN) trong tài chính, Reinforcement Learning (RL), Recurrent deep neural network (RDNN), (BPTT) Backpropagation through time.

1.TỔNG QUAN

- Giao dịch tài chính được coi là một vấn đề ra quyết định trực tuyến, gồm hai bước quan trọng: tóm tắt điều kiện thị trường và thực hiện hành động tối ưu. Tuy nhiên, so với các nhiệm vụ học truyền thống, quyết định động trong giao dịch tài chính đặt ra thách thức hơn do thiếu thông tin giám sát từ các chuyên gia. Điều này đòi hỏi mô hình tự khám phá môi trường chưa biết và đồng thời đưa ra quyết định đúng đắn theo cách trực tuyến. Do đó, các nghiên cứu về học tăng cường đã khuyến khích sự phát triển lâu dài trong lĩnh vực này.
- Trong báo cáo này, nhóm tập trung vào việc sử dụng phương pháp học tăng cường (RL), một khung vi mô lấy cảm hứng từ lĩnh vực sinh học, để giải quyết câu hỏi liệu chúng ta có thể đào tạo một mô hình RL để vượt qua các nhà giao dịch con người có kinh nghiệm trên thị trường tài chính? RL đã được áp dụng thành công trong nhiều nhiệm vụ khác nhau như điều hướng robot, chơi trò chơi Atari và điều khiển trực thăng. Tuy nhiên, áp dụng RL trong giao dịch thuật toán đối mặt với hai thách thức chính:

- Thách thức đầu tiên là khó khăn trong việc tóm tắt và biểu diễn môi trường tài chính. Dữ liệu tài chính thường chứa nhiều nhiễu, bước nhảy và chuyển động không ổn định, gây ra chuỗi thời gian phức tạp. Việc tạo ra một biểu diễn tóm tắt chính xác và khả quan của điều kiện thị trường từ dữ liệu này là một thách thức. Mặc dù các phương pháp truyền thống đã sử dụng các đặc điểm tài chính thủ công như đường trung bình động hoặc các chỉ báo kỹ thuật ngẫu nhiên để tóm tắt điều kiện thị trường, khả năng khái quát hóa của chúng thường kém. Thay vào đó, chúng ta có thể tìm hiểu các biểu diễn đặc trưng mạnh mẽ hơn trực tiếp từ dữ liệu mà không cần phụ thuộc vào các đặc điểm được xác định trước.
- Thách thức thứ hai là do tính động của việc thực hiện hành động giao dịch. Đặt lệnh giao dịch là một quá trình có hệ thống yêu cầu xem xét các yếu tố thực tế. Thay đổi thường xuyên vị trí giao dịch (mua hoặc bán) không đóng góp gì cho lợi nhuận, mà chỉ tăng chi phí giao dịch và trượt giá. Vì vậy, ngoài việc xem xét điều kiện thị trường hiện tại, cần phải mô hình hóa cả các hành động lịch sử và các vị trí tương ứng. Điều này tạo ra một thách thức khác trong việc thiết kế chính sách giao dịch. Làm thế nào chúng ta có thể kết hợp hiện tượng bộ nhớ như vậy vào hệ thống giao dịch?

- Để giải quyết các thách thức trên, trong bài báo này, nhóm giới thiệu một cấu trúc mạng thần kinh học tăng cường sâu (RDNN) mới để đồng thời cảm nhận môi trường và đưa ra quyết định định kỳ cho giao dịch tài chính trực tuyến kết hợp mạng nơ-ron hồi quy (RNN) với (RL) để tăng cường khả năng tóm tắt điều kiện thị trường. Để làm cho mô hình mạnh mẽ hơn, chúng tôi giới thiệu các khái niệm học mờ để giảm độ không chắc của dữ liệu đầu vào. Mô hình học tập toàn diện dẫn đến một mạng nơ-ron phức tạp bao gồm cả cấu trúc sâu và cấu trúc lặp lại.
- Để xử lý cấu trúc lặp lại, nhóm sử dụng phương pháp lan truyền ngược qua thời gian biểu thị mạng RNN như một chuỗi đơn lẻ mà không có phản hồi. Khi áp dụng RL trở lại tất cả các lớp, nhóm giải quyết vấn đề biến mất độ dốc chắc chắn liên quan đến giai đoạn đào tạo. Điều này là do mạng nơ-ron mở rộng cả các cấu trúc sâu sắc trong quá trình học các đặc trưng và các phần mở rộng về thời gian. Do đó, nhóm giới thiệu một phương pháp đào tạo gọi là BPTT để khắc phục vấn đề này. Theo phương pháp tiếp cận của nhóm kết nối một số liên kết ảo từ hàm mục tiêu trực tiếp với các lớp sâu trong quá trình lan truyền ngược (BP). Chiến lược này cho phép các lớp sâu có cơ hội xem xét những gì đang diễn ra trong mục tiêu cuối cùng và cải thiện hiệu suất học tập.
- Hệ thống giao dịch sử dụng RDNN được thử nghiệm trên thị trường tài chính thực để giao dịch hợp đồng tương lai. Nhóm tích lũy dữ liệu giá lịch sử của cả tương lai chỉ số chứng khoán và tương lai hàng hóa. Dữ liệu thị trường thực này được sử dụng trực tiếp để xác minh hiệu suất của hệ thống. Nhóm so sánh hệ thống RL sâu với các hệ thống giao dịch khác trong các điều kiện thử nghiệm khác nhau. Kết quả so sánh cho thấy hệ thống RDNN và mở rộng mờ của nó rất mạnh mẽ đối với các điều kiện thị trường khác nhau và có thể tạo ra lợi nhuận đáng tin cậy trong nhiều tương lai khác nhau.

