Noname manuscript No.
(will be inserted by the editor)

Multi-Agent Reinforcement Learning approach for Hedging Portfolio problem

Uyen Pham · Quoc Luu · Hien Tran

Received: date / Accepted: date

Abstract Developing a hedging strategy to reduce risk of losses for a given set of stocks in a portfolio is a difficult task due to cost of the hedge. In Vietnam stock market, cross-hedge is involved hedging a long position of a stock because there is no put option for the stock. In addition, only VN30 stock index futures contracts are traded on Hanoi Stock Exchange (HNX). Inspired by recently achievement of deep reinforcement learning (DRL), we explore feasibility to construct a hedging strategy automatically by leveraging cooperative multi-agent in reinforcement learning techniques without advanced domain knowledge. In this work, we use 10 popular stocks on Ho Chi Minh Stock Exchange (HSX), and VN30F1M (VN30 Index Futures contracts within one month settlement) to develop a stock market simulator (including transaction fee, tax, and settlement date of transactions) for reinforcement learning (RL) agent training. We use daily return as input data for training process. Results suggest that the agent can learn trading and hedging policy to make profit and reduce losses. Furthermore, we also find that our agent can protect portfolios and make positive profit in case market collapses systematically. In practice, this work can help Vietnam's stock market investors to improve performance and reduce losses in trading, especially when the volatility cannot be controlled.

U. Pham
Economic Mathematics
University of Economics and Law
E-mail: uyenph@uel.edu.vn

Q. Luu
Quantitative and Computational Finance
John von Neumann Institute
E-mail: quoc.luu2015@qcf.jvn.edu.vn

H. Tran
School of Engineering
Tan Tao University
E-mail: hien.tran@ttu.edu.vn

Keywords Deep reinforcement learning · Hedging · Trading · Portfolio

1 Introduction

Hedging a position in stock is an attractive topic for both academics and practitioners. The objective of hedging is to minimize market risk due to price fluctuation, maximize profit by speculation on the basis, and construct a portfolio with reduced risk [15]. Portfolio managers have used stock index futures as a means to adjust desired return of a portfolio and potential loss since the 1980s. Main advantage of index futures as a major hedging tool is liquidity, lower transaction costs [17]. However, hedge strategies are not always effective as expected because relationship of cash price and price of a future contract is usually not perfect, or hedged position in stock is different from the underlying portfolio for the index contract [14], [15]. It is possible to increase risk of potential loss that leads to negative return. Hence, hedgers have to determine the optimal hedge ratio to control the risk of the portfolio.

Differ from supervised learning and unsupervised learning, reinforcement learning mainly relies on experience of repeated interaction to learn optimal policy in order to make sequential decisions to maximum rewards in a given environment [31]. In a complex and dynamic environment, it may require huge amounts of computational power over a long period of time to train. With revolution of deep learning techniques and computer hardware, reinforcement learning has become more feasible by using deep neural network as a function estimator. From long time horizons with high dimensional observation and action spaces in real-time strategy controls game, to self-driving vehicles, data center cooling systems, deep reinforcement learning has been more and more applied to solve many complex real-world challenges [5], [29], [13]. In finance, deep reinforcement learning is also widely adopted. Zhang et.al. [36] make the algorithm to design trading strategies for continuous futures contracts. They use technical indicators such as Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI) as a part of input features. The agent shows that it can deliver profit even under heavy transaction costs. Ganesh et.al. [16] develop a multi-agent dealer market for market marking with different competitive scenarios, market price conditions. The research suggests that trained agent can learn to manage inventory and its competitor's policy for pricing.

However, not many researches carried out to investigate feasibility of stock and futures trading to hedge portfolio at the same time using deep reinforcement learning. In this study, we selected 10 popular stocks on HSX, and one stock index futures contract on HNX to build a simulation of stock market environment with real market data to study learning performance of our agent. Our objective in this work is to investigate if cooperation of multi-agent to determine optimal hedging strategy to protect stock portfolio is achievable.

