PROBABILITY & PARTNERS

Reinforced learning for hedging: transfer learning at work

By Dr Svetlana Borovkova and Alexandru Giurca

www.probability.nl

Summary

We apply reinforcement learning algorithms to option hedging and demonstrate that:

- Reinforcement learning "agents" outperform Black-Scholes – based hedging strategies in presence of trading costs and stochastic volatility;
- Agents show robust performance for a variety of option strikes and maturities – even those they have never seen before;
- Agents can transfer knowledge acquired on synthetic data to the real-world hedging if their training environment is versatile and includes stochastic volatility and jumps in the underlying price process.

Introduction

Machine learning is resolutely marching its way into finance. From credit scoring to option pricing and from fraud detection to algo trading – machine learning algorithms are proving their effectiveness in many financial applications. The massive generalization power of these algorithms allows them to tackle a wide variety of problems. Machine learning is particularly successful in situations where there is a clear underlying nonlinear function (which can be very complex and completely unknown), relating the inputs with the output. An excellent example of such a situation is the problem of option pricing, where a highly nonlinear relationship connects input parameters such as the volatility, price of the underlying, time to maturity, dividends, interest rates to the option price. Machine learning methods, when applied to option pricing, can deal not only with the standard Black-Scholes assumptions, but also with stochastic volatility, exotic options, while providing substantial gain in computational speed. Successful applications of machine learning in this area have been already extensively documented (see, eg., Ferguson and Green (2018), De Spiegeleer et al. (2018), Liu et al. (2019) and other recent papers on this topic).

A related ML application area is hedging of options. Here also, a complex nonlinear function relates the option's parameters to the option's delta and other greeks. This problem is not only nonlinear – it is also a sequential decision-making problem, where hedging decisions must be continuously made throughout the lifetime of an option, and these decisions are accompanied by a clear notion of "reward". A machine learning tool excellently suited for this type of problems is the so-called *reinforcement learning*: a class of machine learning algorithms which sequentially tune their parameters according to the rewards associated with actions.

In this paper, we discuss how reinforcement learning can be successfully applied to hedging of options and show that these machine learning algorithms can "transfer" knowledge obtained from simulated data to the real-world option trading environment. To our knowledge, this is the first study on ML for hedging, which reports a successful knowledge transfer to real data, while previous work in this domain has been done solely on simulated data.

Reinforcement Learning

In reinforcement learning, we typically talk of “agents”, who learn to achieve a goal by learning from past actions and from feedback given by the environment in the form of a reward. Based on the agent's action, the reward is a way of letting the agent know how good an action was at that time. A reinforcement learning agent takes actions in an environment, which are interpreted into a reward and a representation of the state, which are then fed back to the agent. This is shown schematically on Figure 1.

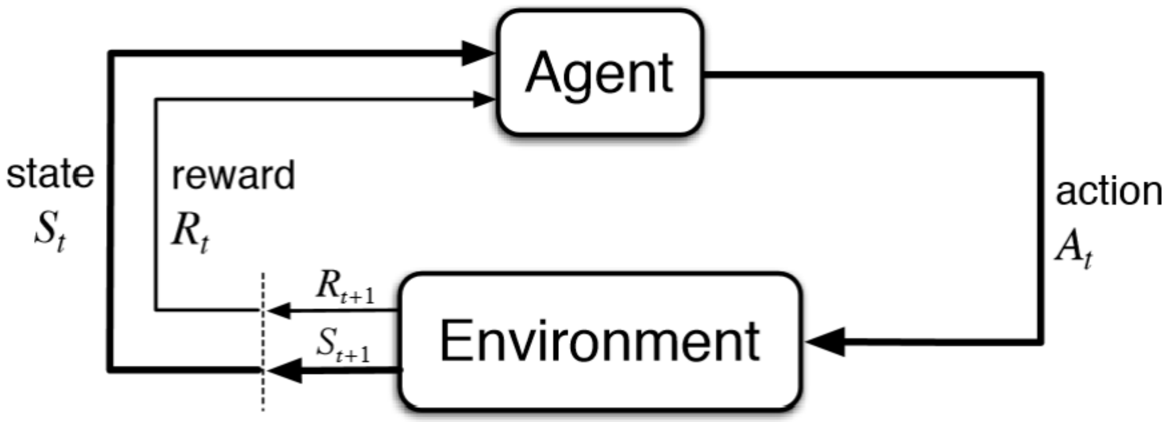


Figure 1: The general schematic operating way of Reinforcement Learning Algorithms. Based on an observed state of the environment and previous reward, the agent performs an action that moves the environment to a new state and generates a reward. The long-term goal of the agent is to maximize the cumulative reward.

