Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/158/2025.TJ23325

Advances and Applications in Multi-Agent Reinforcement Learning

Jingyuan Wang

School of Information Engineering, Capital Normal University, Beijing, China
1221005180@cnu.edu.cn

Abstract: In today's life, reinforcement learning is applied to many fields and has shown good results, such as driverless cars and Warehouse sorting goods robots. From the beginning of simple single-agent intelligent control, researchers gradually turn their attention to multi-agent cooperative games to solve production problems. Compared with the single-agent training algorithm, multi-agent reinforcement learning is more complex and difficult to converge. Therefore, it is particularly important to study the progress and application of multi-agent reinforcement learning. This paper reviews three perspectives, which are the introduction of reinforcement learning algorithms, the application scenarios of multi-agent reinforcement learning (games, path planning, and robot cooperation), the problems faced, and future prospects. It is hoped that this review can provide researchers with more comprehensive information for subsequent in-depth research. At the same time, this paper puts forward more innovative ideas, hoping that researchers can expand their ideas and promote the development of multi-agent reinforcement learning.

Keywords: Multi-agent reinforcement learning, Path planning, Robot cooperative, Game.

1. Introduction

Reinforcement learning is an important part of the field of machine learning. Reinforcement learning is a process in which an agent finds the optimal path or solution by repeatedly trying to obtain rewards or punishments, and finally learning through experience.

Reinforcement learning is now used in all areas of life. Single-agent reinforcement learning applications are often used in areas such as single automation and intelligent control. Intelligent manipulators can finally achieve basic tasks such as grasping, recognition, and welding on the production line through reinforcement learning, thereby improving production efficiency. The application of multi-agent is often seen in more complex tasks involving cooperation, competition and games. For example, the optimization of traffic lights, and multiple intersection lights work together to achieve more intelligent signal control, thereby reducing urban vehicle congestion. The application of multi-agents in the field of warehouse robots should be mainly reflected in the collaboration and coordination between robots, which need to jointly plan the optimal route to avoid collision with each other, and finally realize the improvement of transportation efficiency. MARL can also be applied to games, where multi-agents constantly learn and optimize tactics through interaction with the game environment, improve their decision-making ability, and then play against human players.

DOI: 10.54254/2755-2721/158/2025.TJ23325

In this paper, three aspects are reviewed. Firstly, this paper introduces the classical algorithm of reinforcement learning, from single agent to multi-agent, and introduces the latest algorithm. Secondly, this paper details the application of multi-agent reinforcement learning in games, path planning, and robot cooperation. Finally, this paper discusses the limitations of today's multi-agent reinforcement learning and future directions. The purpose of this review is to provide researchers with a more systematic understanding and provide more development possibilities for the field of reinforcement learning.

2. Basic theory

2.1. Reinforcement learning environments

In reinforcement learning, the agent is in an external world with which it can interact. This external world is called the environment. The environment needs to receive the action taken by the agent and return the updated state and reward after the action taken by the agent.

The environment in reinforcement learning consists of several important parts. The first is the state space, which is the set of all possible states in the environment. The second is the action space, which is the set of all actions the agent can take. The third is the reward function, which is a scalar returned by the environment after the agent performs an action to measure the value of that action. The fourth is the transition function, which defines the probability distribution of the environment transitioning to the next state after the agent performs an action. The fifth is the termination condition, where there is some state in the environment that causes the reinforcement learning task to end, such as the agent finding an exit, or the end of game time.

Various environments will be used in the experiment process, and the more common ones are melting pot and Pogema. meltingpot provides a variety of game environments and can call external libraries for agent training, so that the algorithm can be designed flexibly. Pogema is to design a grid environment to enable agents to achieve local observation, which is closer to the real scene.

2.2. Reinforcement learning algorithms

Reinforcement learning algorithms are defined in terms of different learning methods. Depending on the algorithm, it can be used to better solve specific problems. Some algorithms are value-based, some are policy-based, some are model-based, or a mixture of these to achieve efficient reinforcement learning algorithms.

