A Review on Derivative Hedging Using Reinforcement Learning

Peng Liu

Peng Liu
is an assistant professor of quantitative finance (practice) at Singapore Management University in Singapore.
liupeng@smu.edu.sg

KEY FINDINGS

- Automated optimal hedging solutions using reinforcement learning (RL) have been gaining popularity in recent years because of its flexibility in handling real-world dynamics and restrictions, which are often ignored and difficult to adapt to using traditional approaches.
- Current RL-based option hedging agents are trained to adjust the size of the hedging portfolio using market conditions such as stock price and (optionally) closed-form inducing factors such as delta or vega based on the Black–Scholes model, with the reward function being constructed following problem-specific heuristics.
- Future research directions in RL-based derivative hedging include new training methodologies in model architecture and optimization procedure, a wider coverage of hedging targets and instruments, and different types of hedging strategies.

ABSTRACT

Hedging is a common trading activity to manage the risk of engaging in transactions that involve derivatives such as options. Perfect and timely hedging, however, is an impossible task in the real market that characterizes discrete-time transactions with costs. Recent years have witnessed reinforcement learning (RL) in formulating optimal hedging strategies. Specifically, different RL algorithms have been applied to learn the optimal offsetting position based on market conditions, offering an automatic risk management solution that proposes optimal hedging strategies while catering to both market dynamics and restrictions. In this article, the author provides a comprehensive review of the use of RL techniques in hedging derivatives. In addition to highlighting the main streams of research, the author provides potential research directions on this exciting and emerging field.

Reinforcement learning (RL) is a fast-growing research area with applications in many fields. RL could learn the optimal policy from a set of observable states and determine the optimal action to guide the sequential decision-making process. It is particularly useful in a model-free setting when the dynamics of the learning environment, including the transition probabilities among states and reward function, are unknown. Such a learning framework, backed by Bellman's principle of optimality, shows promise for solving many problems in the realm of finance such as portfolio management and hedging. For example, Halperin (2017) first introduced the Q-Learning Black–Scholes (QLBS) model that applies the Q-learning method to solve for the optimal option hedging strategy in a dynamically replicating portfolio of cash and stock. When the replicating portfolio continuously rebalances the wealth allocation between cash and stock, it could hypothetically match the exact value of the option at all times.

The vanilla European option is probably the most widely studied product when it comes to derivative hedging using RL. The traditional approach relies on the delta hedging strategy based on the option pricing model from Black and Scholes (1973) and Merton (1973), jointly referred to as the BSM model, in which the optimal offsetting position can be derived by differentiating the option price with respect to the underlying stock price, assuming a frictionless market and continuous-time setting with constant volatility. The hedge portfolio is then dynamically rebalanced between cash and stock in a self-financing manner, which ends up generating the same payoff as the derivative. In practice, however, such rebalancing only happens on a periodic basis such as daily, making it a discrete-time exercise and therefore leading to tracking error as the underlying stock moves in value; it is impossible to achieve exact replication at all times. In addition, the transaction cost, first considered in option pricing by Leland (1985), inevitably enlarges the tracking error because of frequent trading of stock. An optimal hedging strategy will thus need to balance between minimizing the mis-hedging risk (caused by imperfect replication) and reducing the transaction cost (because of stock trading), where the relative weight is determined by the risk aversion parameter of the hedger. Indeed, underhedge or overhedge is preferred compared with exact delta neutrality when the transaction cost is nontrivial.

When the system under study becomes complex, deriving the analytic optional solution in closed form is more challenging, as in the case of applying the theoretical delta hedging in a realistic market setting. Specifically, factors such as transaction cost and volatility risk, which are not considered in the original BSM model, will increase the mis-hedging risk of the replicating portfolio. A fully end-to-end approach is thus desired, in which one can automatically learn an optimal hedging strategy from the real market conditions and continuously adjust its strategy based on the latest change in these conditions. To this end, Ritter and Kolm (2019) developed the first fully automated RL solution that optimally hedges an option with transaction cost. By formulating trading cost and hedging variance as the feedback to the RL framework, the learning agent could be designed to optimize over the hedging objective while considering the system constraints such as transaction cost. The RL agent, when properly trained, will then guide the sequential decision-making process to rebalance the hedge position based on the current set of market indicators, such as stock price, so as to minimize the mis-hedging risk.