2. NỘI DUNG LIÊN QUAN

- Trong lĩnh vực học tăng cường (RL), có hai loại phương pháp chính là dựa trên nhà phê bình (các thuật toán học hàm giá trị) và dựa trên diễn viên (các thuật toán học hành động). Các phương pháp dựa trên nhà phê bình ước lượng trực tiếp các hàm giá trị và được sử dụng rộng rãi trong lĩnh vực này. Tuy nhiên, chúng không phù hợp cho vấn đề giao dịch tài chính phức tạp. Thay vào đó, các phương pháp dựa trên diễn viên tập trung vào việc học các hành động trực tiếp từ dữ liệu liên tục của thị trường.
- Trong lĩnh vực giao dịch tài chính, một công trình tiên phong đã đề xuất việc học các hành động trực tiếp từ dữ liệu thị trường, gọi là Deep Reinforcement Learning (DRL). Phương pháp này có hai lợi thế quan trọng: mục tiêu linh hoạt cho việc tối ưu hóa và mô tả liên tục về điều kiện thị trường.
- Để đạt được hiệu suất tốt trong học tăng cường giao dịch, việc biểu diễn đặc trưng dữ liệu là rất quan trọng. Trong lĩnh vực học dữ liệu chứng khoán, đã có nhiều phương pháp biểu diễn đặc trưng từ nhiều quan điểm khác nhau. Một số nghiên cứu đã sử dụng mã hóa thưa (sparse coding) để trích xuất đặc trưng tin cậy hơn so với DRL.
- Tuy nhiên, mã hóa thưa không thể so sánh được với kỹ thuật học sâu (DL) hiện đại, mà có thể tự động học đặc trưng từ dữ liệu lớn. DL đã thành công trong nhiều lĩnh vực như nhận dạng hình ảnh và nhận dạng giọng nói. Tuy nhiên, hiện chưa có nhiều nghiên cứu về DL trong việc phân tích tín hiệu tài chính.
- Bài báo này nhằm tổng hợp sức mạnh của DL vào lĩnh vực xử lý và học tín hiệu tài chính. Mô hình DL sẽ được kết hợp với DRL để thiết kế một hệ thống giao dịch thời gian thực cho giao dịch tài sản tài chính.

3.HỌC CỦNG CỐ SÂU TRỰC TIẾP (*DIRECT DEEP REINFORCEMENT LEARNING)*

3.1 GIAO DỊCH TRỰC TIẾP SỬ DỤNG HỌC TĂNG CƯỜNG

- DRL điển hình về cơ bản là một mạng nơ-ron tái lập một tầng. Chúng tôi định nghĩa p1, p2, ..., pt, ... là các chuỗi giá cả được công bố từ trung tâm giao dịch. Sau đó, lợi nhuận tại thời điểm t dễ dàng xác định bằng zt = pt - pt-1. Dựa trên tình trạng thị trường hiện tại, quyết định giao dịch thời gian thực (chính sách) δt ∈ {mua dài, trung lập, bán ngắn} = {1, 0, -1} được đưa ra tại mỗi thời điểm t. Với các ký hiệu đã được định nghĩa trước đó, lợi nhuận Rt do mô hình giao dịch tạo ra được tính bằng cách
    - **$Rt = \delta t-1zt - c|\delta t - \delta t-1|$**
- Đầu tiên là lợi nhuận/lỗ từ những biến động trên thị trường và thuật ngữ thứ hai là TC (Transaction Cost) khi đổi vị trí giao dịch tại thời điểm t. TC này (c) là khoản phí bắt buộc phải trả cho công ty môi giới chỉ khi δt ≠ δt-1. Khi hai quyết định giao dịch liên tiếp là giống nhau, tức là δt = δt-1, không có TC được áp dụng.
    - **max UT {R1 ... RT}**
    - Ở đây, UT {·} là tổng lượng thưởng tích lũy trong khoảng thời gian từ 1 đến T. Một cách hiển nhiên, thưởng đơn giản nhất là TP (Total Profit) thu được trong khoảng thời gian T, tức là **$UT = \Sigma t=1^T Rt$.** Các hàm thưởng phức tạp khác, chẳng hạn như tỷ suất lợi nhuận được điều chỉnh theo rủi ro, cũng có thể được sử

dụng ở đây như mục tiêu RL. Để dễ dàng giải thích mô hình, chúng tôi ưu tiên sử dụng TP làm hàm mục tiêu trong các phần tiếp theo

- Với hàm thưởng được xác định rõ ràng như vậy, vấn đề chính là làm thế nào để giải quyết nó một cách hiệu quả. Trong các công trình RL truyền thống, các hàm giá trị được xác định trong không gian rời rạc được lặp lại trực tiếp bằng lập trình động. Tuy nhiên việc học trực tiếp hàm giá trị không khả thi đối với vấn đề giao dịch động, vì các điều kiện thị trường phức tạp khó có thể được giải thích trong một số trạng thái rời rạc. Khung công việc này được gọi là DRL (Deep Reinforcement Learning). Cụ thể, DRL sử dụng một hàm phi tuyến để xấp xỉ hành động giao dịch (chính sách) tại mỗi điểm thời gian bằng cách
    - **δxt = tanh [(w, ft) + b + uδt−1]**

