Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 10, No. 3, 2024

A Comprehensive Evaluation and Comparison of Enhanced Learning Methods

Jintong Song[1, *], Houze Liu[2], Keqin Li[3], Jingxiao Tian[4], Yuhong Mo[5]

[1]Boston University, Boston MA USA
[2]New York University, New York NY USA
[3]AMA University, Philippines
[4]San Diego State University, San Diego CA USA
[5]Carnegie Mellon University, Pittsburgh PA USA

[*] Corresponding author: Jintong Song (Email: jintongs@bu.edu)

Abstract: This paper provides a comprehensive evaluation and comparison of current reinforcement learning methods. By analyzing the strengths and weaknesses of the main methods, such as value function-based, strategy gradient-based, and value and strategy-based methods, the differences in their performances on standard problems and the applicable scenarios are explored. Meanwhile, other methods such as Monte Carlo Tree Search (MCTS) and evolutionary methods are also briefly introduced. Through the research and analysis in this paper, it provides reference and guidance for choosing appropriate reinforcement learning methods, and promotes the development and application of reinforcement learning techniques in practical applications.

Keywords: Reinforcement learning, method comparison, advantages and disadvantages analysis.

1. Introduction

With the continuous development and depth of the field of artificial intelligence, reinforcement learning, as an important learning paradigm, is receiving more and more attention and application, which were usually solved by model-based solutions, have an opportunity to be solved with data driven solutions.[1] Reinforcement learning learns through the interaction between the intelligent body and the environment, enabling the intelligent body to maximize the cumulative rewards through trial-and-error learning in unknown environments, thus solving many problems that cannot be solved by traditional machine learning methods, such as decision making and control strategy optimization. However, with the increasing complexity of problems, the advantages and limitations of various reinforcement learning methods have gradually emerged.

The purpose of this paper is to conduct a comprehensive evaluation and comparison of current reinforcement learning methods, to explore the performance differences of different methods on standard problems, and to analyze their applicable scenarios and advantages and disadvantages. An in-depth discussion will be conducted on the main methods, such as value function-based methods, strategy gradient-based methods, and value and strategy-based methods, and their application potentials in solving real-world problems will be assessed. In addition, some other methods, such as Monte Carlo Tree Search (MCTS), evolutionary methods, etc., will be briefly introduced and their strengths and limitations in specific problem domains will be explored.

Through the research and analysis in this paper, it is hoped that it can provide reference and guidance for the selection of appropriate reinforcement learning methods and further promote the development and application of reinforcement learning techniques in practical applications.

2. Theoretical Foundation of Reinforcement Learning

2.1. Reinforcement learning definition and principle

Reinforcement learning is a machine learning paradigm designed to enable an intelligent body to learn optimal behavioral strategies through interaction with the environment in order to maximize cumulative rewards. In reinforcement learning, an intelligent body learns by observing the state of the environment and performing actions, and receives rewards or punishments from the environment to adjust its behavior.

The core principles of reinforcement learning are built on the framework of Markov Decision Processes (MDPs).MDPs provide formal mathematical models describing the interaction of an intelligent with its environment. In MDP, States represent specific situations or configurations of the environment, and intelligences take different actions in different states. States can be discrete or continuous. Actions represent operations or strategies that an intelligent body can perform. In each state, the intelligent body can choose to perform different actions. Rewards denote the feedback that the intelligent body receives from the environment after performing an action. Rewards can be positive, negative, or zero, and are used to indicate the degree of merit of the action. Policies denote the strategy by which an intelligent body chooses an action in a given state, and the strategy can be deterministic or stochastic. Combining the change detection task with reinforcement learning utilizes the framework of Markov Decision Processes (MDPs)[2], describing the interaction between the agent (change detection model) and the environment in image states. Here, states represent specific conditions of the images, actions denote operations taken by the model in different states, rewards signify feedback received from the environment after performing actions, and policies denote the model's strategy for selecting

actions given states, thereby optimizing the performance of the change detection model.