The use of RL techniques in hedging derivatives is a promising area of research. On one hand, the RL community witnesses new methodological innovations frequently, offering more powerful function approximators to optimal hedging strategies. On the other hand, there are many areas yet to be explored when using the fully automated RL solution to hedge derivatives, for example, using RL to hedge other types of product (nonvanilla options) and properties (Greeks other than delta), learn different types of hedging strategies (static, dynamic, or both), and so on. In the rest of the article, we will first cover the fundamentals of derivative hedging using RL, followed by a thorough literature review of the recent advances in this growing area.

FUNDAMENTALS OF REINFORCEMENT LEARNING

RL is a reward-driven approach that aims at guiding a sequential decision-making process under uncertainty. In this section, we first review the basics of RL, including the state and action value function, followed by an in-depth review of a typical environment setting for option hedging, including the choice of the state, action, and reward functions.

RL Basics

RL with a Markov decision process is a sequential learning framework that aims at training an optimal policy to propose an action $a_t \in A$ based on a set of observable state variables in $s_t \in S$ at time step t. Each action will incur a corresponding reward r_t from the environment and transition into the next step s_{t+1}, which completes a tuple record of (s_t, a_t, r_t, s_{t+1}) for either training or evaluation. The optimal policy $\pi: A \rightarrow S$ is the one that maximizes the cumulative long-term return G_t, a utility function defined as the sum of immediate reward r_t and all future discounted rewards until terminal time step T:

$$G_t = \sum_{i=0}^{T-t} \gamma^i r_{t+i+1}$$

where $\gamma \in [0,1]$ is a scalar value that trades off between short-term and long-term rewards and T denotes the terminal time step of the learning system, such as expiration of an option. The decision theory on expected utility then defines an optimal agent as the one that maximizes $\mathbb{E}_\pi[G_t]$ following a policy π, which manifests in both state value function $V_\pi(s_t)$ and action value function $Q_\pi(s_t, a_t)$:

$$\begin{aligned} V_\pi(s_t) &= \mathbb{E}_\pi[G_t | s_t] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma V_\pi(s') \,|\, s_t] \\ &= \sum_a \pi(a|s) \sum_{s',r} P(s',r|s,a)[r + \gamma V_\pi(s')], \forall s \in S \end{aligned}$$

$$\begin{aligned} Q_\pi(s_t, a_t) &= \mathbb{E}_\pi[G_t | s_t, a_t] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma Q_\pi(s', a') | s_t, a_t] \\ &= \sum_{s',r} P(s',r|s,a)[r + \gamma V_\pi(s')], \forall s \in S, \forall a \in A \end{aligned}$$

Based on Bellman's principle of optimality, an optimal action is determined based on the assumption that all future actions also are optimal, which leads to the following Bellman optimality equations for optimal state function $V_*(s)$ and action value function $Q_*(s, a)$:

$$\begin{aligned} V_*(s) &= \max_\pi V_\pi(s) \\ &= \max_a Q_*(s,a) \\ &= \max_a \sum_{s',r} P(s',r|s,a)[r + \gamma V_*(s')] \end{aligned}$$

$$\begin{aligned} Q_*(s,a) &= \max_a Q(s,a) \\ &= \sum_{s',r} P(s',r|s,a)[r + \gamma V_*(s')] \\ &= \sum_{s',r} P(s',r|s,a)[r + \gamma \max_{a'} Q_*(s',a')] \end{aligned}$$

where s' and a' denote the next state and action, respectively. Here, an optimal policy π^* has the maximum state and action function values for any $s \in S$ and $a \in A$ at any time step t:

$$\begin{aligned} V_*(s) &\geq V_\pi(s) \\ Q_*(s,a) &\geq Q_\pi(s,a) \end{aligned}$$

Because the environment dynamics are unknown, function approximation via neural networks is often adopted when training an optimal policy. In the next section, we will review the basic setup that allows us to adjust the RL framework to dynamic delta hedging. See Sutton and Barto (2018) for a comprehensive treatment of RL.