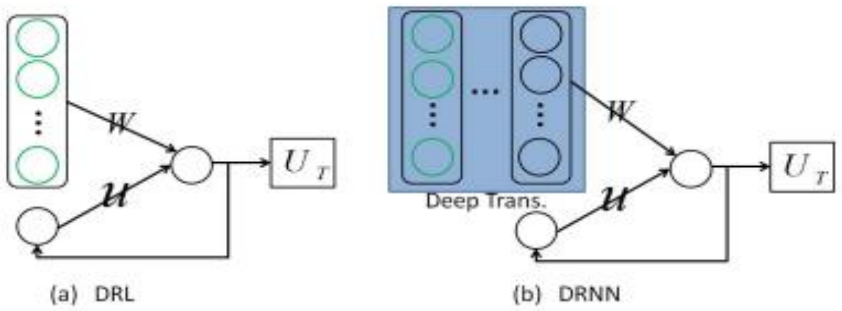


Fig. 1. Comparisons of DRL and the proposed DRNN for joint feature learning and DRT.

- Trong DRL, m giá trị trở lại gần đây được áp dụng trực tiếp như là vector đặc trưng:
    - **ft = [zt−m+1, ..., zt] ∈ Rm.**
- Ngoài các đặc trưng đó, thuật ngữ uδt−1 cũng được thêm vào phương trình để xem xét quyết định giao dịch gần nhất. Thuật ngữ này được sử dụng để ngăn cản tác nhân thay đổi vị trí giao dịch quá thường xuyên và do đó tránh các khoản phí giao dịch lớn. Với phép biến đổi tuyến tính trong dấu ngoặc, hàm tanh (·) dịch chuyển hàm số vào khoảng (-1, 1) để xấp xỉ quyết định giao dịch cuối cùng. Tối ưu hóa của DRL nhằm mục đích học một tập hợp tham số = {w, u, b} có thể tối đa hóa hàm phần thưởng toàn cục trong.

3.2 Mạng Neural Tái Phát Sâu cho DDR *(Deep Recurrent Neural Network for DDR)*

- Trong khi chúng tôi đã giới thiệu DRL dưới dạng một bài toán hồi quy, thì thú vị là nó thực tế là một mạng neural một tầng, như được hiển thị trong Hình 1(a). Thuật ngữ bias không được vẽ rõ ràng trong sơ đồ để đơn giản hóa. Trong việc triển khai thực tế, thuật ngữ bias có thể được hợp nhất vào trọng số w bằng cách mở rộng một chiều của 1 ở cuối vector đặc trưng. Vector đặc trưng ft (nút màu xanh lá cây) là đầu vào trực tiếp của hệ thống. Mạng neural DRL có cấu trúc tái phát, có một liên kết từ đầu ra (δt) đến tầng đầu vào. Một thuộc tính hứa hẹn của RNN là tích hợp bộ nhớ lâu dài vào hệ thống học. DRL lưu giữ các hành động giao dịch quá khứ trong bộ nhớ để ngăn cản thay đổi vị trí giao dịch thường xuyên. Hệ thống trong Hình 1(a) sử dụng một RNN

để tạo ra một cách đệ quy các quyết định giao dịch (học chính sách trực tiếp) bằng cách khám phá một môi trường không rõ. Tuy nhiên, một điểm yếu rõ ràng của DRL là thiếu một phần học đặc trưng để tóm tắt một cách mạnh mẽ các điều kiện thị trường ồn ào.

- Để thực hiện việc học đặc trưng, trong bài báo này, chúng tôi giới thiệu DL phổ biến vào DRL để đồng thời học đặc trưng và giao dịch động. DL là một khung thức học đặc trưng rất mạnh mẽ, tiềm năng của nó đã được chứng minh rộng rãi trong một số bài toán học máy. Cụ thể, DL xây dựng một mạng neural sâu để chuyển đổi thông tin từ tầng này sang tầng khác theo cấp bậc. Việc biểu diễn sâu khuyến khích các biểu diễn đặc trưng thông tin hữu ích hơn cho một nhiệm vụ học cụ thể. Sự biến đổi sâu cũng đã được tìm thấy trong xã hội sinh học khi nghiên cứu các cơ chế khám phá tri thức trong não.
- Những phát hiện này tiếp tục củng cố lý thuyết sinh học để ủng hộ những thành công rộng rãi của DL.
- Bằng cách mở rộng DL vào DRL, phần học đặc trưng (bảng màu xanh) được thêm vào RNN trong Hình 1(a), tạo thành một mạng neural tái phát sâu (DRNN) trong Hình 1(b). Chúng tôi định nghĩa biểu diễn sâu là Ft = gd (ft), được thu được bằng cách biến đổi phân cấp vector đầu vào ft thông qua DNN với một ánh xạ phi tuyến gd (·). Sau đó, hành động giao dịch trong (3) được xác định bởi phương trình sau:

$$\delta t = \tanh [(w, gd (ft)) + b + u\delta t-1].$$

- Trong việc triển khai của chúng tôi, phần biến đổi sâu được cấu hình với nhiều tầng ẩn kết nối tốt, tức là mỗi nút trên tầng (l + 1) được kết nối với tất cả các nút trên tầng l. Để dễ giải thích, chúng tôi định nghĩa:

$$a_i^l = \langle \mathbf{w}_i^l, \mathbf{o}^{(l-1)} \rangle + b_i^l, \quad o_i^l = \frac{1}{1 + e^{-a_i^l}}$$