The main components of reinforcement learning: state/environment → action → reward – are all present in a hedging problem, where the action can be thought of as a delta-hedging decision and the reward – as hedging costs, or a P/L of your hedged portfolio. In the hedging context, an agent learns to maximize the expected profit and loss of the total hedged option portfolio and to minimize the P/L variance at the option's expiration date (a long-term goal). The agent is interacting with a continuously changing market environment. This observable environment is influenced by the actions of the agent: the dynamically changing amount held in the underlying asset, needed to make the position delta neutral in every period. After each trading time, the market transitions into a new state, due to movement in asset prices, volatility and so on. The agent observes this new state and obtains a reward: the profit and loss achieved due to the action, which consequently determines the next action.

Reinforcement learning has been recently applied to the hedging problem by Hull et al. (2019) and Kolm et al. (2019), who demonstrated, based on simulated option prices, that it is possible to train a reinforcement learning algorithm to hedge a particular option in the framework of a specific model (e.g., Black-Scholes or stochastic volatility (SV) model). In this way they demonstrated a potential of this machine learning technique for making hedging decisions.

However, the reality of hedging presents several obstacles to implementing their approach directly. First, we do not know what the exact data-generating mechanism is, i.e., what is the price process of the underlying. Second, typically we do not have enough real trading option data to properly train a reinforcement learning algorithm. Finally, training a separate algorithm for each specific option (i.e., with a specific moneyness and maturity) is infeasible due to time and computational efforts involved.

So, we asked ourselves two questions: will a reinforced learning algorithm, trained on just one type of option, be able to cope with hedging a variety of options? And, if we train a reinforcement learning algorithm on a versatile range of price processes, will it transfer its acquired knowledge to the real-world hedging environment?

This last question, of transferring the knowledge leaned on synthetic, or simulated data to deal with the real market environment, is considered a "holy grail" of machine learning and it even has a special name: *transfer learning*. In other words, we wanted to know: can our reinforcement learning machine learn building with toy bricks and then go and build a real house?

The setup of investigation

We trained two well-known reinforcement learning algorithms – Deep Q-Networks (DQN) and Deep Deterministic Policy Gradient (DDPG) – in various environments: GBM, stochastic volatility (SV) model of Heston and SV model with jumps in the underlying. Such training environments were meant to be as diverse as possible in terms of price and volatility dynamics, to let the agents generalize well on the real market data. Then we compared their performance to the same algorithms but trained in a more specific setting: i.e., in the same type of environment and for the same type of options (as those in the test set). One would suspect that the agents trained to perform a more specific task would do much better than those trained to deal with for more "general" situations – but this turned out not to be the case. Finally, we investigated how well our trained agents can hedge real-life options, or, in other words, how well they can transfer their knowledge acquired on simulated data to the real hedging environment.

In our setting, we consider hedging task as a utility maximization problem, where agents seek an optimal hedging strategy according to some utility function. Such utility function can correspond to maximizing the agent's expected wealth (or P/L) at the end of the option's life, minimizing the hedging error or its variance, or a combination of these (we use such a combination, in the form of a mean-variance utility optimization).

Using P/L-optimizing utility function, we can consider transaction costs involved in hedging (these can be actual trading fees, bid-ask spread in the underlying or market impact) – and these costs can severely affect hedging decisions. On one hand, an option trader wants to hedge as frequently as possible, since only then one can achieve a nearly perfect hedge, but on the other hand, frequent hedging can lead to unacceptable transactions costs.

