CS480 Project (Literature Survey)

Zexuan Jing(z4jing)

Yinong Wang (y2346wan)

Jiahui Cheng (j69cheng)

Introduction:

One of the key reason that artificial intelligence exists is to solve problems for humans. Among all sorts of different problems, gaming is gradually recognized as an ideal type of task to solve because it simulates many challenges in real life. By building AI to learn to succeed in all kinds of games, we can avoid doing unnecessary trial and error in real life and simply migrate strategies that AI learned in games as well as our experiences in training these agents. However, games are sometimes as complex as real-life problems. It still remains a challenging question of how to train agents that perform well in all types of games. In this survey, we studied a series of papers to see how reinforcement learning gradually develops and evolves over recent years from the very basic q-learning method to modern multi-agent deep reinforcement learning. The following is the list of papers that we surveyed:

- Q-learning https://link.springer.com/article/10.1007/BF00992698
- Friend-or-Foe Q-learning in General-Sum Games https://www.researchgate.net/profile/Michael_Littman2/publication/2933305_Friend-or-Foe_Q-learning_in_General-Sum_Games/links/54b66cb80cf24eb34f6d19dc/Friend-or-Foe-Q-learning-in-General-Sum-Games.pdf
- Deep Q Network https://www.nature.com/articles/nature14236/
- Episodic Exploration for Deep Deterministic Policies: An Application to StarCraft Micromanagement Tasks https://arxiv.org/abs/1609.02993
- Markov games as a framework for multi-agent reinforcement learning https://students.cs.byu.edu/~cs670ta/Fall2009/MinimaxQLearning.pdf

- Deep Reinforcement Learning from Self-Play in Imperfect-Information Games https://arxiv.org/pdf/1603.01121.pdf
- Learning to Communicate with Deep Multi-Agent Reinforcement Learning https://arxiv.org/pdf/1605.06676.pdf

Survey:

The first method to be discussed in this survey is the classical method of “q-learning”. The method is described in “Q-Learning” by Watkins and Dayan (Watkins, 1992). Q-learning is a simple way for agents to be able to learn how to act optimally in controlled Markovian domains. It consolidates the idea of dynamic programming and ensures agents to progressively take the most optimal steps, calculated by maximizing the expected future reward of every possible action. Although q-learning is an important milestone and breakthrough in reinforcement learning, it bears many shortages, for example, unwarranted convergence in many cases and potentially large probability table.

While q-learning was the brand new reinforcement learning breakthrough, it is only applicable in single-agent games, which left multi-agent settings still challenging circumstances. The paper “Markov games as a framework for multi-agent reinforcement learning” is to introduce a reinforcement learning strategy named *Minimax-Q algorithm* for multi-agent Markov games. The learning algorithm is a modified version of the *value iteration algorithm* (Bertsekas, 1987). The idea is: similar to Markov decision process, we redefine V(s) to be the expected reward for the optimal policy starting from state s, and Q(s, a, o) as the expected reward for taking action a when the opponent chooses o from state s and continuing optimally thereafter. In the game theory literature, the resolution to this dilemma is to eliminate the choice and evaluate each policy with respect to the opponent that makes it look the worst. So our goal here is to update values $V(s) = \max_{\pi} \min_{o} \sum_{a} Q(s, a, o) * \pi$ of states using the technique of *Minimax-Q algorithm* (adding a linear program to *value iteration*). The article gives a solution

to the zero-sum Markov games with two agents in details and of course we could generalize the idea to multi-agents Markov games. However, some weakness exists:

1. The linear program in the algorithm would increase runtime and makes our training process less efficient.

2. It is not always possible to figure out every possible action of an agent (for example a continuous range of actions)

3. The solution is guaranteed to converge to Nash-equilibrium but always assumes the opponent also uses strategy to converge to Nash-equilibrium. But when the opponent chooses a worse strategy, the algorithm would not generate a better strategy. It would get the same result in the previous case.

Given the shortages of q-learning and minimax-Q, enhancements are brought into the fields over the years. A variation of Q-learning named “Friend-or-Foe Q-learning (FFQ)” is able to successively solve a slightly more general game, the general-sum game (Littman, 2003). The idea of FFQ is motivated by the high restrictions on game assumptions of previous methods. FFQ ensures convergence under more relaxed and realistic conditions. The paper analyzes many types of behaviours when facing different opponent strategies under both adversary and cooperative settings in general-sum games. Compared to previous q-learning methods, besides the better conditional convergence, FFQ does not require learning estimates of the Q function for opponents and can be more easily implemented in multi-agent games. However, like other previous q-learning methods, if neither coordination or adversarial equilibria exist, FFQ still fails to find the equilibria.

