Proceedings of the 5th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/115/2025.18525

Research and Application Analysis of Reinforcement Learning Algorithms

Xiamo Dou[1], Rundong Zhou[2,a,*]

[1]School of Wuxi Huatian Bilingual School, Jiangsu, China
[2]School of Oriental College of Science and Technology, Hunan Agricultural University, Hunan, China
a. zhourundong@ldy.edu.rs
**corresponding author*

Abstract: With the continuous development of the times, there have been many significant breakthroughs in the field of reinforcement learning, not only reflected in the improvement of operational efficiency, but also in the emergence of new methods, all of which contribute to the continuous concretization of reinforcement learning in reality. This article mainly introduces the current research status of reinforcement learning, the introduction of common basic reinforcement learning algorithms such as value function estimation, policy gradient method, and Actor Critic algorithm. In recent years, a review of transfer learning, width planning and active learning based methods Olive, Reptile algorithm, evolutionary strategy (ES), two self play training schemes (Chainer and Pool), and the application analysis of reinforcement learning in the fields of gaming, robot control, transportation, healthcare, finance, and energy is conducted. This article aims to provide a review of the current research status in recent years, in order to provide a reference for the future development of reinforcement learning and indicate the existing problems in current research.

Keywords: Reinforcement learning, Deep Q-Network, Strategy.

1. Introduction

The development process of reinforcement learning as a research field includes early theoretical foundations and subsequent rapid development stages. Reinforcement learning gradually became independent from traditional machine learning methods and became a research field with a unique theoretical and methodological system [1]. Nowadays, deep reinforcement learning (DRL) and deep multi-agent reinforcement learning (Marl) have achieved significant success in many fields such as gaming, autonomous driving, and robotics. However, they still commonly suffer from the problem of low sample efficiency. Even dealing with relatively simple problems usually requires millions of interactions, which seriously hinders their widespread application in practical industrial scenarios. Among them, exploration problems are the key bottleneck challenge in improving sample efficiency, that is, how to efficiently explore unknown environments and collect interaction data that is conducive to strategy learning [2]. With the development of deep learning, the field of reinforcement learning has become a powerful learning framework that can learn complex strategies in a high-dimensional environment and has great application potential in the field of automatic driving. In addition to perception tasks, the auto drive system also has several task scenarios that are not suitable for classical

DOI: 10.54254/2755-2721/115/2025.18525

supervised learning, and reinforcement learning provides a promising solution for these tasks [3]. In multi-agent reinforcement learning, multiple agents interact, learn from each other, and make decisions in the same environment. Compared with single agent reinforcement learning, the environment in multi-agent systems is more complex and dynamic, as the behavior of each agent affects the state of other agents and the entire environment. Multi agent systems are typically described using the theoretical framework of stochastic games or Markov games. In this framework, the probability of state transition, timely rewards, and long-term returns of the environment all depend on the joint actions of multiple agents. Each agent has its own strategy and needs to continuously adjust its strategy based on the behavior of other agents and feedback from the environment to maximize its own benefits [4]. Reinforcement learning provides a framework and toolkit for designing complex and difficult to implement behaviors through traditional engineering methods in the development of robotics related technologies. On the contrary, the challenges posed by the problems of robots also provide inspiration, influence, and validation for the development of reinforcement learning. The relationship between these two fields has great potential, similar to the relationship between physics and mathematics, and the importance of their mutual promotion and collaborative development [5]. This article will analyze the current situation in the field of reinforcement learning by introducing some basic reinforcement learning algorithms, breakthroughs and improvement methods in the reinforcement learning field in recent years, and important areas of reinforcement learning applications.

2. Introduction to the Basic Theory of Reinforcement Learning

2.1. Value function estimation

In the process of implementing reinforcement learning, value function estimation is an important part, often used to evaluate the long-term value of taking corresponding actions in a specific state.

2.1.1. Monte Carlo Methods

The Monte Carlo method estimates the value function through training over multiple periods. During a certain period, the intelligent agent interacts with the environment based on certain strategies from an initial state until reaching a termination state. Its advantage is that it learns directly based on accumulated experience without the need for dynamic model assumptions about the external environment, especially for jobs with termination states. However, it also has disadvantages, such as having to wait until the end of a certain period before updating the value function. This shows the shortcomings of low learning efficiency, as well as the difficulty in applying this method to jobs or tasks without a clear termination state.

