Understanding the future of Deep Reinforcement Learning from the perspective of Game Theory

ZIYI GAO

1 Department of Mathematics, Fordham University, Bronx, NY, 10458, U.S.
* Corresponding author’s e-mail: zgao23@fordham.edu

Abstract. This paper is to discuss the development of Deep Reinforcement Learning and the future of it from the perspective of Game Theory. The relationship and potential interaction between these two areas are also introduced, especially the optimization method. This paper discusses about the situations both under non-cooperative and cooperative game. Recently, Artificial Intelligence (AI) and Machine Learning (ML) have grasp sufficient attention from various research areas. Deep Reinforcement Learning, as one of the most promising ML methods, enlighten more researchers to devote themselves in this area. However, even such accomplishment could not belie that, for most kinds of real-life problems, DRL is still unable to provide with an optimal strategy. Because unlike the well-adjusted environment in laboratory, real life problems are not always able to be converted into mathematical problems. Under such circumstances, most of real-life problems have no nominal “optimal solution”. Game Theory provides potential solutions to covert “real issues” into “mathematical problems”; then it is easier for researchers to handle.

1. Introduction

Deep Reinforcement Learning (DRL) has already become a hot topic in the research of data science, whose shocking performance in recent year gain increasing attention all over the world. In 2016, AlphaGo [1], an agent developed by Deep Mind, defeated Lee Sedol and Ke Jie, two human world champions in go. This astonishing accomplishment of DRL demonstrated its infinite potentials and bright futures.

However, some of the problems in real life are imperfect information problems [2], where the rewards and punishments are inconspicuous. Thus, the traditional DRL (relied on Q-Learning) sometimes has difficulties in converting real problems into mathematical problems for convenience. On the other hand, in the multi-agent strategy games, the number of the players will bring more complexity in the action space and status space. Both drawbacks make designing the optimized strategy complicated. Thus, Game Theory is introduced for understanding and optimizing DRL. Game Theory initially addresses the decision-making process in complete information as well as incomplete information circumstances. One of the classic scenarios in Game Theory is zero-sum game. And more complicated models for complex problems are applied to widely ranges of reality issues, focusing on the strategic interactions between multiple rational agents. The outcomes of Game Theory are influenced by all agents’ own strategies in the game, drawing the analogy of DRL.

Nowadays, single agent’s capability of designing strategy is insufficient. People expect the agents to be able to cooperate with other agents in the same environment and make the best strategy in whole, not only card games but also complex problems in real life [3]. For example, international air traffic is a very complex problem in real life [4]. Airplanes from hundreds of countries would enter the air traffic
system stochastically and fly at different velocities. Thus, keeping the air traffic environment both safe and efficient becomes highly demanded. Another interesting example is DouDiZhu [5], a strategic card game in China, whose model is two players confront the third player together. And this requires agents learn how to make use of good card combinations and cooperate with each other.

This paper aims at introducing how to understand and optimize the Deep Reinforcement Learning (DRL) algorithm with the knowledge of Game theory. Firstly, this paper is going to introduce the development of DRL in recent years and then it will demonstrate the relationship between the Game Theory and DRL.

2. The development of Reinforcement Learning and Deep Learning

Reinforcement Learning (RL) is a traditional learning strategy for solving policy problems and is often described as Markov Decision Processes (MDPs). It lets an agent to interact with the environment, to receive a status in status space, and to choose an action from action space, following a policy, receiving a reward, then to transition to next status [6]. In order to solve more complicated problems and to save training costs, Deep Learning (DL) is introduced.

The origin of RL can be traced back to 1957, when Bellman first introduced the definition of “dynamic programming”, which is known as Bellman Equation [7]. It breaks a complex problem down into several sub-problems in a recursive manner and finds optimal solutions for each sub-problem. Then, if the sub-problems are able to be recursively nested in bigger problems, the methods are applicable. Even though, as Bellman indicated that as there are small increments in state variables, the computational workload increases exponentially, dynamic programming still outperforms general methods and is widely applied to numerous areas, especially in engineering. Almost every problem that is able to be solved by Optimal Control theory could be solved via analyzing appropriate Bellman Equation.

Q-Learning, an off-policy Temporal Difference control algorithm was firstly introduced in 1989 where “Q” represents quality of an action, and then explained in 1992 by Watkins [8]. Q-Learning is able to deal with stochastic problems without environment adaptions (model-free). It later becomes the most basic algorithm in RL for drastically simplifying analysis of algorithm and contributing to convergence early proofs [9]. For any given finite Markov Decision Processes, Q-Learning can search for a policy which can maximize the rewards for all steps through acquiring Q-Matrix. The advantage of Q-Learning is that it does not require model constructions for any given problems. On the contrary to Q-Learning, an off-policy algorithm, Sarsa is an on-policy algorithm in RL, firstly created in 1995 by Rummery [10]. Both its action policy and its evaluation policy rely on ε-greedy policy. The difference between these two algorithm is that, Q-Learning firstly assumes the next action with the best reward, then updates value function, and then chooses the next action through ε-greedy policy, while Sarsa executes the next action via ε-greedy first, and then updates value function. The advantage of these two algorithms is that they directly pursuit the best policy.

