2021

Real-World Reinforcement Learning: Observations from Two Successful Cases

Philipp Back

Real-World Reinforcement Learning: Observations from Two Successful Cases

Philipp Back

Aalto University School of Business, Department of Information and Service Management, Helsinki, Finland; e-mail: philipp.back@aalto.fi

Abstract Reinforcement Learning (RL) is a machine learning technique that enables artificial agents to learn optimal strategies for sequential decision-making problems. RL has achieved superhuman performance in artificial domains, yet real-world applications remain rare. We explore the drivers of successful RL adoption for solving practical business problems. We rely on publicly available secondary data on two cases: data center cooling at Google and trade order execution at JPMorgan. We perform thematic analysis using a pre-defined coding framework based on the known challenges to real-world RL by Dulac-Arnold, Mankowitz, & Hester (2019). First, we find that RL works best when the problem dynamics can be simulated. Second, the ability to encode the desired agent behavior as a reward function is critical. Third, safety constraints are often necessary in the context of trial-and-error learning. Our work is amongst the first in Information Systems to discuss the practical business value of the emerging AI subfield of RL.

Keywords: reinforcement learning, AI adoption, thematic analysis, machine learning, self-learning agents

University of Maribor Press

DOI https://doi.org/10.18690/978-961-286-485-9.20
ISBN 978-961-286-485-9

1 Introduction

Reinforcement Learning (RL) is a machine learning technique that enables artificial agents to learn optimal strategies for sequential decision-making problems (Sutton & Barto, 2018). No direct supervision is provided; the agent learns by trial-and-error while interacting with a (virtual) environment and receiving feedback on its actions.

Over the past decade, RL has achieved superhuman performance in a series of artificial domains. In a seminal paper by Google DeepMind (Mnih et al., 2015), an RL system learned how to play different Atari games, such as Breakout and Space Invaders, on a human or even superhuman level. This breakthrough started an RL frenzy in the scientific community and was presumably a major reason for Google to acquire DeepMind for $650 million in 2014. The next milestone came with AlphaGo, the first computer program to master the ancient Chinese board game of Go (Silver et al., 2017); a feat that experts believed was still decades away as the number of possible game states exceeds the number of atoms in the known universe. Over 200 million people watched online as AlphaGo beat the world's best Go player, Lee Sedol, 4 to 1 during The Google DeepMind Challenge. In the aftermath of the game, Lee Sedol described his opponent as following: *"I thought AlphaGo was based on probability calculation and that it was merely a machine. But when I saw this move, I changed my mind. Surely, AlphaGo is creative"* (Silver, Hubert, Schrittwieser, & Hassabis, 2018).

Despite its many opportunities, RL also presents significant challenges that have prevented wide-spread adoption in the real world (Dulac-Arnold et al., 2019). Google's DeepMind unit, conqueror of Atari, Go, and StarCraft II has been losing an estimated $1 billion over the past three years while trying to transfer its impressive work from artificial domains to real-world products - so far with little success.

Interestingly, Information Systems (IS) research has largely ignored the emerging AI subfield of RL. The research landscape remains dominated by the computer science community who continues to present algorithmic improvements that are usually evaluated on artificial problems. IS often treats AI technologies as mere **tools** for certain applications, such as decision making, supply chain management, market predictions, or innovation, rather than as objects of research themselves (Shmueli & Koppius, 2011). Specific AI subfields, such as machine learning, big data analytics, or predictive modeling, remain under-researched despite their rising importance to

management (Nascimento, da Cunha, de Souza Meirelles, Scornavacca Jr, & de Melo, 2018). Case in point: RL has already been discussed in IS as a **tool** for music recommendation (Liebman, Saar-Tsechansky, & Stone, 2019), but not yet as novel AI subfield **(object)**. However, amidst the ongoing RL hype, IS would serve practitioners well by treating RL as on object of study itself. Already 30 years ago, King (1984) pointed to a gap between inflated claims on what AI-based expert systems may be able to do, and what has actually been delivered. More recently, Nascimento et al. (2018) confirmed that claims made by AI vendors and the media seem ahead of what is supported by research findings. By critically examining RL's potential business value, IS would continue its long history of educating practitioners about the possibilities and challenges of novel AI technology. To quote Noel Sharkey, emeritus professor of Artificial Intelligence and Robotics at the University of Sheffield: *"[...] the wrong idea of what robotics can do and where AI is at the moment it's very, very dangerous."* (Delcker, 2018).

