Deep Hedging: Hedging Derivatives Under Generic Market Frictions Using Reinforcement Learning

Hans Buehler[*] **Lukas Gonon**[†] **Josef Teichmann**[†] **Ben Wood**[*] **Baranidharan Mohan**[*]

Jonathan Kochems[*]

Abstract

This article discusses a new application of reinforcement learning: to the problem of hedging a portfolio of "over-the-counter" derivatives under under market frictions such as trading costs and liquidity constraints. It is an extended version of our recent work [3], here using notation more common in the machine learning literature.

The objective is to maximize a non-linear risk-adjusted return function by trading in liquid hedging instruments such as equities or listed options.

The approach presented here is the first efficient and model-independent algorithm which can be used for such problems at scale.

1 Introduction

Modern quantitative finance has developed a rich toolkit for handling derivative pricing and risk management under the idealized "complete markets" assumption of perfect hedgability in the absence of any trading restrictions or cost. It has not yet succeeded in providing a scalable industrial approach under more realistic conditions which take into account such market frictions. As a consequence, the practical risk management of non-electronic over-the-counter[3] derivatives is still to a large extent manual, driven by the trader's intuitive understanding of the shortcomings of the existing derivative tools.

In this article, we take a first step towards a more integrated, realistic and robust approach to automated derivative risk management by applying modern deep reinforcement learning policy search. In the context of derivatives risk management, a policy means a hedging strategy. We propose to use neural networks to represent our hedging strategies.

The networks are trained on simulations of future states of the market, including all relevant hedging instruments. Designing a market simulator is not the focus of this paper, and we use toy simulators

[*]J.P. Morgan, London, UK *Disclaimer: Opinions and estimates constitute our judgement as of the date of this Material, are for informational purposes only and are subject to change without notice. It is not a research report and is not intended as such. Past performance is not indicative of future results. This Material is not the product of J.P. Morgan's Research Department and therefore, has not been prepared in accordance with legal requirements to promote the independence of research, including but not limited to, the prohibition on the dealing ahead of the dissemination of investment research. This Material is not intended as research, a recommendation, advice, offer or solicitation for the purchase or sale of any financial product or service, or to be used in any way for evaluating the merits of participating in any transaction. Please consult your own advisors regarding legal, tax, accounting or any other aspects including suitability implications for your particular circumstances. J.P. Morgan disclaims any responsibility or liability whatsoever for the quality, accuracy or completeness of the information herein, and for any reliance on, or use of this material in any way. Important disclosures at: www.jpmorgan.com/disclosures*

[†]Department of Mathematics, Eidgenössische Technische Hochschule, Zürich, Switzerland

[3]"Over-the-counter" products are traded directly between counterparties. We use the term to refer to products that are not liquidly traded on an exchange, and hence that do not have observable prices.

Preprint. Work in progress.

for the experiments presented here; an advantage of our approach is that the procedure for learning the optimal hedging policy is independent of the choice of market simulator. This is not the case for standard approaches to hedging derivatives.

We will show results for test cases ranging from a basic vanilla option to a real portfolio of barrier options. We use simple market simulators based on standard models (Black-Scholes [1], Heston [7]) with the addition of transaction costs and liquidity constraints. Classical approaches to the derivative hedging problem scale poorly with the size of the portfolio; in contrast, our method becomes more efficient as the portfolio grows.

The main contribution of this paper is the first application of reinforcement learning techniques to a particular problem in mathematical finance, namely the risk management of illiquid derivatives in a non-frictionless market. The key advantages of the reinforcement learning approach over existing techniques can be summarized as follows:

- Scalability. We are aware of no existing method of assessing the impact of market frictions that can be applied to an entire portfolio of derivatives with reasonable computational effort.
- Model independence. Our proposal decouples the choice of market dynamics from the architecture of the hedging engine; we assume only that we have some means of simulating market dynamics.

This article is an extended version of our recent work [3]; it contains additional results on the performance of our approach for a portfolio of barrier options, and the notation has been adapted to be in line with conventions in the machine learning literature.

1.1 Reinforcement Learning in Finance

There are several related applications of reinforcement learning in finance which face similar challenges. We want to highlight two such areas: firstly, classic portfolio optimization,[4] where, as in our case, non-linear objective functions are required, e.g., [16] and [11]; and secondly, algorithmic trading, where several authors have shown promising results, e.g. [4] and [15].

This article differs in that it focuses on the hedging of derivatives, and in particular, over-the-counter derivatives which do not have an observable market price. In this area, [6] has proposed hedging using Q-learning in a simpler setting (using only the stock price, with Black-Scholes assumptions and without transaction costs); [10] proposed to use machine-learning techniques trained on real equity index data for pricing and delta-hedging.

1.2 Existing Approaches to Hedging With Frictions

There is a vast literature on hedging in market models with frictions. Existing approaches[5] have the following shortcomings in common: they are numerically challenging, making application to large portfolios impractical; and the solution method is tightly coupled to the market dynamics. Our approach aims to address these drawbacks.

