Proceedings of the 2023 International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/8/20230255

Studies advanced in artificial intelligence based game

Jiayu Zhou

The School of Software Engineering Chongqing University, Chongqing University, Chongqing, 401331, China

20204372@stu.cqu.edu.cn

Abstract. Thanks to the rapid development of high-performance computing equipment and pattern recognition technology, artificial intelligence technology has gradually matured in recent years and is widely used in many fields. Due to its potentially huge economic value, AI-based game production has attracted a lot of research attention from academia and industry in recent years. AI has progressed from using simple rules to playing a variety of games, and it has proven to be a formidable opponent, defeating human champions in some instances. This paper provides an overview of the algorithms used in different games and highlights the power of AI demonstrated in chess games. However, despite its successes, there are still some problems that exist. This paper concludes by discussing possible solutions to these issues.

Keywords: deep learning, artificial intelligence, gaming.

1. Introduction

Since the inception of artificial intelligence, algorithms have been designed for their application in games, as shown in Figure 1. For example, in the Theseus game, the mouse learns through recorded paths and eventually succeeds in finding the route [1]. In the NIM game, a classic model in game theory, the machine searches for a winning state [2]. However, with the large number of solutions generated as the game progresses, a reasonable search approach aims to achieve the target solution at a small cost. Over the following decades, AI has matured, especially since IBM Deep Blue defeated human experts in 1989 [3]. The technology for developing game AI has become increasingly advanced, and has rapidly developed in recent years. Through continuous learning, AI makes appropriate decisions and can ultimately defeat even the most specialized human experts. Moreover, AI has expanded from board games (such as AlphaGo, AlphaGo Zero, and AlphaZero) to games with more complex rules [4-5]. For example, DeepStack is designed for card games, AlphaStar for strategy games, Suphx for mahjong, and DouZero for landlords. AI is learning and proficiently using more and more rules [6].

Games are becoming increasingly important research objects due to the combination of various technologies, software, and hardware. The fusion of artificial intelligence and machine learning is one of the most notable algorithms in digital gaming research, which enabled the algorithm to defeat a human chess champion [7,8]. The small search space of games and the complete definition of variables and rules make algorithms highly applicable to games [9]. The key component of AI/ML interaction is the choice and role of the data or content used or generated. Data obtained from players' gameplay (including visual information and behavior) can be used not only to study player behavior but also to generate data for AI learning or game content generation [10-12]. AI/ML algorithms have become

© 2023 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

important research objects in games because they can increase player engagement, generate diverse game content, and provide information about player actions to select sources of big data. By combining game design concepts and player behavior, AI/ML can provide information about the player experience and generate predictive content to maximize engagement [13]. Games will no longer be the only applications for AI/ML algorithms as players provide the systems more information through gameplay and obvious choices.

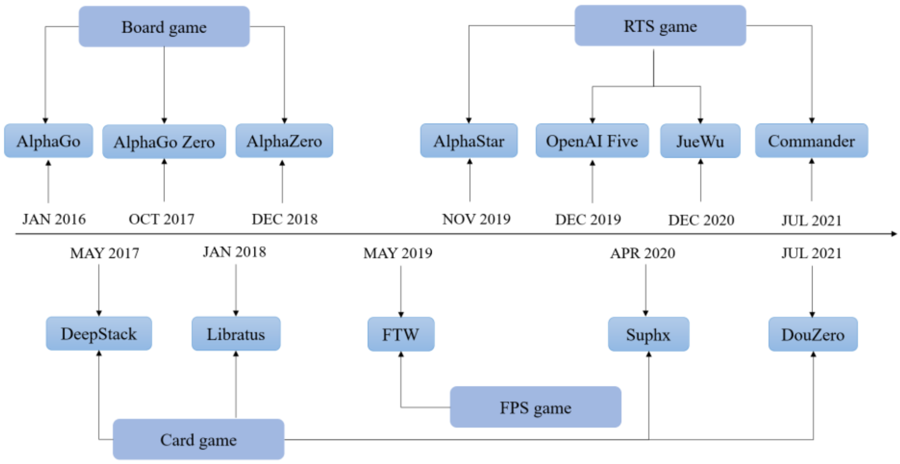

Figure 1. A timeline of AI applications in games.

2. Methods

Machine learning is a primary way to understand games by learning the rules, players, and taking advantage of their strengths, while also being adaptive to have a wide range of capabilities. Building a machine learning model requires a method for model space bias, which is crucial for learning efficiency and accuracy. Performance is also important, as the machine needs enough memory to be motivated to learn. In the following, we will introduce relevant aspects from multiple theories and techniques.