In reinforcement learning, the goal of the intelligent body is to learn an optimal policy that maximizes the cumulative reward through interaction with the environment. To achieve this goal, the intelligent body needs to explore different strategies through trial and error and guide the learning process through reward signals. Just as mentioned in the "Detect Any Deepfakes (DADF)" framework, the Reconstruction Guided Attention (RGA) module is introduced to better identify forged traces and enhance the model's sensitivity to manipulated areas[3]. Such a framework seamlessly integrates end-to-end forgery localization and detection optimization, allowing the agent to progressively optimize its behavioral strategies through continuous experimentation and feedback, thereby adapting to various environments and tasks.

Reinforcement learning methods usually include key techniques such as value function estimation, strategy optimization, and exploration and exploitation. Through these techniques, intelligent bodies can effectively learn and make decisions in complex environments, thus realizing the goal of autonomous intelligent behavior.

In practice, reinforcement learning is widely used in au tomatic control, game design, financial trading and other fields, and plays an important role in the field of artificia l intelligence. With the development of deep learning and other technologies, reinforcement learning has great poten tial for solving complex tasks and realizing the generalize d goals of artificial intelligence.

2.2. Classification of common reinforcement learning algorithms

2.2.1. Classification based on learning mode

Reinforcement learning algorithms can be classified into two main types according to how the intelligences learn and optimize their strategies: value function-based methods and strategy gradient-based methods.

Value function-based methods: These algorithms optimize strategies by estimating the value function of states or state-action pairs. Among them, Q-learning and SARSA are the most classical value function-based algorithms that improve the policy by updating the state-action value function. In addition, there are value-based approximation methods such as DQN (Deep Q Networks), which utilize neural networks to approximate the value function to deal with continuous state space problems.

Policy gradient based methods: These algorithms learn the policy function directly without estimating the value function. They update the policy parameters by maximizing the gradient of the cumulative reward. Typical policy gradient based algorithms include REINFORCE, PPO (Proximal Policy Optimization), TRPO (Relative Entropy Policy Optimization), etc. These algorithms have better performance and generalization ability in dealing with problems in continuous action space.

2.2.2. Classification based on policy updating method

Offline updating algorithms: These algorithms use the experience of the whole round to update the strategy parameters after the complete round is over, Q-learning and SARSA are representatives of offline updating algorithms.

Online updating algorithms: These algorithms update the policy parameters at each step or time step without waiting until the end of the complete round. REINFORCE and DDPG (Deep Deterministic Policy Gradient) are typical online updating algorithms that are able to adapt to continuous environments and update the policy in real time.

2.2.3. Classification based on value function estimation approach

Exact Value Function Estimation Algorithms: These algorithms attempt to directly estimate an exact value function, such as an exact state value function or state action value function. Typical algorithms include Q-learning and SARSA.

Approximate Value Function Estimation Algorithms: These algorithms use function approximators (e.g., neural networks) to approximate the value function. These algorithms are capable of handling large-scale state-space and action-space problems and have performed well in practice. For example, DQN utilizes deep neural networks to approximate the state-action value function, thus enabling efficient estimation of continuous state spaces.In the field of digital asset trading, researchers have begun exploring the integration of Distributed Ledger Technology (DLT) and DQN to further enhance the security and reliability of transactions. This integration lays a solid foundation for the development of digital asset trading[4]. In this process, reinforcement learning, as a learning paradigm based on the interaction between agents and the environment, also plays a crucial role.

3. Evaluation Criteria for Reinforcement Learning Methods

3.1. Performance Evaluation Metrics

3.1.1. Cumulative reward

Cumulative reward is one of the core evaluation indexes of the reinforcement learning algorithm, representing the sum of rewards obtained by the intelligent body during the execution of the task. A higher cumulative reward means that the intelligent body can perform the task effectively and make appropriate decisions based on the rewards provided by the environment. By comparing the cumulative rewards of different algorithms on the same task, it is possible to determine which algorithm is better suited to solve a particular problem.

3.1.2. Speed of convergence

The convergence speed metric measures how quickly the reinforcement learning algorithm learns the optimal policy. A faster convergence speed means that the algorithm can quickly converge to the optimal policy or reach a steady state, thus achieving better performance in the same training time. By comparing the convergence speeds of different algorithms with the same number of training rounds, it is possible to determine which algorithm is more efficient.