2 Derivatives Risk Management as a Reinforcement Learning Problem

We are looking for an algorithm to determine the optimal trading strategy for *hedging instruments* to mitigate the risk of holding a portfolio of illiquid *contingent claims*. We begin by clarifying these terms.

2.1 Hedging instruments

The defining property of a hedging instrument is that it can be traded daily with sufficient liquidity (but not necessarily at zero cost); we assume that our trading does not affect the price.

[4] Here "classic" means without options and under the assumption that market prices are available for all hedging instruments.

[5] Examples are discussed in Appendix A; see, e.g., [19], [18], and [17].

Remark 2.1. For simplicity of exposure, we will restrict our explanation to the equities market, where we will assume flat interest rates and that spot FX can be traded cost-free. We would like to stress that the framework we present is by no means limited in this way.

Our hedging instruments will be equities and listed equity futures and options. For simplicity we assume that the same d hedging instruments are used in each time step. This does not imply a limitation of the model: we will show below how to implement restrictions on trading in any of our instruments at any time.

We will broadly follow [20] in our notation. The *market* observed at some point t represents our state space $\mathcal{S}_t$. It will contain all current and past prices, cost estimates, news and anything else we might deem necessary for determining our risk management strategies. It will also contain past trading decisions and, when required, the internal state of our policy (e.g. for the hidden state of a recurrent neural network). States are denoted by $s_t \in \mathcal{S}_t$.

Holding a hedging instrument will at some point trigger a cash flow[6], either positive (received) or negative (paid). These payments are denoted by the d-dimensional function vector $h_t \equiv h_t(s_t)$. We denote by $H_t \equiv H_t(s_t)$ the vector of observable mid-market prices at time t, and by δ_t our current position in our hedging instruments.

2.2 Contingent Claims

A *contingent claim* is a generic name for a financial agreement on a sequence of cash flows at future times, determined by the performance of the underlying assets. A cash flow at time t must be a function of the observed market $\mathcal{S}_t$ at that time.

Example 2.2. *A simple example is a "knock-out put option" with maturity T of one year, a strike of $K = 100\%$ and a barrier of $B = 80\%$, written on a stock. We denote by H_t the price of the equity today, and by H_T the unknown price of the equity in one year. Intermediate prices are referred to as H_s. This option will then pay the holder* $\max(0, H_t K - H_T) 1_{\min_{s\in[t,T]} H_s > B H_t}$.

Some standard contingent claims (e.g., "vanilla" options) trade liquidly in the market; their prices are observable to within a bid-offer spread, and we categorize them as hedging instruments. Other contingent claims are more bespoke, and their prices are not observable; at any time, a successful over-the-counter derivatives business will hold a large portfolio of positions in such illiquid instruments. Before selling a bespoke derivative, we must determine the minimal price we are willing to accept to compensate us for the additional risk we are taking on; this price will depend on our existing portfolio (our "book"), and our appetite for risk. We will formalize this idea in Section 2.5 below.

The cash flows generated by our book are denoted by $z_t \equiv z_t(s_t)$; we use a generic z to refer to a specific set of derivatives. Since all these derivatives have a maturity, we may assume that there is a maximum maturity T.

2.3 Action Space

Given a book, we engage in active risk management by trading in hedging instruments. At any given time t, there is a range of hedging instruments available in the market, subject to transaction costs, and typically limited in their liquidity. A *policy* π is therefore a trading strategy which decides in each point in time how much to trade in each hedging instrument.

We denote our trading *actions* at time t by a_t. These actions are subject to constraints, such as liquidity limits, "risk" limits, market rules preventing naked short-selling, etc. Restrictions on trading are assumed to be functions of the state s_t, which includes our current position δ_t in our hedging instruments; as an example, $-a_t \leq \delta_t$ prevents naked short selling. They thus give rise to the set $\mathcal{A}_t$ of admissible actions at t, which we assume is non-empty.

In this note, we assume that the actions a_t of any trading strategy π are a deterministic functions of the state s_t, in other words $a_t \equiv a_t^\pi(s_t) \in \mathcal{A}_t$. For consistency, we also write $\delta_t^\pi := a_t^\pi + \delta_{t-1}^\pi$. The goal is to find the strategy π which is optimal for a given book and trading objective.

[6] In general the cash flow will occur at the end of our trading horizon, when all positions are liquidated. For an option, it will come earlier if the option expires before then.