Fig. 1 Daily Movement of VN30 Index from 19 December 2019 to 16 April 2020

2 Related Work

2.1 Hedging Effectiveness of Stock Index Futures

Motivated by risk reduction, hedging a stock portfolio with index futures has been an active research topic since it was introduced [14], [17], [9]. A hedger supposes that return of a hedged position (e.g. stock portfolio) can be close to risk-free interest rate. Optimal hedge ratio hr was estimated using one-to-one hedge, the beta hedge, and minimum variance hedge ratio (MVHR) [11], [6]. With cash prices and futures moving closely together assumption, one-to-one hedge strategy suggests $hr = -1$. Beta hedge strategy uses negative of the beta cash portfolio as hr. The hedger expects the overall beta of the portfolio is zero. However, in practice, change of price of spot and futures does imperfectly correlated. Especially in case of cross-hedge, one-to-one and beta hedge may not reduce risk. In contrast, futures hedging can lead to unexpected loss. The MVHR was introduced to workaround for the problem by taking the imperfect relationship of prices into account and determine the optimal ratio hr:

$$hr = -\frac{Cov(R_s, R_f)}{Var(R_f)} \tag{1}$$

Furthermore, by using ordinary least squares (OLS) regression to estimate minimum risk hedge, Figlewski [14] found that hedging effectiveness of a large capitalization portfolio can yield “fairly good” for a one week holding period (p. 663). However, with diversified portfolio of small stocks, the effectiveness is reduced significantly. Basis risk is also not negligible even if the spot is hedged with index futures itself. When basis risk arises, it can generate profit or loss. It is suggested that one day hedge strategy can potentially increase basis risk and reduce risk effectiveness than one week hedge.

Stating that traditional methods to estimate optimal hedge ratio are mis-specified, error correction model (ECM) was proposed to estimate optimal hedge ratio and forecast out-of-sample for evaluation as in [17]. Firstly, it carries out cointegration test. Secondly, it use OLS regression to estimate error correction model. The model incorporates relationship of the long-run equilibrium as well as the short-run dynamics. The result shows that optimal hedge ratio is significantly improved with adjusted R^2 from ECM is higher than traditional methods. Also by comparing root mean squared error (RMSE), out-of-sample forecasts from the ECM are found to be better than other methods.

Beyond variance and standard deviation, value at risk (VaR) and conditional conditional value at risk (CVaR) are extensively applied to measure market risk for hedging strategies of portfolio [10], [21]. VaR was introduced by J.P. Morgan in the 1990s and widely adopted to summarize risk of an entire portfolio at the end of each day [25]. However, VaR is not a coherent risk measure. To be coherent, it must be monotonicity, positive homogeneity, translation invariance, and subadditivity [4], [3]. CVaR was constructed with these properties as a new valid practical alternative to VaR [1]. Espeholt et.al. [2] show that CVaR is applicable to a wide range of derivatives portfolio including American options and exotic options. In addition, it is found that CVaR risk metric is suitable for asymmetric return distributions and expected loss of portfolio can be minimized in many circumstances [32].

2.2 Deep Reinforcement Learning in Trading

Reinforcement learning was proposed to train trading systems to make profit and adjust risk [28], [27]. Recurrent learning and Q-Learning with neural networks were used to optimize financial performance functions including risk-adjusted return, immediate utility for online learning [28]. Furthermore, portfolio with continuous quantities of multiple assets was consider. The result shows that reinforcement learning can avoid large losses when market crashed. Basis risk hedging strategy was developed using reinforcement learning as in [34]. Without assets modeling requirement, State–action–reward–state–action (SARSA) based algorithm was applied to find an optimal trading policy to hedge a non-traded asset. Q-Learning is proposed to extend Black-Scholes-Merton (BSM) model for option pricing and hedging in [18]. In an attempt to escape Greeks and complete market assumptions in risk management, by leveraging deep reinforcement learning, a greek-free approach is proposed to focus on on realistic market dynamics and out-sample testing performance for optimizing hedging of a portfolio of derivatives [7]. Deep reinforcement learning is further investigated for hedging a portfolio of over-the-counter derivatives under generic market frictions as in [8]. Trading costs and liquidity constraints are considered in the approach.