3.3. Mở rộng Mờ để Giảm Bớt Sự Bất Ngờ

- Cấu hình sâu đã giải quyết tốt nhiệm vụ học đặc trưng trong RNN. Tuy nhiên, một vấn đề quan trọng khác cần được xem xét cẩn thận, đó là sự không chắc chắn trong dữ liệu tài chính. Khác với các loại tín hiệu khác như hình ảnh hoặc âm thanh, các chuỗi tài chính chứa một lượng không chắc chắn không thể đoán trước do sự đánh bạc ngẫu nhiên trong giao dịch. Ngoài ra, một số yếu tố khác như tình hình kinh tế toàn cầu và một số tin đồn về công ty cũng có thể ảnh hưởng đến hướng của tín hiệu tài chính trong thời gian thực. Do đó, giảm bớt sự không chắc chắn trong dữ liệu gốc là một phương pháp quan trọng để tăng tính ổn định cho việc khai thác tín hiệu tài chính.
- Trong cộng đồng trí tuệ nhân tạo, học mờ là một mô hình lý tưởng để giảm bớt sự không chắc chắn trong dữ liệu ban đầu .Thay vì sử dụng mô tả chính xác về một số hiện tượng, hệ thống mờ thích gán các giá trị ngôn ngữ mờ cho dữ liệu đầu vào. Các biểu diễn mờ như vậy có thể dễ dàng thu được bằng cách so sánh dữ liệu thế giới thực với một số tập

mờ khác nhau, sau đó suy ra các mức độ thành viên mờ tương ứng. Do đó, hệ thống học chỉ làm việc với các biểu diễn mờ này để đưa ra quyết định kiểm soát mạnh mẽ.

- Đối với vấn đề tài chính được thảo luận ở đây, các tập mờ có thể tự nhiên được định nghĩa dựa trên các chuyển động cơ bản của giá cổ phiếu. Cụ thể, các tập mờ được xác định trên các nhóm tăng, giảm và không có xu hướng. Các tham số trong hàm thành viên mờ sau đó có thể được xác định trước theo ngữ cảnh của vấn đề được thảo luận. Hoặc chúng có thể được học theo một cách hoàn toàn dựa trên dữ liệu. Vấn đề tài chính là vô cùng phức tạp và khó có thể thiết lập hàm thành viên mờ một cách thủ công.

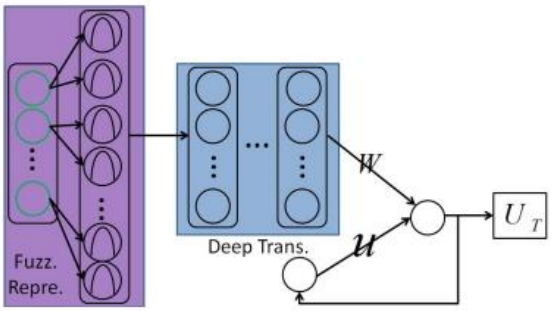


Fig. 2. Overview of fuzzy DRNNs for robust feature learning and self-taught trading.

- Trong mạng neural mờ, phần đại diện mờ thường được kết nối với vector đầu vào ft (các nút màu xanh lá cây) bằng các hàm thành viên mờ khác nhau. Trong hệ thống của chúng tôi, chúng tôi tuân theo một công trình tiên phong và gán k bậc mờ khác nhau cho mỗi chiều của vector đầu vào. Trong biểu đồ của Hình 2, chỉ có hai nút mờ (k = 2) được kết nối với mỗi biến đầu vào do hạn chế không gian. Tuy nhiên, trong việc triển khai thực tế của chúng tôi, giá trị k được cố định là 3 để mô tả các điều kiện tăng, giảm và không có xu hướng. Toán học, hàm thành viên mờ thứ i vi (·): R → [0, 1] ánh xạ giá trị đầu vào thứ i thành một bậc mờ trong khoảng [0, 1].

$$o_i^{(l)} = v_i\left(a_i^{(l)}\right) = e^{-\left(a_i^{(l)} - m_i\right)^2 / \sigma_i^2} \quad \forall i.$$

- Hàm thành viên mờ Gaussian với trung bình m và phương sai σ^2 được sử dụng trong hệ thống của chúng tôi theo các đề xuất của. Sau khi có các biểu diễn mờ, chúng được kết nối trực tiếp với lớp biến đổi sâu để tìm kiếm các biến đổi sâu.
- Tóm lại, FDRNN (mạng neural mờ sâu đệm) bao gồm ba phần chính là biểu diễn mờ, biến đổi sâu và DRT. Khi xem FDRNN là một hệ thống thống nhất, ba phần này lần lượt đóng vai trò trong việc tiền xử lý dữ liệu (giảm không chắc chắn), học đặc trưng (biến đổi sâu) và tạo ra chính sách giao dịch (RL). Toàn bộ khung tối ưu hóa được xác định như sau:

$$\max_{\{\Theta, g_d(\cdot), v(\cdot)\}} U_T(R_1..R_T)$$
$$\text{s.t. } R_t = \delta_{t-1} z_t - c|\delta_t - \delta_{t-1}|$$
$$\delta_t = \tanh(\langle \mathbf{w}, \mathbf{F}_t \rangle + b + u\delta_{t-1})$$
$$\mathbf{F}_t = g_d(v(\mathbf{f}_t))$$

4. Học DRNN

4.1 Khởi tạo Hệ thống

- Khởi tạo tham số là một bước quan trọng để huấn luyện DNN. Chúng tôi sẽ giới thiệu các chiến lược khởi tạo cho ba phần học. Phần biểu diễn mờ (hình 2, khung màu tím) dễ dàng khởi tạo. Các tham số duy nhất cần được chỉ định là trung tâm mờ (mi) và độ rộng (σ2_i) của các nút mờ, trong đó i là chỉ số của nút thứ i trên lớp biểu diễn mờ. Chúng tôi áp dụng phương pháp k-means để chia các mẫu huấn luyện thành k lớp. Tham số k được cố định là 3, vì mỗi nút đầu vào được kết nối với ba hàm thành viên. Sau đó, trong mỗi cụm, giá trị trung bình và phương sai của mỗi chiều trên vectơ đầu vào (ft) được tính toán tuần tự để khởi tạo mi và σ2_i tương ứng.