2.1.2. Temporal Difference Learning

The time difference learning algorithm is mainly based on the idea of combining dynamic programming and Monte Carlo methods. The specific implementation is to update the value function of the current state by estimating the value of the current state and the value of the next state. It solves the disadvantage of having to wait for a period of time to update the value function compared to Monte Carlo methods, and has the advantage of high learning efficiency. At the same time, the idea of dynamic programming's bootstrapping is combined to update the value of the current state using the estimated value of subsequent states. However, the combination of bootstrapping may introduce estimation bias.

DOI: 10.54254/2755-2721/115/2025.18525

2.1.3. State-Action-Reward-State-Action(SARSA)

SARSA is an algorithm based on the idea of time difference learning, which is similar to Q-learning algorithm. The difference is that SARSA updates the action value function by using the value of the next action actually executed, rather than the action with the highest value among all possible actions in the next state. Compared to Q-learning, SARSA is more conservative and less affected by estimation problems, but has lower learning efficiency in tasks with strong exploratory nature.

2.2. Policy Gradient

The policy gradient method in reinforcement learning is an algorithm that optimizes policies to maximize cumulative rewards, with the aim of finding policy parameters that maximize (or minimize) the policy objective function. Strategy is the basis for intelligent agents to select actions based on the current state, and for actions determined by a given state output, it can reflect the determinacy of the strategy; According to the probability distribution of the output action, it can be seen that the strategy also has randomness. By continuously adjusting strategy parameters, agents can obtain better strategies through continuous learning and gain more rewards in their interaction with the environment. The gradient represents the rate of change of the policy objective function with respect to the policy parameters, and updating the parameters along the direction of the gradient can gradually optimize the value of the objective function.

2.3. Actor Critic algorithm

The Actor Critic algorithm is also an important algorithm in reinforcement learning, which combines policy based and value based methods. The Actor Critic algorithm selects an action based on the current state, which generates an action decision. It is usually represented by a parameterized policy function, which includes parameters representing states and actions. The ultimate goal of an actor is to learn an optimal strategy that maximizes the cumulative rewards obtained over the long term. The role of Critic is to evaluate the quality of the current strategy, usually reflected in the value function or action value function, used to estimate the long-term value of taking a certain action in a given state. Critic provides guidance for Actor updates by observing the interaction process between the agent and the environment, and learning how to accurately evaluate the value of the current strategy during this process. The Actor Critic algorithm exhibits strong adaptability to different types of tasks and environments, can handle problems in discrete and continuous action spaces, and can adapt to different reward functions and environmental dynamics. It has high learning efficiency and faster convergence speed due to the combination of Actor and Critic.

3. Improvement and optimization of reinforcement learning

3.1. Reinforcement learning is a method of improving model performance and reducing training time through transfer learning

Sourabh Prakash's team from the University of California, USA, studied the application of reinforcement learning in Atari games[6], The validation method adopted is to compare the performance of RL models trained from scratch and models trained through transfer learning, and explore how to use transfer learning to improve performance and reduce the training time of reinforcement learning models through the above validation method. DQN improves the performance of Atari games and enables other reinforcement learning tasks to achieve optimal performance by approximating the optimal Q-function using deep neural networks and learning the Q-function through replay buffer and strategy objective network architecture. In the implementation process, use

DOI: 10.54254/2755-2721/115/2025.18525

the vector module of the gym library to create a vectorization environment and implement asynchronous parallel execution, improve sample generation efficiency, and use replay buffers to ensure algorithm stability. When learning from a specific environment, a pre trained encoder from a similar environment can be used, and the weights of the feature extractor can be frozen first. Then, a new header layer can be added and the model fine tuned, or a new uninitialized policy network can be added and trained from scratch. Choose a pre trained encoder, initialize the weights of the feature layer and header layer first, and fine tune the model on the new task to enable the encoder to converge quickly. The training model was trained on multiple Atari environments with the same action space and tested in new, unseen environments to explore the development of general game agents. The experimental results showed that transfer learning is more effective than training from scratch, significantly reducing training time and improving the performance of reinforcement learning models.

3.2. Olive's method based on width planning and active learning

Benjamin Ayton's team from the Watson AI Laboratory at MIT studied an online planning and learning agent called Olive (Online VAE IW)[7], Used for Atari games, the performance of width based planning methods in Atari games has been improved by combining Iterative Width (IW) and Active Learning. Olive has solved the challenges of data collection, reward attention, and dataset size in VAE-IW through online learning, including novelty based pruning, Bayesian estimation based search guidance, and uncertainty sampling based active learning. Introduced prior distribution of rewards and Best Arm Identification (BAI) strategy, including algorithms such as Top Two Thompson Sampling (TTTS) and Upper Confidence Bound (UCB1). Updating the screen dataset through uncertainty sampling, including passive random selection and active uncertainty based selection, to improve the accuracy of VAE. Through the above research, it has been found that Olive has gradually improved the quality of learning representations by actively searching for new and rewarding states and based on good Bayesian statistical principles. Olive has shown excellent performance in competition with width based planning methods with much larger training budgets, demonstrating high sample efficiency.