Up to FFQ, Q-learning is only capable of solving a very narrow range of games which are limited to domains that can be represented in low dimensions. The family of Q-learning methods encountered their bottleneck in complex games until the revival of artificial neural networks, especially the prosper of deep neural network. In 2015, a paper named “Human-level control through deep reinforcement learning” from Nature introduced the promising idea of “deep Q-network (DQN)” (Mnih, et al., 2015). The DQN method

consolidates the power of deep convolutional networks (CNN) with the biologically inspired mechanism - “experience replay”. CNN provides the ability to learn representations of convoluted input from games and generalizations of policies. The experience replay mechanism allows the DQN to discover long-term strategies and critically ensures the successful integration of deep network architectures. By forming this end-to-end reinforcement learning pipeline, the DQN method outperforms the best existing reinforcement learning methods at that time on 43 of Atari 2600 games and beat the human scores on more than half of the games. Despite the promising breakthrough of DQN, temporally extended planning strategies remains a major challenge. DQN is also lacking the optimizations in its model structure as well as the use of biasing experience replay towards salient events.

Great potentials have been exploited in individual agents because of the birth of DQN.
As each agent becomes much more competent and intelligent, large interests have shifted toward more complicated multi-agent games that are closer to real-life problems. Specifically, we studied three major aspects of modern multi-agent games. The first and most critical challenge is to control each agent simultaneously. Once the control is gained, cooperation is the next necessary step and communication, in particular, is the start of cooperation. Last but not least, just as information is never fully exposed in real life, it is crucial for agents to success with imperfect information. In order to achieve better AI agents, we must tackle these three challenges.

The very first difficulty to overcome is searching for the optimal method for coherently controlling multiple agents, the solution, which is addressed in the paper “Episodic Exploration for Deep Deterministic Policies”. The contribution is twofold: At first, it transfers the winning goal of StarCraft as benchmarks for reinforcement learning with actions lasting long enough, postponed rewards, and large action spaces making random exploration infeasible. Secondly, it introduces a reinforcement learning algorithm with zero-order optimization technique claiming to outperform all previous methods in terms of efficiency for discrete action spaces by applying robust training and episodically consistent exploration in

policy space. The RL algorithm combines direct exploration in the policy space and backpropagation for weighting the optimization efficiency and update of parameters. It also allows for the collection of traces for learning using deterministic policies, which appears more efficient than most other algorithms.

In addition to coherently control multiple agents, allowing efficient communication among multiple agents is of vital importance. A deep learning approach of learning communication protocols is firstly introduced in “Learning to Communicate with Deep Multi-Agent Reinforcement Learning”. The benchmark is to maximize the combined utility between multiple agents against the environment. Two key requirements are participations of multiple agents and imperfect information each is capable of receiving. The paper introduces two approaches: Reinforced Inter-Agent learning (RIAL) and Differentiable Inter-Agent Learning (DIAL). The former combines deep Q-learning with a recurrent network DRQN to address the requirement that each agent receives imperfect information and exploits the advantage of parameter sharing. The latter addresses the limitation of RIAL, no feedback loop during communication, by combining centralized learning and Q-networks, as centralized learning affords more opportunities to improve learning than just parameter sharing by allowing real-valued messages to pass between agents. It follows that gradients can be back-propagated through communication channels, resulting in a system trainable. While during execution, it adopts a decentralized approach as it allows real-valued messages to be discretized and mapped to the discrete set of communication actions allowed by the task. The implementation adopts a deep learning approach by exploiting the opportunities of centralized learning.