3.1.3. Algorithm Stability

Algorithm stability evaluates the consistency of a reinforcement learning algorithm's performance under different conditions. A stable algorithm should be able to produce consistent results under different environments and parameter settings without being susceptible to noise or randomness. By evaluating the stability of an algorithm, the reliability and robustness of the algorithm can be determined, as well as its applicability in real-world environments.

3.2. Training and testing environment

When evaluating reinforcement learning methods, it is critical to select appropriate training and testing environments. These environments should be able to adequately reflect the characteristics of the problem to be solved and provide suitable challenges as well as effective performance evaluation.

The training environment is the environment with which the intelligence interacts, learns and optimizes. This environment should simulate the key features of the problem to be solved and provide enough information and feedback so that the intelligences can gradually improve their strategies and learn the optimal behavior. The training environment should be well controlled and reproducible so that researchers can experiment and compare under different conditions. The testing environment is used to evaluate the performance and generalization ability of the trained intelligences. This environment is usually different from the training environment to ensure that the intelligences are able to make accurate decisions in unseen situations. The test environment should contain a variety of possible scenarios and challenges to fully evaluate the performance of the intelligences. At the same time, the test environment should be set up to match the real-world application scenarios to ensure the usefulness and reliability of the algorithms. When designing training and testing environments, there is a need to weigh the degree of interface between the simulation environment and the real world. Although simulation environments can provide better controllability and debugging, they may not fully reflect the complexity and uncertainty of the real world. Therefore, it is necessary to consider the possibility of validating and tuning algorithms in the real world during the testing phase to ensure their reliability and effectiveness in real-world applications. The selection of suitable training and testing environments is crucial for evaluating the performance of reinforcement learning methods. By designing suitable environments and rationally utilizing training and testing data during experiments, the effects of algorithms can be more accurately assessed and their feasibility and reliability in real-world applications can be improved.

3.3. Algorithm Complexity and Scalability

Evaluating the algorithmic complexity and scalability of reinforcement learning methods is an important consideration to ensure the feasibility and efficiency of the algorithms in practical applications. The algorithmic complexity metric assesses how much the reinforcement learning algorithm consumes in terms of computational and memory resources. This includes the algorithm's time complexity, space complexity, and computational resource requirements. Algorithms with low algorithmic complexity are able to execute efficiently with limited computational resources, making them more suitable for deployment and operation in real-world applications. The scalability metric evaluates the ability of a reinforcement learning algorithm to handle large-scale problems. An algorithm with good scalability can efficiently handle a large number of state spaces and action spaces without suffering from performance degradation or resource exhaustion. When evaluating scalability, it is necessary to consider the performance of the algorithm on problems of different sizes and determine its applicability and efficiency on large-scale problems.The recently proposed Parameter Efficient Fine-Tuning (PEFT) strategy for language models has demonstrated lower algorithmic complexity during implementation, along with the potential for efficient execution under limited computational resources.[5]

Parallelism enables the use of parallel computing resources to accelerate the process of algorithm execution, thereby increasing its efficiency and performance on large-scale problems. The scalability and parallelism of algorithms can be effectively improved by designing algorithmic structures and optimizing algorithmic implementations that support parallel computing.

For large-scale problems, distributed computing techniques can further improve the scalability and performance of algorithms. By assigning computational tasks to multiple computing nodes for parallel processing, computational time and resource consumption can be effectively reduced. When evaluating the scalability of an algorithm, it is necessary to consider whether it supports distributed computing and verify its performance in a distributed environment through experiments. Comprehensive consideration of factors such as algorithm complexity and scalability can help assess the feasibility and efficiency of reinforcement learning methods in practical applications. By choosing algorithms with low algorithm complexity and good scalability, and optimizing the algorithm implementation to improve parallelism and distributed computing capability, the practicality and performance performance of the algorithms can be effectively improved.

4. Reinforcement Learning Algorithm Performance Comparison, Advantages and Disadvantages Analysis and Its Applicable Scenarios

4.1. Algorithm Performance Comparison

Both Q-learning and SARSA are value function based methods for learning optimal policies. In general, Q-learning is more inclined to explore actions that maximize rewards during offline learning, while SARSA is more concerned with the stability of online strategies. SARSA may outperform Q-learning in terms of stability, especially in environments with a high degree of randomness.