2.1. FSP algorithm

FSP algorithm is a virtual algorithm that developed from the need to traverse all the states to the machine to take policy independently [14]. This algorithm is based on self-gaming and uses data generated from sampling for learning, combining reinforcement learning and supervised learning to find the appropriate strategy.

2.2. Adversarial learning in machine learning

Adversarial learning in machine learning is a classification game between a classifier and an adversary, where the optimal strategy is to maximize the expected outcome while minimizing the advantage of the adversary. In sequential games, the leader makes a strategy first, taking advantage of the priority action, and the follower adopts the optimal strategy based on the action.

2.3. Policy networks

Policy networks alternate between convolutional layers and nonlinear rectifiers in supervised learning, with the input being the state of the board and the final output being the probability distribution of all legal moves [15]. The policy network is trained on randomly sampled state-action pairs, using stochastic gradient ascent to maximize the likelihood of selecting a particular action in a given state. Policy gradient reinforcement learning (RL) seeks to enhance reinforcement learning by preventing the present policy network from becoming overfit by playing a game with a prior iteration of a randomly chosen policy network.

DOI: 10.54254/2755-2721/8/20230255

2.4. Deep neural networks

Deep neural networks have revolutionized the field of game playing AI by enabling machines to play board games with unprecedented mastery. These networks are designed to take in current and historical board positions as input and output move probabilities and a scalar evaluation that estimates the current probability of winning.

One key advantage of deep neural networks is that they combine the roles of strategic and value networks into a single architecture, resulting in more efficient and effective search strategies. The architecture of these networks typically consists of residual blocks of convolutional layers, which are updated continuously based on the outcomes of self-play games. These networks are trained to get better over time by reducing the error between the predicted value and the self-play game winner and maximising the similarity of the neural network move probability to the search probability.

Two prominent examples of deep neural network-based game playing AI systems are AlphaGo Zero and AlphaZero [16]. While both systems use deep neural networks, they differ in their algorithms. AlphaZero estimates and optimizes the expected outcome, taking into account draws or potential alternative outcomes, while AlphaGo Zero transforms the board positions before evaluating them with the neural network in order to average over different deviations. Overall, deep neural networks have proven to be a highly effective tool for advancing the field of game playing AI.

2.5. Policy space response prediction

Policy space response prediction starts with a single policy and grows through continuous iterations, with empirical responses as inputs to the meta-strategy solution. Although PSRO is general, it takes a long time to converge, so a practical parallel form of deep cognitive hierarchy is introduced.

2.6. Suphx

Suphx is an AI system designed for playing Mahjong games, which involves three main learning steps. As shown in Figure 2, in the first step, the system learns from the state and behavior of top human players using supervised learning. In the second step, reinforcement learning is used to further improve the supervised model to handle complex rules. In the third step, the system adopts a gradient strategy algorithm along with a global reward prediction mechanism, which predicts the final reward based on the current round and previous information using a recursive neural network. Additionally, prophetic guidance is utilized to provide information about all pieces to the AI, allowing it to better handle the challenges posed by the game [17]. During gameplay, the AI can adjust its strategy based on the current situation and observe the opponent's actions during their turn to facilitate better decision-making.

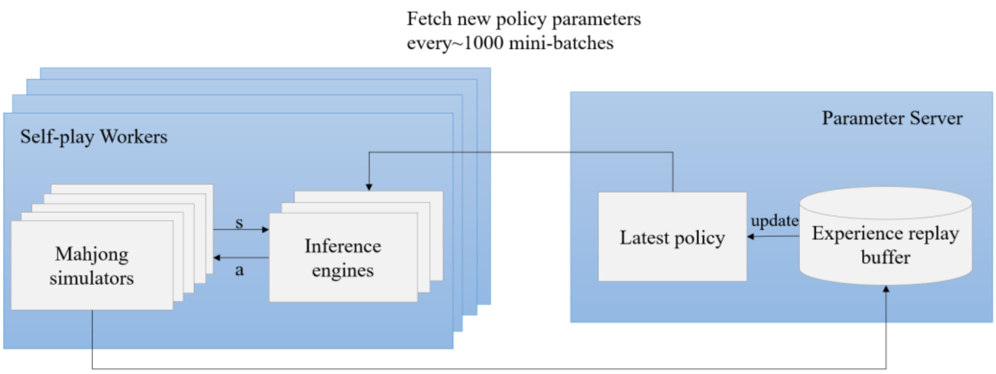

Figure 2. Distributed RL system in Suphx.