3 Multi-Agent Reinforcement Learning Approach

3.1 Deep Reinforcement Learning

3.1.1 Single Agent Reinforcement Learning

For a given stochastic environment ε, an agent interacts with the environment by choosing to take a legal action a_t from many actions at time step t, $a_t \in A \equiv \{1, \ldots, L\}$. Action space can be discrete or continuous. When the selected action is passed to the environment ε, internal state s_t is switched to another state in many states S. In other words, the process of sequential interactions between the agent and the environment is result of mapping from perceived states s_t to actions a_t by policy π. For instance, in Dota 2 game, internal state can be all the available information for human player including positions, health, map [5]. In this research, internal state is asset return in percentage, position of each asset. In return, the agent receives reward r_t of the passed action as feedback, and new internal state s_{t+1} for each time step until reaching terminate state. The ultimate goal of deep reinforcement learning is to find an policy π that can select optimal action to maximize reward signal for each state s_t. Value of a state measures total expected return by predicting future reward with discount-rate $\gamma \in [0, 1]$. The total accumulated discounted return G_t from time t with k time steps in the future is defined:

$$G_t = \sum_{\infty}^{k=0} \gamma^k r_{t+k+1} \tag{2}$$

The state value $V_\pi(s)$ is defined as in [31]:

$$\begin{aligned} V_\pi(s) \doteq& \mathbb{E}_\pi[G_t|S_t = s] \\ =& \mathbb{E}_\pi\left[\sum_{\infty}^{k=0} \gamma^k R_{t+k+1}|S_t = s\right] \end{aligned} \tag{3}$$

Similarly, action value $Q_\pi(s, a)$ is the expected return for state s from selecting action a following policy π.

$$\begin{aligned} Q_\pi(s, a) =& \mathbb{E}_\pi[G_t|S_t = s, A_t = a] \\ =& \mathbb{E}_\pi\left[\sum_{\infty}^{k=0} \gamma^k R_{t+k+1}|S_t = s, A_t = a\right] \end{aligned} \tag{4}$$

In value-based reinforcement learning, off-policy Q-learning was introduced to estimate the action value function $Q_\pi(s, a)$, defined as [33].

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right] \tag{5}$$

The algorithm directly approximate the optimal action value function $Q_*(s, a)$. By extending neural network as a function approximator, the function can be estimated as $Q_*(s,a) \approx Q(s,a,\theta)$. The approach is referred as Q-network with weights θ [26].

In contrast to value-based methods, policy-based can select actions directly by parameterizing the policy $\pi(a|s,\theta)$ and using gradient ascent to optimize $\mathbb{E}[R_t]$ to find the best θ that can produce the highest reward. In term of probability, we can express the policy as $\pi(a|s,\theta) = Pr\{A_t = a|S_t = s, \theta_t = \theta\}$ for the probability of a given environment ε in state s at time t with parameter θ to take action a. Actor-critic algorithms use both value and policy functions to learn approximations [22]. To improve performance, the critic learns a value function (e.g. state value) and is used to update policy parameters of actor.

3.1.2 Multi-Agent Reinforcement Learning

Extent from single agent, multi-agent learning is considered n agents interacting with the environment ε. At state s_t of time step t, each agent selects action a_t^i to react to the state and receive reward r_t^i, where $i \in \{1, \dots n\}$. Hence, for any given joint policy $\pi(a|s) \doteq \prod_i^n \pi^i(a^i|s)$ with state $s \in S$, value state function can be defined as in [35]:

$$V^i_{\pi^i,\pi^{-i}}(s) \doteq \mathbb{E}\left[\sum_{\infty}^{k=0} \gamma_k^i R^i_{t+k+1} \middle| a_t^i \sim \pi^i(\cdot|s_t), s_0 = s\right] \tag{6}$$

where $-i$ indicates that all agents except agent i.