RL Environment for Delta Hedging

The success of an RL agent heavily relies on the proper setup of a learning environment in the learning framework. In this case, a common setup is to simulate paths that represent stock price movements and use the discrete-time BSM model as a benchmark to measure the quality of an RL-based policy. Assume the agent sells one European option at $t = 0$ and wishes to hedge this position using two assets in the hedge portfolio, stock and cash, where the latter represents a riskless component that can be saved in a bank account to grow interest. The stock price S_t is assumed to be a log normally distributed random variable and follows a geometric Brownian motion:

$$\delta S_t = \mu S_t \delta t + \sigma S_t \delta W_t$$

where μ and σ are constants that denote the percentage drift and percentage volatility, respectively, and W_t is a Wiener process. The transaction cost associated with change in stock position $\delta N_t = N_{t+1} - N_t$ is denoted by $f(S_t, \delta N_t)$.

Under the self-financing constraint, all hedging operations on the stock position are supported by the cash account, and there is no external cash injected or withdrawn from the hedge portfolio since initiation. This gives the following remaining cash balance B_{t+1} after changing the stock position from N_t to N_{t+1}:

$$B_{t+1} = B_t e^{\rho \delta t} - (N_{t+1} - N_t) S_t - f(S_t, \delta N_t)$$

where ρ denotes the fixed risk-free interest rate. The change in the hedge portfolio before and after changing the number of stocks on hand can be characterized as follows. Representing the cash amount, stock price, and number of stocks as a tuple, the state at the start of time step t can be expressed as $s_t = (B_t, S_t, N_t)$, which becomes $s_t = (B_t e^{\rho\delta t}, S_t, N_t)$ at the end of time step t. When entering the next time step, the next state becomes $s_{t+1} = (B_{t+1}, S_{t+1}, N_{t+1})$, with B_{t+1} calculated based on the aforementioned closed-form expression.

Because the option price is available under the continuous-time BSM model, a common practice is to include the option price C_t, its delta Δ_t, or both as part of the state, as in the case of Halperin (2017) and Ritter and Kolm (2019). Specifically, the state variable is expressed as $s_t = (t, W_t, N_t, C_t, \Delta_t)$, where W_t is used to calculate the underlying stock price $S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}$ and $\Delta_t = \frac{\partial C_t}{\partial S_t}$ is the option delta at discrete time step t available in closed-form via the BSM formula. The option delta is also the action from the optimal policy π^*, giving $a_t^* = \pi^*(s_t) = \Delta_{t+1}$. A general policy π determines the number of shares of the stock N_{t+1} based on the state s_t, which leads to $a_t = \pi(s_t) = N_{t+1}$.

Note that the cash account B_t is ignored in the state because it can be uniquely derived by the information contained within the state. We also note that the state variable includes the option price C_t and delta Δ_t, which constitute the right answers as part of the state for the next action. However, this self-fulfilling property may not always stand in practice, especially when the closed-form solutions are not available. Learning the optimal policy without these hints in the state is a more challenging problem and requires further research.

When using a dynamic replicating portfolio for hedging, the portfolio value is expected to match exactly the option price at each time step. In other words, we can use the following net portfolio value to represent the mis-hedging risk:

$$\Pi_t = N_t S_t + B_t e^{\rho\delta t} - C_t$$

The maximum net portfolio value is thus zero, which indicates perfect hedge. When the agent makes an action to adjust the number of shares of the underlying stock, the immediate reward can be defined as the difference between the net portfolio value Π_t and Π_{t+1}:

$$\begin{aligned}
\delta\Pi_t &= \Pi_{t+1} - \Pi_t \\
&= [N_{t+1}S_{t+1} + B_{t+1} - C_{t+1}] - [N_t S_t + B_t e^{\rho\delta t} - C_t] \\
&= (N_{t+1}S_{t+1} - N_t S_t) - \{(N_{t+1} - N_t)S_t - f(S_t, \delta N_t)\}e^{\rho(T-t)} - (C_{t+1} - C_t)
\end{aligned}$$

where we plug in the definition of B_{t+1} at the start of $t + 1$ and account for its future value based on the remaining maturity. Note that this also is referred to as the accounting profit and loss (P&L) of the hedging portfolio, where both the changes in the replicating portfolio and the option price are compared to provide immediate feedback on the quality of the hedge in the presence of the transaction cost. Setting up such immediate reward ensures faster convergence of an RL-based learning algorithm, as used in both Cao et al. (2021) and Cao et al. (2023).

