2024 6th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI) | 979-8-3503-8697-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/IoTAAI62601.2024.10692624

Leverage of generative adversarial model for boosting exploration in deep reinforcement learning

Shijie Chu*
School of Computer Engineering and Science,
Shanghai University, Shanghai, China
*Corresponding author's e-mail: csj2021@126.com

***Abstract*-Deep Reinforcement Learning (DRL) has made remarkable progress in a range of domains. However, its inherent random exploration mechanism leads to low efficiency, especially in complex environments where the time complexity of strategy exploration is high. To overcome this challenge, we propose a Deep Q-Learning based on the Generative Adversarial Model algorithm (DQN-GAN), aiming to improve exploration efficiency in deep reinforcement learning. The core of the DQN-GAN algorithm is to generate more diverse actions, thereby effectively simplifying the complex exploration process. Specifically, the proposed algorithm utilizes a Generative Adversarial Network (GAN) to train a generator that generates diverse actions, enriching the exploration space. Meanwhile, a discriminator is trained with real demonstration data to ensure that the generated actions are highly consistent with actual data. Experimental data demonstrates that the integration of GAN into deep Q-learning exhibits significant performance improvements in OpenAI Gym environments. The proposed algorithm not only optimizes the exploration process but also enhances learning efficiency.**

Keywords-Deep reinforcement learning, generative adversarial model, sample efficiency, exploration

I. INTRODUCTION

Deep reinforcement learning (DRL) has emerged as a powerful learning paradigm and has made remarkable progress in recent years. The autonomous behavior planning capability of DRL makes it an essential tool for solving behavior planning challenges in various domains, such as game control [1], robotics [2], and financial trading [3-4]. However, despite its significant success, DRL still faces the challenges of low efficiency when exploring complex environments.

In traditional DRL methods, the agent interacts with the environment through random exploration and optimizes policies through the experience gained from interactions. To explore more states, some researchers propose to introduce noise [5], curiosity [6], and entropy regularization [7]. However, these exploration methods still suffer from low efficiency as they require a significant amount of trial and error and time to find effective action. Particularly in high-dimensional state and action spaces, the time complexity of policy exploration grows exponentially, limiting the widespread application of DRL in the real world.

Generative Adversarial Network (GAN) [8] is a commonly used generative model. Meng et al. [9] proposed to integrate safety constraints into adversarial training to improve the anti-interference ability of DRL. Dong et al. [10] proposed using safety critics based on GAN to prevent the agent from making unsafe decisions. Inspired by these successes, we propose an innovative algorithm, Deep Q-Learning based on Generative Adversarial Models (DQN-GAN), to produce more diverse samples. Unlike traditional DRL methods, DQN-GAN aims to enrich the exploration space of agents by leveraging the powerful generation of more diverse samples by GANs. This approach can enhance the efficiency of policy learning. The main contributions of this work are summarized as follows:

1. This work combines the deep Q network with the generative adversarial network and uses a generator to generate diverse actions, thereby enriching the exploration space of the agent.

2. This work uses real interaction data to train the discriminator, thereby improving the quality and effectiveness of generated actions.

3. Experimental results show that compared with the traditional DRL method, the proposed algorithm has achieved significant performance improvements in various complex environments.

II. FUSION OF DRL AND DQN

This section introduces the foundations of DQN and GAN, including predefined representations and fundamental principles underlying their respective architectures.

A. DQN

DQN is a commonly used algorithm for enabling autonomous decision-making by the agent. This approach utilizes neural networks to approximate the Q function, which estimates the future expected reward for each action given a state. The Q function is typically defined as $Q(s,a)$, where s is the state and a is the action. The objective of DQN is to iteratively update the Q-values through temporal-difference learning, enabling the agent to select actions with the highest Q-values to maximize long-term rewards. The update process for the Q function is as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)). \quad (1)$$

where $Q(s',a')$ is the Q-value corresponding to the next state and the actual action taken; $Q(s,a)$ is the Q-value for taking action a in the state s; r is the immediate reward obtained after taking action a; γ is the discount factor, balancing the importance of current and future rewards; α is the learning rate, controlling the step size of updates.

Traditional DRL methods typically employ a $\epsilon - greedy$ strategy to guide the exploration process, by which the agent randomly selects actions with probability ϵ to explore unknown areas of the environment. However, this strategy may lead to excessively random actions, particularly when dealing with high-dimensional state and action spaces, resulting in a sharp increase in the time complexity. Therefore, it is necessary to refine exploration strategies to enhance the efficiency and performance of reinforcement learning.