3.2 High Throughput Architecture with Importance Weighted Actor-Learner Architecture

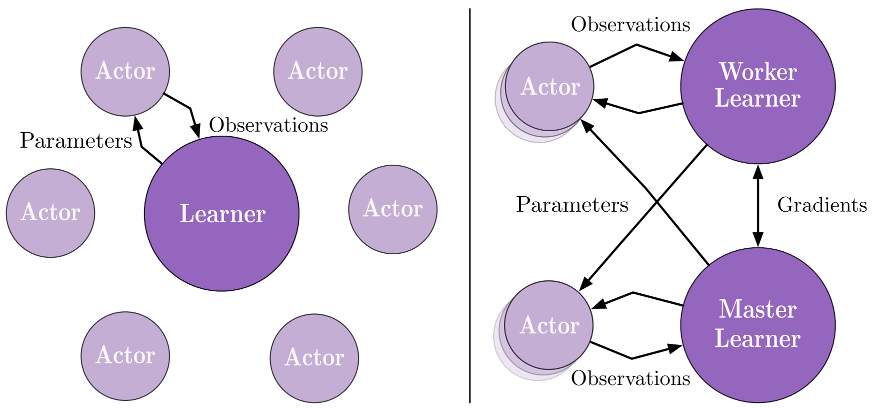

Fig. 2 Left: Single Learner. Right: Multiple Synchronous Learners. Adopted from [12]

Importance weighted actor-learner architecture (IMPALA) is a decoupled actor-critic style learner with introduction of V-trace off-policy to learn a policy π and a baseline function V^{π} that achieves stability, high data throughput and efficiency for agent training [12]. Moreover, deep neural networks can be train efficiently with IMPALA as suggested in Fig. 2. Suppose at time t, a given local actor policy μ generates trajectory $(s_t, a_t, r_t)_{t=k}^{t=k+n}$, The n-step V-trace target for value approximation $V(s_k)$ at state s_k, is defined as:

$$
\begin{aligned}
v_k &\doteq V(s_k) + \sum_{t=k}^{k+n-1} \gamma^{t-k} \big(\prod_{i=k}^{t-1} c_i\big) \delta_t V \\
\delta_t V &\doteq \rho_t (r_t + \gamma V(s_{t+1}) - V(s_t)) \\
\rho_t &\doteq \min\big(\bar{\rho}, \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}\big) \\
c_i &\doteq \min\big(\bar{c}, \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}\big)
\end{aligned}
\tag{7}
$$

where $\delta_t V$ is temporal difference for V, ρ_t and c_i are truncated importance sampling. It is worth to note that the truncation levels are assumed $\bar{c} \leq \bar{\rho}$.

Furthermore, value function V_θ and policy π_ω with θ and ω parameters respectively, can be updated in the direction of:

$$
\begin{aligned}
\Delta\theta &= (v_k - V_\theta(s_k)) \nabla_\theta V_\theta(s_k) \\
\Delta\omega &= \rho_k \nabla_\omega log\pi_\omega(a_k|s_k)(r_k + \gamma v_{k+1} - V_\theta(s_k)) - H(\omega) \\
H(\omega) &= \nabla_\omega \sum_a \pi_\omega(a|s_k) log\pi_\omega(a|s_k).
\end{aligned}
\tag{8}
$$

Entropy $H(\omega)$ is added to avoid immature convergence and encourage exploration in agent training process. IMPALA algorithm can be used to concurrently train for multiple tasks with one set of weights due to efficiency of the architecture.

4 Experiments

4.1 Data

We collect daily historical stock prices and volumes data from Ho Chi Minh stock exchange (HSX) for equity and Ha Noi stock exchange (HNX) for derivatives. We use data of the stock markets from 25 September 2017 to 21 May 2020 for our training and evaluating purposes. The time range covers high fluctuation of price periods as impact of market events.