Last but not least, it is natural to consider improving the models we have so that they become appropriate for the more severe environment because the real world is not always fully observable and deterministic. In the paper “Deep Reinforcement Learning from Self-Play in Imperfect-Information Games”, a deep reinforcement learning algorithm is introduced: using self-play in an imperfect-information game. This procedure is the first scalable end-to-end

approach to learning approximate Nash equilibria without prior domain knowledge (for example Leduc poker) that is called *Neural Fictitious Self-Play* in short *NFSP*. NFSP combines *FSP* (Fictitious Self-Play) (Heinrich et al., 2015) with neural network function approximation. The idea is: An NFSP agent interacts with its fellow agents and memorizes its experience of game transitions and its own best response behaviour in two memories. NFSP treats these memories as two distinct datasets suitable for deep reinforcement learning and supervised classification respectively. The agent trains a neural network to predict action values from data in the first memory. The resulting network defines the agent's approximate best response strategy. A separate neural network to imitate its own past best response behaviour on the data in the other memory. This network maps states to action probabilities and defines the agent's average strategy. During play, the agent chooses its actions from a mixture of its two strategies. NFSP also ensure the stability of the resulting algorithm as well as enable simultaneous self-play learning using *reservoir sampling* (Vitter, 1985) and *anticipatory dynamics* (Shamma and Arslan, 2005). NFSP agents choose their actions from the mixture policy. In this way, NFSP is scalable without prior domain knowledge and is the first deep reinforcement learning method known to converge to approximate Nash equilibria in self-play. The advantage for NFSP: using a slowly changing (anticipated) average policy to generate self-play experience. Thus agents' experience varies more smoothly, resulting in more stable data distribution and more stable neural networks.

Analysis:

Applying deep learning to learning communication protocol is the current state of the art model for multi-agent communication and zero ordering is the most update to date optimization technique for reinforcement learning for discrete action spaces by applying robust training and episodically consistent exploration in policy space.

Future directions include optimizing unit movement, variations of CNN-based model preserving the 2G geometry of the game while embedding the discrete components of the state-action pair, zero ordering optimization technique, exploring domains besides StarCraft,

e.g: Atari, experiment including self-play, multi-map training, other complex scenarios except circumstance in which actions are either moving or attacking, e.g: recruiting units and manipulating them. Moreover, understanding communication and language of agents, covering compositionality, making more conversational agents and many other open problems.

Conclusion:

This survey should be seen as a summary of these seven papers, it is evident that they are highly related. From old to new: we could see both innovations in the learning method and improvement of the scope of application, in other words, could be more widely used. Generally, reinforcement learning being used from single-agent games to multi-agent games. In detail, from Q-learning to DDPG, the Q-learning model keeps being updated, consequently becomes more expressive and efficient; from MARL to NFSP, Markov model could be applied to more cases in practice (zero-sum game to general-sum game to imperfect-information game). Learning all this knowledge, we not only understand reinforcement learning better but also become capable to apply reinforcement learning to solve problems in the game industry like (designing game AI) or even relevant fields. The RL procedure could be applied in analogical systems that have similar features such as multi-agents, blind(imperfect-information) environments and so on.

Moreover, aiming at currently existing problems, we recommend future research about understanding agents' communication and language in their full splendour, conversational agents, optimizing the unit movement of agents and exploring more domains in games because the environment in games is in development and becomes more and more complicated. There are many other problems still lie ahead and we are optimistic that the approaches proposed in this survey would play a substantial role in tackling new challenges in the future.

Reference:

[1] Watkins, Christopher JCH, and Peter Dayan. "Q-learning." *Machine learning* 8.3-4 (1992):

279-292.

[2] Littman, Michael. (2003). Friend-or-Foe Q-learning in General-Sum Games.

[3] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. Nature 64(2), p. 10-12.

[4] Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In Machine learning proceedings 1994 (pp. 157-163). Morgan Kaufmann.

[5] Heinrich, J., & Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. arXiv preprint arXiv:1603.01121.

[6] Usunier, N., Synnaeve, G., Lin, Z., & Chintala, S. (2016). Episodic exploration for deep deterministic policies: An application to starcraft micromanagement tasks. arXiv preprint arXiv:1609.02993.

[7] Foerster, J., Assael, I. A., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. In Advances in Neural Information Processing Systems (pp. 2137-2145).

[8] Bertsekas, D. P. 1987. Dynamic Programming: Deterministic and Stochastic Models. Prentice-Hall.

[9] Heinrich, J., Lanctot, M., and Silver, D. (2015). Fictitious self-play in extensive-form games. In Proceedings of the 32nd International Conference on Machine Learning.

[10] Heinrich, J. and Silver, D. (2015). Smooth UCT search in computer poker. In Proceedings of the 24th International Joint Conference on Artificial Intelligence.

[11] Vitter, J. S. (1985). Random sampling with a reservoir. ACM Transactions on Mathematical Software.

[12] Shamma, J. S. and Arslan, G. (2005). Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria. IEEE Transactions on Automatic Control, 50(3):312–327.