3.3. Using Reptile algorithm for reinforcement learning to train neural networks

Sanyam Jain from the University of Stafford studied an experimental method of using Reptile algorithm [8] for reinforcement learning to train neural networks to play Super Mario Bros. By comparing it with Proximal Policy Optimization (PPO) and Deep Q-Network (DQN) algorithms, the potential of Reptile algorithm for few sample learning in video game AI was demonstrated. First, perform preprocessing to convert game frames into grayscale images, downsample and stack them to capture motion information. Next, a convolutional neural network is used to define the model and input the preprocessed frames to obtain the output action probability distribution. Using meta learning algorithms to learn neural network models by calculating the gradient of expected rewards to update initial parameters and obtain a good initialization. Fine tune the initial parameters of the neural network model using the Reptile algorithm and adapt to the corresponding task by updating task specific parameters and initial parameters. First, store the experience tuples in the replay buffer for training the neural network model, and evaluate the performance of the trained agent on the test set using the average total reward as a metric. The performance of PPO, DQN, and RAMario (Mario using Reptile algorithm) in Super Mario Bros. was compared through experiments, and RAMario outperformed PPO and DQN in terms of movement count and distance. The advantage of RAMario lies in its meta learning method that can adapt to specific tasks, update weights using gradient descent, and quickly adapt to new tasks and environments. However, Reptile algorithm relies on

DOI: 10.54254/2755-2721/115/2025.18525

hyperparameter selection, has limited effectiveness in complex environments, is affected by data quality, and may not be as effective as other algorithms in certain scenarios.

3.4. The advantages and potential of evolutionary strategy (ES) in optimizing neural network weights

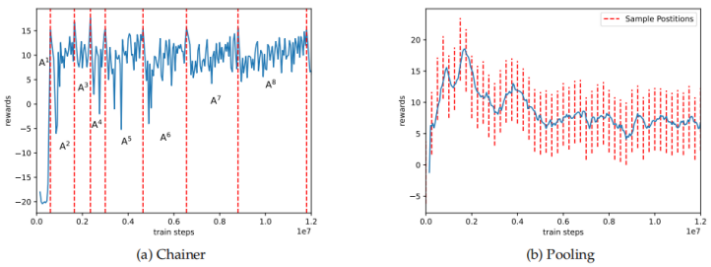
Annie Wong's team from Leiden Institute for Advanced Computer Science (LIACS) at Leiden University in the Netherlands studied the application of evolutionary strategies (ES) in reinforcement learning [9]. They compared ES with gradient based deep reinforcement learning methods and explored its performance in optimizing linear policy networks. Three ES (CSA ES, sep CMA-ES, CMA-ES), three gradient based DRL methods (Deep Q-learning, Proximal Policy Optimization, Soft Actor Critic), and ARS were benchmarked. For ES and ARS, only linear strategies are trained, which are linear mappings from states to actions. Train the same linear strategy for gradient based methods and compare it with the original network architecture. Experiments were conducted in different environments, including classical control tasks, MuJoCo simulated robot tasks, and Atari learning environments. Simple methods such as classical ES and enhanced random search (ARS) can achieve similar results to NES, and linear strategies can effectively solve continuous control tasks. Although achieving good results, ES variants are often suitable for problems with a small number of parameters due to computational complexity limitations.

3.5. Improve agent performance through two self play training schemes (Chainer and Pool)

Bowen He's team from the Department of Computer Science at Brown University studied the method of training agents in Atari Pong game using two self play training schemes (Chainer and Pool) [10] and comparing them with standard DQN agents to explore whether DQN agents truly learn the content and the impact of adversarial training schemes on agent performance. This article proposes two adversarial training schemes, Chainer and Pool. Pool randomly selects previous agents from a fixed size queue as opponents and regularly updates them, while Chainer only allows the current agent to play against its direct predecessor. When a certain evaluation threshold is reached, the current agent replaces the opponent's strategy and continues to train subsequent agents.

The learning curve graph of Chainer (a)[10] shows that as training progresses, the number of steps required for the agent to confront the opponent increases, indicating that the opponent's strength is constantly improving.

