FINANCIAL
INNOVATION
SERIES

GLOBAL RISK INSTITUTE

FINANCIAL INNOVATION SERIES

Reinforcement Learning: Risks and Challenges for Financial Institutions

Author: Alexey Rubtsov,
PhD, Senior Research Associate, Global Risk Institute

January 2022

EXECUTIVE SUMMARY

Machine Learning is generally viewed in four main categories: Supervised, Unsupervised, Semi-Supervised, and Reinforcement Learning. Each type of machine learning will be covered individually in our Financial Innovation series. This paper addresses Reinforcement Learning (RL) in finance and discusses the implementation issues for financial institutions who want to employ RL in their business.

Financial institutions around the world already apply RL to solve challenging problems in their businesses.[1] It has been reported that RL-based solutions were used to recognize the sudden changes in the market, adapt to them, and preserve performance for clients during the recent COVID-19 pandemic.[2]

We first illustrate key concepts of RL by a simple example. Next, we demonstrate the significant potential of RL by listing some of the current applications in finance. In the Implementation Issues section, we address the following questions:

- *What is needed to apply RL?*
- *How challenging is it to specify the objective for an RL algorithm?*
- *What risks should be addressed when RL is combined with other Machine Learning tools?*
- *What role can quantum technologies play in speeding up the development of RL-based solutions?*
- *Is it feasible to have different RL algorithms to efficiently interact with one another to achieve one common goal (e.g., to maximize returns for the entire organization)?*
- *What are the adverse consequences of combining domain human expertise with RL algorithms?*
- *What advantages will it bring to financial institutions when RL algorithms become capable of learning how to learn?*

We close this paper with key questions that should be addressed by financial institutions before they deploy RL-based solutions.

1. WHAT IS REINFORCEMENT LEARNING?

The word "Reinforcement" in Reinforcement Learning refers to the action that during the learning process certain behaviours of a learning algorithm are encouraged (or reinforced). To understand the idea behind RL, consider a simple example of chess.

Imagine a situation where one has to learn how to play chess. The only information that is available to the player is the moves that are allowed/not allowed, i.e., the rules of the game. Further assume that if the player wins the game, he/she is rewarded with $1; the draw yields the payoff of $0; if the opponent wins, the player's payoff is-$1. This payoff at the end of each game is termed "Reward". The "Objective" of the player is to maximize the total payoff received after playing many games.

In the absence of any other information (e.g., books, coaches, etc.), the only way to master the game is to start playing and learn as you play the game. It is intuitive that specific moves and strategies followed in certain situations during the game that eventually lead to a win should be remembered by the player and followed in future games when similar situations occur.

The following are the key insights from the example that are relevant to RL:

1. **Learning is accomplished through experience.** The more games are played, the more experience the player acquires.

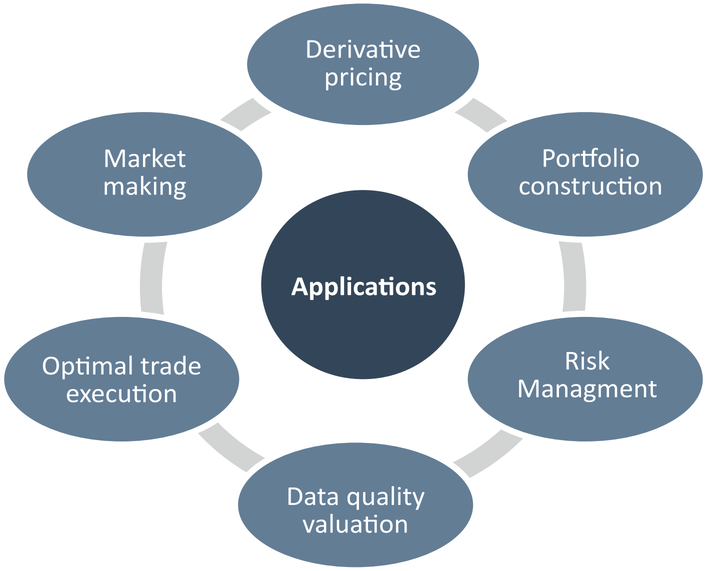
2. **Payoffs reinforce desired behaviour.** Successful strategies are identified because a player's strategy is driven by the desire to maximize payoff.