Both DQN and DDPG are deep neural network-based approaches for dealing with continuous states and action spaces.DQN is mainly used for problems in discrete action spaces, while DDPG is suitable for continuous action spaces. DDPG tends to achieve better performance when dealing with continuous action space. However, DQN also performs well on simple discrete action space problems.

REINFORCE and PPO are both policy gradient based methods, but differ slightly in the way they update the policy and their optimization goals. REINFORCE is based on Monte Carlo estimation of the samples, while PPO uses proximal policy optimization to ensure the stability of the updates. In most cases, PPO outperforms REINFORCE in terms of performance and convergence speed.

Both TRPO and PPO are policy-based methods that aim to learn the optimal policy by optimizing the policy function.TRPO uses relative entropy constraints to ensure the stability of the policy updates, while PPO uses proximal policy optimization to achieve similar goals. In terms of

performance and stability, both usually have similar performance, but PPO is easier to implement and tune.

Monte Carlo Tree Search (MCTS) and DDPG are both methods for dealing with continuous action spaces, but differ in policy search and action selection.MCTS is a heuristic search method that allows policy search based on rewards and state values in a search tree, while DDPG is a value function-based method that learns optimal policies directly. MCTS may perform better on problems that require long-term planning and exploration, while DDPG is more suitable for continuous action space problems.

4.2. Analysis of Algorithm Advantages and Disadvantages

Value Function Based Approaches For discrete state and action space problems, such as grid worlds, value function based approaches usually have better convergence and stability. The value functions of states or state-action pairs can be learned directly, leading to a better understanding of the value of the environment and actions. However, when dealing with continuous state and action space problems, such as real robot control, the value function-based methods may suffer from dimensional catastrophe, leading to learning difficulties and performance degradation.

Policy gradient-based methods can learn the policy function directly without explicitly estimating the value function, thus enabling the handling of continuous action space problems. For highly stochastic environments and tasks, policy gradient methods are usually more robust and stable. However, they are less efficient in sample utilization during training and may require more samples and training time to obtain good performance. Due to the direct optimization of the strategy, it is easy to be affected by hyperparameters such as the initialization of the strategy parameters and the update step size, and it is more difficult to tune the parameters.

The value and strategy-based approach combines the advantages of the value function-based and strategy-based approaches, which can effectively deal with the continuous state and action space problems, and has better learning performance and stability. The value function and the strategy function can be learned at the same time, leading to a more comprehensive understanding of the environment and the behavior of the intelligences. However, the algorithm is more complex and requires more computational resources and training time to obtain good performance, and the algorithm parameters and hyperparameters need to be carefully adjusted to ensure the stability and convergence of the algorithm.

Monte Carlo Tree Search (MCTS) is suitable for problems that require long-term planning and exploration, and can effectively search for optimal strategies. However, the computational complexity is high and may not perform well on large-scale problems.

Evolutionary Methods (EMs) can globally search the policy space and are suitable for complex and high dimensional problems. However, the convergence speed is slow, sensitive to algorithm parameters and population settings, and difficult to tune the parameters.

4.3. Applicable Scenarios of Algorithms

Based on value function methods, these methods usually perform well on problems with discrete state and action spaces, such as board games and mazes. Since these problems have clear state and action definitions, value function-based methods can effectively learn and infer optimal policies. For example, Q-learning and SARSA usually achieve good performance in such problems because they are able to achieve policy optimization by estimating the value functions of states or pairs of state-action pairs.

Policy gradient-based methods are mainly applicable to problems with continuous action spaces, such as robot control and continuous control systems. Since the action space of these problems is continuous, policy gradient-based methods are able to learn the policy function directly with good generalization ability and robustness. For example, REINFORCE and DDPG typically achieve better performance in such problems because they are able to update their strategies in real time and adapt to continuous environments.

Value- and policy-based methods combine the advantages of value function-based and policy-based methods, and are able to handle more complex problems with better learning performance and stability. This type of approach is suitable for problems with both discrete and continuous state spaces or action spaces. For example, PPO (Proximal Policy Optimization) and A3C (Asynchronous Advantageous Actor-Commentator Algorithm) usually achieve good performance in such problems because they are able to learn both the value function and the policy function, and understand the environment and the intelligence's behaviors more comprehensively.