2.4 Rewards

The first source of rewards are the cash flows from our portfolio of hedging instruments and illiquid derivatives:

$$R_t^\pi := \delta_t^\pi h_t + z_t \, .$$

The second source of (negative) rewards are transaction costs. For $s \in \mathcal{S}_t$ and $a \in \mathcal{A}_t(s)$ we model these as $C_t(a, s) := aH_t(s) + c_t(a, s)$, where $c_t(a, s)$ is a non-negative function and $c_t(0, s) = 0$. For a trading strategy π we then set

$$C_t^\pi(s) := C_t(a_t^\pi(s), s) \, .$$

Example 2.3. *A simple way to specify transaction costs is to assume that they are proportional to the value of the hedging instrument. Assume we wish to trade $a \in \mathbb{R}$ units of an equity with price H_t today (so $a > 0$ implies we are buying, while $a < 0$ implies we are selling). Proportional transaction costs are then given by*

$$c_t(a, s) := \gamma H_t(s)|a| \tag{2.1}$$

where $\gamma > 0$.

The sum of future rewards to some horizon $T > t$ when following a trading strategy π is given by the episodic representation

$$G_t^\pi := (R_{t+1}^\pi - C_{t+1}^\pi) + \ldots + (R_T^\pi - C_T^\pi)$$

Like R^π, the functional G^π depends on z.

2.5 Trading Goals

The classic goal of reinforcement learning is finding an optimal strategy π^* which maximizes expected rewards. However, when managing a book of financial contracts, we need to take into account the risk arising from any position: any financial institution is subject to internal and regulatory constraints which restrict the risk they can and want to take.

Such restrictions can often be expressed as target returns under a risk-adjusted measure. Reasonable minimal assumptions on such a risk-adjusted return measure E are:

- **Monotonicity**: for $X \geq Y$, we have $\mathrm{E}(X) \geq \mathrm{E}(Y)$ (a strictly better position has a higher value);
- **Concavity**: for X, Y and $\alpha \in [0, 1]$ we have $\mathrm{E}(\alpha X + (1-\alpha)Y) \geq \alpha \mathrm{E}(X) + (1-\alpha)\mathrm{E}(Y)$ (we are risk-averse);
- **Cash-Invariance**: $\mathrm{E}(X + c) = \mathrm{E}(X) + c$ for any $c \in \mathbb{R}$ (adding cash increases the return accordingly); and
- **Normalization**: $\mathrm{E}(0) = 0$.

Classic examples of such measures[7] are:

- **Worst-Case**: $\mathrm{E}(X) := \inf X$ measures the worst-case loss we could experience.
- **Expectation**: $\mathrm{E}(X) := \mathbb{E}[X]$.
- **Entropy**: let $\lambda \geq 0$. The functional $\mathrm{E}(X) := -\frac{1}{\lambda} \log \mathbb{E}\left[\exp(-\lambda X)\right]$ is called the *entropic risk-adjusted return*, also referred to as the *certainty equivalent of expected utility*.
- **Conditional Value At Risk "CVaR"** or the **Expected Shortfall** for the confidence level $\alpha \in [0, 1]$ is given by

$$\mathrm{E}(X) := \sup_{w \in \mathbb{R}} \left\{ w - \lambda \, \mathbb{E}[(w - X)^+] \right\} \tag{2.2}$$

  with $\lambda := 1/(1-\alpha)$ and where $x^+ := \max\{0, x\}$.

Remark 2.4. The most famous risk-adjusted return metric in portfolio management, the mean-variance metric $\mathrm{E}(X) := \mathbb{E}[X] - \lambda(\mathbb{E}[X^2] - \mathbb{E}[X]^2)$ is in general not monotone.

[7] Note that $(-\mathrm{E})$ is a *convex risk measure* [5].

For the Entropy and CVaR measures, we see that the function E decreases as risk aversion rises, with natural limits $\lim_{\lambda\uparrow\infty} \mathrm{E}(X) = \mathbb{E}[X]$ and $\lim_{\lambda\downarrow 0} \mathrm{E}(X) = \inf X$.

The value of a strategy π is given in our risk-adjusted framework as

$$v_t^\pi(z|s) := -C_t^\pi(s) + \mathrm{E}\left(G_t^\pi \mid s_t = s \right) .$$

Our goal is to find an optimal strategy π^* with value function

$$v_t^*(z|s) = \sup_{\pi\in\mathcal{A}} v_t^\pi(z|s) . \tag{2.3}$$

Remark 2.5. If our cost functions c_t are convex in their actions, then our objective is convex in π.

We then define the *minimal price* as the "indifference value"

$$p^*(z|s) := v_t^*(0|s) - v_t^*(z|s) \tag{2.4}$$

such that $v_t^*(z + p^*|s) = v_t^*(0|s)$.

3 Deep Hedging Algorithm

We propose to solve the optimization problem (2.3) by direct policy search.[8] We approximate the action function of our optimal strategy using neural networks: assuming $\hat{a}_t^{\mathbf{w}_t} : \mathcal{S}_t \to \mathbb{R}^d$ is a neural network with (deterministic) weights $\mathbf{w}_t \in \mathbb{R}^M$, we define the associated policy for $s \in \mathcal{S}_t$ by $a_t^\pi(s) := \hat{a}_t^{\mathbf{w}_t}(s)$. Next, we give an overview of the algorithm.