Pooling Learning Curve (b) Pooling [10] shows that over time, as the agents in the pool become stronger, the reward curve obtained by the current agent decreases, but it can still defeat the standard DQN agent.



(a) Chainer (b) Pooling

Figure 1: Learning curve result chart[10]

From figure 1, it is clear that adversarial trained agents can easily defeat standard DQN agents, and their relative strength steadily increases as training progresses.

DOI: 10.54254/2755-2721/115/2025.18525

4. The fields of reinforcement learning applications

In the gaming field, intelligent agents can achieve human level control abilities in Atari games. Despite the progress of the times, DQN algorithm is still a classic algorithm for reinforcement learning in the gaming field, laying the foundation for the subsequent progress of reinforcement learning in games. It was not until the proposal and implementation of MuZero algorithm [11] that it can perform well in various games without knowing the learning rules, which has promoted the further development of reinforcement learning in the gaming field. In the field of robotics, the effectiveness, universality, and low cost of reinforcement learning in flexible robot operations are of great significance for the application of robots in complex operational tasks[12].In the field of transportation, a research team from Tsinghua University has proposed a new framework ActorRL [13] to address the curse of dimensionality and instability challenges faced by multi-agent deep reinforcement learning in autonomous intersection management. The actor allocation mechanism assigns different roles to vehicles in intersection management, enabling them to better collaborate in driving. In the field of healthcare, reinforcement learning is applied to optimize treatment plans for chronic diseases [14]. By learning the patient's historical data and treatment feedback, personalized treatment plans are developed to improve treatment outcomes. In the financial field, reinforcement learning is applied to portfolio optimization [15] by learning market data and the historical performance of investment portfolios, which can provide investors with optimal investment portfolio strategies and improve investment returns. In the field of energy, reinforcement learning is applied to the control of energy storage systems in microgrids [16], which optimizes the control of energy storage systems and improves the stability and reliability of microgrids by learning the operating status of the grid and the electricity demand of users.

5. Challenge and Analysis

Large state space action: As the state action space increases, the difficulty of DRL naturally increases. For example, robots in the real world often have high-dimensional sensory input and numerous degrees of freedom, and recommendation systems have graph-structured data and a large number of discrete actions. In addition, the state action space may have complex underlying structures, such as causal dependence between states, compositionality and mixed nature of actions, which makes it difficult to effectively explore in the large state action space [17].

Sparse, delayed rewards: Exploring in an environment with sparse, delayed rewards is another major challenge. For example, in environments like Chain MDP and Montezuma's Revenge, basic exploration strategies struggle to find meaningful states or get valuable feedback. In this case, the key to effective exploration is to explore the environment using information unrelated to rewards as a dense signal, along with the ability to conduct time-extended exploration (also known as time-consistent exploration or deep exploration).

The white noise problem: The real world environment is usually highly random, and white noise will appear in the observation or action space, such as the visual observation of an autonomous car containing a lot of irrelevant information. In the exploratory literature, white noise is often used to generate high-entropy states and inject randomness into the environment, which makes it difficult for agents to build accurate dynamical models to predict the next state. For example, in a "Noisy-TV" environment, the addition of Gaussian noise makes the agent attracted to stay in the current room and unable to pass through more rooms [18].

Multi-agent exploration: Multi-agent exploration faces the challenges of exponentially growing joint state action space, coordinated exploration, and local and global exploration balance. Specifically, the increase of joint state action space makes it more difficult to explore the environment, and individual exploration based on local information may lead to bias and non-stationarity of

DOI: 10.54254/2755-2721/115/2025.18525

exploration measurements, making it difficult to achieve collaborative exploration. In addition, achieving a balance between local and global perspectives is also a key issue, otherwise it can lead to under-exploration or redundancy [19].

Lack of interpretability: Existing methods typically formalize reinforcement learning problems through deep neural networks, which are black boxes that take sequential data as input and policies as output. They struggle to reveal the internal relationships between states, actions, or rewards behind the data and provide intuition about the characteristics of the strategy.