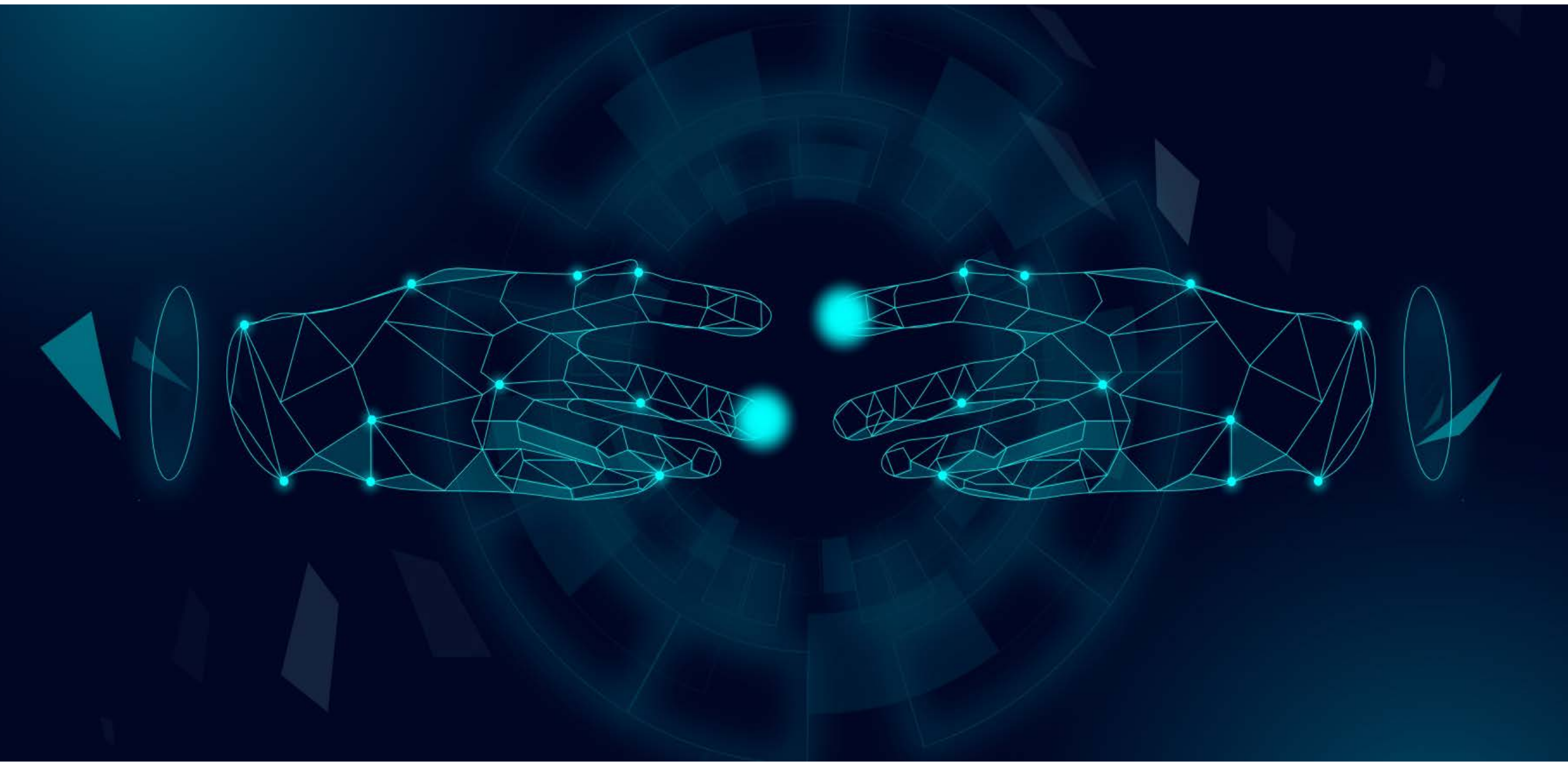
We train our reinforcement learning algorithms to delta-hedge European call options. A "state" at each point in time is characterized by the underlying asset price, volatility, the number of units of the underlying in the hedging portfolio, the value of the option and its delta. Furthermore, we specify the reward as the increment in P/L, penalized by its variance (which corresponds to our mean-variance optimization problem). Trading costs are calculated based on the tick size, the number of units of underlying bought or sold and a parameter reflecting the market friction for a particular underlying (which can be more liquid or less liquid, hence different friction parameter can be used). Finally, the action is naturally the amount of the underlying held in the hedging portfolio until the next re-hedging date. For mathematical details in all these choices and definitions, we refer the reader to the full paper (Giurca and Borovkova (2021)).