3.1 Sampling

There is insufficient historical market data for us to use it directly to train our model.[9] We therefore begin by generating a large number of simulated sample episodes for training and for analyzing out-of-sample performance.

Next, we evaluate the future cash flows of our book of derivatives in each of the sampled states. An advantage of our approach is that this computation is not part of the policy search; the computation of cash flows can be performed in parallel using modern Big Data map/reduce methods. The computational cost of adding another derivative to our portfolio is therefore linear and, when compared to the cost of finding an optimal strategy, marginally irrelevant, in contrast to existing methods.

3.2 Network Construction

For practical purposes it not feasible to allow the policy to depend on the entire state space $\mathcal{S}_t$. It is reasonable to assume that the hedging decision at some t mainly depends on the prices of the hedging instruments and the current position in them. We may choose to work with transformed versions of the raw prices: for example, option prices are typically converted to "implied volatilities", and equity prices are represented in terms of their logarithms. Our basic feature vector is therefore of the form $f_t \equiv f_t(H_t)$. We label this network *Simple*; its policy can be written in the form $\hat{a}_t^{\mathbf{w}_t}(f_t(H_t))$.

We know that in certain cases, current market prices do not provide enough information to hedge effectively. For example, in the presence of transaction costs, we expect that our decisions should be influenced by our existing position in the hedging instruments. A policy that includes this information has the form $\hat{a}_t^{\mathbf{w}_t}(f_t(H_t, \delta_{t-1}))$; we label it *Recurrent*[10].

Similarly, to manage path-dependent options, information about the past state of the market is needed (e.g., for the portfolio of section 4.4, we need to know whether the equity price has crossed a barrier). We can encode this information explicitly; for example, we may include the running maximum and minimum of the equity price: $\hat{a}_t^{\mathbf{w}_t}(f_t(H_t, \min_{t'<t} H_{t'}, \max_{t'<t} H_{t'}))$. We label this network *FFN*.

[8] Please refer to Appendix B for motivation of this choice.

[9] Each sample covers a period of duration equal to our time horizon (typically measured in months or years); the number of relevant historical samples available is therefore relatively tiny.

[10] Note that a fully recurrent network would be achieved by using the same network to hedge at each time step, so that $\mathbf{w}_t \to \mathbf{w}$.

An alternative approach to the need for past state information is to allow the network to pass on internal states; the extraction of relevant past data is then learned as part of the training process. The policy then takes the form $a_t^\pi(s) := \hat{a}^{\mathbf{w}}(f_t(H_t), y_{t-1})$, where y_t denotes the internal state: $y_t = \hat{y}^{\mathbf{w}}(f_t(H_t), y_{t-1})$. In our experiments below we use Long Short-Term Memory [8] cells to achieve this, and label the network *LSTM*.

3.3 Training

Having evaluated the portfolio and hedge instrument cash flows for all the samples, the sum of future rewards for each sample can be expressed as a function of the policy. The network can then be trained using standard methods based on stochastic gradient descent.

We wish to emphasize the importance of the proper formulation of the hedging objective in terms of a risk-adjusted measure. One could instead simply write down an heuristic penalty function such as variance to add to the expected reward, but this would obscure the meaning of the objective, and likely lead to violations of the reasonable minimal assumptions listed in section 2.5.

4 Applications & Results

In the following sections we present the results of applying our algorithm. For convenience, and for benchmarking purposes, we use standard theoretical models to simulate the market; this is not a limitation of our approach, and our optimization algorithm is separate from and orthogonal to the process for generating market data.

4.1 Settings

We consider two settings for our tests:

1. Market simulated using Heston stochastic volatility model (see, e.g., [2]); two hedging instruments (equity and variance swap). Time horizon of 30 trading days with daily rebalancing. *Simple* and *Recurrent* fully connected feed-forward networks with three layers; two hidden layers of dimension 15; either two (*Simple*) or four (*Recurrent*) input nodes, and two output nodes. ReLU activation functions throughout. Trained with the Adam optimizer (see, e.g., [13]) with a learning rate of 0.005 and batch size of 256.
2. Market simulated using Black-Scholes model; one hedging instrument (equity). Time horizon of one year with daily rebalancing. *FFN* network with three fully connected layers of width 20; ReLU activation functions for the first two layers; linear activation for the third layer; 1000 parameters per day in total. *LSTM* network with two LSTM cells of size 30 plus one linear layer; parameters shared across time steps (11,000 in total). Trained with the Adam optimizer with a learning rate of 0.01 and a batch size of 1000.