3. **The payoff is the way to communicate *what* we want to achieve, but not *how* we want to achieve it**. Indeed, it was the outcome of the entire game that was given an award of ±$1 or $0, but not the specific decisions made during the game. It is left to the player to learn the way to achieve the best results. This point is subtle but critical for RL.

2. APPLICATIONS

RL has been successfully applied to a variety of problems in finance. Below are some notable applications (Figure 1).

Figure 1: Applications of RL



DERIVATIVE PRICING

A call option confers on its owner the right, but not the obligation to buy one share of the stock for a certain price. An American call option can be exercised (i.e., used) at any time before the contract's expiration date. These American-type derivatives are found in all major financial markets, including the equity, commodity, foreign exchange, insurance, energy, sovereign, agency, municipal, mortgage, credit, real estate, convertible, swap, and emerging markets. The valuation and optimal exercise of American options remain one of the most challenging problems in derivatives finance, particularly when more than one factor affects the value of the option. Li et al. (2009) apply RL to value American options and find the optimal exercise strategy.[3] Their results show that the exercise policies discovered by RL gain larger payoffs than those by the current industry standard, Longstaff-Schwartz approach.

PORTFOLIO CONSTRUCTION

Many attempts have been made recently to apply RL to portfolio construction (see Cong et al. (2020), Liang et al. (2018), Jiang et al. (2017)).[4,5,6] One of the advantages of the RL approach is that it takes the distribution of asset returns as unknown, observes the risk-adjusted returns (e.g., realized Sharpe ratio), tests various asset allocations, and then maximizes the investors' objective directly without modelling asset-return distributions. This “distribution/model-free” approach can incorporate variable positions, transaction costs, risk appetite, path dependence, etc. The results are overall superior to traditional approaches used in the industry (e.g., mean-variance optimization). Cong et al. (2020) has employed RL to construct a portfolio that outperforms other successful benchmark portfolios developed over the recent years in the literature.

RISK MANAGEMENT (HEDGING)

Researchers from one of the major US banks applied RL to hedge a portfolio of derivatives in presence of market frictions such as transaction costs, market impact, liquidity constraints or risk limits (see Bühler et al. (2018)).[7] Of note is the computational performance of the developed approach, which is largely invariant in the size of the portfolio as it depends mainly on the number of hedging instruments available. Researchers from the University of Toronto similarly applied RL to find hedging strategies for derivatives in the presence of transaction costs (see Cao et al. (2021)).[8] In both research papers, the RL approach has seen better results than some well-known benchmarks.

DATA QUALITY VALUATION

Data is essential for all models. However, noisy and low-quality data may worsen the performance of an algorithm. In this respect, one needs to determine the most useful data for the target task. A rather straightforward approach is the so-called Leave-One-Out which evaluates the performance difference when a specific sample of the data is removed and assigns it as that sample’s data value. The major problem with this approach is its prohibitively high computational cost. Yoon et al. (2019) apply RL to adaptively learn data values based on the performance relative to a certain task (e.g., sales prediction).[9] The authors found the proposed algorithm significantly outperforms many traditional techniques.

OPTIMAL TRADE EXECUTION

How does one sell a large block of shares within a given timeframe? Selling all shares at once will have an adverse price impact, and some shares could be sold at a very low price. Yet, splitting the order into smaller blocks slows down the trade and could expose it to adverse market fluctuations. Ning et al. (2018) develop an RL algorithm that learns the best execution strategy.[10, 11]

MARKET MAKING

RL has also been applied to the problem of market making (see Ganesh et al. (2019)).[12] Market makers provide liquidity to markets by continuously quoting prices at which they are willing to buy and sell. A marker maker’s policies around pricing and risk management depend on its objectives and preferences (e.g., risk appetite), the policies of competing market makers, the overall market environment (e.g., volatility), and trade flow from investors. The researchers show that the RL-based market maker is able to learn about its competitor’s pricing policy. It also learns to manage inventory by selecting asymmetric prices on the buy and sell sides and maintaining a positive (or negative) inventory depending on the market price.