B. GAN

GAN is a classic generative model consisting of two neural networks, the generator and the discriminator. The core of GAN is to train the generator to produce samples that resemble the real data distribution through adversarial training. Through iterative training, the generator gradually enhances the quality and diversity of the generated samples, bringing them closer to the distribution of real data. The objective function of GANs can be expressed as:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \quad (2)$$

where G is the generator network, attempting to generate realistic samples; D is the discriminator network, attempting to distinguish between generator-generated samples and real samples; x represents real data samples; $p_{data}(x)$ is the distribution of real data; z is random noise samples drawn from the noise distribution $p_z(z)$.

By alternately updating the parameters of the generator and the discriminator, GAN can reach an equilibrium state, so that the quality of the samples generated by the generator and the similarity of the real data are optimized. Combining DQN and GAN can provide a richer exploration space for reinforcement learning tasks. By utilizing the diverse samples generated by GAN, the agent can overcome exploration problems in high-dimensional state and action space, and improve learning efficiency and performance.

III. Implementation of DQN-GAN

A. Overview of DQN-GAN

Our objective is to reconstruct the exploration space of the agent and enhance the exploration probability of feasible actions. The algorithm flowchart is shown in Figure 1. DQN-GAN mainly consists of two parts and the learning process is illustrated in Figure 2 and Figure 3.

B. Reconstruction of the exploration space

Compared to traditional DRL methods, DQN-GAN offers a distinct advantage by focusing on reconstructing the agent's exploration space. This approach effectively reduces the interaction costs associated with learning, making it more efficient. Similar to imitation learning techniques, DQN-GAN relies heavily on real data as its foundation, as illustrated in Figure 2. These real data are collected from tasks with well-defined reward functions, comprising both states and actions. Each data set includes the state observed at each time step and the corresponding action taken by the agent in that state.

By leveraging the data generated during the agent's interactions with the environment, we create a robust dataset that serves as the backbone for the generative model within DQN-GAN. This dataset provides essential data support, ensuring that the DQN-GAN can operate effectively and efficiently. The use of real interaction data not only enhances the accuracy and relevance of the generated samples but also helps the model to better replicate and anticipate the dynamics of the actual environment, leading to improved performance in reinforcement learning tasks.

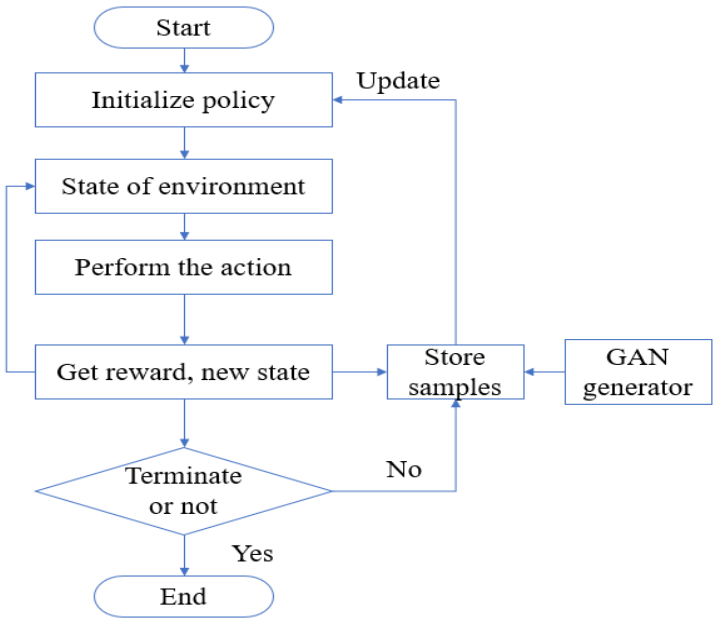

Figure 1. Flowchart of DQN-GAN.

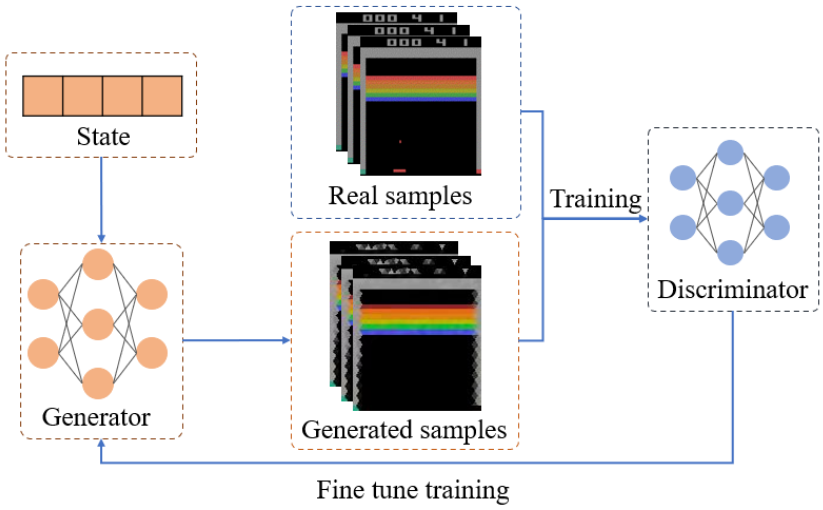

Figure 2. The architecture of GAN.