4.2 Data Prepossessing and Network Architecture

Training data from raw inputs rather than handcrafted features is often recommended for feeding data to deep neural networks to achieve higher performance [23]. Likewise, researches show that reinforcement learning can exceed human capabilities without human expert data or domain knowledge [26], [30]. As a result, in this study, instead of applying advanced quantitative finance theories to develop trading and hedging strategy, daily return data of each asset collected from HSX and HNX exchanges is used as main components of environment observation. Specifically, we conducted a set of 12 different periods for out-sample evaluating. We use data from 25 September 2017 to 16 May 2019 for training, and from 17 May 2019 to 21 May 2020 for evaluating (see Table. 1). We also provide position and unrealized profit of each asset (P&L) in portfolio to our neural network.

Table 1 Evaluation Periods

	First Trading Date	**Last Trading Date**
1	2019-05-17	2019-06-20
2	2019-06-21	2019-07-18
3	2019-07-19	2019-08-15
4	2019-08-16	2019-09-19
5	2019-09-20	2019-10-17
6	2019-10-18	2019-11-21
7	2019-11-22	2019-12-19
8	2019-12-20	2020-01-16
9	2020-01-17	2020-02-20
10	2020-02-21	2020-03-19
11	2020-03-20	2020-04-16
12	2020-04-17	2020-05-21

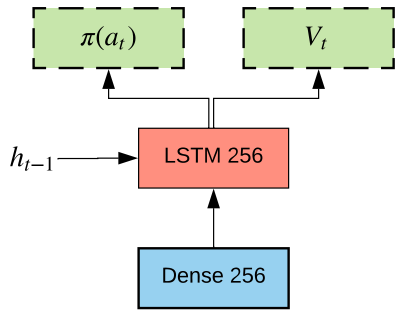

Fig. 3 Model architecture for policy and value networks

Actor-critic based algorithms use policy and value networks. We can use policy network and value network separately, or combined these two networks. We choose the combined architecture due to improvement of computational efficiency. Furthermore, we use LSTM [20] beside dense layer in the shared network.

As suggested in Fig. 3, we use a shallow network architecture for this study. In detail, in term of shared network, a trajectory is feed to the first fully-connected hidden layer with 256 units and apply *tanh* activation function. The next hidden layer is stateful LSTM with 256 unit. The tanh function is also applied to the LSTM layer. Finally, policy and value heads are fully-connected linear layer for single output of each action and state value respectively.

4.3 Result

We use same network architecture and hyper-parameters all trading agent without tweaking. The agent learn to decide to long buy or short sell assets on its own. For instance, agent can cut loss or hold positions overnight without any constraint.

Input : Selected valid action $\pi(a)$ of learned agent π, Transaction fee F, Price of asset m, Asset p in portfolio P, Total time step T
Output: Portfolio value PV

```
for t=1 to t=T do
    for p in P do
        if π(a_t) ← 0 then
            Hold position;
        else
            if π(a_t) ← 1 then
                Trade return r_p ← Long buy at m_t;
            else
                Trade return r_p ← Short sell at m_t;
            end
            r_p -= F;
            PV += r_p;
        end
    end
end
```

Algorithm 1: Simulation of stock market environment

Our experiment use discrete actions. For equity, trading agent can hold, buy, or sell stocks without considering amount of volume. Stocks are only sold after T+2 settlement. Likewise, derivatives trading agent can hold, long, or short futures contracts. However, the agent can trade continuously as it is T+0 settlement market. We include transaction fees for every trade return (see Algorithm. 1). For every agent, reward can be 1 if overall profit of an episode is positive and -1 in case of negative profit. In addition, we discount reward for every long position of agent in future market to encourage hedging.

Finally, we use RLLib [24] with 32 workers to train the agent. RMSProp algorithm is used as optimizer for training.