To this end, we conduct a qualitative exploratory study on two successful cases of real-world RL adoption: data center cooling by Google and trade order execution by JPMorgan. The research question that this paper seeks to address is: *What factors drive successful real-world adoption of Reinforcement Learning for practical business problems?* By studying "how others did it", we hope to provide decision-makers with a better understanding of RL's opportunities and pitfalls; and how to overcome them.

2 Theoretical Background

2.1 Reinforcement Learning Overview

RL is one of the three machine learning paradigms, alongside supervised and unsupervised learning (Sutton & Barto, 2018). An RL agent learns by interacting with its (virtual) environment (Figure 1). At each time step t, the agent observes the environment state s_t and chooses an action a_t from the set of available actions. The environment moves to the next state s_{t+1} and the agent receives a reward r_{t+1} based on the transition (s_t, a_t, s_{t+1}). The goal of an RL agent is to learn an optimal strategy (policy) which maximizes the expected cumulative reward. No direct supervision is provided to the agent; it is never directly told the best course of action. Rather, the agent has to learn from the consequences of its actions via trial-and-error. The only guidance comes from the numerical reward, a reinforcement signal

that encodes how good it is to take an action in a certain state. The agent must reason about the long-term consequences of its actions although the immediate reward associated with its action might be negative. For example, a stock trading agent may learn to accept small daily losses in exchange for a significant payoff when the market sentiment changes. Thus, RL is particularly well suited for sequential decision-making problems that include long-term versus short-term reward trade-offs[1].

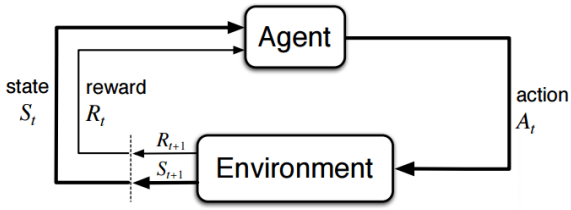

Figure 1: Reinforcement Learning framework
Source: Sutton & Barto (2018)

2.2 Known Challenges to Real-World Reinforcement Learning

Dulac-Arnold et al. (2019) identified nine challenges that – if present in an underlying real-world problem - must be addressed to productize RL:

1. training off-line from limited (historic) logs of the system's behavior
2. learning on the real system from limited (historic) data samples
3. high-dimensional continuous state and action spaces that become computational infeasible to search over (curse of dimensionality)
4. safety constraints that should never or at least rarely be violated
5. tasks that may only be partially observable (non-stationary/stochastic)
6. reward function design (unspecified, multi-objective, or risk-sensitive)
7. system operators who desire explainable policies and actions
8. inference that must happen in real-time at the system's control frequency[2]
9. large and/or unknown delays in the system actuators, sensors, or rewards

[1] Given the scope and limitation of this study, we refrain from a more detailed technical overview of RL and kindly refer the interested reader to Sutton & Barto (2018).
[2] The task can be run neither faster nor slower than real-time. This limits the quick generation of massive amounts of training, as well as slow, computationally expensive modeling approaches.

3 Methodology

To examine the drivers of successful RL adoption for practical business problems, we conducted a qualitative exploratory study on two cases: data center cooling by Google and trade order execution by JP Morgan. We selected the two cases based on the simple fact that they are – to the best of our knowledge – some of the only instances where RL has been successfully deployed in practice. We intentionally did not consider cases of self-driving cars and user-recommendation systems, as they are either not yet production-ready (Osiński et al., 2020), or use simpler forms of RL, such as contextual bandits (Amat, Chandrashekar, Jebara, & Basilico, 2018).

3.1 Data Collection

We rely on publicly available material for exploring the use of RL at Google and JPMorgan. Our sources include research publications, blog posts, newspaper articles, and published interviews with company representatives. We chose this form of qualitative secondary analysis (Heaton, 2008) because both case companies are reluctant to give interviews on the inner workings of their RL systems that would go beyond what they already chose to disclose. Nevertheless, useful insights can be gained from a careful analysis of various publicly available sources.