C. Policy optimization

We integrate GAN into the policy optimization process to significantly enhance the agent's exploration capabilities, as illustrated in Figure 3. The inclusion of GAN allows the generator to produce a diverse set of potential actions for the agent in any given state. By leveraging these generated actions, the agent gains access to a broader exploration space, enabling it to explore the environment more thoroughly and gather a wider range of data samples.

Once these exploratory actions are generated, the agent can execute them to interact with the environment, thereby collecting a diverse array of samples that reflect different possible outcomes. These generated samples, when combined with the actual actions taken by the agent, create a rich dataset that is essential for robust learning. During the training of the DQN model, both the generated and real interaction samples are used, with each type of sample being selected with a probability of 50%. This approach ensures that the model benefits equally from the diversity of GAN-generated data and the authenticity of real-world interactions.

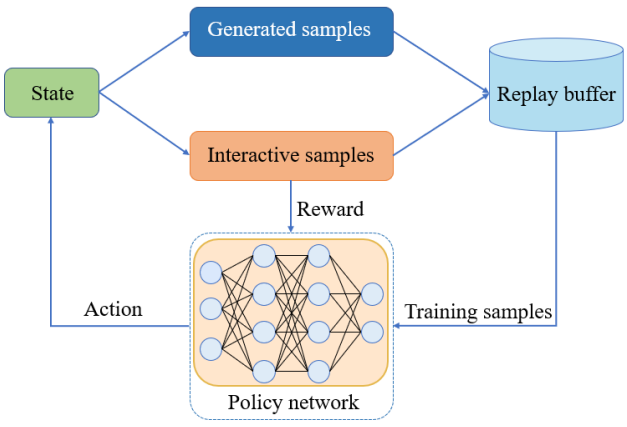

Figure 3. The process of DRL.

IV. Experiment and Analysis

A. Tasks and setting

We selected three Atari games from OpenAI Gym to evaluate the proposed algorithm. The three Atari games are Breakout, MsPacman, and Seaquest. Their screenshots are shown in Figure 4. We compare DQN-GAN with three classic algorithms, DQN, Dueling DQN, and Prioritized Replay DQN. All algorithms are reproduced exactly as the original text. We perform comparison experiments on the NVIDIA V100 GPU.

B. Performance comparison

Throughout the extensive training process, which involved one million iterations, we conducted approximately two hundred evaluations for each specific task. These evaluations were primarily based on the reward values achieved at the conclusion of the training, as illustrated in Figure 5. The results clearly demonstrate that across the three distinct tasks we tested, the DQN-GAN algorithm consistently outperformed others by reaching the highest reward values at the end of the training period.

Notably, when analyzing the entire training process, it becomes evident that our algorithm exhibits remarkable stability, particularly in the Breakout and MsPacman tasks. This stability is reflected in the performance trends, where the DQN-GAN algorithm maintains a superior trajectory compared to other methods. As highlighted by the black line in Figure 4, the algorithm also showcases the highest sample efficiency among the tested approaches. This efficiency is a critical metric, as it indicates how effectively the algorithm can learn from a limited number of samples, thereby speeding up the training process and improving overall performance.

However, for Seaquest, the instability of the compared methods may be attributed to the high complexity of the task, which poses difficulties for the algorithm in learning effectively from the available samples. In future work, we will consider adopting a priority-based approach for selecting high-reward experiences, thereby enhancing its performance stability in complex tasks.

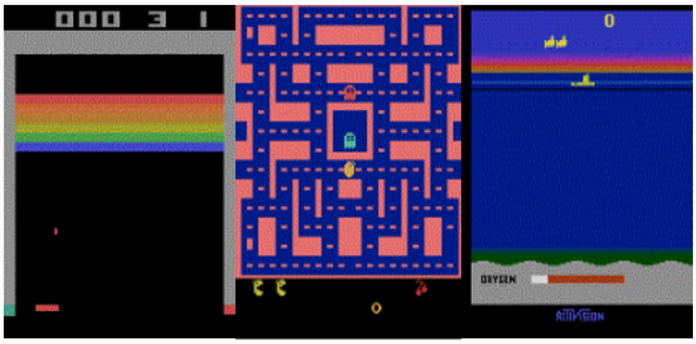

Figure 4. Screenshots of Breakout, MsPacman, and Seaquest.

