The Application of Reinforcement Learning in Video Games

Xuyouyang Fan

Maynooth International Engineering College, Fuzhou University, No.2, Wulongjiang North Ave., Fuzhou, China
832003115@fzu.edu.cn

Abstract. With the continuous development of reinforcement learning, artificial intelligence has reached a level of sophistication where it can effectively handle complex situations, making it a valuable component in video game development and gameplay experiences. Recent applications have demonstrated the potential of reinforcement learning in controlling game agents to achieve victory, as well as its role in facilitating game development processes for human developers. This paper presents a comprehensive review of the methods employed in video games to support in-game artificial intelligence and game development. A comparison is made between traditional methods and the emerging paradigm of reinforcement-learning-based techniques. By exploring the advantages of applying reinforcement learning, this study highlights the potential benefits it brings to the video game industry. Furthermore, real-world cases are examined to showcase successful applications of reinforcement learning in video games. It concludes that the future and advancement of reinforcement learning in video games hold great promise.

Keywords: Reinforcement Learning, Video Games Playing, Video Game Development

1 Introduction

The reinforcement learning (RL) has emerged as an important method of artificial intelligence. The agent in RL learns experience by interacting with the environment. This branch of machine learning has developed a lot of applications across various domains, with the video game playing being a field that interests many scholars. Video games provide simulated world where players can interact with the environment and other players, and this world can be seen an environment for agent of RL. Traditionally, game-playing agents were programmed with fixed rules or heuristics, limiting them to some specific situations or games. However, with the development of RL algorithms and computational power, agents can now autonomously acquire skills and improve their performance through trial and error.

The application of RL in video game playing has opened up exciting possibilities for developing highly intelligent and adaptive game-playing agents. These agents can

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_21

learn optimal strategies, navigate through complex dungeons, and even compete against human players. Moreover, they can continually adapt and evolve their game-play based on feedback and rewards, leading to enhanced player experiences and challenging game-play scenarios. Moreover, RL in video games might laid the groundwork for creating realistic non-player characters (NPCs) that can exhibit human-like behaviours and adapt to the variable situation in games. These intelligent NPCs can provide players with engaging and immersive gameplay experiences, making the gaming environment more dynamic and challenging.

Since RL has demonstrated its adaptability to the gaming environment, has there been a number of applications? Driven by curiosity about this question, the author undertook a certain degree of exploration. This paper aims to explore the application of RL in video game, focusing on the advancements, challenges, and potential future directions in the field. We will discuss various RL algorithms used in video game playing, highlight notable achievements, and examine the impact of these developments on the game industry.

2 The Basic of RL

RL provides a paradigm of self-regulated learning based on the principles of Markov Decision Process (MDP). In RL, the agent interacts with the environment and receives feedback in the form of rewards or penalties based on its actions [1]. Typical RL algorithms commonly involve three main types: Value-based Methods, Policy-based Methods, and Actor-Critic Methods.

Value-based methods: This type of methods like value iteration and Q-learning requires a function called value function to evaluate the "value" of each policy and chose the policy that rewards the max value, a little similar to the heuristic search except the value function will learn from the environment. The performance of this type of methods very depends on the value function (Fig.1).

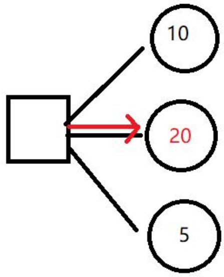

Fig. 1. The value-based methods（Photo/Picture credit: Original）

Policy-based methods: Algorithms such as Policy Gradient and the REINFORCE algorithm fall under this category (Fig.2). These algorithms do not evaluate the value of each policy, instead, the evaluate the potential of the policies. It is possible that the agent chose a node that has the smallest value. It provides the agent possibility to find another way that might leads to a higher value.

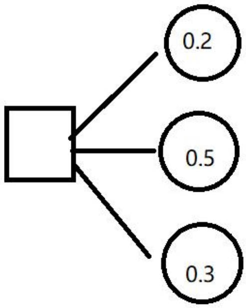

Fig. 2. The policy-based methods (Photo/Picture credit: Original)

Actor-Critic methods: In Actor-Critic algorithms, the Actor learns strategy while the Critic evaluates it. Examples include Deterministic Policy Gradient and Soft Actor-Critic (SAC). These methods train both actor and critic, optimizing policy and assessing effectiveness using value functions. These approaches balances learning rate and search precision in finding the optimal strategy [2].

3 The Traditional Approaches to Build AI and RL Methods

3.1 The Traditional Artificial Intelligence Implementations in Games

Traditional, the artificial intelligence in video games is implemented manually through coding fixed rules (like if A happens then do B else do C). The rules are programmed by the human who writes the rules based on the experience or knowledge of the game world. There are some implementations of the said process. Like the state machine or the behaviour tree.