3.2 Data Analysis

We performed a thematic analysis using a pre-defined coding framework. Thematic analysis is a method for systematically identifying, organizing, and offering insights into patterns of meaning (themes) across a data set (Braun & Clarke, 2012). This method allowed us to identify what is common to the way in which the topic – what drives successful RL adoption for practical business problems - was described across multiple data items, and to make sense of those commonalities. As **themes**, we used the nine challenges to real-world RL (Section 2.2) by Dulac-Arnold et al. (2019). After becoming familiar with the data, we consolidated all sources in text format and systematically searched for the pre-defined themes. For example, the excerpt *"We send simulated orders to the exchange, we simulate how they execute, we simulate market impact, and then we feed the reward and batches of execution back to the agent's brain […]"* (QuantMinds365, 2018) related to "data efficiency" (theme 2) and "reward function design" (theme 6). Finally, we synthesized our findings in three key observations.

4 Findings

Here we present our findings to answer the research question of w*hat factors drive successful real-world adoption of Reinforcement Learning for practical business problems*. We start with a general overview of optimal trade execution at JPMorgan and data center cooling at Google. Next, we present the findings of our thematic analysis, i.e. how RL has been productized in each case. Finally, we highlight the business value that RL offered over competing solutions.

Overall, the thematic analysis of secondary data sources produced observations on four out of the nine themes: training from fixed logs (1), learning from limited samples (2), safety constraints (4), and reward function design (6).

4.1 Case 1: Optimal Trade Execution (JPMorgan)

Banks and financial service firms have long been using algorithms to make equity trading more efficient. In 2017, JPMorgan, one of the leading global financial services firm with assets of $2.6 trillion, announced LOXM, an AI-based limit-order placement engine that takes efficient trade execution to new heights. LOXM can execute equity trades at maximum speed and at optimal prices, and allows clients to offload large equity positions without causing market swings (Terekhova, 2017).

Optimal trade execution is a non-trivial problem: client needs differ, market conditions vary, and legal regulations must be met. An AI agent must learn to operate in the environment of bid/ask prices, and monitor the liquidity on both sides of the order book (QuantMinds365, 2018).

JPMorgan addressed these challenges by using RL. They constructed a market simulator from billions of past trades to provide the RL agent with a learning environment (Nevmyvaka, Feng, & Kearns, 2006; Vyetrenko et al., 2019). This simulator approach eliminated the need to learn directly from limited historic samples (theme 2). The artificial environment receives orders from the agent, simulates market impact, and sends rewards back to the agent who then updates its understanding of what actions are good/bad. In an interview, Vaslav Glukhow, Head of EMEA e-Trading Quantitative Research at JPMorgan, explained the process as following: *"In this approach we have an action, and the action is how much to place,*

what price, and for how long. It makes sense for the agent to be intelligent about the quantity that it asks for, and it needs to be smart in terms of how long it needs to place that order – if it gives the order for too long it will lose the opportunity and it will need to execute at a higher price. All these things need to be taken into consideration and the agent needs to be aware of the consequences of each action." (QuantMinds365, 2018). The agent is rewarded (theme 6) for being efficient in the market in the form of a scalar reward signal that encodes how well the agent splits large orders into smaller, more efficient "child" orders compared to executing the large "parent" order at once (Vyetrenko & Xu, 2019). LOXM makes use of RL's ability to balance long- and short-term rewards; the total rewards are not necessarily the sum of local rewards.

Since LOXM's depute in 2017, RL has proven its real-world worth for optimal trade execution. According to JPMorgan, the system provides significant savings and far outperforms both manual and existing automated trading methods (Terekhova, 2017).

4.2 Case 2: Data Center Cooling (Google)

Cooling is a critical component of data center operations. Servers produce considerable amount of heat, and high temperatures may lead to lower IT performance or equipment damage (Lazic et al., 2018). Dealing with excess heat is one of the biggest, most expensive factors when running a modern data center. Past solutions to the temperature problem have included moving data centers to cooler climates, or even situating them at the bottom of the ocean.

In 2018, Google announced that it has handed control over the cooling of several of its behemoth data centers to an AI that optimizes effective power management. The system learns how to adjust fans, ventilation, and other equipment to lower power consumption (Knight, 2018). Two years earlier, in 2016, Google had already presented an earlier version that made recommendations to human data center managers, who would then decide whether to implement them (Evans & Gao, 2016). The new system is managing cooling all by itself, although a human can still intervene. *"It's the first time that an autonomous industrial control system will be deployed at this scale, to the best of our knowledge",* said Mustafa Suleyman, head of applied AI at DeepMind, an AI company that was acquired by Google in 2014 (Knight, 2018).

