HỌC SÂU TRỰC TIẾP TĂNG CƯỜNG CHO BIỂU DIỄN TÍN HIỆU TÀI CHÍNH VÀ GIAO DỊCH

TRƯƠNG VĂN THÔNG (*), VÕ HOÀNG MINH TRÍ

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

truongvantthong@gmail.com, vhmtri2401@gmail.com

Tóm tắt: Chúng ta có thể đào tạo máy tính để vượt trội hơn các nhà giao dịch có kinh nghiệm trong giao dịch tài sản tài chính không? Bài viết này giải quyết thách thức này bằng cách giới thiệu một mạng thần kinh sâu tái phát (NN) để biểu diễn và giao dịch tín hiệu tài chính theo thời gian thực. Mô hình của chúng tôi kết hợp các khái niệm về học sâu (DL) và học tăng cường (RL). Thành phần DL nắm bắt các điều kiện thị trường năng động để tìm hiểu tính năng thông tin, trong khi mô-đun RL đưa ra quyết định giao dịch và tối đa hóa phần thưởng trong một môi trường không xác định. Hệ thống thần kinh kết hợp cả cấu trúc sâu và cấu trúc lặp lại, đồng thời chúng tôi đề xuất một phương pháp lan truyền ngược nhận thức nhiệm vụ thông qua phương pháp thời gian để xử lý vấn đề biến mất độ dốc trong quá trình đào tạo sâu. Tính mạnh mẽ của hệ thống thần kinh được đề xuất được xác minh thông qua các thử nghiệm trên thị trường tương lai chứng khoán và hàng hóa trong các điều kiện thử nghiệm đa dạng. Từ khóa: Học sâu, xử lý tín hiệu tài chính, mạng nơron tài chính, học tăng cường.

Từ khóa: Deep learning, xử lý tín hiệu tài chính, mạng nơ-ron cho tài chính, reinforcement learning.

DEEP DIRECT REINFORCEMENT LEARNING FOR FINANCIAL SIGNAL REPRESENTATION AND TRADING

Faculty of Information Technology, Industrial University of Ho Chi Minh City

Abstract: Can we train computers to outperform experienced traders in financial asset trading? This paper addresses this challenge by introducing a recurrent deep neural network (NN) for real-time financial signal representation and trading. Our model combines the concepts of deep learning (DL) and reinforcement learning (RL). The DL component captures dynamic market conditions for informative feature learning, while the RL module makes trading decisions and maximizes rewards in an unknown environment. The neural system incorporates both deep and recurrent structures, and we propose a task-aware backpropagation through time method to handle the issue of gradient vanishing during deep training. The robustness of the proposed neural system is verified through experiments on stock and commodity future markets under diverse testing conditions.

Keywords: Deep learning, financial signal processing, neural network for finance, reinforcement learning.

Lĩnh vực: Công nghệ thông tin

1. TỔNG QUAN

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

2.1 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, trung lập, bán} = {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:

$$R_t = \delta_{t-1} z_t - c|\delta_t - \delta_{t-1}|. \quad (1)$$

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.

$$\max_{\Theta} U_T\{R_1 \dots R_T|\Theta\} \quad (2)$$

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 $U_T = \sum_{t=1}^{T} R_t$. 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

$$\delta_t = \tanh[\langle \mathbf{w}, \mathbf{f}_t \rangle + b + u\delta_{t-1}]. \quad (3)$$

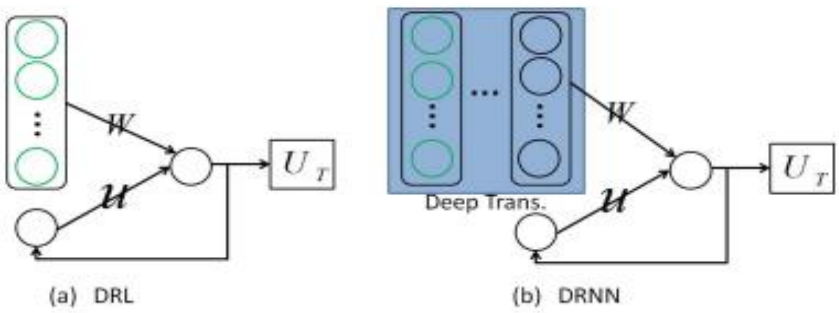


Fig. 1. Comparisons of DRL and the proposed DRNN for joint feature learning and DRT.