4.3.1 Buy and Hold Strategy Baseline

We compare our proposed trading strategy result with performance of buy and hold strategy to determine effectiveness of the approach. During the evaluation periods, the stock market is highly volatile due to impact of COVID 19 pandemic. Buy and hold strategy may lead to negative return (see Table. 2),

4.3.2 Multi-Agent Reinforcement Learning

Learned deep RL agent was deployed to trade out-of-sample market data from 17 May 2019 to 21 May 2020. The result shows that the learned agent can protect portfolio by short selling in futures market. In addition, in some cases, our agent can cut loss and achieve higher performance than market return in equity market even market plunged as traders had panic-sold out of COVID-19 pandemic fear (see Table. 3).

Table 2 Performance of Market in Percentage

	FPT	**GAS**	**SSI**	**MSN**	**NVL**	**VN30F1M**
Period 1	-0.15	-7.41	-3.49	-6.04	-1.88	-4.14
Period 2	4.79	5.01	2.82	-4.36	5.4	2.22
Period 3	12.54	-9.09	-15.86	-4.82	-0.83	-0.8
Period 4	9.06	6.43	1.16	7.72	5.83	4.95
Period 5	1.42	-1.65	3.51	-3.83	-2.36	1.32
Period 6	-1.05	2.67	-3.02	-6.17	-5.81	-0.65
Period 7	-2.29	-7.69	-11.96	-23.29	-5.65	-6.1
Period 8	2.7	-2.08	0	0.89	-0.18	2.78
Period 9	-1.58	-6.38	-3.81	-9.91	-1.45	-2.14
Period 10	-15.15	-36.36	-24.3	-4.72	-5.9	-20.99
Period 11	4.41	18.93	14	22.89	1.37	4.53
Period 12	15.23	11.86	9.47	7.05	2.9	12.85

Table 3 Performance of Trained RL Agent in Percentage

	FPT	**GAS**	**SSI**	**MSN**	**NVL**	**VN30F1M**	**Portfolio**
Period 1	2.5	-3.48	-2.99	-5.33	-2.67	6.54	2.07
Period 2	1.06	3.02	1.35	-3.76	2.89	1.65	1.28
Period 3	11.47	-2.01	-11.91	-1.11	-1.61	1.9	0.43
Period 4	5.11	0.06	2.44	6.84	5.32	-1.35	1.3
Period 5	-1.66	-5.18	-0.16	0.29	-2.85	4.38	1.23
Period 6	-0.92	2.66	-0.2	0.6	-4.04	1.63	0.06
Period 7	-0.72	-0.9	-6.72	-10.38	-0.95	6.33	1.2
Period 8	0.85	-2.03	-0.65	6.99	5.32	0.74	1.42
Period 9	2.16	1.79	-1.98	-9.91	-0.75	2.93	0.6
Period 10	-12.37	-21.19	-2.83	-0.34	-3.99	19.17	5.53
Period 11	11.03	11.19	6.45	17.61	-0.84	-0.74	4.17
Period 12	4.35	3.82	11.83	2.63	0.36	11.2	7.89

Specifically, VN30 Index was lost about 300 points (33%) during the first three months (Period 9, Period 10, Period 11) of 2020 (see Fig. 1). In term of equity trading, every stock in portfolio had negative market return in the periods. In equity market, after transaction and commission fees, the RL agent can not maintain positive return. In contrast, our agent executed many orders for opening and closing position to hedge equity assets dynamically in futures market. It leads to positive return of the portfolio (see Figure 4). The portfolio profit did not decrease when the market rebounded due to dynamic hedge strategy. However, in some cases, the agent can not achieve higher performance than return of market. As a result, we show that our method can reduce