2.7. Virtual neural self-gam

Virtual neural self-game combines fictitious self-play (FSP) with neural network function approximation to learn to play games. The algorithm maintains two datasets: one containing experience obtained during self-play, and the other containing the best response behavior observed during self-play. These datasets are then used to train two neural networks. The first network is trained using non-policy reinforcement learning to predict action values from data in the experience dataset, which defines an approximate best response strategy. The second network uses supervised classification on data in the behavior dataset to mimic the agent's past best response behavior. Finally, the two strategies are combined to select actions. This approach has been shown to be effective in learning to play complex games such as poker and Go.

2.8. AlphaHoldem

AlphaHoldem is an end-to-end learning framework used to efficiently and effectively make decisions in imperfect information games (IIG), such as poker [18]. The goal is to reduce the computational cost of Counterfactual Regret Minimization (CFR) in both the training and testing phases.

As shown in Figure 3, the framework uses deep reinforcement learning (RL) to learn the underlying relationships between the game's information, actions, and rewards, and capture the complex dynamics of the game. The inputs to the framework include both action and card state information. A concatenated architecture is used to isolate and process different kinds of information, allowing the model to learn adaptive feature representations. The fully concatenated layers are then fused to produce the desired actions.

To obtain valid and appropriate state representations, new multidimensional features are used to represent current and historical information in the game. The elements that drive the model learning are the game's rewards. A loss function is proposed to update the model and train this pseudo-connected architecture. A non-policy RL is used to update the model parameters and draw different training data from the game state space.

Overall, AlphaHoldem combines deep learning with RL to perform efficient and effective decision-making in IIGs, and represents a promising approach to developing intelligent agents that can play complex games.

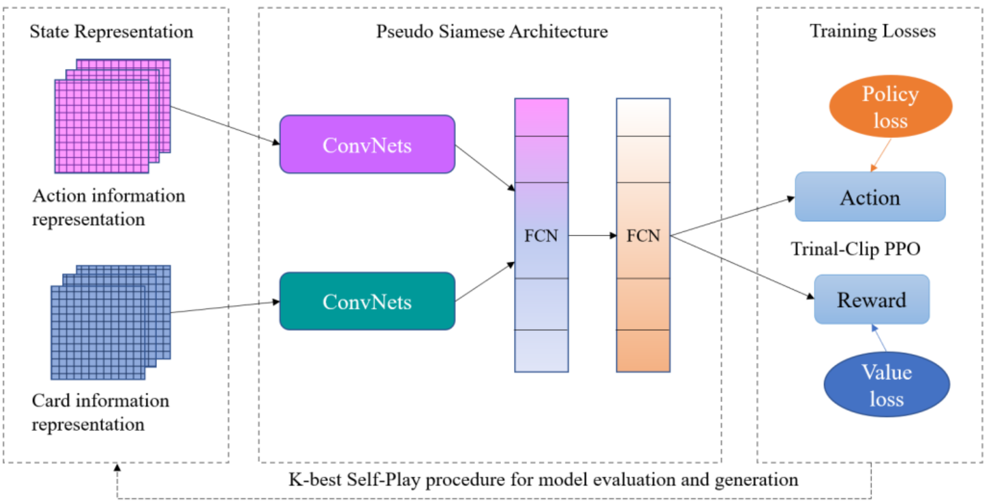

Figure 3. End-to-end learning architecture of AlphaHoldem.

3. Experiments

Considering only the context of a board game, the different algorithms also demonstrated strong performance, with a relatively high win rate against their opponents.

AlphaGo: To evaluate AlphaGo's performance, variants of AlphaGo were played against other Go programs, including robust commercial programs and programs with higher search performance. As shown in Table 1, with a winning percentage of 494 out of 495 games (99.8%), stand-alone AlphaGo

DOI: 10.54254/2755-2721/8/20230255

outperformed all previous Go programs. The distributed AlphaGo, on the other hand, had a much stronger performance, winning 77% of the games against single-player AlphaGo and 100% against other programs. The AlphaGo variants evaluated positions using only the value network (λ=0) or only the rollover (λ=1). Even without rolls, AlphaGo outperformed all other Go programs, suggesting that the value network provides a viable alternative to Monte Carlo evaluation in Go. However, the hybrid evaluation (λ = 0.5) performed the best, winning ⩾95% of its games against the other variants.

AlphaGo zero: Again, a fully trained AlphaGo Zero went through an internal competition with other Go programs. Without using any foresight, the original neural network had an Elo score of 3,055. In comparison to AlphaGo Master (4,858), AlphaGo Lee (3,739), and AlphaGo Fan (3,144), AlphaGo Zero scored 5,185. Finally, AlphaGo Zero was evaluated against AlphaGo Zero won 89 games to 11.