- AE được sử dụng để khởi tạo phần biến đổi sâu [Hình 2 (khung màu xanh lam)]. AE nhằm mục tiêu tái tạo tối ưu thông tin đầu vào trên một lớp ảo đặt sau các biểu diễn ẩn. Để dễ hiểu, ba lớp được xác định ở đây, tức lớp đầu vào thứ (l), lớp ẩn thứ (l+1) và lớp tái tạo thứ (l+2). Ba lớp này đều được kết nối tốt. Chúng tôi xác định hθ(·) [tương ứng, hγ(·)] là phép biến đổi feedforward từ lớp thứ l đến lớp thứ (l+1) [tương ứng, từ lớp (l+1) đến lớp (l+2)] với tập tham số θ [tương ứng, γ]. Tối ưu hóa AE giảm thiểu tổn thất sau đây:

$$\sum_t \|\mathbf{x}_t^{(l)} - h_\gamma\left(h_\theta(\mathbf{x}_t^{(l)})\right)\|_2^2 + \eta\|\mathbf{w}^{(l+1)}\|_2^2.$$

Lưu ý rằng x(l)t là trạng thái của các nút của lớp thứ l với mẫu huấn luyện thứ t là đầu vào. Trong (9), một thuật ngữ bậc hai được thêm vào để tránh hiện tượng overfitting. Sau khi giải quyết tối ưu hóa AE, tập tham số θ = {w(l+1), b(l+1)} được ghi lại trong mạng như là tham số khởi tạo của lớp (l+1). Lớp tái tạo và các tham số tương ứng γ không được sử dụng. Điều này là vì lớp tái tạo chỉ là một lớp ảo, hỗ trợ việc học tham số của lớp ẩn [28], [39]. Quá trình tối ưu hóa AE được thực hiện trên từng lớp ẩn tuần tự cho đến khi tất cả các tham số trong phần biến đổi sâu đã được thiết lập.

- Trong phần DRL, các tham số có thể được khởi tạo bằng cách sử dụng biểu diễn sâu cuối cùng Ft làm đầu vào cho mô hình DRL. Quá trình này tương đương với việc giải quyết RNN nông ở Hình 1(a), đã được thảo luận trong [17]. Lưu ý rằng tất cả các chiến lược học được trình bày trong phần này liên quan đến việc khởi tạo tham số. Để làm cho toàn bộ hệ thống DL hoạt động mạnh mẽ trong việc giải quyết các nhiệm vụ khó khăn, một bước điều chỉnh tinh chỉnh cần được thực hiện để điều chỉnh chính xác các tham số của mỗi lớp. Bước điều chỉnh tinh chỉnh này có thể được coi là việc học đặc trưng phụ thuộc vào nhiệm vụ.

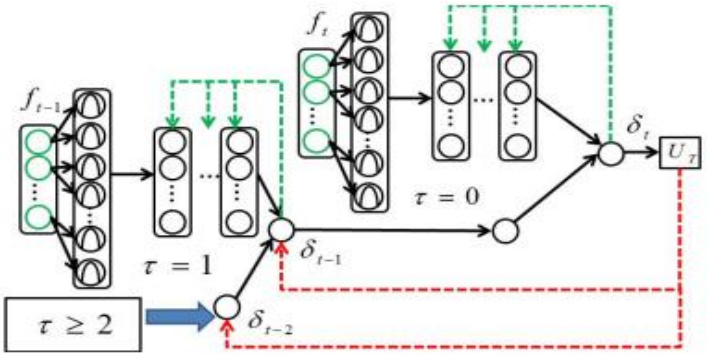


Fig. 3. Task-aware BPTT for RDNN fine tuning.

4.2 Task-Aware BPTT

- Trong phương pháp truyền ngược ngắn hạn biết về tác vụ (Task-Aware BPTT), chúng ta áp dụng phương pháp lỗi BP vào bước tinh chỉnh tinh chỉnh DNN. Tuy nhiên, FRDNN phức tạp hơn một chút và có cấu trúc đệ quy và sâu. Chúng tôi ký hiệu θ là tham số chung trong FRDNN, và đạo hàm của nó được tính bằng quy tắc chuỗi.

$$\frac{\partial U_T}{\partial \theta} = \sum_t \frac{dU_t}{dR_t}\left\{\frac{dR_t}{d\delta_t}\frac{d\delta_t}{d\theta} + \frac{dR_t}{d\delta_{t-1}}\frac{d\delta_{t-1}}{d\theta}\right\}$$
$$\frac{d\delta_t}{d\theta} = \frac{\partial \delta_t}{\partial \theta} + \frac{\partial \delta_t}{\partial \delta_{t-1}}\frac{d\delta_{t-1}}{d\theta}.$$