The state machine just another implementation of the if-else statements. It can closely control the behaviour of the agent, if the conditions are set very specific, the behaviour of the agent can handle the complex situation and act like a smart player. But with the increase of the number of states, the complexity of the state machine grows, and the difficulty of the maintenance becomes a significant cost [3].

3.2 The RL Based Artificial Intelligence in Games

RL, a type of machine learning that has rapidly gained popularity in the game industry, unlike traditional machine AI implementations, RL allows the agents to learn and adapt to variant in-game situations through trail-and-error and feedback based on rewards and penalties. Such a process can be only be used in the time of the game development, that means that the AI does not need to be specific designed, the developers can use the RL to train an AI (model) to control the agents, the model might find some technics that even not be awarded by the developers themselves.

If it is possible to continuously train the AI and distribute the updated model to the released game, the developers can gain the data from the actual AI interaction with the player to make the AI more powerful [4]. It means that the AI can adapt the actual game-play of players to provides players customised experience: adjust the difficulty to make the player comfortable, or learn the player's playing pattern and find the way to against the pattern to make the game more challenge.

RL-based intelligence can be applied to a wide range of gaming genres, including strategy games, action games, and sports games. For example, RL can be used to train NPCs to play as opponents that adapt and learn from the player's strategies in real-time. It can also be used to control the behaviour of teammates in multiplayer games, making them more cooperative and strategic.

4 The Applications of RL in Video Game Field

The video game, which build a simulated world that accepts user inputs and outputs the corresponding result to the users, naturally provides an environment for the interaction of agents in the RL. And there are many actual applications of the RL in the video game playing. The RL shows its power on playing video games that has a goal to achieve, like the real-time strategy (RTS), platform jumping, chess games and other fields.

4.1 The Application in Go

The most famous application is the AlphaGo, who defeated the top human players. The AlphaGo using the CNN to extract the features from the game status and use the extracted features to assist the decision (Fig.3). It using two deep neural networks, the policy network and the value network, to make a decision. The success of the AlphaGo shows that the ability of the RL to evolve self by playing games with itself, to finally achieve the level that no one has ever reached [5].

Years after, David Silver et al. extended the method in AlphaGo Zero to a single AlphaZero algorithm, which can achieve outstanding performance in many challenging chess games. AlphaZero uses deep neural network, general RL algorithm and general tree search algorithm to replace the manual knowledge and domain-specific enhancement used in traditional game programs. Based on the only game rules, AlphaZero overwhelmingly defeated world champion programs in the fields of Go, Chess, and Shogi [6].

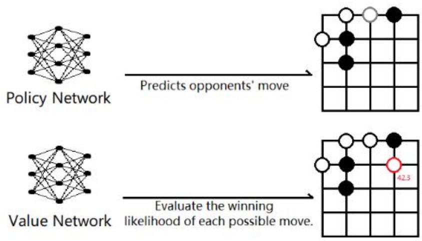

Fig. 3. AlphaGo uses two neural network to make decision [5].

On the other hand, the developer of AlphaGo, DeepMind, provides a teaching tool that demonstrate the evaluation result of AlphaGo to help people examine their moves of playing Go and find a new way of playing Go [7]. This tool is helpful for training Go players with a certain level of proficiency by showing how AlphaGo "thinks" of a certain step (Fig.4). It can help players discover the flaws and problems in their policies, and also assist them in finding new strategies to improve their Go skills.

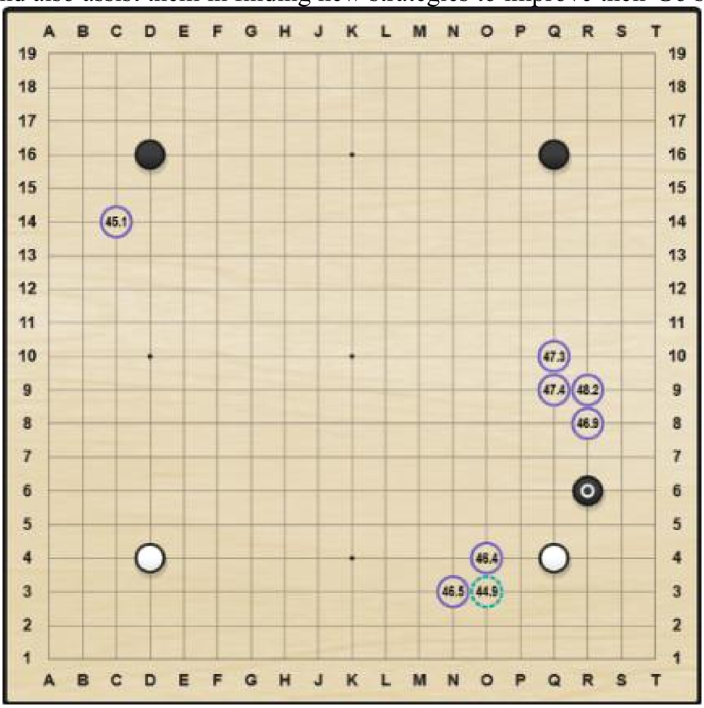

Fig. 4. The teaching tool demonstrates how the AlphaGo "Thinks" [7]