Google uses RL to find optimal control policies for their complex, large-scale data center systems. Google does not explicitly disclosure how the RL learning environment has been constructed (Lazic et al., 2018), only that historic control data has insufficient information to directly train an RL agent (theme 2). However, earlier work has shown that neural networks can model Google's data center dynamics with 99.6% accuracy (Kava, 2014). Thus, it can be assumed that Google is using the latter model as a simulator to construct the RL environment. To ensure safe operation already during training (theme 4), Google used historic control logs (theme 1) to limit each control variable to a safe range. In the absence of such data, the safe range could be initialized conservatively and gradually expanded (Lazic et al., 2018).

According to Google, deep RL has proven highly effective at operating cooling systems: the system consistently achieves a 40% reduction in the amount of energy used for cooling (Evans & Gao, 2016).

5 Discussion

In both applications - optimal trade execution and data center cooling - the goal was to find an optimal strategy for a sequential decision-making problem. Despite very different domains, both applications share certain characteristics that allowed RL to be applied in practice. In the following, we discuss some of the key challenges in RL (Dulac-Arnold et al., 2019) and how Google and JPMorgan overcame them.

5.1 Observation 1: Learning Directly from Historic Data is Still Difficult

Standard supervised learning is teaching by example, whereas RL is teaching by experience. The agent gathers experience by interacting through trial-and-error with its environment; like a virtual playground. The learning by trial-end-error framework can be applied to almost any sequential decision-making problem, but this generality comes at a price: RL is hugely sample inefficient, meaning it requires a lot of training data. Imagine trying to learn a new board game, but instead of studying the rule book or recalling your experience from other board games, you simply take random moves while only receiving the final game score as feedback. What quickly strikes humans as a ludicrous endeavor is precisely how many RL problems are framed. Indeed, given enough time and computational power, the trial-and-error approach should converge to a (near-) optimal strategy, but what is enough? For AlphaGo it took 5

billion games of self-play and around $35 million in computing costs to beat the best human Go player.

Since RL agents require a lot of experience, and experience is gathered by interacting with an environment, the way in which a (real-world) problem can be represented as an RL environment is critical. Most of RL's successes have been with artificial systems that can be simulated, such Atari games (Mnih et al., 2015), Go (Silver et al., 2017), DOTA (Berner et al., 2019), and StarCraft II (Vinyals et al., 2019). The reason for this is that simulators allow us to generate unlimited training data - an advantage that can hardly be overstated in the context of sample inefficient learning. An RL agent might not be able to gather sufficient experience from a set of finite training data, but with a simulated environment it can continue to interact and learn until a (near-) optimal policy is found. Moreover, simulators eliminate much of the messiness that an agent may face in the real world and thus provide stable benchmarks against which new algorithms can be compared. Simulated environments therefore also play an important role in algorithmic research.

Unfortunately, out-of-the-box simulators rarely exist for real-world applications. The two herein presented applications addressed this issue by improving the fidelity of the simulations to a point where the gap between simulations and the real world was so small that things learned in simulation were directly transferable to the real world. JPMorgan used billions of past trades to construct a market simulator that closely reflected the true market dynamics (Vyetrenko et al., 2019). Similarly, Google managed to simulate data center dynamics with 99.6% accuracy (Kava, 2014). In contrast to training directly on a (limited) historic dataset (theme 2), simulators offer unlimited opportunities to learn. Assuming that the simulator accurately reflects the real system, it then becomes possible to transfer the learned optimal strategy directly from the artificial environment to the real world. The existence of an environment simulator, or the ability to construct one from historic data or first principles, is therefore a common characteristic of successful real-world RL applications.

Observation 1: If the tasks in a field can be accurately simulated, RL may dramatically improve the state of the art in that field over the next few years.