In addition, the variance of the hedging risks also is considered in Ritter and Kolm (2019), following the mean variance optimization framework. Specifically, we can express the variance of the hedging difference as follows:

$$\begin{aligned}
\mathrm{Var}[\delta\Pi_t] &= \mathrm{Var}[(N_{t+1}S_{t+1} - N_t S_t) - \{(N_{t+1} - N_t)S_t - f(S_t, \delta N_t)\} - (C_{t+1} - C_t)] \\
&= \mathrm{Var}[N_{t+1}\delta S_t - \delta C_t] \\
&= \mathrm{Var}[N_{t+1}\delta S_t - \frac{\partial C_t}{\partial S_t}\delta S_t] \\
&= \mathrm{Var}[(N_{t+1} - \Delta_t)(\mu S_t \delta_t + \sigma S_t \delta W_t)] \\
&= [\sigma S_t (N_{t+1} - \Delta_t)]^2 \delta_t
\end{aligned}$$

where we ignored the accrued interest because of single-stage variance and the transaction cost, which is mainly considered in the hedging difference. We also used first-order Taylor expansion for option price in the derivation, where $C_{t+1} \approx C_t + \frac{\partial C_t}{\partial S_t}\delta S_t$.

The mean and variance could then be combined as a weighted sum into a single objective function as the discounted immediate reward r_t from the learning environment:

$$r_t = \gamma^{-t}\left[\mathbb{E}[\delta\Pi_t] - \frac{\lambda}{2}\mathrm{Var}[\delta\Pi_t]\right]$$

where γ is the discount factor for future reward and often is set to 0.99, and λ is a hyper-parameter that balances off between tracking error and hedging variance. Note that when $t = T$, the terminal reward is given by

$$r_T = \gamma^{-T}[C_0 e^{\rho T} - C_0]$$

where we assume an income of C_0 from selling the option at the beginning and start with zero shares of stock on hand. Here, the hedge position is closed upon expiration by setting $N_T = 0$, and the stocks bought/sold are converted to the cash account, making $\Pi_T = B_T$.

Because hedging is a sequential decision-making process that requires an action from the agent at each discrete time step until option expiration, the overall quality of a given policy π can be measured as a weighted sum of individual rewards, that is, the long-term return at $t = 0$ can be calculated as

$$R_0 = \sum_{t=0}^{T} \gamma^t r_t$$

LITERATURE REVIEW

In this section, we highlight the main research streams in the literature.

Halperin (2017) first applied Q-learning to option hedging in the Black–Scholes world, referred to as QLBS, with finite state and action spaces. By minimizing the terminal variance of the hedge portfolio based on the Markowitz portfolio theory, the QLBS algorithm can obtain a separate semi-analytic solution for each derivative position based on a careful choice of basis functions used to approximate the value function. Note that both the state and action spaces are discrete and the market is assumed to be frictionless, that is, transaction cost is not considered. The state space on the stock price is further smoothed in Halperin (2019), in which the author compared QLBS with a model-based solution using dynamic programming (DP) and an analytic solution using the BSM model. The author also introduced an inverse RL setting for learning the reward function based on observed state and action variables.

Buehler et al. (2019) introduced a neural network as the function approximator under convex risk measures, which also include proportionate transaction cost, in their deep hedging approach for hedging over-the-counter derivatives. The optimal hedging strategy is then obtained by considering both portfolio cash flow and transaction cost.

Ritter and Kolm (2019) proposed an RL-based solution to learn automatically the optimal hedging strategy in a realistic setting, given that the pricing of the derivative is available in closed form or via Monte Carlo simulation. An autonomous agent was developed to adjust sequentially the stock position to minimize deviation from the optimal hedge while minimizing the transaction cost. Using experiments based on simulations of multiple paths of stock price, the authors show that the RL-based solution is comparable to the delta hedging strategy in terms of tracking variance but produces a much lower cost. Specifically, a continuous state space $s_t = (t, S_t, N_t, C_t)$ is used in the experiment, where S_t is the stock price at t and the option delta Δ_t is ignored to increase the level of difficulty for the learning task. The action space is set to be discrete and bounded by the maximum number of shares of stock needed for option hedging, and the cost is proportionate to the change in stock position. The single-stage reward function $r_t = \delta\Pi_t - \frac{\lambda}{2}\mathrm{Var}[\gamma\Pi_t]$ is defined in a similar vein, where $\delta\Pi_t = q_t - c_t$ is the hedge cost q_t (treated as a random walk term) subtracted by the nonlinear transaction cost c_t (including commissions, bid-offer spread cost, market impact cost, and other sources of slippage). The authors applied a Q-learning algorithm with an ϵ-greedy policy to train the RL agent, using SARSA as the training targets, and performed in a batch mode. Here, SARSA is an on-policy algorithm that learns the Q-value based on the current policy, as opposed to the Q-learning algorithm that learns the Q-value in a greedy and off-policy fashion. The authors also provide a more comprehensive overview of portfolio risk management using modern RL-based methods in Kolm and Ritter (2019).