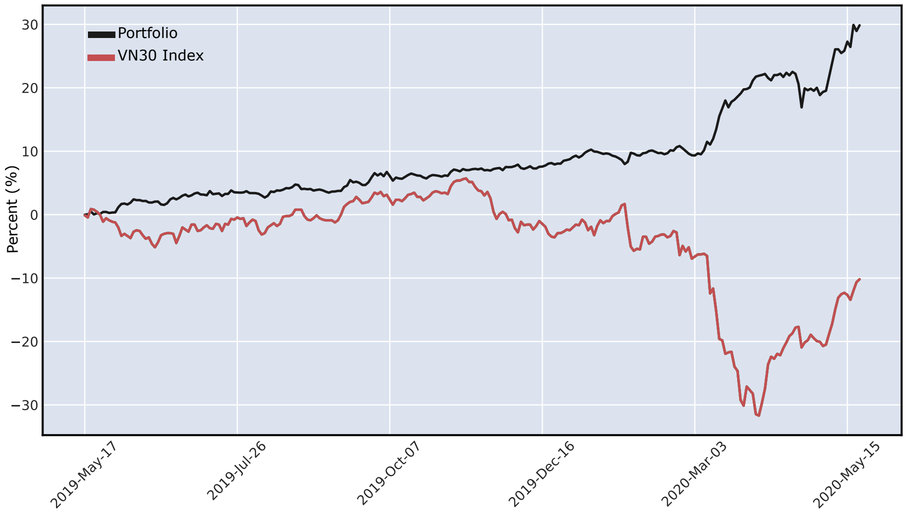

Fig. 4 Cumulative return of portfolio and VN30 Index in percentage from 17 May 2019 to 21 May 2020

losses and achieve positive profit in trading. Furthermore, the trading data also suggests that dynamic hedging strategy for equity in portfolio is feasible in cross-hedging case. The futures trading agent generated far profit than losses. Overall, during evaluation periods, our deep RL agent earned about 30% profit of portfolio value and maintain positive return in case market collapsed systematically.

5 Conclusion

This study proposed a feasible approach for cross-hedging in trading without domain knowledge by applying deep reinforcement learning. Our result also suggests that the approach can cut loss efficiently when market is in selling panic as happening in Covid 19 event. Overall, the proposed method can generate positive profit with dynamic hedge strategy. The result is desirable as our approach earns higher performance than the risk-free rate [19].

It is important to develop a deterministic behavior of agent to maintain reliable outcome. In future work, we should further study stability and safety in reinforcement learning for trading.

References

[1] Carlo Acerbi and Dirk Tasche. "Expected shortfall: a natural coherent alternative to value at risk". In: *Economic notes* 31.2 (2002), pp. 379–388.

[2] Siddharth Alexander, Thomas F Coleman, and Yuying Li. "Derivative portfolio hedging based on CVaR". In: *New Risk Measures in Investment and Regulation: Wiley* (2003).

[3] Philippe Artzner et al. *Coherent measures of risk, mathematical Finance 9 (3): 203–228.* 1999.

[4] Ph Artzner et al. "Thinking coherently. RISK, 10". In: *November, 68* 71 (1997).

[5] Christopher Berner et al. "Dota 2 with Large Scale Deep Reinforcement Learning". In: *arXiv preprint arXiv:1912.06680* (2019).

[6] Stewart L Brown. "A reformulation of the portfolio model of hedging". In: *American Journal of Agricultural Economics* 67.3 (1985), pp. 508–512.

[7] Hans Buehler et al. "Deep hedging". In: *Quantitative Finance* 19.8 (2019), pp. 1271–1291.

[8] Hans Buehler et al. "Deep Hedging: Hedging Derivatives Under Generic Market Frictions Using Reinforcement Learning-Machine Learning Version". In: *Available at SSRN* (2019).

[9] Darren Butterworth and Phil Holmes. "The hedging effectiveness of stock index futures: evidence for the FTSE-100 and FTSE-Mid250 indexes traded in the UK". In: *Applied Financial Economics* 11.1 (2001), pp. 57–68.

[10] Zhiguang Cao, Richard DF Harris, and Jian Shen. "Hedging and value at risk: A semi-parametric approach". In: *Journal of Futures Markets: Futures, Options, and Other Derivative Products* 30.8 (2010), pp. 780–794.