Trong DRL, m giá trị trở lại gần đây được áp dụng trực tiếp như là vector đặc trưng:

$$\mathbf{f}_t = [z_{t-m+1}, \ldots, z_t] \in \mathbb{R}^m. \quad (4)$$

Ngoài các đặc trưng đó, thuật ngữ $u\delta_{t-1}$ 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 $\Theta = \{w, u, b\}$ có thể tối đa hóa hàm phần thưởng toàn cục trong.

2.2 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 f_t (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 Fig (1) 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 $\mathbf{F}_t = g_d(\mathbf{f}_t)$ được thu được bằng cách biến đổi phân cấp vector đầu vào f_t thông qua DNN với một ánh xạ phi tuyến g_d (·). Sau đó, hành động giao dịch trong (3) được xác định bởi phương trình sau:

$$\delta_t = \tanh[\langle \mathbf{w}, g_d(\mathbf{f}_t)\rangle + b + u\delta_{t-1}]. \quad (5)$$

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:

$$a_i^l = \langle \mathbf{w}_i^l, \mathbf{o}^{(l-1)}\rangle + b_i^l, \quad o_i^l = \frac{1}{1+e^{-a_i^l}} \quad (6)$$

2.3. 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.

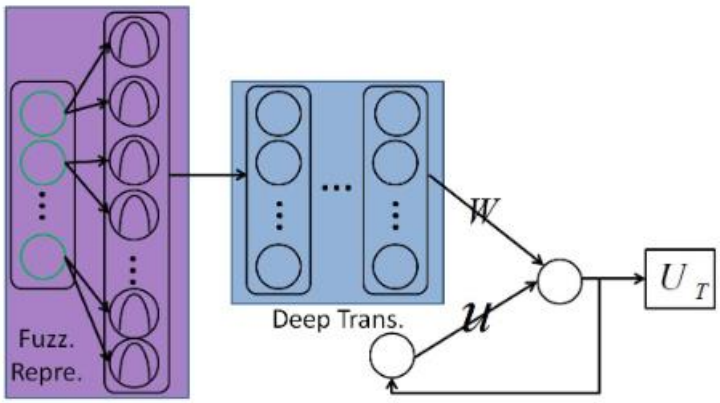


Fig. 2. Overview of fuzzy DRNNs for robust feature learning and self-taught trading.

Trong mạng nơ-ron mờ, phần biểu diễn mờ truyền thống được kết nối với vectơ đầu vào ft (nút xanh lá cây) bằng các hàm thành viên khác nhau. Lưu ý rằng trong cài đặt của chúng tôi, chúng tôi tuân theo một công trình tiên phong [35] để gán k giá trị mờ khác nhau cho mỗi chiều của vectơ đầu vào. Trong hình đại diện ở Fig 2, chỉ có hai nút mờ (k=2) được kết nối với mỗi biến đầu vào do giới hạn không gian. Trong thực tế, chúng tôi sử dụng k cố định là 3 để mô tả các điều kiên tăng, giảm và không có xu hướng. Về mặt toán học, hàm thành viên mờ thứ i $v_i(\cdot) : R \rightarrow [0, 1]$.ánh xạ bởi đầu vào thứ i dưới dạng một giá trị mờ:

$$o_i^{(l)} = v_i\left(a_i^{(l)}\right) = e^{-\left(a_i^{(l)} - m_i\right)^2 / \sigma_i^2} \quad \forall i. \qquad (7)$$