Monte Carlo Tree Search (MCTS) is suitable for problems that require long term planning and exploration, such as Go, Poker, etc. MCTS usually achieves good performance in these problems because it enables long term planning and exploration through heuristic search.

Evolutionary Methods are suitable for problems with large strategy space and high complexity. Evolutionary Methods usually achieve better performance in these problems because they are able to optimize the policy through global search.

5. Conclusion.

Currently, reinforcement learning methods play an important role in the field of artificial intelligence and are widely used in a variety of complex problem domains, such as the gaming domain, robot control, and traffic management. While different reinforcement learning methods have their own strengths and weaknesses, overall, they have made significant strides in addressing a variety of highly uncertain and complex decision-making problems. For instance, when dealing with challenges such as bad data detection (BDD) and neural attack locations (NAL), leveraging key designs[6] like perturbation of state variables, customized loss function design, and variable transformation becomes crucial to tackle real-world complexities.

Value function-based methods perform well on problems with discrete states and action spaces, and are able to achieve better policy optimization through value function estimation. The strategy gradient-based methods are suitable for problems in continuous action space with good generalization ability and robustness. In contrast, the value and strategy-based methods are able to combine the advantages of both to deal with more complex problems and have better learning performance and stability.

Despite the remarkable progress of reinforcement learning methods, they still face some challenges, such as the

convergence speed of algorithms, the stability of the training process, and the demand for computational resources. In the future, further in-depth research on the optimization and improvement of various reinforcement learning methods is needed to improve the efficiency and reliability of the algorithms in solving complex practical problems. Meanwhile, exploring more efficient and flexible reinforcement learning frameworks by combining deep learning, probabilistic modeling and other methods will provide more possibilities and opportunities for solving complex problems in the real world.

References

[1] Y.-T. Hsieh, Z. Qi, and D. Pompili, "ML-based Joint Doppler Estimation and Compensation in Underwater Acoustic Communications," in Proceedings of the 16th International Conference on Underwater Networks & Systems (WUWNet '22), New York, NY, USA, 2022, pp. 1–8.

[2] Z. Li, Y. Huang, M. Zhu, J. Zhang, J. Chang, and H. Liu, “Feature Manipulation for DDPM based Change Detection,” arXiv.org, Mar. 23, 2024.

[3] Y. Lai, Z. Luo, and Z. Yu, "Detect Any Deepfakes: Segment Anything Meets Face Forgery Detection and Localization," in Biometric Recognition, W. Jia, Ed., vol. 14463, Lecture Notes in Computer Science, Springer, Singapore, 2023.

[4] Q. Cheng et al., “Secure Digital Asset Transactions: Integrating Distributed Ledger Technology with Safe AI Mechanisms”, AJST, vol. 9, no. 3, pp. 156–161, Mar. 2024.

[5] S. Zhao et al., “Defending Against Weight-Poisoning Backdoor Attacks for Parameter-Efficient Fine-Tuning,” arXiv.org, Mar. 29, 2024.

[6] J. Tian et al., "LESSON: Multi-Label Adversarial False Data Injection Attack for Deep Learning Locational Detection," in IEEE Transactions on Dependable and Secure Computing.

[7] Xiaotian Zhang,Yawen Wang,Zhiqing Xie et al. Reinforcement Learning Approach for Sequence Determination of Class Integration Tests [J]. Computer Engineering, 2024, 50 (01): 68-78.

[8] Rongyun Li. Reinforcement Learning Approach for Game Manipulation Behavior Imitation [D]. University of Electronic Science and Technology, 2022.

[9] Yin Hang. Epistemic modeling of self-reinforcement learning methods based on deep residual networks[D]. Liaoning University of Engineering and Technology, 2022.

[10] Wei Minatong. Deep network-based video repair and reinforcement learning methods in unstable environments[D]. University of Science and Technology of China, 2021.

[11] Zhang Wei. Deep reinforcement learning for target localization and recognition[D]. Xi'an University of Technology, 2021.

[12] Lei Xu. Deep Reinforcement Learning for Text Games[D]. Heilongjiang University, 2021.