2.2.1. Deep Q-Network algorithm

The deep Q-Network (DQN) algorithm is a relatively mature algorithm, which uses neural networks in deep learning to mitigate fluctuations in the reinforcement learning process. The DQN algorithm is optimized based on Q-learning. The Q-learning algorithm uses a table to store each state and its corresponding reward value, and the DQN algorithm makes an improvement on this. The DQN algorithm uses a neural network to predict each state and its corresponding reward value, which solves the problem that Q-learning cannot deal with high-dimensional state space.

The training steps of DQN are as follows: First, the current state is obtained from the environment, and the Q network is used to predict the Q value in the current state. Then an action is selected, and the selected action is executed to get the next state and reward, and the experience is stored in the experience replay pool. Finally, a batch of samples were randomly sampled from the replay pool for training, the parameters of the Q network were updated, and the parameters of the Q network were copied to the target network for learning at intervals [1].

DOI: 10.54254/2755-2721/158/2025.TJ23325

2.2.2. Proximal Policy Optimization algorithm

The predecessor of the Proximal Policy Optimization (PPO) algorithm is the TRPO algorithm, which is based on policy gradient. In this algorithm, it needs to find the maximum expected reward and update it to the stored network. During the training process, the PPO algorithm specifies the advantage function, so that the agent will be closer to the behavior that obtains more rewards in the strategy, avoiding the occurrence of local optima and missing better solutions. The PPO algorithm also specifies a truncation function to limit the step size of the training iteration, which also makes the reinforcement learning training of the algorithm more stable [2].

2.2.3. MuZero algorithm

MuZero is a relatively cutting-edge algorithm, based on AlphaZero. The MuZero algorithm, directly constructs the MDP model, which directly learns and predicts the relevant data without relying on the environment that the agent depends on.

The MuZero algorithm is able to show great learning ability in the absence of an environment, so it is suitable for environments where only local observations can be made, and it does not rely on predefined models, and it performs well in more complex games. Compared with the DQN algorithm and PPO algorithm, the algorithm is more efficient.

2.3. Multi-agent reinforcement learning

The process of multi-agent learning, compared with the single agent will be more complex. Because the process of multi-agent reinforcement learning involves competition and cooperation between agents, as well as more complex interactions. In multi-agent learning, researchers need to maximize the agent's own benefit or the global benefit. This also greatly improves the research difficulty of multi-agent reinforcement learning.

There are many kinds of states between multi-agents, including perfect competition, full cooperation, mixed scenarios and partial environment observability. Based on this, researchers can invoke different algorithms for reinforcement learning training.

For example, in the perfect competition scenario, it means that one party must be punished while one party gains. The interests of the two parties are mutually exclusive, which is often seen in confrontations and board games. At this time, adversarial extension algorithms of DQN and PPO can be used. In the case of full cooperation, the task to be accomplished is to maximize the benefit obtained by the multi-agent population.

3. Application areas

3.1. Games

Reinforcement learning training has been successful in two-agent games, but little work has been done on multi-agent Settings. In today's research, more people turn their attention to the multi-agent game method.

In the game of the Three Kingdoms killing, there is a scene of four agents fighting against each other. Researchers model the cooperation of multiple agents based on the idea of strategy gradient, and the cooperation and confrontation between multi-agents are included in the decision-making process. This algorithm is proposed to solve the instability problem in a multi-agent environment, And the final team reward is at least 12% higher than the winning rate [3].

In the research of reinforcement learning at home and abroad, there is always a problem of lack of generality in multi-agent reinforcement learning. Researchers have proposed the idea and algorithm

DOI: 10.54254/2755-2721/158/2025.TJ23325

of parallel games. This algorithm incorporates psychology, information, simulation and decision making. It performs better than single-role games and multi-role games in solving problems[4].

The application of multi-agent in the field of games can make researchers better apply the experimental results of virtual space to the real field, which is a very important research.

3.2. Path planning

Reinforcement learning is widely used in path planning. Multi-agents need to cooperate to learn and be more efficient in daily life and military fields.