Note that the above framework is suitable not just for (a portfolio of) European call options, but can be extended to puts, options with other, more exotic payoffs as well as other options (path-dependent, American options or options on assets that provide dividends).

The evolution of the market (i.e., of the underlying asset) is simulated by the combination of Black-Scholes model, Heston stochastic volatility model and Bates model of stochastic volatility with jumps. Especially for the goal of transfer learning, we construct a rich training set that exposes the agents to enough versatile training environments, to enable them to hedge options in a real market environment. We do this by including episodes generated by a mixture of different market simulators in our training set. Compared to a training set based solely on realizations of one model, where the learned optimal policies are specific to this environment, in our setting the agent strategies are (hopefully) robust to an unknown testing environment. Instead of assuming one model that reflects the asset price dynamics of real market data best, we use a multitude of models: Black-Scholes, Heston and Bates.

The performance of the reinforcement learning agents is compared with two benchmark hedging strategies: the discrete time Black-Scholes delta hedging and Wilmott delta hedging (this is a variant of Black-Scholes hedging strategy but considering the balance between transaction costs and quality of discrete delta hedging, as measured by Gamma).

For simulated-to-real knowledge transfer analysis, we apply our trained algorithms to a rich set of daily S&P500 options data from 2019. We chose not to include the data from 2020, as this year can be described as a period of abnormal market conditions due to the Covid-19 pandemic, which introduced large fluctuations in prices and volatility in stock markets. We test our algorithms on those options that are close to being at-the-money and have up to 6 months until maturity. S&P500 price evolution is simulated, for all considered models, using model parameters calibrated on the market data from 2009 until 2018.



Results

Our full paper (Giurca and Borovkova (2021)) contains a multitude of interesting results – of which we will mention here just a few most significant ones.

First, in the "idealized" Black-Scholes environment, the reinforcement learning algorithms show particularly superior performance (when compared to benchmark strategies) in the most realistic situations: when transaction costs are high, and hedging is done daily (rather than weekly or monthly). Also, agents perform particularly well for long-maturity options. Once we depart from Black-Scholes environment and move into stochastic volatility territory, reinforcement learning agents show superior performance in all situations – indicating they are much better in dealing with more complex price processes than either BS-based hedging strategies. In practice, asset prices exhibit stochastic volatility, but hedging is done predominantly by Black-Scholes delta – so this result of superior performance of reinforcement learning agents is particularly useful in practice.

Next, we tested our agents on options different than those they were trained on. It turns out that agents perform quite well (in terms of expected P/L) when hedging European options with a variety of characteristics (moneyness and maturities), even when they were trained to hedge one specific European call option. So, they can "generalize" their acquired hedging knowledge to a wider range of options than just those they have been trained on. The agents' robustness to different strikes (rather than maturities) is particularly pronounced.

Furthermore, agents trained on a rich data set containing scenarios of different volatility levels have a much better hedging performance than BS or Wilmott hedging strategies, making them more robust in changing volatility environments. Also, we compared the agents trained on a variety of volatility environments ("versatile" agents) to those trained on a one specific volatility level ("specific" agents). The comparison is done on the test set with the same volatility level as that where "specific" agents were trained on. One would expect that agents trained on one specific task would outperform – but it turns out that "versatile" agents deliver the same performance as the "specific" ones. This means that using a versatile training set, which covers many volatility regimes, makes the agents robust in the real application environment.

Finally, what about the performance of the agents in the real market environment (i.e., when hedging real traded options), after they were trained on scenarios generated from a multitude of price models? It turns out the reinforcement learning algorithms reduce the hedging costs by 30% compared to Black-Scholes and 10% to Wilmott hedging strategies and had up to 14% lower variance, as Table 1 shows.