To further extend the power of state-of-the-art deep RL algorithms, Du et al. (2020) introduced a deep Q-learning network (DQN), DQN with Pop-Art, and proximal policy optimization (PPO) to approximate the action function or the policy directly. In their implementation, the state variable is configured as $s_t = (t, S_t, N_t, K)$, where K denotes the strike price and the action space remains a discrete one. Incorporating the strike price allows the agent to learn and propose hedging strategies for options with different strike prices in one shot. In addition, the option price C_t and its delta Δ_t, which are removed from the state vector, are automatically learned by these function approximators, given enough training budget. Using the same mean–variance-based risk-adjusted reward function in the presence of nonlinear transaction costs, Ritter and Kolm (2019) show that PPO outperforms other algorithms and the baseline delta hedging strategy in terms of delta neutrality, training time, and the amount of training data required.

Treating the action space as discrete inevitably introduces approximation error in the Q-learning–based algorithms. Indeed, rounding off the hedge position fails to differentiate numerically close action values. To eliminate the discretization error and therefore adopt a continuous action space, Cao et al. (2021) introduced the deep deterministic policy gradient (DDPG) method, which allows both the state space and action space to be continuous and offer better numerical results compared with other RL architectures. The authors also introduced a new architecture that decouples the original composite reward into two dedicated Q value targets to encourage better differentiation: the expectation and standard deviation of the hedge cost, where the optimal combination is learned by RL. In this case, the reward function becomes $r_t = \mathbb{E}(\Pi_t) - \lambda\sqrt{\mathbb{E}(\Pi_t^2) - \mathbb{E}^2(\Pi_t)}$, where the volatility of the hedge portfolio is used. When considering transaction cost, the authors showed superior performance of DDPG-based RL over using the practitioner delta (delta hedging with fixed volatility) and Bartlett delta (with stochastic volatility) when the stock price follows a geometric Brownian motion (Black and Scholes 1973) and stochastic volatility process (Hagan et al. 2002), respectively.

In addition, the authors also argued in favor of the accounting P&L over the cash flow approach, which aligns with our choice of reward function from the previous section. Such choice is sensible in other RL settings as well. Compared with a delayed reward received upon completion of an episode using the cash flow approach, which gives rise to the temporal credit assignment problem, the P&L reward received at every step provides immediate feedback for the quality of the current action, thus facilitating faster learning. The experiments use $s_t = (t, S_t, N_t)$ as the continuous state variable and $a_t = \pi(s_t)$ as the action variable that is continuous in the policy-gradient algorithm and discrete in the Q-learning algorithm.

Building on formulating new and better objective functions, Gu (2022) introduced the function property term as a regularizer in the reward function, where the regularization serves to induce specific bias in the model estimation process according to domain-specific knowledge. Specifically, the reward function becomes $r_t = \delta\Pi_t - \lambda_1 \text{Var}[\delta\Pi_t] - \lambda_2 f_{mr}$, where f_{mr} is an indicator function that assumes the value of one when a prescribed function property (in this case, mean reversion) is violated, and λ_1 and λ_2 are hyperparameters to be manually fine-tuned. For example, when holding a long European call with a current stock price S, strike price K, and time to maturity T, the indicator function on option price C becomes $f_{mr} = \mathbb{I}_{\{C<0\}\cup\{C<S-Ke^{-rT}\}}$. In the experiments, the state variable is defined as $s_t = (t, S_{t-1}, S_t, N_t)$, where S_{t-1} denotes the stock price at the previous time step and is included to facilitate learning of the underlying price dynamics. The action a_t is continuous when using an off-the-shelf PPO algorithm.