Chúng tôi sử dụng hàm thành viên mờ Gaussian với giá trị trung bình m và phương sai σ^2 trong hệ thống.Sau khi có được biểu diễn mờ, chúng được kết nối trực tiếp với lớp biến đổi sâu để tìm các biến đổi sâu.Tóm lại mạng nơ-ron mờ và nơ-ron hồi quy sâu (FDRNN) bao gồm ba phần chính là biểu diễn mờ, biến đổi sâu, và DRT. Khi xem FRDNN như một hệ thống nhất, ba phần

này lần lượt đóng vai trò xử lý dữ liệu trước(giảm sự không chắc chắn),học đặc trưng(biến đổi sâu, và tạo chính sách giao dịch (RL). Mô hình tối ưu hóa được đưa ra như sau:

$$\begin{aligned} &\max_{\{\Theta, g_d(\cdot), v(\cdot)\}} U_T(R_1..R_T) \\ &\text{s.t. } R_t = \delta_{t-1} z_t - c|\delta_t - \delta_{t-1}| \\ &\quad \delta_t = \tanh(\langle \mathbf{w}, \mathbf{F}_t \rangle + b + u\delta_{t-1}) \\ &\quad \mathbf{F}_t = g_d(v(\mathbf{f}_t)) \end{aligned} \tag{8}$$

Trong đó có ba nhóm thông số cần học:

$\Theta = (\mathbf{w}, b, u)$:các tham số giao dịch.

$v(\cdot)$:biểu diễn mờ

$g_d(\cdot)$:biến đổi sâu.

Trong hàm tối ưu trên UT là tổng lợi nhuận cuối cùng của hàm RL , δ_t là quyết định giao dịch được xấp xỉ bằng FDRNN và Ft là biểu diễn đặc trưng cấp cao của điều kiện thị trường hiện tại được tạo ra bởi DL.

3. DRNN

3.1 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 (m_i) và độ rộng (σ_i^2) 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 σ_i^2 tương ứng.
- AE được sử dụng để khởi tạo phần biến đổi sâu [Fig 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:

$$\sum_t \left\| \mathbf{x}_t^{(l)} - h_\gamma\left(h_\theta\left(\mathbf{x}_t^{(l)}\right)\right) \right\|_2^2 + \eta \|\mathbf{w}^{(l+1)}\|_2^2. \tag{9}$$

  - Lưu ý rằng $\mathbf{x}_t^{(l)}$ 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 $\theta = \{\mathbf{w}^{(l+1)}, b^{(l+1)}\}$ 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 Fig1(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.

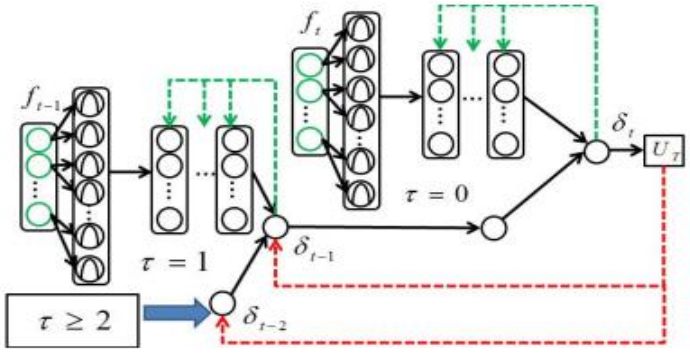


Fig. 3. Task-aware BPTT for RDNN fine tuning.