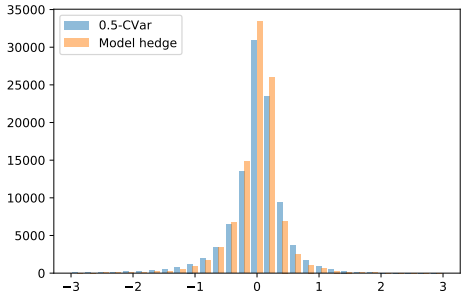
In both cases we take proportional transaction costs. We use 4×10^6 and 10^6 samples for training and out-of-sample testing, respectively. All results are computed using out-of-sample data.

The algorithms are implemented in Python, using Tensorflow to build and train the neural networks; network parameters are initialized randomly.

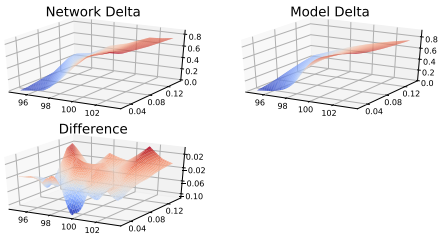
4.2 Benchmarking: Without Transaction Costs

We first consider hedging a simple European call option ($z_T = (S_T - K)^+$ with $K = S_0$) without transaction costs in setting 1. In this example, we use the CVaR risk-adjusted return (2.2) with $\lambda = 50\%$, and compare with the model hedge corresponding to our toy market dynamics.

In the continuous-time limit, the perfect replication strategy given by the model is also optimal when the objective is given as a convex risk measure. In our discrete-time setting, the model hedge is not perfect; for example, it generates a CVaR(50%) score of 0.25 (when charging the model price of 1.69). The CVaR(50%)-implied price of using the model hedge is therefore $1.69 + 0.25 = 1.94$. The *Simple* network price is 1.94, exactly in line with the model; the out-of-sample CVaR(50%) score for the network is -0.0001. We conclude that the network is able to reproduce the model hedge.

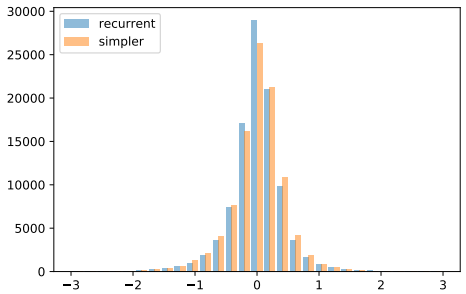


(a) P& L distribution

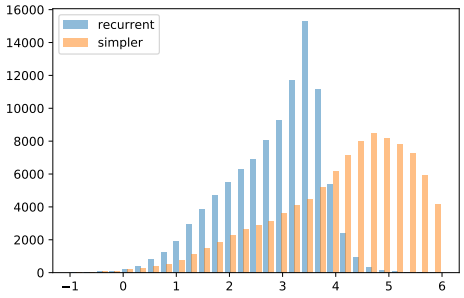


(b) Stock hedge δ_t for $t = 15$ days

Figure 1: P&L distribution and hedge visualisation, comparing the *Simple* network optimized for CVaR(50%) with the model hedge; model and network give very similar results in the absence of transaction costs.



(a) Without transaction costs



(b) With transaction costs

Figure 2: Network architecture matters in the presence of transaction costs: comparison of P&L distributions for *Recurrent* and *Simple* networks with CVaR(99%).

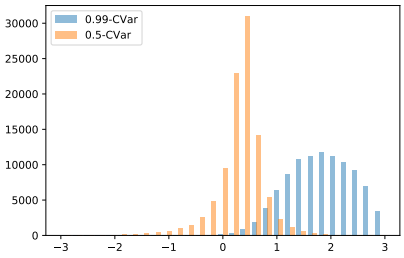
The return distributions are shown in Figure 1; for ease of comparison, we have assumed that we charge the same model price in each case. We also plot in Figure 1 the model and *Simple* network hedging strategies, as functions of the market state (i.e., the stock and variance swap prices), at a time point in the middle of the life of the option. We see that the model and network action functions are very similar.

4.3 Impact of Transaction Costs and Risk Aversion

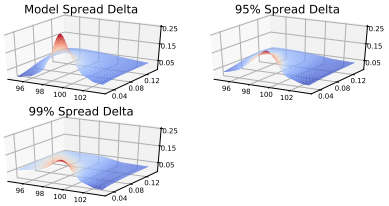
When there are no transaction costs, the previous examples show that the *Simple* network performs well; when we add transaction costs, we expect this to change. Figure 2 compares the performance of the *Simple* and *Recurrent* networks (again in setting 1), first without transaction costs (where they give very similar results), then with proportional transaction costs ($\gamma = 0.01$), where the two networks perform very differently for CVaR(99%). The simpler non-recurrent network yields a much higher, and therefore less optimal minimal price.

Next, we illustrate how different levels of risk aversion affect the P&L distribution. Figure 3a shows the P&L distribution for 99% and 50% CVaR parameters; in each case, we charge the respective CVaR-implied network price. We can clearly see the effect of risk aversion: the CVaR(50%) strategy is centered more closely on zero and has a smaller mean hedging error, while the more strongly risk-averse CVaR(99%) strategy yields fewer extreme losses (but, of course, requires a higher price).