6. Conclusion

This paper describes the background, research status, improvement and optimization, application fields and challenges of multi-agent reinforcement learning. Reinforcement learning is of great significance in solving complex decision problems, promoting intelligence in various industries, and allowing agents to learn optimal strategies in interaction with the environment. However, there are many challenges, such as the exploration problem caused by the growth of state action space, sparse delay reward, white noise interference, multi-agent coordination difficulties, data inefficiency, and lack of interpretability. At the same time, there are some problems such as environmental uncertainty, low sample efficiency, limited generalization ability, difficult reward design and large computing resource demand. When it comes to improving optimization, researchers are constantly exploring new algorithms and techniques. Its wide range of applications, although facing challenges, but with the progress of technology, reinforcement learning is expected to play a greater role, provide strong support for the intelligent future, become an important means to solve complex decision-making problems, promote the industry to a higher level of intelligence, create a more efficient and intelligent future development pattern.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] *Vouros, G. A. (2022). Explainable Deep Reinforcement Learning: State of the Art and Challenges. ACM Computing Surveys 55: 1-39.*

[2] *Yang, T. P. et al. (2021). Exploration in Deep Reinforcement Learning: From Single-Agent to Multiagent Domain. IEEE Transactions on Neural Networks and Learning Systems 35: 8762-8782.*

[3] *Kiran, B. R. et al. (2020). Deep Reinforcement Learning for Autonomous Driving: A Survey. IEEE Transactions on Intelligent Transportation Systems 23: 4909-4926.*

[4] *Huh, D. and Prasant, M. (2023). Multi-agent Reinforcement Learning: A Comprehensive Survey. ArXiv abs/2312.10256: n.pag.*

[5] *Kober, J. et al. (2013). Reinforcement learning in robotics: A survey. The International Journal of Robotics Research 32: 1238-1274.*

[6] *Ashrya, A., Priyanshi, S., Sourabh, P. (2023). Pixel to policy: DQN Encoders for within & cross-game reinforcement learning.*

[7] *Benjamin, A., Masataro, A. (2022). Is Policy Learning Overrated?: Width-Based Planning and Active Learning for Atari. arxiv.*

[8] *Sanyam, J. (2023). RAMario: Experimental Approach to Reptile Algorithm -- Reinforcement Learning for Mario. arxiv.*

[9] *Annie, W., Jacob, N., Thomas, B., et al. (2024). Solving Deep Reinforcement Learning Benchmarks with Linear Policy Networks. arxiv.*

[10] *Bowen, H., Sreehari, R., Jessica, F., et al. (2022). Does DQN really learn? Exploring adversarial training schemes in Pong. arxiv.*

[11] *Schrittwieser, J., Antonoglou, I., Hubert, T. et al. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. Nature 588, 604–609.*

DOI: 10.54254/2755-2721/115/2025.18525

[12] *Zhu, H. et al. (2018). Dexterous Manipulation with Deep Reinforcement Learning: Efficient, General, and Low-Cost. 2019 International Conference on Robotics and Automation (ICRA): 3651-3657.*

[13] *Li, G. Z., Wu, J. P. and He, Y. J. (2022). HARL: A Novel Hierachical Adversary Reinforcement Learning for Automoumous Intersection Management.*

[14] *Saranya, G. and Sivaraman, K. (2024). Applying Reinforcement Learning to Optimize Treatment Plans for Chronic Disease Management." 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI): 1-6.*

[15] *Sen, J. (2023). Portfolio Optimization Using Reinforcement Learning and Hierarchical Risk Parity Approach. In: Rivera, G., Cruz-Reyes, L., Dorronsoro, B., Rosete, A. (eds) Data Analytics and Computational Intelligence: Novel Models, Algorithms and Applications. Studies in Big Data, vol 132. Springer, Cham.*

[16] *Lin, Y. J., Chen, Y. C., Hsieh, S. F., Liu, H. Y., Chiang, C. H. and Yang, H. T. (2023). Reinforcement Learning-based Energy Management System for Microgrids with High Renewable Energy Penetration, 2023 IEEE International Conference on Energy Technologies for Future Grids (ETFG), Wollongong, Australia, pp. 1-6, doi: 10.1109/ETFG55873.2023.10407803*

[17] *Li, C., Zhang, X. R. Ding, Y. M. (2021). Flexible Job-shop Scheduling Problem Based onDeep Reinforcement Learning. 2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT), Haikou, China, pp. 660-666.*

[18] *Fan, W. T., Zeng, Y. C., Guo, Y. W. et al. (2024). Research on Optimization Method of High speed Railway Train Operation Diagram Compilation Model Based on Reinforcement Learning Railway Transportation and Economy, 1-12 http://kns.cnki.net/kcms/detail/11.1 949.U.20241017.1623.004.html.*

[19] *Bi, Q., Qian, C., Zhang, K., (2022). etc Design of multi-agent angle tracking method based on deep reinforcement learning. Computer Engineering, 1-9. https://doi.or g/10.19678/j.issn.1000-3428.0069710.*