3.2 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.

$$\frac{\partial U_T}{\partial \theta} = \sum_t \frac{dU_t}{dR_t}\left\{\frac{dR_t}{d\delta_t}\frac{d\delta_t}{d\theta} + \frac{dR_t}{d\delta_{t-1}}\frac{d\delta_{t-1}}{d\theta}\right\} \qquad (10)$$
$$\frac{d\delta_t}{d\theta} = \frac{\partial \delta_t}{\partial \theta} + \frac{\partial \delta_t}{\partial \delta_{t-1}}\frac{d\delta_{t-1}}{d\theta}.$$

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

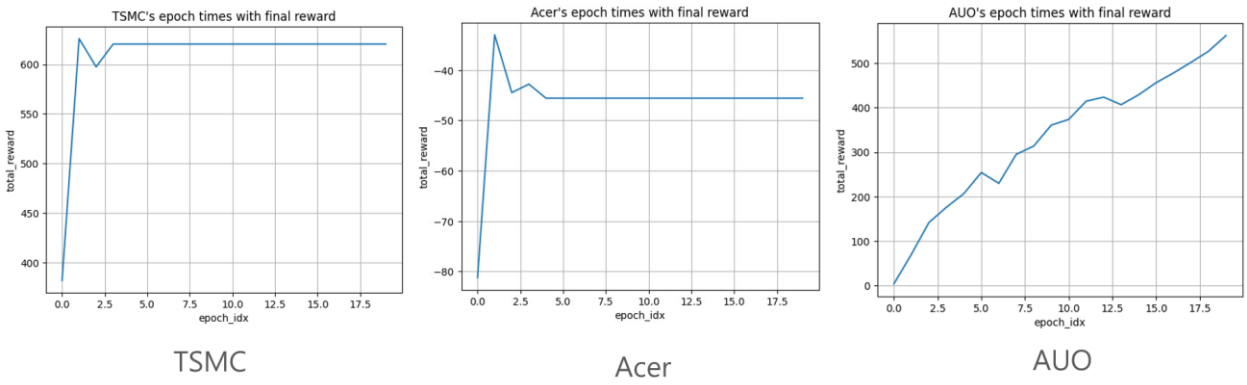
- Các đường kẻ đứt chấm màu đỏ được kết nối từ nhiệm vụ U_T đến nút đầu ra của mỗi ngăn xếp thời gian. Với cài đặt này, thông tin đạo hàm truyền ngược của mỗi ngăn xếp thời gian đến từ hai phần tương ứng: 1) ngăn xếp thời gian trước đó (độ trễ thời gian thấp hơn) và 2) hàm thưởng (nhiệm vụ học). Tương 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 Θ 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.