In order to cope with the complex sea battlefield environment, taking multiple ships as the research object, on the basis of the DQN algorithm, researchers add networks with the same structure and different parameters and update the actual value and estimated value of Q to realize the convergence of the value function. This method uses the experience replay and the target network double parameter update mechanism, effectively solving the problem of unstable neural network convergence and so on. This method improves the ship's obstacle avoidance ability by more than two times [5].

In the complex environment of multiple UAVs, the path planning problem is particularly important. In order to realize the autonomous route planning of UAVs, a flight path planning method based on multi-agent reinforcement learning theory is proposed. In this algorithm, two agents with different functions corresponding to local and global path planning are used to divide and abstract the state and action space, which effectively reduces the number of states and solves the problem of excessive dimensionality in reinforcement learning. The final algorithm shows a faster convergence speed and successfully solves the task of path planning [6].

Researchers also proposed the use of Multi-agent Proximal Policy optimization based on the Network Pruning (NP-MAPPO) algorithm. This algorithm improves the training efficiency, and the simulation results also show the superiority of the algorithm in training time [7].

In the search and rescue of multiple UAVs, the researchers also used a partially observable Markov decision process (Dec-POMDP) and Double deep Q-network architecture (Double DQN) to make the UAV control strategy optimal. And multiple UAVs can achieve effective cooperation [8].

3.3. Robot cooperation

Robot collaboration is also an important application scenario for multi-agent learning. In complex areas such as warehouses, which are difficult for humans to navigate, robots need to cooperate to find and deliver goods and prevent collisions with other robots in the process of delivery.

When a robot agent selects an action, it is often influenced by the actions performed by other agents. Therefore, reinforcement learning in the field of robot cooperation needs to consider multi-agent joint states and actions. Researchers propose to use probabilistic neural networks to predict the actions of other agents. This behavior can constitute a joint action of multiple agents [9].

In the traditional path planning method, the robot can not achieve high handling efficiency. Researchers proposed the QTRAN Plus algorithm based on multi-agent reinforcement learning to participate in path planning training. This algorithm improves the convergence speed, The overall performance is also better than the traditional algorithm [10].

DOI: 10.54254/2755-2721/158/2025.TJ23325

4. Existing limitations and future prospects

4.1. Limitations

4.1.1. Generalization ability

In the existing reinforcement learning training, there are some problems, such as the generalization ability of the reinforcement training results of multi-agent is not strong. For example, in the grid environment of Pogema, multi-agents are trained to find exits. In the state of the fixed maze, good training results can be shown. However, in the random generation of the new environment, the cooperation between multi-agents and the search for exits will appear oscillation and other conditions, which cannot adapt to the new environment well.

Therefore, researchers need to design algorithms with stronger generalization ability when training multi-agents, so that multi-agents can flexibly solve problems in more complex and diverse environments.

4.1.2. Training speed

In the training of multi-agent reinforcement learning, too long training time is also a problem. Too slow training speed will affect the efficiency of the product, resulting in a waste of time and computing power resources.

The main reason for the slow training speed is that as the number of multi-agents increases, the state space and action space of the whole environment algorithm will grow exponentially, and small changes in each individual agent can affect the global result. So the convergence rate will be slower and the training time will be longer.

In order to solve the problem of slow training speed, researchers can design more optimized and efficient algorithms that use computing resources with faster training speed.

4.1.3. Stability

During MARL training, it's easy to fall into local optima, such as oscillating behavior somewhere and not moving forward to get more rewards.

In the training process of multi-agent training, in complex scenarios such as multi-agent cooperation and competition, the multi-agent needs to adjust its strategy many times, which leads to the strategy failing to converge in a short time, producing oscillating behavior, and falling into the suboptimal solution, unable to find the global optimal solution.

4.2. Future prospects

To address the generalization problem, researchers can explore the incorporation of large-scale models, such as large language models (LLMs) or vision-language models, which hold significant promise for enhancing the efficiency and effectiveness of multi-agent reinforcement learning (MARL). LLMs, in particular, can facilitate natural language communication among agents, enabling more intuitive and flexible coordination in complex tasks. This capability is especially valuable in scenarios involving human-agent collaboration or tasks that require symbolic reasoning.