  - rõ ràng khi tính đạo hàm dδt/dθ, ta cần tính đạo hàm đệ quy cho dδt−τ /dθ, ∀τ = 1, ..., T. Việc tính toán đạo hàm đệ quy như vậy gây khó khăn đáng kể. Để đơn giản hóa vấn đề, chúng tôi giới thiệu phương pháp BPTT [40] nổi tiếng để xử lý cấu trúc đệ quy của NN.
  - Bằng cách phân tích cấu trúc FRDNN trong Hình 2, liên kết đệ quy xuất phát từ phía đầu ra đến phía đầu vào, tức là δt−1 được sử dụng làm đầu vào của các neuron để tính toán δt. Hình 3 cho thấy hai bước đầu tiên của việc mở rộng FRDNN. Chúng tôi gọi mỗi khối với các giá trị khác nhau của τ là một ngăn xếp thời gian, và Hình 3 chỉ ra hai ngăn xếp thời gian (với τ = 0 và τ = 1). Sau khi được mở rộng bằng BPTT, hệ thống hiện tại không có cấu trúc đệ quy nào và phương pháp BP thông thường được áp dụng dễ dàng. Khi lấy đạo hàm của các tham số tại mỗi ngăn xếp thời gian riêng biệt, chúng được lấy trung bình để tạo thành đạo hàm cuối cùng của mỗi tham số.
  - Theo Hình 3, DNN gốc trở nên sâu hơn do quá trình mở rộng dựa trên thời gian. Để làm rõ điểm này, chúng tôi nhắc nhở độc giả để chú ý các ngăn xếp thời gian sau khi được mở rộng. Điều này dẫn đến một cấu trúc sâu theo các khoảng thời gian khác nhau. Hơn nữa, mỗi ngăn xếp thời gian (với các giá trị τ khác nhau) chứa phần học tính đặc trưng sâu riêng của nó. Khi áp dụng trực tiếp BPTT, sự biến mất đạo hàm trên các tầng sâu không được tránh trong bước tinh chỉnh . Vấn đề này trở nên ngày càng nghiêm trọng trên các ngăn xếp thời gian cấp cao và các tầng phía trước.
  - Để giải quyết vấn đề nêu trên, chúng tôi đề xuất một giải pháp thực tế hơn để mang thông tin đạo hàm trực tiếp từ nhiệm vụ học tới mỗi ngăn xếp thời gian và mỗi tầng của phần

DL. Trong phần mở rộng thời gian, các đường kẻ đứt chấm màu đỏ dùng để dịch giúp truyền ngược từ ngăn xếp thời gian hiện tại lên các ngăn xếp thời gian trước đó và truyền xuống các tầng sâu hơn.

Algorithm 1 Training Process for the FRDNN

Input : Raw price ticks $p_1, \ldots, p_T$ received in an online manner; ρ, c_0 (learning rate).
Initialization: Initialize the parameters for the fuzzy layers (by fuzzy clustering), deep layers (auto-encoder) and reinforcement learning part sequentially.

1 **repeat**
2 c = c + 1;
3 Update learning rate $\rho_c = \min(\rho, \rho\frac{c_0}{c})$ for this outer iteration;
4 **for** $t = 1 \ldots T$ **do**
5 Generate Raw feature $\mathbf{f}_t$ vector from price ticks;
6 BPTT: Unfold the RNN at time t into $\tau + 1$ stacks;
7 Task-aware Propagation: Add the virtual links from the output to each deep layer;
8 BP: Back-propagate the gradient through the unfolded network as in Fig. 3;
9 Calculated $\nabla(U_t)_\Theta$ by averaging its gradient values on all the time stacks.;
10 Parameter Updating: $\Theta_t = \Theta_{t-1} - \rho_c \frac{\nabla(U_t)_\theta}{||\nabla(U_t)_\theta||}$;
11 **end**
12 **until** *convergence*;

- Các đường kẻ đứt chấm màu đỏ được kết nối từ nhiệm vụ UT đến nút đầu ra của mỗi ngăn xếp thời gian. Với cài đặt này, thông tin đạo hàm truyền ngược của mỗi ngăn xếp thời gian đến từ hai phần tương ứng: 1) ngăn xếp thời gian trước đó (độ trễ thời gian thấp hơn) và 2) hàm thưởng (nhiệm vụ học). Tương tự, đạo hàm của nút đầu ra trong mỗi ngăn xếp thời gian được đưa trở lại các tầng DL thông qua đường kẻ đứt chấm màu xanh lá cây. Phương pháp BPTT như vậy với các đường kẻ ảo kết nối với hàm mục tiêu được gọi là BPTT nhận biết nhiệm vụ.
- Quá trình chi tiết để huấn luyện FRDNN đã được tóm tắt trong Thuật toán 1. Trong thuật toán, chúng tôi ký hiệu ———————————— là ký hiệu chung để biểu diễn các tham số. Nó đại diện cho toàn bộ gia đình tham số ẩn liên quan đến FRDNN. Trước khi thực hiện giảm đạo hàm ở dòng 10, vector đạo hàm tính toán được chuẩn hóa thêm để tránh giá trị cực lớn trong vector đạo hàm

5. KIỂM TRA THỰC NGHIỆM

Không áp dụng Fuzzi

- Khởi tạo:

Thiết lập độ dài của chuỗi thời gian là TIME_STEPS.

Ban đầu, đặt $\delta 0 = 0$, tức là không mua hay bán.

- *Chính sách (Policy):*

Sử dụng một mạng nơ-ron (neural network) để đưa ra quyết định giao dịch từ phản hồi hiện tại và được ký hiệu là δt.