4. KIỂM TRA THỰC NGHIỆM

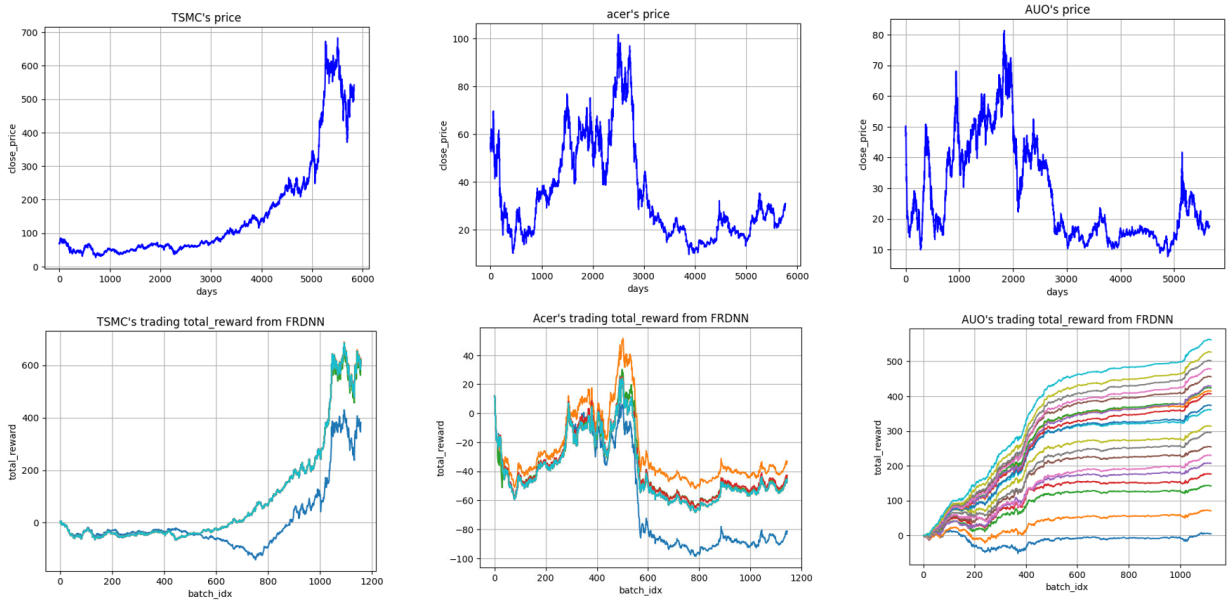
4.1 HIỆN THỰC BIỂU ĐỒ

4.1.1 STOCK TARGET CHOOSE

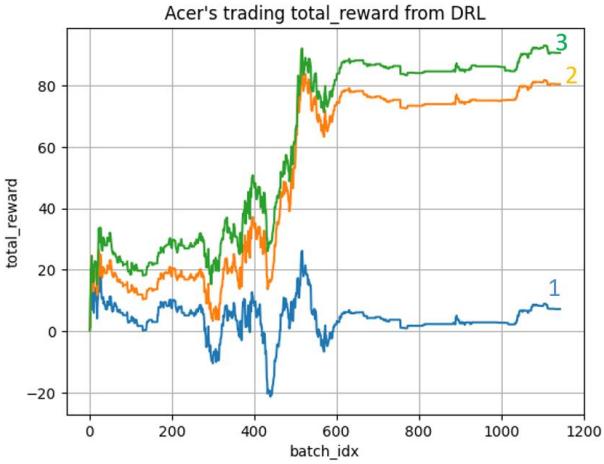
4.1.2 Epochs total reward

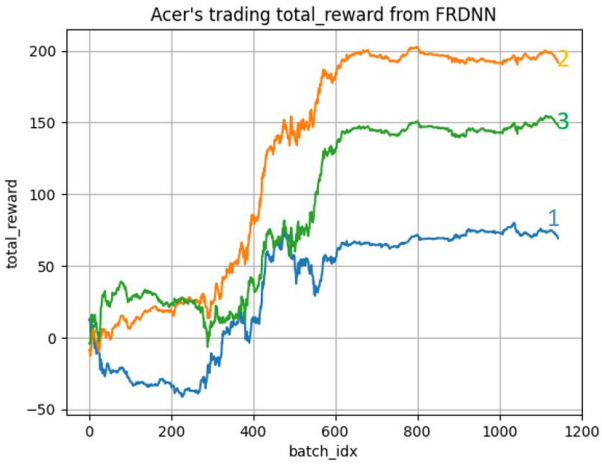


4.1.3 Epochs different



4.1.4 Fuzzy or not?





REFERENCES

[1] A. Alpher, Frobnication, *Journal of Foo*, vol. 12, no. 1, pp. 234-778, 2002.

[2] A. Alpher and J. P. N. Fotheringham-Smythe, Frobnication revisited, *Journal of Foo*, vol. 13, no. 1, pp. 234-778, 2003.

[1] E. W. Saad, D. V. Prokhorov, and D. C. Wunsch, II, "Comparative study of stock trend prediction using time delay, recurrent and prob- abilistic neural networks,"IEEE Trans. Neural Netw., vol. 9, no. 6, pp. 1456–1470, Nov. 1998.

[2] D. Prokhorov, G. Puskorius, and L.Feldkamp, "Dynamical neural net- works for control," inA Field Guide to Dynamical Recurrent Networks. New York, NY, USA: IEEE Press, 2001.