Regarding the issue of training speed, environment modeling becomes a critical area of focus. By leveraging large generative models to construct rich, dynamic, and realistic simulated environments, it is possible to significantly improve training efficiency. These environments more accurately capture the complexity and variability of the real world, helping to bridge the sim-to-real gap and accelerating policy learning.

DOI: 10.54254/2755-2721/158/2025.TJ23325

For the challenge of stability, integrating large models with strong generalization and representation capabilities can lead to more stable policy updates. Such integration allows agents to better interpret environmental signals and the behavior of other agents, ultimately promoting convergence and coordination in multi-agent systems.

Looking ahead, the synergy between MARL and large-scale models is expected to play a pivotal role in developing robust, adaptive, and intelligent agent systems that can operate effectively in complex and ever-changing real-world environments.

5. Conclusions

This paper revolves around multi-agent reinforcement learning. This paper reviews multi-agent reinforcement learning from three perspectives. Firstly, the basic algorithms and novel algorithms of multi-agent reinforcement learning are introduced. After the training algorithm matures, many multi-agent reinforcement learning applications are applied in real life.MARL has been integrated into many fields such as games, path planning, and robot cooperation, which shows the powerful application potential of multi-agent. However, when researchers apply multi-agent in real-world scenarios, they will also face many challenges. In the face of these limitations, people need to continue to innovate and envision the future.

In the future, reinforcement learning needs more researchers to address some limitations, such as weak generalization ability, slow training time, or inability to be directly applied to real-world problems. In the development of other fields, such as large models, researchers can combine MARL with other fields to bring new breakthroughs in the development of reinforcement learning.

This paper aims to provide a comprehensive overview of the current progress, challenges, and future directions in multi-agent reinforcement learning, serving as a foundation for further research and application in this evolving field.

References

[1] *Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.*

[2] *Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.*

[3] *Luo Furong, Wang Yisong, Qin Jin & Yu Xiaomin (2024). A Reinforcement learning based Three States kill Multi-agent game method. Computer Simulation,41(07),484-490.*

[4] *Shen Yu, Han Jin-Peng, Li Lin-xi & Wang Fei-yue (2020). AI in Game Intelligence: From multi-role game to parallel Game. Journal of Intelligent Science and Technology,2(03),205-213.*

[5] *Yu Zhou, Bi Jing & Yuan Haitao (2022). Path planning method for complex sea battlefield based on improved DQN algorithm. Journal of Intelligent Science and Technology,4(03),418-425.*

[6] *Li Donghua, Jiang Ju & Jiang Changsheng (2009). Flight path planning algorithm based on multi-agent reinforcement learning. Electrooptics and Control,16(10),10-14.*

[7] *Si Pengbo, Wu Bing, Yang Ruizhe, Li Meng & Sun Yanhua (2023). Uav Path Planning based on multi-agent deep reinforcement learning. Journal of Beijing University of Technology,49(04),449-458.*

[8] *Guo Tianhao, Zhang Gang, Yue Wenyuan, Wang Qian& Da-bo guo. (2022). Based on the multi-agent reinforcement learning of unmanned aerial vehicle (uav) indoor auxiliary rescue. Computer system application, 31 (02) 88-95. The doi: 10.15888 / j.carol carroll nki. Csa. 008302.*

[9] *Duan Yong & Xu Xinhe.(2014). Research on Multi-robot cooperation Strategy Based on Multi-agent reinforcement Learning. Systems Engineering Theory and Practice,34(05),1305-1310.*

[10] *Liao Dengyu, Zhang Zhen, Zhao Dejing & Cui Haoyan. (2023). Based on the depth of multi-agent reinforcement learning method of robot cooperation handling. Electronic design engineering, 31 (23), 7 to 11. Doi: 10.14022 / j.issn1674-6236.2023.23.002.*