To further illustrate the implications of risk aversion on hedging, we consider selling a "call spread" (the difference between two call options). The terminal cash flow is then $z_T = [(S_T - K_1)^+ - (S_T - K_2)^+]/(K_2 - K_1)$ (with strikes $K_1 < K_2$; we take $K_1 = S_t$, $K_2 = 101\% S_t$). In a risk-neutral world, the value of the difference of two call options is given by the difference in prices; with imperfect hedging and risk aversion, this is no longer true. We compare the model hedge to the more risk-averse CVaR-based network hedging strategies in Figure 3b. The plot shows the strategy flattening for higher levels of risk-aversion. From a practical perspective, this corresponds to a "barrier shift", i.e., a more risk-averse hedge for a call spread with strikes K_1 and K_2 actually aims

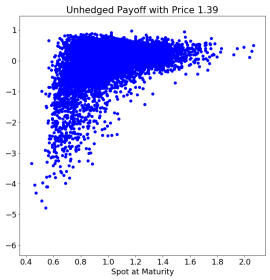


(a) Comparison of P&L distributions

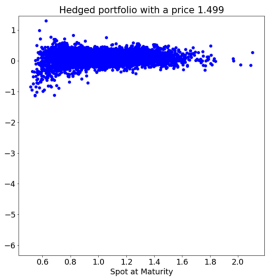


(b) Model and network equity "delta" for a call spread

Figure 3: Impact of risk aversion. P&L distributions and network "delta" for networks with different



(a) Unhedged P&L (charging the model price)



(b) Hedging with *LSTM* under CVaR(50%), CVaR price

Figure 4: The P&L arising from the barrier portfolio as a function of terminal equity price.

at hedging a spread with strikes $\tilde{K}_1$ and K_2 for $\tilde{K}_1 < K_1$. This qualitative behavior is in line with heuristics employed in manual trading.

4.4 Portfolio Hedging

Finally, we test our approach on a prototype real-life application: a historic snapshot of a real portfolio of daily-observed barrier options. The portfolio contains 69 trades, with a mixture of knock-in and knock-out call and put options, and maturities up to one year. We work in setting 2 (Black-Scholes model, hedging with equity).

Barrier options bring the added complication of path-dependence: to hedge efficiently, the network must learn that the final payoff depends on whether the spot price crossed the barrier level during the life of the trade.

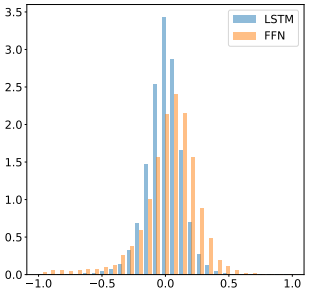
We compare *FFN* and *LSTM* networks; note that the feature vector of the *FFN* network is augmented with the running maximum and minimum of the spot price.

The model price of the portfolio is 1.39; with no transaction costs, the *FFN* network achieves a CVaR(50%) price of 1.596, while the *LSTM* obtains 1.499. Remarkably, the *LSTM* network is able to efficiently hedge this portfolio of options without the feature vector engineering, and outperforms the *FFN*. We show the effectiveness of the *LSTM* hedge by plotting the hedged and unhedged P&L as a function of terminal spot price in Figure4.

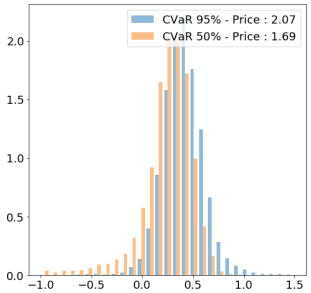
The P&L distributions are compared in Figure 5, along with the case of non-zero transaction costs. With transaction costs, the *LSTM* price rises to 1.69 under CVaR(50%) optimization and to 2.07 under CVaR(99%); again, we see that changing risk-aversion has the expected effect on the hedging strategy.

5 Conclusions

We have presented a novel approach to calculate the price and optimal hedging strategies for portfolios of derivatives under market frictions using reinforcement learning methods. The approach is model-independent and scalable. Learning the optimal hedge for the portfolio is faster than for a single



(a) Without transaction costs (CVaR(50%)).



(b) With transaction costs (*LSTM* only, $\gamma = 0.01$).

Figure 5: P&L distributions for a real portfolio of barrier options, hedged with *FFN* and *LSTM* networks. The price charged is the risk-neutral model price of 1.39.

instrument; the real efficiencies of scale that apply to the management of a derivatives portfolio also apply to our network.

References

[1] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, 1973.

[2] Hans Buehler. Heston model. In *Encyclopedia of Quantitative Finance*. John Wiley & Sons, Ltd, 2010.