Mạng nơ-ron $g\theta$ (với các tham số θ) nhận vào đặc trưng hiện tại của thị trường ft và quyết định trước đó $\delta t-1$ để đưa ra quyết định giao dịch mới.

- *Hàm phần thưởng (Reward function):*

Hàm này định nghĩa lợi nhuận mà mô hình giao dịch đạt được.

Công thức của hàm phần thưởng là:

$Rt = R(zt, \delta t, \delta t-1, c) = \delta t-1 * zt - c * |\delta t - \delta t-1|$, trong đó:

zt là giá trị trả về (return) tại thời điểm t,

δt là quyết định được đưa ra bởi chính sách (policy) tại thời điểm t,

$\delta t-1$ là quyết định được đưa ra bởi chính sách tại thời điểm t-1,

c là chi phí giao dịch.

- *Hàm phần thưởng tích lũy (Cumulative reward function):*

Để có được giá trị hàm giá trị (value function) tại mỗi điểm thời gian, giá trị tích lũy trong suốt quá trình huấn luyện được định nghĩa là:

$UT(R1,...,RT|\theta) = \sum Tt=1\ Rt = \sum Tt=1\ R(zt, \delta t, \delta t-1, c)$.

Mục tiêu là tìm chính sách tốt nhất để tối đa hóa hàm phần thưởng tích lũy này.

- *Khung học tăng cường trực tiếp (Direct Reinforcement Learning framework):*

Đây là bài toán tối ưu hóa:

$\max\theta\in\Theta\ UT(R1,...,RT|\theta)$, trong đó:

$\{Rt = \delta t-1 * zt - c * |\delta t - \delta t-1|, \delta t = \tanh(g\theta(ft, \delta t-1))\}$,

với θ là tập hợp các tham số của mạng nơ-ron $g\theta$.

Sau khi tối ưu hóa phương trình trên, ta sẽ thu được tập hợp các tham số $\theta* = \text{argmax}\theta\in\Theta\ UT(R1,...,RT|\theta)$, đây là tập hợp các tham số của chiến lược giao dịch tối ưu được huấn luyện cho môi trường giao dịch được xem xét.

- *Triển khai thuật toán:*

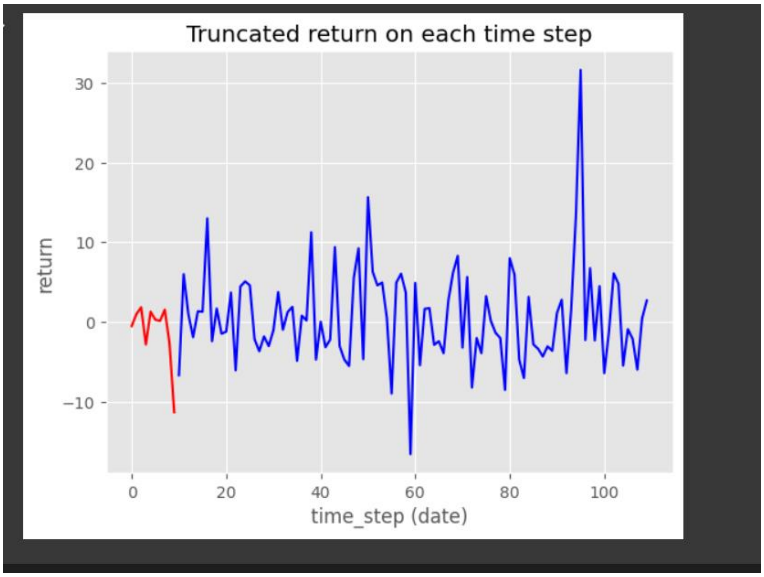
Đặt độ dài của mỗi chuỗi thời gian là TIME_STEPS.

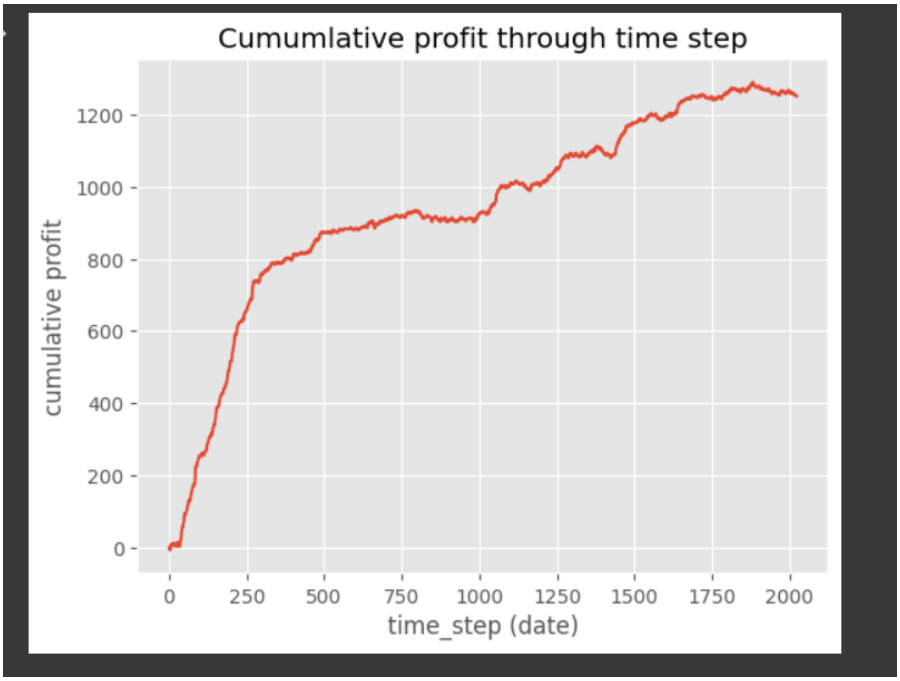
Ở thời điểm ban đầu t0, khởi tạo $\delta 0 = 0$, tức là không mua hoặc bán.