	π_{DQN}			π_{DDPG}	
$\mathbb{E}_\pi[PL_T]$	$\mathrm{Var}_\pi[PL_T]$	$\min(PL_T)$	$\mathbb{E}_\pi[PL_T]$	$\mathrm{Var}_\pi[PL_T]$	$\min(PL_T)$
-2.74	5.23	-11.42	-2.85	5.49	-12.53
	π_{BS}			π_W	
$\mathbb{E}_\pi[PL_T]$	$\mathrm{Var}_\pi[PL_T]$	$\min(PL_T)$	$\mathbb{E}_\pi[PL_T]$	$\mathrm{Var}_\pi[PL_T]$	$\min(PL_T)$
-3.97	6.36	-9.43	-3.02	5.31	-8.01

Table 1: Expected P/L, its variance and minimum P/L for two reinforcement learning algorithms and for two benchmark hedging strategies.

If we examine the minimum profit and loss at maturity, we see that, in some situations, the reinforcement learning agents show less than optimal performance. This could be the consequence of them encountering situations never seen before – which can be dealt with by a more precise calibration of the models to market data and by fine-tuning model's hyperparameters such as the risk aversion coefficient. Recall that this parameter is responsible for the variance and outliers in agents' P/L, and hence, increasing this parameter could reduce this downside risk.

The testing period was the year 2019, which was a relatively stable market. An in-depth testing on a more volatile market period (such as 2020) should be performed to assess the behavior of the agents under such circumstances. We imagine that, to deal with market distress, it can be useful to train the agents on two types of markets separately - a stable and a volatile period - and to use them according to the market environment encountered by traders in a precise situation, instead of using "universal" agents, trained on one training set.

While in simulated environment, we saw that DDPG algorithm outperformed DQN in terms of average profit and loss, on empirical data this difference fades away. As the training of DDPG is much more time consuming and sensitive to hyper-parameter tuning, a discrete action space algorithm like DQN would suffice in practice, to achieve a strong hedging performance.

To conclude

Here and in our extended paper (Giurca and Borovkova (2021)), we demonstrated that reinforcement learning algorithms can be successfully applied to the option hedging problem, We performed a multitude of experiments showing their superior performance in realistic situations – such as high transaction costs and stochastic volatility. Furthermore, we demonstrated that the "transfer knowledge" – being able to deal with real data while training the algorithms on the simulated ones - was successful. To our knowledge, this is the first such study, since all other studies in this area rely on synthetic data both for training and testing.

Our results are promising: we saw that reinforcement learning is a robust and flexible tool that can be used in the real-world hedging. However, the results can be improved further, by tuning model and hyperparameters and by separating stable and volatile market environments in the training phase.

Due to scarcity and high costs of historical option data and the computational resources required to train the algorithms, transfer learning is of fundamental importance in applications of machine learning to pricing and hedging of options. The ideal goal – to train an algorithm on synthetic data and for one specific derivative and use it for different derivatives and in real hedging environment – seems within reach.

References

Cao, J., Chen, J., Hull, J.C. and Poulos, Z. (2019). Deep Hedging of Derivatives Using Reinforcement Learning. Available at SSRN: https://ssrn.com/abstract=3514586

Giurca and Borovkova (2021). Delta Hedging of Derivatives using Deep Reinforcement Learning. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3847272

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

Ferguson, R. and Green, A. (2018). Deep Learning Derivatives. Available at SSRN: https://ssrn.com/abstract=3244821

Liu, S., Oosterlee, C.W. and Bohte, S.M. (2019). Pricing Options and computing implied volatilities using neural networks. Risks, Volume 7, Issue 1.

De Spiegeleer, J., Madan, D.B., Reyners, S. and Schoutens, W. (2018). Machine Learning for Quantitative Finance: Fast Derivative Pricing, Hedging and Fitting. Available at SSRN: https://ssrn.com/abstract=3191050

Contact

To find out more, please contact:

Dr Svetlana Borovkova, Head of Quantitative Modelling
svetlana.borovkova@probability.