3. IMPLEMENTATION ISSUES

The application of RL in finance has a variety of challenges, considerations and risks when applying the ML model. We discuss several of these implementation issues in this section.

DATA AVAILABILITY

RL algorithms are data-hungry and have to be trained on a large amount of data. Applications that do not have many observations are not suitable for RL. One of the ways to overcome the data scarcity problem is to generate synthetic data. It becomes critical to use data generators that preserve the most important features of the real data (means, variances, tail correlations, etc.) of risk the miscalibration of the model. Recently, machine learning techniques have been applied to generate market scenarios with complex dependence structures (see Kondratyev and Schwartz (2020)).[13]

REWARD CHOICE

Defining rewards and the objective for an RL algorithm requires domain-specific expertise. For example, the risk of a portfolio can be measured in a variety of ways, such as Value-at-Risk, Conditional Value-at-Risk, the standard deviation of returns, etc. The objective that defines a proper risk-return trade-off should generally depend on internal and regulatory constraints. In this regard, we note that the results of RL could be sensitive to different specifications of the rewards and the objective.

COMBINATION OF MACHINE LEARNING TOOLS

In most applications, RL algorithms are used in conjunction with other ML tools such as Artificial Neural Networks (ANNs). As such, the additional risks inherent to ANNs should also be addressed for such applications.[14]

IMPROVING LEARNING TIMES

Another challenge is the relatively long time it takes to train an RL algorithm. For instance, it took a few days to train AlphaGo, a computer program developed by Google to play the board game Go. One exciting solution pathway is the emerging quantum technologies. It has been recently reported that a combination of RL with quantum technologies can significantly speed up the learning process for RL applications (see Saggio et al. (2021))[15] who saw learning time improvements of more than 60%.

ARTIFICIAL GENERAL INTELLIGENCE

The time it takes to train an RL-based algorithm becomes critical for the development of the so-called Artificial General Intelligence when the algorithm is trained to perform more than one task. For example, in 2017, researchers from DeepMind announced an RL-based system, AlphaZero, that taught itself how to play chess, shogi (Japanese chess), and Go (a board game). This line of research is particularly important because decisions made by multiple entities within the same institution could be coordinated as a system as opposed to the case when each entity maximizes its individual objective.

In 2019 DeepMind's research team developed an RL-based algorithm to play complex games that involve teamwork, such as the capture the flag "game mode" inside the video game Quake III Arena. The algorithm successfully learned skills specific to the game and, most importantly, how to collaborate with other teammates. This feature of RL can have beneficial implications. In the future, algorithms developed by different teams within a financial institution can be designed to interact with one another to achieve a common goal in the most efficient way.

COMBINING HUMAN KNOWLEDGE WITH RL

For many applications of RL, the rewards are sparse in the sense that the algorithm must make thousands of decisions before the reward becomes available. For example, the reward in chess comes only when the game is won/lost after typically many moves. However, if the game is won, what were the critical moves that made it successful? To answer this question, an RL-based algorithm requires a very large number of games to learn from.[16] In the case of the board game Go, a few million games were played before the system learned to play well. One way to overcome this problem is to let a human intervene in the learning process and employ, for example, Reward Shaping or Imitation Learning.

Reward Shaping takes advantage of domain expertise where an expert introduces intermediate rewards in addition to the final reward. In the chess game example, intermediate rewards could be made whenever the learner captures enemy pieces. On the other hand, instead of tinkering with the reward scheme, Imitation Learning has an expert provide a set of data that demonstrate how decisions should be made, and the goal of the algorithm is to learn by imitating the expert's behaviors. It should be emphasized that both approaches are risky in that they are quite subjective and could introduce biases and distract the algorithm from finding the optimal solution. Indeed, the algorithms can discover novel and unexpected ways to maximize rewards.