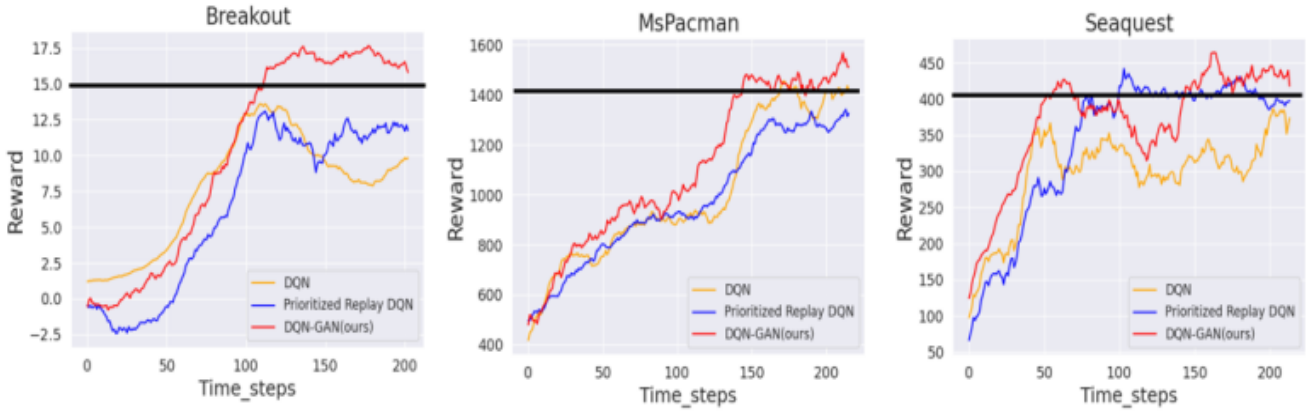

Figure 5. Comparison of DQN-GAN(ours), Prioritized Replay DQN, and DQN.

V. Conclusion

In this work, we propose the DQN_GAN algorithm, which successfully combines the GAN model with the DRL framework and uses the generation ability of GAN to provide richer exploration actions for the agent. We adopt a unique sample generation and training method and use the generated and interactive samples to train the DQN model. This method not only makes full use of the diversity of generated samples but also ensures the authenticity and effectiveness of the training data.

In the future, we will continue to explore the potential of GANs in deep reinforcement learning and promote the widespread application of deep reinforcement learning.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski. (2015). Human-level control through deep reinforcement learning. Nature, 518: 529-533.

[2] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, and D. de Las Casas. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. Nature, 602: 414-419.

[3] J. B. Chakole, M. S. Kolhe, G. D. Mahapurush, A. Yadav, and M. P. Kurhekar. (2021). A Q-learning agent for automated trading in equity stock markets. Expert Systems with Applications, 163: 113, 761.

[4] B. Yang, T. Liang, J. Xiong, and C. Zhong. (2023). Deep reinforcement learning based on the transformer and U-Net framework for stock trading. Knowledge-based Systems, 262: 110, 211.

[5] S. Han, W. Zhou, J. Lu, J. Liu, and S. Lü. (2022). NROWAN-DQN: A stable noisy network with noise reduction and online weight adjustment for exploration. Expert Systems with Applications, 203: 117, 343.

[6] J. Zhang, Z. Zhang, S. Han, and S. Lü. (2022). Proximal policy optimization via enhanced exploration efficiency. Information Sciences, 609: 750-765.

[7] Y. Matsuo, Y. LeCun, M. Sahani, D. Precup, D. Silver, M. Sugiyama, E. Uchibe, and J. Morimoto. (2022). Deep learning, reinforcement learning, and world models. Neural Networks, 152: 267-275.

[8] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. (2018). Generative adversarial networks: An overview. IEEE Signal Processing Magazine, 35: 53-65.

[9] J. Meng, F. Zhu, Y. Ge, and P. Zhao. (2023). Integrating safety constraints into adversarial training for robust deep reinforcement learning. Information Sciences, 619: 310-323.

[10] W. Dong, S. Liu, and S. Sun. (2023). Safe batch-constrained deep reinforcement learning with the generative adversarial network. Information Sciences, 634: 259-270.