In 21st century, RL gains more and more heed from researchers, and it develops into a new status. Before the 21st century, Artificial Intelligence had successfully challenged human professionals in chess, but they were still not able to defeat human professionals in 19 x 19 go game because it is very complicated and very deceptive for traditional algorithms. In 2006, two Hungarian researchers introduced Upper Confidence Bound Apply to Tree (UCT) [11], which combines Monte Carlo Method Tree Search and Upper Confidence Bound functions. It outperformed in searching information in large scale. This new method then successfully defeated human professionals in 9x9 go game.

Before 2010, RL is however mainly applied in intelligent robotics controlling. After 2010s, RL gained more and more interests. In 2014, Ian J. Goodfellow with his colleagues created a new class of machine learning system, called Generative Adversarial Network (GANs) [12]. It lets two different neural networks to contest with each other and generated new data set. GANs also had very broad applications, such as improving images and outperforming in video games. In order to generate real enough images, GANs has two components, one generator and one discriminator, where discriminator is to find out whether an image is real or artificial and generator is to generate image real enough to fool
the discriminator. This method is similar to Actor Critic Method in Reinforcement Learning [13]. When comparing GANs and Actor-Critic method, researchers find out there is similarity between these two algorithms, where generator acts like actor, while discriminator performs like critic. Another similarity is that both algorithms are known for difficulty for optimizing. Hence, it is possible and reasonable for applying technique in one field to another.

In the same Year, Silver brought forward the Policy Gradient algorithm [14], which targets at directly obtaining the optimal policy. In 2015 DeepMind introduced Deep Q Network (DQN) [15]. Because in some applicable situations, rewards are hard to evaluate. Unlike the previous two value-based algorithms, Policy Gradient is a policy-based algorithm, aiming at solving problems that is hard to find exact value for rewards. In 2016, AlphaGo defeated human professional in 19x19 go game, which marked a huge step in RL development.

Even though it is under fast development, RL actually still stays in a rudimentary status. Its high cost of money and time of learning process, its low reproductivity, and catastrophic instability are still obstacles in applying to real problems.

3. Game Theory
Game Theory is one branch of modern Mathematics and an important subject of Operational Research, which focuses on problems with contentious properties. It studies every participant’s policy and real behavior in the same game, and furthermore studies how to optimize the policy. One of the most key assumption of Game Theory is that every player must be rational and every person must aim at maximizing his or her own benefits.

Game Theory was firstly introduced in 1928 by Emile Borel and Von Neumann for their studying whether ideas of Game Theory could be applied to the military. In 1944, Neumann and Morgenstem published “Theory of Games and Economic Behavior” [16], which was a hallmark in development in Game Theory. Initially, game theory’s applications were limited to its low level of theoretical frameworks. But after several decades of deepening, generalizing, and refining, Game Theory has already become one of the most crucial tools in solving problems within broad branches of subjects. In 1950, the mathematical discussion about the famous Prisoner’s Dilemma emerged. John Nash’s “Nash Equilibrium” had a far-reaching influence in modern social science [17]. Since 1970s, Game Theory has been extensively applied in biology and economics, and so many mathematicians won Nobel Prize in Economics for their breakthrough in studies of Game Theory. In 1994, John Nash, John Harsanyi, and Reinhard Selten were awarded Nobel Prize in Economics for their pioneering contribution to “analysis of equilibrium in non-cooperative game”. In 2005, Aumann won Nobel Prize in Economics for “having enhanced the understanding of conflicts and cooperation through game theory”. And in 2012, Shapley also won the Nobel Prize in Economics for “the theory of stable allocations and the practice of market design”.

One important concept in Game Theory is Zero-Sum Game, in other words, non-cooperative game. It means that under the strict contention, there will not be a mutually beneficial situation in the game. The net revenues of both players in the game equal to zero, and this is why it is impossible for players to cooperate in this situation. Zero-Sum Game attracted such attention from mathematicians because the model in theory assembles to various kinds of real-life problems. Neumann proved that, assuming each player in the game aims at constraining the opposite’s benefits as much as possible, then via linear calculations, it is possible for obtaining a result in which each step is displayed as probability distribution.

4. Cooperative Game and Non-cooperative Game’ model applied in DRL
4.1. DRL under non-cooperative game
Usually, obtaining human’s expertise knowledge is extremely difficult and time consuming [18]. However, the fictitious self-play is an efficient method for training an agent to acquire such knowledge without any pre-studied data. Under non-cooperative game model, each agent targets at undermining opponents’ benefits as much as possible. Fictitious agents [19], assuming their opponents’ behaviors for
next step, then choose the best response behaviors. One advantage of self-play is that this method always ensures the difficulty is appropriate for AI agent to improve because the winning rate is always 50 percent. Hence, here is a smooth path by which the agent will gradually learn better strategies. Another advantage is that self-play does not rely on environment. It could be operated in a simple environment. For the above advantages mentioned, self-play is pervasively used in one-to-one non-cooperative competition.