5.2 Observation 2: It's All about Rewards

The desired behavior of an RL agent has to be encoded in a reward function. Without direct expert guidance, the reward function is the only source of feedback that tells the agent what action in which state was good. Unfortunately, reward function design is notoriously difficult as it must capture *exactly* the desired behavior. An ill-defined reward function will cause the RL agents to break in surprising, counterintuitive, and sometimes amusing ways. One of the most (in-) famous examples of faulty reward functions is an RL agent learning to play the CoastRunners game (Clark & Amodei, 2016). The goal of the game - as understood by most humans - is to finish a boat race quickly and (preferably) ahead of others. Players are not directly rewarded for progressing around the course but earn points by hitting targets laid out along the racetrack. The human who designed the reward system simply assumed that the overall game score would implicitly reflect the goal of finishing the race. However, the targets were positioned so that the RL agent could gain a high score without ever having to finish the race. Instead, the agent finds an isolated lagoon where it can turn in a large circle and repeatedly knock over three targets, timing its movement so as to always knock over the targets just as they repopulate. Despite repeatedly catching on fire, crashing into other boats, and going the wrong way on the track, the agent manages to achieve a higher score using this strategy than is possible by completing the race in the normal (human) way (Clark & Amodei, 2016). The counterintuitive strategy leads to scores that are on average 20% higher than what is achieved by human players.

Reward function design becomes even more complex when systems have multi-dimensional costs, or when product owners cannot even articulate clearly what they want to minimize (Dulac-Arnold et al., 2019). In both herein presented cases, however, it was possible to encode the desired agent behavior as a scalar reward signal (theme 6): minimize trading costs and energy consumption. Neither JPMorgan nor Google had to perform any tricks to encode a complex behavior or multi-dimensional objective as a scalar reward. The natural representation of the desired system behavior thus greatly contributed to the suitability of RL.

Observation 2: An agent's desired behavior must be able to be expressed exactly as a reward function.

5.3 Observation 3: Safety First

Most physical systems can damage themselves and their environment if operated incorrectly. When neither a simulator nor sufficient training data are available, and when the RL agents has to gather experience by interacting directly via trial-and-error with the real system, safety becomes important; not only during the final system operation, but also during the exploratory learning phase. For example, a self-driving car cannot be allowed to explore all possible actions, including crashing full speed into a wall, until it has eventually mastered to stay within its lane. (Un)fortunately, safety violations will likely be very rare in historic logs and thus do not provide sufficient opportunity to learn from them (Dulac-Arnold et al., 2019). Furthermore, safety constraints are often assumed ("it's just common sense!") and are not even specified explicitly. Moreover, it may not be possible to impose strict limits by describing the action or state space directly; instead, hazard-avoiding behavior must be learned.

Safety constraints were crucial in both herein discussed applications of RL. Google's data center represents a large-scale system that can suffer catastrophic damage when operated incorrectly. JPMorgan had to consider legal, regulatory, and ethical constraints that may prohibit certain trading activities. As discussed in Section 5.2, it is notoriously difficult to express fine-grained behaviors in the reward function. Constrained RL is thus a viable alternative to prohibit the agent from learning, or even exploring, certain actions.

Observation 3: Trial-and-error learning makes safety considerations during training and control highly important.

6 Conclusion and Further Directions

Reinforcement Learning has achieved superhuman performance in many artificial domains, yet real-world applications remain rare. We conducted a qualitative study to explore the drivers of successful RL adoption for practical business problems. We used pre-defined themes based on the known challenges to real-world RL by Dulac-

Arnold et al. (2019) to perform a thematic analysis on secondary data that described how Google and JPMorgan successfully deployed RL for data center cooling and optimal trade execution. Three success drivers emerged: simulatability of the problem dynamics, a natural representation of the desired agent behavior as a reward function, and compliance with safety constraints during trial-and-error learning. Our work is amongst the first in IS to approach RL as a separate object of study. By exploring two successful real-world RL applications, we hope to shine some light on the practical business value of this emerging AI subfield.

Notwithstanding some useful insights, this study is limited by the use of publicly-available secondary data. For example, we were only able to identify four out of the nine themes (challenges) in our dataset. Especially the Google case could have benefited from more details. This study is thus a mere first step towards a more complete understanding of the practical business value of RL. For the future, we hope to collect primary data through interviews with key people who were involved in the design and deployment of the herein described RL solutions at Google and JPMorgan. Such data would allow us to directly analyze the technical details of the RL systems, rather than using high-level (textual) descriptions. Our study is further limited by having only two case companies. It will be interesting to extend our study to future use cases of RL once this method has become more widely adopted. Finally, it would be interesting to tackle a new application area with RL, and to document how the known challenges are addressed during the design process.

References

Amat, F., Chandrashekar, A., Jebara, T., & Basilico, J. (2018). Artwork personalization at Netflix. Proceedings of the 12th ACM Conference on Recommender Systems, 487–488. Vancouver, Canada.

Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., … Hesse, C. (2019). Dota 2 with large scale deep reinforcement learning. ArXiv Preprint ArXiv:1912.06680.

Braun, V., & Clarke, V. (2012). Thematic analysis. In APA handbook of research methods in psychology, Vol 2 (pp. 57–71). Washington, DC: American Psychological Association.

Clark, J., & Amodei, D. (2016). Faulty Reward Functions in the Wild. Retrieved September 20, 2020, from https://openai.com/blog/faulty-reward-functions/

Delcker, J. (2018). Europe divided over robot 'personhood.' Retrieved November 10, 2020, from https://www.politico.eu/article/europe-divided-over-robot-ai-artificial-intelligence-personhood/

Dulac-Arnold, G., Mankowitz, D., & Hester, T. (2019). Challenges of real-world reinforcement learning. ArXiv Preprint ArXiv:1904.12901.

Evans, R., & Gao, J. (2016). DeepMind AI reduces Google data centre cooling bill by 40%. Retrieved September 20, 2020, from https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40

Heaton, J. (2008). Secondary analysis of qualitative data: An overview. Historical Social Research/Historische Sozialforschung, 33–45.

Kava, J. (2014). Better data centers through machine learning. Retrieved August 21, 2020, from https://blog.google/inside-google/infrastructure/better-data-centers-through-machine/

King, W. R. (1984). Editor's comment: Decision support systems, artificial intelligence, and expert systems. MIS Quarterly, 8(3), iv–v.

Knight, W. (2018). Google just gave control over data center cooling to an AI. Retrieved September 5, 2020, from https://www.technologyreview.com/2018/08/17/140987/google-just-gave-control-over-data-center-cooling-to-an-ai/

Lazic, N., Lu, T., Boutilier, C., Ryu, M. K., Wong, E. J., Roy, B., & Imwalle, G. (2018). Data center cooling using model-predictive control. Advances in Neural Information Processing Systems. Montreal, Canada.

Liebman, E., Saar-Tsechansky, M., & Stone, P. (2019). The right music at the right time: Adaptive personalized playlists based on sequence modeling. MIS Quarterly, 43(3).

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., … Ostrovski, G. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529.

Nascimento, A. M., da Cunha, M. A. V. C., de Souza Meirelles, F., Scornavacca Jr, E., & de Melo, V. V. (2018). A literature analysis of research on artificial intelligence in management information system (MIS). AMCIS.

Nevmyvaka, Y., Feng, Y., & Kearns, M. (2006). Reinforcement learning for optimized trade execution. Proceedings of the 23rd International Conference on Machine Learning, 673–680. Pittsburgh, USA.

Osiński, B., Jakubowski, A., Zięcina, P., Miłoś, P., Galias, C., Homoceanu, S., & Michalewski, H. (2020). Simulation-based reinforcement learning for real-world autonomous driving. 2020 IEEE International Conference on Robotics and Automation (ICRA), 6411–6418. Paris, France.

QuantMinds365. (2018). The Latest in LOXM and why we shouldn't be using single stock algos. Retrieved July 3, 2020, from https://informaconnect.com/the-latest-in-loxm-and-why-we-shouldnt-be-using-single-stock-algos/

Shmueli, G., & Koppius, O. R. (2011). Predictive analytics in information systems research. MIS Quarterly, 553–572.

Silver, D., Hubert, T., Schrittwieser, J., & Hassabis, D. (2018). AlphaZero: Shedding new light on chess, shogi, and Go. Retrieved September 18, 2020, from https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., … Bolton, A. (2017). Mastering the game of go without human knowledge. Nature, 550(7676), 354–359.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). Cambridge: MIT Press.

Terekhova, M. (2017). JPMorgan takes AI use to the next level. Retrieved August 19, 2020, from https://www.businessinsider.com/jpmorgan-takes-ai-use-to-the-next-level-2017-8?r=US&IR=T

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., … Georgiev, P. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 575(7782), 350–354.

Vyetrenko, S., Byrd, D., Petosa, N., Mahfouz, M., Dervovic, D., Veloso, M., & Balch, T. H. (2019). Get real: Realism metrics for robust limit order book market simulations. ArXiv:1912.04941.

Vyetrenko, S., & Xu, S. (2019). Risk-sensitive compact decision trees for autonomous execution in presence of simulated market response. ArXiv Preprint ArXiv:1906.02312