LEARNING HOW TO LEARN

People who know how to ride a bike are likely to learn fast how to ride a motorcycle. However, it is still recognized as an open problem on how to design RL algorithms to rapidly master new tasks. This approach to learning was termed Meta-Learning. The idea here is that the algorithm gains knowledge by learning on a large set of tasks, and as this knowledge accumulates, it allows the algorithm to adapt more quickly to each new task it encounters.[17] Given that financial markets are constantly changing, advancing Meta-RL is particularly important for financial applications where algorithms should not only adapt to new environments but also be expected to learn and perform new tasks.

4. KEY QUESTIONS

In this section, we list key questions that senior executives should ask their teams before an RL-based model is deployed. A number of these questions are based on the implementation issues that were discussed in the previous section. The questions are:

- *How were the data used by the RL algorithm generated (e.g., historical, simulated or expert-based)?*
- *Are there any simplifying assumptions made in the model (e.g., no transaction costs, no liquidity constraints, no market impact)?*
- *How were the rewards and the objective determined?*
- *Were any other ML models used in conjunction with RL and are their risks also being accounted for?*
- *What is the advantage of the chosen model over others that might also be used?*
- *Is interpretability an issue for the model?*
- *Are there any conditions when the model is not expected to work well?*
- *How was the model tested regarding biases?*

APPENDIX

To understand the idea that underlies RL, consider the Game of Nim example from Hull (2020).[18]

There are two parties in the game: the player and his opponent. The two parties take turns to pick matches from a pile of matches: each person must take only 1 or 2 or 3 matches at a time. The person who has no choice but to pick the last match loses the game and pays $1 to the other party. The question is: Can a computer learn the best strategy in this game?

The number of matches left in the pile is the only piece of information that one needs to consider before deciding on how many matches to pick. If the pile contains, say, 8 matches, the total number of possibilities that may occur can be represented by the following table.

Table A 1: Situations that can occur in the Game of Nim when the game starts with 8 matches

Matches to pick	Number of matches left						
	2	3	4	5	6	7	8
1	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓
3		✓	✓	✓	✓	✓	✓

To find the best strategy, one can simply let the computer simulate the game as follows.

1. **Player:** Most of the time selects the strategy that worked best in the past, and only occasionally makes random decisions to see if less explored strategies have higher payoffs. For instance, if the player has to decide about the number of matches to pick when there are 6 matches left in the pile, he can choose 1 match because in the past it was more likely to lead to a win. Alternatively, he could try picking 2 matches just to see if this could be a better choice although two matches led to fewer wins in the past.

2. **Opponent:** Always makes random decisions about the number of matches to pick (i.e., no specific strategy to follow).

The strategy adopted by the player is known as Exploitation vs. Exploration choice: either follow the strategy that was successful in the past or explore other possibilities in hope of better outcomes. In either case, additional experience is generated.

The following table shows the average payoffs after simulating the game 25,000 times.

Table A 2: Average payoff (in dollars) for each situation that can occur in the Game of Nim that starts with 8 matches. The game is played 25,000 times

Matches to pick	Number of matches left						
	2	3	4	5	6	7	8
1	1.000	0.080	0.104	0.069	0.936	0.000	0.741
2	-1.000	-1.000	0.103	0.412	-0.059	0.000	0.835
3		-1.000	1.000	-0.106	0.041	0.000	1.000

The best strategy is now clear from Table 2. The player should always start with selecting 3 matches from the original pile of 8 matches because this decision has the highest average payoff of $1 (bottom right corner of the table). This leaves the opponent with 5 matches. If the opponent selects 1 match, then the player now faces 4 matches in the pile and the best decision is to pick 3 matches again because this decision has the highest average payoff of $1. The opponent now loses the game because there is only 1 match left in the pile. The computer algorithm (or machine) learned the best strategy.