[3] D. Zhao and Y. Zhu, "MEC—A near-optimal online reinforcement learning algorithm for continuous deterministic systems,"IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 2, pp. 346–356, Feb. 2015.

[4] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward,"Science, vol. 275, no. 5306, pp. 1593–1599, 1997.

[5] H. R. Beom and K. S. Cho, "A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning,"IEEE Trans. Syst., Man, Cybern., vol. 25, no. 3, pp. 464–477, Mar. 1995.

[6] V. Mnihet al., "Human-level control through deep reinforcement learn- ing,"Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[7] H. J. Kim, M. I. Jordan, S. Sastry, and A. Y. Ng, "Autonomous helicopter flight via reinforcement learning," inProc. Adv. Neural Inf. Process. Syst., 2003, pp. 799–806.

[8] Y.-D. Song, Q. Song, and W.-C. Cai, "Fault-tolerant adaptive control of high-speed trains under traction/braking failures: A virtual parameter- based approach,"IEEE Trans. Intell. Transp. Syst., vol. 15, no. 2, pp. 737–748, Apr. 2014.

[9] C. J. Neely, D. E. Rapach, J. Tu, and G. Zhou, "Forecasting the equity risk premium: The role of technical indicators,"Manage. Sci., vol. 60, no. 7, pp. 1772–1791, 2014.

[10] J. J. Murphy,Technical Analysis of the Financial Markets: A Compre- hensive Guide to TradingMethods and Applications.NewYork,NY, USA: New York Institute of Finance, 1999.

[11] J. M. Poterba and L. H. Summers, "Mean reversion in stock prices: Evidence and implications,"J. Financial Econ., vol. 22, no. 1, pp. 27–59, 1988.

[12] R. S. Sutton and A. G. Barto,Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 1998. [13] R. S. Sutton and A. G. Barto,Introduction to Reinforcement Learning. Cambridge, MA, USA: MIT Press, 1998. [14] M. L. Puterman,Markov Decision Processes: Discrete Stochastic Dynamic Programming. Hoboken, NJ, USA: Wiley, 2014.

[15] G. Tesauro, "TD-Gammon, a self-teaching backgammon program, achieves master-level play,"Neural Comput., vol. 6, no. 2, pp. 215–219, 1994.

[16] D. P. Bertsekas,Dynamic Programming and Optimal Control. Belmont, MA, USA: Athena Scientific, 1995.

[17] J. Moody and M. Saffell, “Learning totrade via direct reinforcement,” IEEE Trans. Neural. Netw., vol. 12, no. 4, pp. 875–889, Jul. 2001.

[18] M. A. H. Dempster and V. Leemans, “An automated FX trading system using adaptive reinforcement learning,”Expert Syst. Appl., vol. 30, no. 3, pp. 543–552, 2006.

[19] Y. Deng, Y. Kong, F. Bao, and Q. Dai, “Sparse coding-inspired optimal trading system for HFT industry,”IEEE Trans. Ind. Informat., vol. 11, no. 2, pp. 467–475, Apr. 2015.