The list of hedging instruments can be expanded further to include derivatives such as forwards, swaps, futures, and options. For example, Buehler, Murray, and

Wood (2022) introduced a deep Bellman hedging framework that provides RL-based hedging in a realistic setting: continuous state and action spaces, risk-adjusted reward function, and market frictions such as transaction costs and liquidity constraints. The reward function is defined as a sum of change in book values (mark-to-market price of the financial instrument to be hedged), cash flow generated because of change in the position of hedging derivatives, and the cost of hedging. The authors provided a theoretical justification of the existence of a unique finite solution for the Bellman equation of the value function based on this reward function and proposed an actor-critic architecture for further experiments.

Hedging can go beyond delta and cover more Greek letters such as gamma, the second-order derivative of option price to change the underlying stock price. For example, Cao et al. (2023) used a finite differencing approach to approximate the second order term $\frac{\partial^2 C_t}{\partial S_t^2}(\delta S_{t+1})^2$ with $(N_{t+1} - N_t)\delta S_{t+1}$, when approximating the option price C_{t+1} at time step $t + 1$:

$$C_{t+1} \approx C_t + \frac{\partial C_t}{\partial S_t}\delta S_{t+1} + \frac{1}{2}\frac{\partial^2 C_t}{\partial S_t^2}(\delta S_{t+1})^2$$

The authors then added the square of this gamma-like term as part of the reward function to reduce the sensitivity of N_{t+1} to changes in the asset price, in addition to maximizing the account profits of the hedge portfolio. We note that this has an effect similar to penalizing the variance of the hedge portfolio, covered in the previous section. Also, the authors propose a more comprehensive state space $s_t = (t, S_t, N_t, B_t, C_t, K, \Delta_t, \gamma_t, \upsilon_t)$, where common Greek letters Δ_t, γ_t, and υ_t are also included. Using a continuous action space for N_t and considering a linear transaction cost, the authors used the state-of-the-art DDPG method to train RL agents and demonstrated superior performance on various option datasets.

To further expand the scope of RL-based hedging strategies, the same authors proposed to hedge the gamma (the second partial derivative of portfolio value with respect to the underlying asset price) and vega (the partial derivative of portfolio value with respect to the volatility of the underlying asset) using option as the hedging instrument (Cao et al. 2023). Compared with delta hedging using stock, hedging the gamma and vega is often much more costly and requires the use of other derivatives such as option in the hedging portfolio. Such extension shifts from the mainstream focus on stock-based hedging strategy in the literature and highlights an avenue toward hedging the portfolio's exposure to additional properties such as large movements in asset price (big gamma) and large changes in its volatility (large vega).

The authors also harnessed the latest development in the RL community and employed a distributional neural network architecture that extends the mean–variance risk-aware reward function to a full distribution, allowing for the formulation of alternative risk measures such as value-at-risk and conditional value-at-risk. As shown in the numerical experiments, having access to the full distribution of Q values at different quantiles provides better hedging quality at different levels of transaction cost and maturity. In terms of the environment setting, the authors choose $s_t = (t, S_t, \gamma_t, \upsilon_t, \gamma_t^*, \upsilon_t^*)$, where γ_t^* and υ_t^* denote the gamma and vega of the at-the-money option used for hedging, respectively. The action is the proportion of maximum allowable hedging in the range of $[0, 1]$, which is clipped to prevent an arbitrary position in the hedging option during training. The reward function measures the same accounting P&L as before, in which the change in portfolio includes both the option being hedged and options used for hedging.

FUTURE RESEARCH DIRECTION

As shown in the previous section on literature review, the past few years have witnessed exciting developments in building autonomous strategies for derivative hedging, which is mostly sparked by the advances in the RL community. In this section, we highlight a few research directions in derivative hedging using RL.

Adopting New Training Methodologies in Deep RL for Derivative Hedging

The methodological innovation in the model architecture and optimization procedure, which is a research hot spot in the RL community, also could extend to the specialized domain of derivative hedging. Since the adoption of basis-function–based Q-learning (Halperin 2017) and DQN (Du et al. 2020), which are considered to be basic RL training procedures, more advanced model architectures, such as DDPG (Cao et al. 2021) and distributional RL architecture (Cao et al. 2023) have been proposed. In this respect, we foresee that more advanced, domain-specific but not necessarily complex RL model architectures deserve further research.