[11] Louis H Ederington. "The hedging performance of the new futures markets". In: *The journal of finance* 34.1 (1979), pp. 157–170.

[12] Lasse Espeholt et al. "Importance weighted actor-learner architectures: Scalable distributed deep-rl with importance weighted actor-learner architectures". In: *arXiv preprint arXiv:1802.01561* (2018).

[13] Richard Evans and Jim Gao. "Deepmind AI reduces Google data centre cooling bill by 40%". In: *DeepMind blog* 20 (2016), p. 158.

[14] Stephen Figlewski. "Hedging performance and basis risk in stock index futures". In: *The Journal of Finance* 39.3 (1984), pp. 657–669.

[15] Christos Floros* and Dimitrios V Vougas. "Hedge ratios in Greek stock index futures market". In: *Applied Financial Economics* 14.15 (2004), pp. 1125–1136.

[16] Sumitra Ganesh et al. "Reinforcement Learning for Market Making in a Multi-agent Dealer Market". In: *arXiv preprint arXiv:1911.05892* (2019).

[17] Asim Ghosh. "Hedging with stock index futures: Estimation and forecasting with error correction model". In: *Journal of futures markets* 13.7 (1993), pp. 743–752.

[18] Igor Halperin. "Qlbs: Q-learner in the black-scholes (-merton) worlds". In: *Available at SSRN 3087076* (2017).

[19] GD Hancock and PD Weise. "Competing derivative equity instruments: empirical evidence on hedged portfolio performance". In: *The Journal of Futures Markets (1986-1998)* 14.4 (1994), p. 421.

[20] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[21] Markus Huggenberger, Peter Albrecht, and Alexandr Pekelis. "Tail risk hedging and regime switching". In: *Asian Finance Association (AsianFA) 2015 Conference Paper*. 2016.

[22] Vijay R Konda and John N Tsitsiklis. "Actor-critic algorithms". In: *Advances in neural information processing systems*. 2000, pp. 1008–1014.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[24] Eric Liang et al. "RLlib: Abstractions for distributed reinforcement learning". In: *arXiv preprint arXiv:1712.09381* (2017).

[25] Michael B Miller. *Quantitative Financial Risk Management*. John Wiley & Sons, 2018.

[26] Volodymyr Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[27] John Moody et al. "Performance functions and reinforcement learning for trading systems and portfolios". In: *Journal of Forecasting* 17.5-6 (1998), pp. 441–470.

[28] John Moody et al. "Reinforcement learning for trading systems and portfolios: Immediate vs future rewards". In: *Decision Technologies for Computational Finance*. Springer, 1998, pp. 129–140.

[29] Matthew O'Kelly et al. "Scalable end-to-end autonomous vehicle testing via rare-event simulation". In: *Advances in Neural Information Processing Systems*. 2018, pp. 9827–9838.

[30] David Silver et al. "Mastering the game of go without human knowledge". In: *nature* 550.7676 (2017), pp. 354–359.

[31] RS Sutton and AG Barto. *Reinforcement learning: An introduction*. 2nd. MIT press, 2018.

[32] Nikolas Topaloglou, Hercules Vladimirou, and Stavros A Zenios. "CVaR models with selective hedging for international asset allocation". In: *Journal of Banking & Finance* 26.7 (2002), pp. 1535–1561.

[33] Christopher John Cornish Hellaby Watkins. “Learning from delayed rewards”. In: (1989).

[34] Samuel Watts. *Hedging basis risk using reinforcement learning*. Tech. rep. Working Paper, University of Oxford, 2015.

[35] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. “Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms”. In: *arXiv preprint arXiv:1911.10635* (2019).

[36] Zihao Zhang, Stefan Zohren, and Roberts Stephen. “Deep Reinforcement Learning for Trading”. In: *The Journal of Financial Data Science* (2020).