[20] K.-I. Kamijo and T. Tanigawa, “Stock price pattern recognition: A recurrent neural network approach,” inProc. Int. Joint Conf. Neural Netw., San Diego, CA, USA, 1990, pp. I-215–I-221.
[21] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, “Low-rank structure learning via nonconvex heuristic recovery,”IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 3, pp. 383–396, Mar. 2013.
[22] K. K. Ang and C. Quek, “Stock trading using RSPOP: A novel rough set-based neuro-fuzzy approach,”IEEE Trans. Neural. Netw., vol. 17, no. 5, pp. 1301–1315, Sep. 2006.
[23] Y. Deng, Y. Liu, Q. Dai, Z. Zhang, and Y. Wang, “Noisy depth maps fusion for multiview stereo via matrix completion,”IEEE J. Sel. Topics Signal Process., vol. 6, no. 5, pp. 566–582, Sep. 2012.
[24] Y. Deng, Q. Dai, and Z. Zhang, “Graph Laplace for occluded face completion and recognition,”IEEE Trans. Image Process., vol. 20, no. 8, pp. 2329–2338, Aug. 2011.
[25] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid match- ing using sparse coding for image classification,” inProc. IEEE Conf. Comput. Vis. Pattern Recognit., Miami Beach, FL, USA, Jun. 2009, pp. 1794–1801.
[26] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” inProc. 26th Annu. Int. Conf. Mach. Learn., Montreal, QC, Canada, Jun. 2009, pp. 609–616.
[27] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre- trained deep neural networks for large-vocabulary speech recogni- tion,”IEEE Trans. Audio, Speech, Language Process., vol. 20, no. 1, pp. 30–42, Jan. 2012.
[28] Y. Bengio, “Learning deep architectures for AI,”Found. Trends Mach. Learn., vol. 2, no. 1, pp. 1–127, 2009. [29] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” inProc. IEEE Int. Conf. Acoust., Speech Signal Process., Vancouver, BC, Canada, May 2013, pp. 6645–6649.
[30] J. Moody, L. Wu, Y. Liao, and M. Saffell, “Performance func- tions and reinforcement learning for trading systems and portfolios,” J. Forecasting, vol. 17, nos. 5–6, pp. 441–470, 1998.
[31] J. D. Bransford, A. L. Brown, and R. R. Cocking,How People Learn: Brain, Mind, Experience, and School. Washington, DC, USA: National Academy Press, 1999.
[32] T. Ohyama, W. L. Nores, J. F. Medina, F. A. Riusech, and M. D. Mauk, “Learning-induced plasticity in deep cerebellar nucleus,”J. Neurosci., vol. 26, no. 49, pp. 12656–12663, 2006.
[33] G. J. Klir and T. A. Folger,Fuzzy Sets, Uncertainty, and Information. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.
[34] N. R. Pal and J. C. Bezdek, “Measuring fuzzy uncertainty,”IEEE Trans. Fuzzy Syst., vol. 2, no. 2, pp. 107–118, May 1994.
[35] C.-T. Lin and C. S. G. Lee, “Neural-network-based fuzzy logic con- trol and decision system,”IEEE Trans. Comput., vol. 40, no. 12, pp. 1320–1336, Dec. 1991.
[36] Y. Deng, Y. Li, Y. Qian, X. Ji, and Q. Dai, “Visual words assignment via information-theoretic manifold embedding,”IEEE Trans. Cybern., vol. 44, no. 10, pp. 1924–1937, Oct. 2014.
[37] C.-T. Lin, C.-M. Yeh, S.-F. Liang, J.-F. Chung, and N. Kumar, “Support- vector-based fuzzy neural network for pattern classification,”IEEE Trans. Fuzzy Syst., vol. 14, no. 1, pp. 31–41, Feb. 2006.

[38] F.-J. Lin, C.-H. Lin, and P.-H. Shen, “Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive,”IEEE Trans. Fuzzy Syst., vol. 9, no. 5, pp. 751–759, Oct. 2001.
[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,”J. Mach. Learn. Res., vol. 11, no. 12, pp. 3371–3408, Dec. 2010.
[40] P. J. Werbos, “Backpropagation through time: What it does and how to do it,”Proc. IEEE, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
[41] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,”IEEE Trans. Neural Netw.,vol.5, no. 2, pp. 157–166, Mar. 1994.
[42] O. Ledoit and M. Wolf, “Robust performance hypothesis testing with the Sharpe ratio,”J. Empirical Finance, vol. 15, no. 5, pp. 850–859, 2008.
[43] W. F. Sharpe, “The Sharpe ratio,”J. Portfolio Manage., vol. 21, no. 1, pp. 49–58, 1994.
[44] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,”Neural Comput., vol. 12, no. 10, pp. 2451–2471, 2000.