[3] Hans Buehler, Lukas Gonon, Josef Teichmann, and Ben Wood. Deep hedging. *arxiv*, 2018. https://arxiv.org/pdf/1802.03042.pdf.

[4] Xin Du, JinJian Zhai, and Koupin Lv. Algorithm trading using Q-learning and recurrent reinforcement learning. *arxiv*, 2009. https://arxiv.org/pdf/1707.07338.pdf.

[5] Hans Föllmer and Alexander Schied. *Stochastic finance: An introduction in discrete time*. De Gruyter, 2016.

[6] Igor Halperin. QLBS: Q-learner in the Black-Scholes (-Merton) worlds. *arxiv*, 2017. https://arxiv.org/abs/1712.04609.

[7] Steven Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327343, 1993.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[9] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[10] James M Hutchinson, Andrew W Lo, and Tomaso Poggio. A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance*, 49:851–889, 1994.

[11] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A deep reinforcement learning framework for the financial portfolio management problem. *arxiv*, 2017. https://arxiv.org/abs/1706.10059.

[12] N. El Karoui, S. Peng, and M. C. Quenez. Backward stochastic differential equations in finance. *Mathematical Finance*, 71(1), 1997.

[13] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[14] Francis Longstaff and Eduardo Schwarz. Valuing american options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.

[15] David Lu. Agent inspired trading using recurrent reinforcement learning and LSTM neural networks. *arxiv*, 2017. https://arxiv.org/pdf/1707.07338.pdf.

[16] John Moody and Lizhong Wu. Optimization of trading systems and portfolios. *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, 1997.

[17] H. Mete Soner Peter Bank and Moritz Voß. Hedging with temporary price impact. *Mathematics and Financial Economics*, 11(2):215–239, Mar 2017.

[18] L. C. G. Rogers and Surbjeet Singh. The cost of illiquidity and its effects on hedging. *Mathematical Finance*, 20(4):597–615, 2010.

[19] H. M. Soner, S. E. Shreve, and J. Cvitani. There is no nontrivial hedging portfolio for option pricing with transaction costs. *The Annals of Applied Probability*, 5(2):327–355, 1995.

[20] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, 1998.

A Existing Approaches to Hedging With Frictions

We highlight a few examples of work on this topic to demonstrate the complex character of the problem. For example, [19] prove that in a Black-Scholes market with proportional transaction costs, the cheapest super-hedging price for a European call option is the spot price of the underlying; the bound is therefore too weak to be of relevance to practitioners. In [18] the authors study a market in which trading a security has a (temporary) impact on its price. A one-dimensional Black-Scholes model is used for the price process; the optimal trading strategy can then be calculated by solving a system of three coupled non-linear PDEs. In [17] a more general tracking problem covering the temporary price impact hedging problem is addressed for a Bachelier model; a closed-form solution for the strategy is obtained, involving conditional expectations of a time integral over the optimal frictionless hedging strategy.

B Determining the Optimal Risk-Adjusted Price

The standard approach to solving optimisation problems of the type 2.3 in quantitative finance is the use of (backward) dynamic programming which yields its value function. Typical methods are Finite Difference and, much less common, Finite Elements. The associated hedging strategy is then obtained by choosing the locally optimal action, which amounts to taking an infimum over the value function. Such an operation is usually not robust w.r.t. the usual value function approixmators. Either way, such methods are not feasible here because of the high dimensionality of our state space.

Another popular technique for similar problems in finance in a "complete market" setting is (forward) value function approximation; see, e.g., [14] for an early application. In complete markets, the optimal policy may be obtained by taking the first derivative of the value function by with respect to the prices of our hedging instruments.

With market frictions, more complicated methods using backward stochastic differential equations (BDSEs) have been proposed. However, none of these methods can be scaled up to a large class of instruments, nor are they independent of the underlying market dynamics; see, e.g. [12] for an overview.

A plain action-value function approach would also suffer from our lack of knowledge of the (optimal) distribution of the states $\delta_t \in \mathbb{R}^d$. Policy learning, on the other hand, is readily available and fits most naturally with our stated objective of finding the optimal hedging strategy; should we wish to compute the value function for a given strategy, doing so is relatively inexpensive.

We therefore focus on direct policy search, as described in the paper.

C Indifference pricing

We defined our *minimal price* for a portfolio z as the "indifference value"

$$p^*(z|s) := v_t^*(0|s) - v_t^*(z|s) \tag{C.1}$$

such that $v_t^*(z+p^*|s) = v_t^*(0|s)$.

To understand why this is a sensible choice, consider that $v_t^*(z|s)$ inherits from E the notion of cash invariance, i.e. $v_t^*(z+c|s) = v_t^*(z|s) + c$. That means we could try to interpret $-v_t^*(z|s)$ as the minimal price which we would need to charge to have zero risk, in the sense that $v_t^*(z - v_t^*(z|s)|s) = 0$. However, this is not quite correct, since for $z = 0$ the optimal value $v_t^*(0|s)$ may be positive, even if E is normalized. This is the case if there are "statistical arbitrage" opportunities, i.e. trading strategies which yield in expectation a positive return, and whose risks are acceptable with our level of risk aversion.