4.2 The Applications in the RTS Games

A multi-agent RL algorithm was constructed by Oriol Vinyals et al. utilizing data from both human players and computer players. The agent AlphaStar was trained using a combination of supervised learning and RL. However, in the context of RL, certain challenges were encountered. On the one hand, due to the complexity of the maps and the scarcity of rewards, exploration became difficult for the agents. On the other hand, the game had a large time span, and the action space of the units was complex, making off-policy learning a formidable task. Nonetheless, AlphaStar was still rated as a Grand Master in all three StarCraft II race trials, surpassing 99.8% of human players at the same level [8].

4.3 The Applications in the Platform Jumping Games

The platform jumping game is a type of games that holds simple rules. Players need to control the game characters jumping over platforms to avoid obstacles, traps and monsters.

A type of intelligent agent called the Deep Q Network (DQN) has been developed by Volodymyr Mnih et al. The DQN combines RL with deep neural networks, a class of artificial neural networks, which addressed the instability issue in RL when using nonlinear function approximators. In this case, the deep neural network replaces the role of Q function.

Additionally, the DQN achieved significant success in six games (Boxing, Breakout, Crazy Climber, Demon Attack, Krull, and Robotank) run on the Atari consoles. It outperformed professional players in these games, showcasing the effectiveness of the DQN approach [9].

4.4 The Application in Multiplayer Online Battle Arena (MOBA) Games

MOBA games are known for their high complexity, posing significant challenges in AI action prediction and decision-making. Tencent's team focused on training a 1v1 AI model for Honor of Kings, a popular MOBA game, and achieved impressive results in real player confrontations.

The experimental team at Tencent developed an AI model by dividing their deep learning architecture into four sub-modules: the Artistic Intelligence Server, Dispatch Module, Memory Pool, and RL Learner. To model MOBA action decisions, they designed an actor-critic neural network. The network was optimized using a multi-label proximal policy optimization (PPO) objective and incorporated techniques such as decoupling methods for action dependencies, attention mechanisms for target selection, action masks for efficient exploration, LSTM for learning skill combinations, and an enhanced version of PPO called dual-clip PPO for improved training convergence [10].

During the training process, Tencent quantitatively evaluated the results by measuring the beating speed against a weaker model and imposed an APM (actions per minute) limit of 133ms for the AI. The AI model achieved an impressive win rate of 0.998 in 2100 games against top amateur players and only lost one set against professional players. When benchmarked against the ELO rating system, the AI's decision-making and gameplay abilities in MOBA games reached the level of professional players [11].

4.5 The Application in Game Balancing

Not only directly participate in the game to compete with human players in the competitive games, the RL can be also applied to the game system itself the learn from the human players' playing and adjust the difficulty of the game itself to provides the player a smoother and comfortable game experience while not losing the challenge.

A simple game, called *RoguelikeRL*, developed by Matt Gray to demonstrate such a concept, is a game that has a built-in RL model to monitor the performance of the player (Fig.5). If the agent find it set too many obstacles that might defeat player, it will put more items to help player overcome the difficulties. And if it finds that the player conquers the dungeon too easy, the agent stops setting helpful items and set difficulties. The developer also set a global model to initialise the model. At the most beginning, the global model set up the stages and starts to learn from the player. Through the playing of player, the model eventually be trained to fit the personal style of the player to provides a customised, challenging and smooth experience of playing for the player [4].

This demonstration shows the potential of the RL in the dynamic game balancing, the RL-based balancing system can provide each player with a unique and delighting game experience.

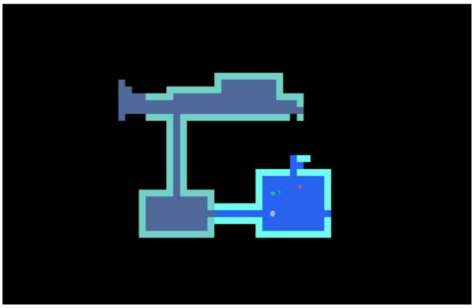

Fig. 5. The *RoguelikeRL*, a rough prototype of roguelike game [4]

5 The Future of RL in the Game Field

In the future, video games will increasingly incorporate features such as multiplayer confrontations, team building, and challenging levels with numerous small monsters and bosses. These specific scenarios create a demand for well-designed game AI that can quickly generate a large number of high-quality intelligent agents.