A similar approach is followed by data scientists who design RL algorithms for machines to learn. For instance, to learn how to play the board game Go, a computer (a machine) played millions of games against itself before it learned how to play well.

How can this example be used to approach problems in finance? In this respect, the following generalization is helpful:

- *position of pieces on the board are States of the environment*
- *possible moves represent Actions that are available in each state*
- *±$1 or $0 payoff is the Reward*
- *Objective is to maximize average reward achieved by playing many games*

Many problems can be approached with RL. We can think of stock trading where actions represent the number of shares to buy/sell and states represent the values of some economic variables. However, the trading example is much more complex than chess in that it has more states and actions that need to be considered. Furthermore, in the stock trading example, the economic variables that describe the states are not uniquely determined and different experts may have different sets of economic variables.

ENDNOTES:

1. For example, the Royal Bank of Canada developed Aiden, an RL-based trading platform. JP Morgan Chase applies RL in its financial derivatives business. OP-Trust employs RL to better manage the risk of their portfolio.

2. See the discussion "RBC Capital Markets wins 2021 Celent Model Sell Side Award for its Aiden® AI-powered electronic trading platform" of Aiden at http://www.rbc.com/newsroom/news/2021/20210309-rbccm-celent-award.html

3. Li, Y., Szepesvari, C., Schuurmans, D.: Learning Exercise Policies for American Options. Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

4. Cong, L., Tang, K., Wang, J., Zhang, Y. : AlphaPorfolio for Investment and economically Interpretable AI (2020)

5. Liang, Z., Chen, H., Zhu, J., Jiang, K., Li, Y.: Adversarial Deep Reinforcement Learning in Portfolio Management (2018) https://arxiv.org/abs/1808.09940

6. Jiang, Z., Xu, D., Liang, J.: A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem (2017) https://arxiv.org/abs/1706.10059

7. Bühler, H., Gonon, L., Teichmann, J., Wood, B.: Deep Hedging (2018) https://arxiv.org/abs/1802.03042

8. Cao, J., Chen, J., Hull, J., Poulos, Z.: Deep Hedging of Derivatives Using Reinforcement Learning (2021) https://arxiv.org/abs/2103.16409

9. Yoon, J., Arik, S., Pfister, T.: Data Valuation Using Reinforcement Learning (2019) https://arxiv.org/pdf/1909.11671.pdf

10. Ning, B., Lin, F., Jaimungal, S.: Double Deep Q-Learning for Optimal Execution (2018) https://arxiv.org/abs/1812.06600

11. On October 14, 2020 RBC launched Aiden, an AI-based electronic trading platform that uses the computational power of deep reinforcement learning in its pursuit of improved trading order execution. See https://www.borealisai.com/en/blog/aiden-reinforcement-learning-for-order-execution/

12. Ganesh, S., Vadori, N., Xu, M., Zheng, H., Reddy, P., Veloso, M.: Reinforcement Learning for Market Making in a Multi-agent Dealer Market (2019) https://arxiv.org/abs/1911.05892

13. Kondratyev, A., Schwarz, C.: The Market Generator, SSRN (2020)

14. See also "Artificial Neural Networks in Financial Modelling", Financial Innovation series, Global Risk Institute (2019)

15. Saggio, V., Asenbeck, B.E., Hamann, A. et al. Experimental quantum speed-up in reinforcement learning agents. Nature 591, 229-233 (2021)

16. This is known as 'sample efficiency' which reflects the amount of data that an algorithm needs to learn well. The more an algorithm can take out of every datum (a sample) available to it, the more sample efficient it is.

17. In February 2021, a research team from DeepMind and University College London have released Alchemy, a 3D video game which is proposed to serve as a benchmark for meta-RL research. Previously, there have been no benchmarks for testing new meta RL algorithms.

18. Hull, J.: Machine Learning in Business: An Introduction to the World of Data Science, 2nd Edition, 2020