C.1 Dynamic risk measures

The entropy risk-adjusted return given by

$$\mathrm{E}(X) := -\frac{1}{\lambda} \log \mathbb{E}\left[\exp(-\lambda X)\right] \tag{C.2}$$

has the specific property that it is time-dynamic: it can be written as a recursive non-linear dynamic programming problem. We may define

$$v_t^\pi(s) := -C_t^\pi(s) - \frac{1}{\lambda} \log \mathbb{E}\left[\, \exp(-\lambda G_t^\pi) \mid s_t = s \,\right]$$

which satisfies the dynamic programming representation

$$v_t^\pi(s) = -C_t^\pi(s) - \frac{1}{\lambda} \log \mathbb{E}\left[e^{-\lambda\left\{R_{t+1} + v_{t+1}^\pi(s_{t+1})\right\}} \middle| s_t = s \right].$$

The CVaR functional (2.3) is not directly time-consistent, but we can still write it as a modified dynamic programming problem. Let

$$v_t^\pi(s) := -\inf_{w \in \mathbb{R}} \nu_t^{\pi,\lambda}(w|s)$$

for $\nu_t^\pi(w|s) := C_t^\pi(s) + \lambda\, \mathbb{E}[(w - G_t^\pi)^+ | s_t = s] - w$. Then, $\nu^{\pi,\lambda}$ has the dynamic programming representation

$$\nu_t^\pi(w|s) = C_t^\pi(s) + \mathbb{E}\left[\nu_{t+1}^\pi\left(w|s_{t+1}\right) - R_{t+1}^\pi \middle| s_t = s\right].$$

In other words, it is sufficient to learn ν to solve for v.

D Universal Policy Approximation For Constrained Policies

Here we present an extension of the universal approximation theorem to the case of constrained policies.

Fix a bounded activation function and consider a sequence of neural networks $(\alpha^M)_{M\in\mathbb{N}}$ with $\alpha^M = (\alpha_t^M, \ldots, \alpha_T^M)$ which has the following properties:

- Each α_t^M is a feed-forward neural network with at most M possible network weights and which maps $\mathcal{S}_t$ to $\mathbb{R}^d$.
- The networks are increasing in the sense that for all weights $\mathbf{w} \in \mathbb{R}^M$ there exist weights $\mathbf{w}' \in \mathbb{R}^{M+1}$ such that $\alpha_t^M(\mathbf{w}, \cdot) \equiv \alpha_t^{M+1}(\mathbf{w}', \cdot)$.

As shown in [9], any function on our discrete probability space can be approximated arbitrarily close with such networks.

We also construct a continuous projection $\eta_t(s_t) : \mathbb{R}^d \to \mathcal{A}_t(s_t)$ which restricts any action to the feasible region $\mathcal{A}_t(s_t)$ by setting $\hat{a}_t^{\mathbf{w}_t}(s_t) := \eta_t(\alpha_t^{\mathbf{w}_t}(s_t), s_t)$. We denote by $\mathcal{A}^M \subset \mathcal{A}$ the set of policies π which might be represented as $a_t^\pi(s) = \hat{a}_t^{\mathbf{w}_t}(s_t)$ for $\mathbf{w}_t \in \mathbb{R}^M$. We label quantities referring to such a strategy with weights $\mathbf{w} = (\mathbf{w}_t, \ldots, \mathbf{w}_T)$ with the superscript $\mathbf{w}$.

Proposition D.1. *Define the numerical objective*

$$v_t^M(s) := \sup_{\mathbf{w}} : -C_t^{\mathbf{w}}(s) + \mathrm{E}\left(G_t^{\mathbf{w}} | s_t = s\right) \ . \tag{D.1}$$

Assume that our cost-functions c_t are upper semi-continuous in their actions. Then, the universal approximation property holds: for $M \uparrow \infty$

$$v_t^M(s) \longrightarrow v_t^*(s)$$

Proof. Thanks to the universal approximation theorem of [9] we find for any strategy $\pi \in \mathcal{A}$ a sequence $\pi^M \in \mathcal{A}^M$ such that $a_t^M \to a_t^\pi$. Fix $\varepsilon > 0$ and chose some $\pi \in \mathcal{A}$ such that $v_t^\pi \geq v_t^* - \varepsilon/2$. Continuity of E when defined over a finite probability space, and upper semi-continuity of $c_t(\cdot, s)$ yield that $\limsup_{M\uparrow\infty} v_t^M \geq v_t^\pi$. We have therefore shown that for any $\varepsilon > 0$ there exists an M' such that $v_t^{M'} \geq v_t^* - \varepsilon$, as claimed. □