4.2. DRL under cooperative game
Cooperative game, or known as positive-sum game, is much more attractive recently because cooperation becomes more dominant in modern society. The benefits after cooperation should exceed the sum of benefits of each agent before cooperation. In other words, cooperation is seeking for minimizing the total lost. Then the DRL problem is naturally converted into a mathematical problem. The global minimum of the total lost is actually the saddle point of the total lost function. The followings are some popular methods to obtain such a saddle point.

One intriguing method is called Simulate Anneal (SA) [20], which was created by Kirkpatrick, S.; Gelatt Jr, C. D.; Vecchi, M. P. in 1983. Its name comes from the metallurgy, a technique used for heating and cooling under control for increasing the material’s crystalizing. It actually works as a probabilistic technique to approximate the global minimum of any given function. To be more specific, in large scale search space for optimal problem, especially Travelling Salesman Problem, SA is quite popular. On the contrary, if an approximate global optimum is more preferable than an accurate local optimum, then another method, Gradient Descent [21], is a good alternative. Gradient Descent is a first-order iterative optimization algorithm for finding the minimum of a function. In game theory, this method is to obtain the global minimum of the lost function. Because what it obtains is a local minimum and not all of the parameters are tested, it is impossible to determine the global minimum. It is necessary to randomize the initial parameters numerous times to acquire sufficient amount of local minimum in order to approximate global minimum. In order to obtain higher quality and higher speed of convergence, researchers sometimes will randomly add disturbance to make the range of data larger [22].

5. Conclusion
Nowadays. The most challenging problem in DRL is the difficulty in reproducing the experimental results [23]. To achieve general AI, an agent should be able to finish tasks in different environments, but actual experiments’ variances are still too large to be stably applied to real life. Furthermore, the majority of the most recent accomplishment in RL are model-base algorithm. However, in real life, it is hard to build up an appropriate model for every problem encountered. Hence, effective and efficient scenario-based algorithm is more expected for researchers. In the future, training cost saving and reproducing the experiment results will be the main targets for the researchers in studies of Artificial Intelligence.

References
[1] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel& Demis Hassabis (2017). Mastering the game of Go without human knowledge. Nature volume 550, pages 354–359
[2] Henry Charlesworth (2018). Application of Self-Play Reinforcement Learning to a Four-Player Game of Imperfect Information. CoRR, vol. abs/1808.10442
[3] Gharehshiran, O.N., Hoiles, W., & Krishnamurthy, V. (2015). Reinforcement Learning and Nonparametric Detection of Game-Theoretic Equilibrium Play in Social Networks. ArXiv, abs/1501.01209.
[4] Brittain, M., & Wei, P. (2019). Autonomous Air Traffic Controller: A Deep Multi-Agent Reinforcement Learning Approach. ArXiv, abs/1905.01303.
[5] You, Y., Li, L., Guo, B., Wang, W., & Lu, C. (2019). Combinational Q-Learning for Dou Di Zhu. ArXiv, abs/1901.08925.

[6] Li, Y. (2017). Deep Reinforcement Learning: An Overview. ArXiv, abs/1701.07274.

[7] Bellman, R. (1957). Dynamic Programming. Princeton University Press, Princeton.

[8] Watkins, C. (1989). Learning from delayed rewards.

[9] Sutton, R.S., & Barto, A.G. (1988). Reinforcement Learning: An Introduction. IEEE Transactions on Neural Networks, 16, 285-286.

[10] Rummery, G.A., & Niranjan, M. (1994). On-line Q-learning using connectionist systems.

[11] Kocsis, L., & Szepesvári, C. (2006). Bandit Based Monte-Carlo Planning. ECML.

[12] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). Generative Adversarial Nets. NIPS.

[13] Pfau, D., & Vinyals, O. (2016). Connecting Generative Adversarial Networks and Actor-Critic Methods. ArXiv, abs/1610.01945.

[14] Silver, D., Lever, G., Heess, N.M., Degris, T., Wierstra, D., & Riedmiller, M.A. (2014). Deterministic Policy Gradient Algorithms. ICML.

[15] Hasselt, H.V., Guez, A., & Silver, D. (2015). Deep Reinforcement Learning with Double Q-learning. AAAI.

[16] Barna, T., Neumann, J.V., & Morgenstern, O. (1944). Theory of Games and Economic Behaviour.

[17] Nash, J.F. (1951). NON-COOPERATIVE GAMES.

[18] Heinrich, J., & Silver, D. (2016). Deep Reinforcement Learning from Self-Play in Imperfect-Information Games. ArXiv, abs/1603.01121.

[19] Brown, G.W. (1951). Activity analysis of production and allocation. Iterative Solutions of Games by Fictitious Play. 374-376.

[20] Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. Science, 220 4598, 671-80

[21] Sutton, R. (1986). Two problems with backpropagation and other steepest-descent learning procedures for networks.

[22] Rueda, M.D., & Gonzá lez, S. (2008). A new ratio-type imputation with random disturbance. Appl. Math. Lett., 21, 978-982.

[23] Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2017). Deep Reinforcement Learning That Matters. ArXiv, abs/1709.06560.