Currently, AI in games can achieve or even surpass the gaming level of amateur human players across various game types. Moving forward, the focus for game companies will be on creating diverse gaming styles for AI to offer unique experiences to players in games. For instance, when AI serves as the rivals, the training model should aim to enrich the AI's styles of decision-making. Some AIs may exhibit aggressive behaviour towards the player, while others may adopt a cautious style of hiding and ambushing. As a teammate, AI can play the role of a supporter, serving the player character (PC), providing support and cooperation with the PC to allow the player to confidently charge into battle; It can also act like a defender, appearing braver, leading the way, taking damage and requiring the support from the PC, making the player feel needed. In the role of scene Non-Player's Characters (NPCs), AI should respond differently to the player's actions, creating a more lifelike interaction rather than relying on fixed actions.

It is predicted that as RL continues to advance in the gaming industry, AI with personalised and distinctive characteristics will become a key selling point for new games, attracting players. The gaming field serves as an excellent testing ground for AI, and the demands of players will drive further development and breakthroughs in AI technology. Based on the deficiencies observed in training models, the future development of AI in gaming is expected to focus on the following directions:

Autonomous Learning and Innovation: The aim is to reduce the reliance on preestablished rules and algorithms by enhancing AI's ability to innovate and learn autonomously. This would enable AI to adapt and respond dynamically to new gaming scenarios, leading to more unpredictable and engaging gameplay experiences.

Optimisation of Training Methods: Efforts will be made to improve and optimize training methods to minimize the computational resources and time required for training AI models. This would allow for more efficient and scalable training processes, making it easier to develop high-quality intelligent agents within reasonable timeframes.

Enhance Generalisation Ability: The goal is to enhance the generalization ability of AI, enabling it to handle complex scenarios and adapt to various game types. This would involve training AI models that can effectively transfer learned knowledge and skills from one game to another, leading to more versatile and adaptable AI agents.

Multi-Agent Learning and Collaboration: Emphasis will be placed on improving swarm intelligence through multi-agent learning and collaboration. This involves training AI agents to work together, communicate, and coordinate their actions, leading to more sophisticated and realistic behaviours in multiplayer gaming scenarios.

6 Conclusion

This essay explores the fundamental concepts of RL and its various applications in the video game industry. The focus is on its utilization in RTS games, platformer games, MOBA games, chess Games and the interplay between game intelligence and RL. Traditional intelligence implementation in games, the integration of RL-based intelligence, and specific use cases of RL in games are also discussed. Lastly, the future of RL in the gaming industry is examined.

To summarize, RL exhibits great potential and is making notable advancements in the gaming field. However, there are certain limitations in the current application of AI in games. Existing AI models face challenges in handling complex situations and long-term planning, particularly in RTS games. Training RL models often necessitates substantial computational resources and time, which can impede their widespread adoption in real-world game development. Additionally, some AI models rely heavily on predefined rules and algorithms, limiting their creativity and adaptability in unfamiliar environment.

References

1. R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.
2. P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup and D. Meger, "Deep reinforcement learning that matters," Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1, 2018.
3. D. Jagdale, "Finite State Machine in Game Development," Algotithms, vol. 10, no. 1, 2021.
4. M. Gray, "Developing a Roguelike Game with Reinforcement Learning using GCP," 27 Jan. 2021. Available: https://towardsdatascience.com/developing-a-roguelike-game-with-reinforcement-learning-using-gcp-46a9b2f5ca3.
5. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot and others, "Mastering the game of Go with deep neural networks and tree search," nature, vol. 529, no. 7587, pp. 484--489, 2016.
6. D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel and others, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," Science, vol. 362, no. 6419, pp. 1140--1144, 2018.
7. DeepMind, "AlphaGo Teach: Discover new and creative ways of playing Go," DeepMind Technologies Limited, 2017. [Online]. Available: https://alphagoteach.deepmind.com/. [Accessed 30 6 2023].
8. O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. a. D. A. Mathieu, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev and others, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," Nature, vol. 575, no. 7782, pp. 350--354, 2019.
9. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski and others, "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529--533, 2015.
10. S. Huang, W. Chen, L. Zhang, S. Xu, Z. Li, F. Zhu, D. Ye, T. Chen and J. Zhu, "TiKick: Towards Playing Multi-agent Football Full Games from Single-agent Demonstrations," arXiv e-prints, p. arXiv:2110.04507, Oct. 2021.
11. D. Ye, Z. Liu, M. Sun, B. Shi, P. Zhao, H. Wu, H. Yu, S. Yang, X. Wu, Q. Guo, Q. Chen, Y. Yin, H. Zhang, T. Shi, L. Wang, Q. Fu, W. Yang and L. Huang, "Mastering Complex Control in MOBA Games with Deep Reinforcement Learning," arXiv e-prints, p. arXiv:1912.09729, Dec. 2019.