Để phù hợp với thuật toán, thay vì giải quyết bài toán tối đa hóa, ta sẽ chuyển đổi bài toán tối thiểu hóa bằng cách thêm dấu âm vào hàm mục tiêu: $\max\theta\in\Theta$ UT(R1...,RT|θ) = $\min\theta\in\Theta$ -UT(R1,...,RT|θ).

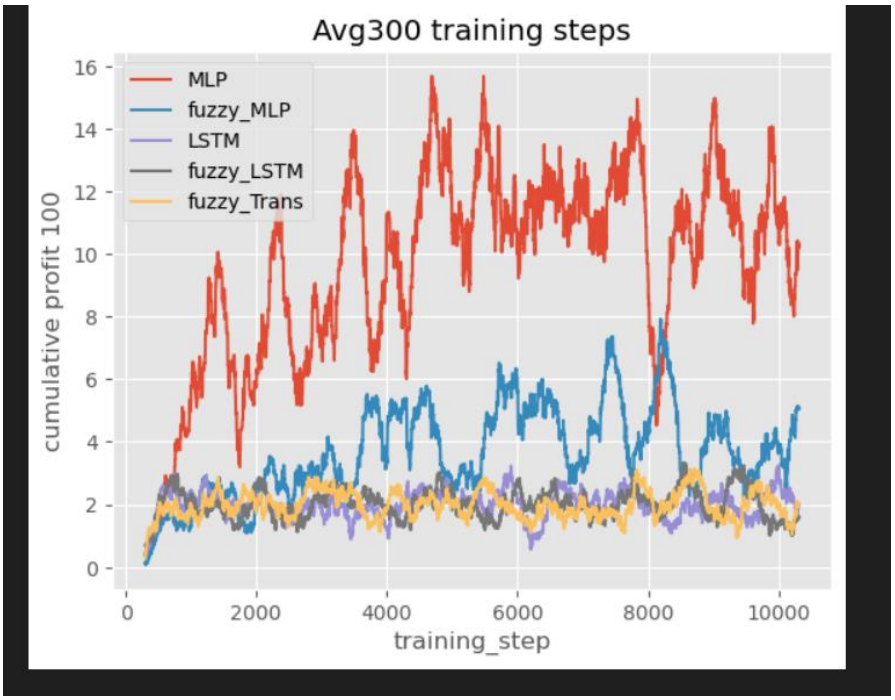
Vui lòng lưu ý rằng mô tả trên chỉ là một tóm tắt tổng quan dựa trên thông tin bạn cung cấp, và có thể thiếu một số chi tiết cụ thể trong thuật toán. Để hiểu rõ hơn về thuật toán này, tốt nhất là đọc nguyên văn của bài báo hoặc tham khảo các tài liệu liên quan.

- Kết quả đạt được:

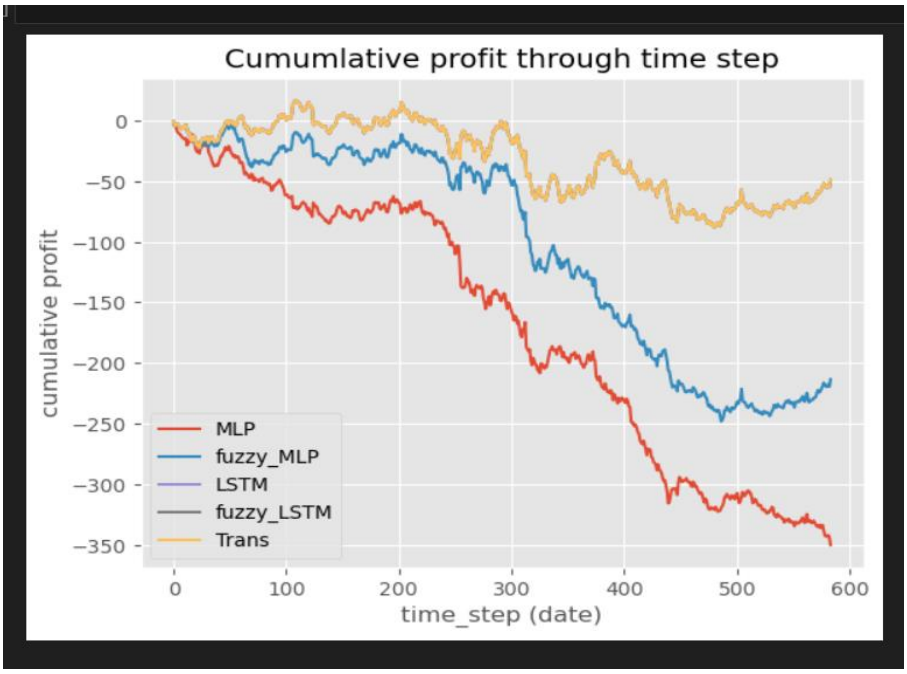




- ***Kết quả khi thêm bước Fuzzi vào train với 5 model khác nhau với (2 model có Fuzzi và 3 model không có Fuzzi)***



- ***Test model với tập data Amazon***



6. TÀI LIỆU THAM KHẢO

https://github.com/nithinsethu/Deep-Direct-Reinforcement-Learning-for-Financial-Signal-Representation-and-Trading