Exploring More Hedging Targets and Instruments

The most widely used example for RL-based option hedging is to achieve delta neutrality by adjusting the stock position, such as (Kolm and Ritter 2019) and others. Other Greek letters also can be set as the hedging target, also using option as the hedging instrument as shown in Cao et al. (2023). Note that all existing work focus on hedging an European option, while the more complex case of hedging an American option has not been studied. Therefore, we believe a more exciting research direction is to expand the hedging space (both the target derivative to be hedged and the instruments used for hedging) and spread the use of RL-based autonomous hedging agent to more real-life trading scenarios.

Expanding the Type of Hedging Strategies

Option hedging is a dynamic exercise and requires constant rebalancing; while forward hedging is a static one, the optimal action is to perform the hedge at day one. In other words, the sooner the RL agent realizes the superiority of static hedge over dynamic hedge, the better the hedging quality in the case of forward hedging. Recognizing different types of hedging strategies gives the RL agent a unique advantage in tackling more complex challenges, such as hedging a binary option.

CONCLUSION

In this article, we provided a comprehensive review of the current state of building an autonomous derivative hedging agent using RL. We first illustrated the fundamental framework of option hedging using RL, including the setting of state, action, and reward functions, followed by a detailed review of major research papers in this emerging and exciting field. We then provided a few directions for future research, focusing on the RL methodology, hedging targets, instruments, and strategies. We hope this article provides a quick and essential overview on the current development of this field and sparks more interest in adopting RL in derivative hedging.

ACKNOWLEDGMENTS

I thank Tee Chyng Wen for the helpful discussion on derivative hedging.

REFERENCES

Black, F., and M. Scholes. 1973. “The Pricing of Options and Corporate Liabilities.” *Journal of Political Economy* 81 (3): 637–654.

Buehler, H., L. Gonon, J. Teichmann, and B. Wood. 2019. “Deep Hedging.” *Quantitative Finance* 19: 1–21.

Buehler, H., P. Murray, and B. Wood. 2022. “Deep Bellman Hedging.” *arXiv* 2207.00932.

Cao, J., J. Chen, S. Farghadani, J. Hull, Z. Poulos, Z. Wang, and J. Yuan. 2023. “Gamma and Vega Hedging using Deep Distributional Reinforcement Learning.” *Frontiers in Artificial Intelligence* 6: 1129370.

Cao, J., J. Chen, J. Hull, and Z. Poulos. 2021. “Deep Hedging of Derivatives Using Reinforcement Learning.” *The Journal of Financial Data Science* 3 (1): 10–27.

Du, J., M. Jin, P. Kolm, G. Ritter, Y. Wang, and B. Zhang. 2020. “Deep Reinforcement Learning for Option Replication and Hedging.” *The Journal of Financial Data Science* 2: 44–57.

Gu, S. 2022. “Deep Reinforcement Learning with Function Properties in Mean Reversion Strategies.” *The Journal of Financial Data Science* 4 (3): 54–65.

Hagan, P., D. Kumar, A. Lesniewski, and D. Woodward. 2002. “Managing Smile Risk.” *Wilmott Magazine* 1: 84–108.

Halperin, I. 2017. QLBS: Q-Learner in the Black–Scholes (–Merton) Worlds. *arXiv* 1712.04609.

——. 2019. “The QLBS Q-Learner Goes Nuqlear: Fitted Q Iteration, Inverse RL, and Option Portfolios.” *Quantitative Finance* 19: 1–11.

Kolm, P., and G. Ritter. 2019. “Modern Perspectives on Reinforcement Learning in Finance.” *SSRN* 3449401.

Leland, H. E. 1985. “Option Pricing and Replication with Transaction Costs.” *The Journal of Finance* 40: 1283–1301.

Merton, R. 1973. “The Theory of Rational Option Pricing.” *The Bell Journal of Economics and Management Science* 4: 141–183.

Ritter, G., and P. Kolm. 2019. “Dynamic Replication and Hedging: A Reinforcement Learning Approach.” *The Journal of Financial Data Science* 1 (1): 159–171.

Sutton, R. S., and A. G. Barto. *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, Massachusetts, United States: MIT Press, 2018.