AlphaZero: For Chess, Shogi, and Go, the same algorithm setup, network structure, and hyperparameters were used for all three games, with separately trained instances used for each game. In chess, AlphaZero outperformed Stockfish in only 4 hours (300,000 moves); in shogi, AlphaZero outperformed Elmo in less than 2 hours (110,000 moves); and in Go, AlphaZero outperformed AlphaGo Lee in 8 hours (165,000 moves). AlphaZero searches at a lower depth but has a more efficient thinking time by focusing on certain variables through deep neural networks.

Table 1. The performance of several algorithms.

Game	White	Black	Win	Draw	Loss
Chess	AlphaZero	Stockfish	25	25	0
	Stockfish	AlphaZero	3	47	0
Shogi	AlphaZero	Elmo	43	2	5
	Elmo	AlphaZero	47	0	3
Go	AlphaZero	AG0 3-day	31	-	19
	AG0 3-day	AlphaZero	29	-	21

LCZERO: By studying the positions searched, due to the prior probability, once LCZero determines that all moves seem to fail, it tries to focus on the move that has the most promise to extend the game. Since the remaining nodes require a deeper search, the engine prioritizes searching other lines in the face of an unfavourable position. The checkmate was achieved after searching about 5.5M nodes, S, while Stockfish searched about 500M nodes before finding the checkmate. Even though Stockfish's search was significantly faster than LCZero, LCZero could find the best checkmate route after searching order of magnitude lower number of nodes.

Table 2. The performance of LCZero.

	d8	c6	d8+	f6+	c2+	xe3
Q-value	-4.67	-5.02	-5.44	-5.61	-6.46	-8.88
Win Prob.	5.86%	5.44%	5.02%	4.86%	4.2%	3.01%
Policy	3.11%	7.23%	13.33%	7.40%	9.22%	15.75%
Visits	0.53%	2.39%	1.27%	0.36%	1.07%	92.35%
Moves left	73.6	79.3	73.8	63.7	80.5	95.9

As shown in Table 2, comparing the four algorithms, they all demonstrate great computational power while choosing different decision-making approaches in the face of the opponent's actions. As technology evolves, the strategies adopted by AI in chess games become more and more efficient.

4. Discussion

In the current era of AI technology development, while many techniques have advanced significantly, there are still several areas that require improvement in the learning process, as outlined below.

For games with similar rules, the generalization ability of AI still needs to be improved. Due to rule differences, AI can usually only perform well in a particular game, requiring a new algorithm to be designed for each game category. To build more generalized deep learning models that learn higher-level strategies for decision-making, any helpful information that can be provided should be used for training. The training data also needs to be increased to make the model sustainable for learning. Thus, a single model can solve problems that need to be tackled individually in multiple games.

In deep reinforcement learning, excessive data and computation become a complex problem. Searching for the best result requires significant computing power, and the intelligent strategy will gradually decline as the computation is passed. Additionally, the current distributed training framework faces significant challenges in performing computation with large amounts of data. Therefore, the training framework must be made more computationally powerful to ensure correct decisions as much as possible.

In reinforcement learning, there are cases where untrustworthy information is learned, leading to errors. Imperfect and misleading information can result in learning biases, leading to incorrect optimal solutions. Therefore, it is essential to establish a system for evaluating various learned information, enabling input information to find the optimal result while discharging interference effects and making decisions to find the best solution, even in cases of incomplete information.

5. Conclusion

This paper has provided a comprehensive overview of the current AI techniques employed in various fields, including games, machine learning, and deep learning. The performance of AI has significantly improved over time, and several approaches have been compared in this paper. However, despite these advancements, there are still challenges that need to be addressed.For instance, the generalization of AI for games with similar rules needs to be improved to create more generalized models capable of learning higher-level strategies for decision-making. Additionally, the massive computational requirements for deep reinforcement learning pose a significant challenge. Furthermore, there is a need to establish a system for evaluating the information learned in reinforcement learning to ensure that the model can output optimal solutions accurately, even in cases of incomplete or misleading data. In conclusion, the field of AI has made remarkable progress, but there is still room for improvement in addressing these challenges.

References

[1] Yin Q, Yang J, Huang K, Zhao M, Ni W, Liang B, Huang Y, Wu S, Wang L. 2022. J. Ai in human-computer gaming: Techniques, challenges and opportunities. arXiv.org. Retrieved March 13, 2023.

[2] Karpouzis K, Tsatiris G.A. 2022. J. AI in (and for) Games. In: Tsihrintzis, G.A., Virvou, M., Jain, L.C. (eds) Advances in Machine Learning/Deep Learning-based Technologies. Learning and Analytics in Intelligent Systems, vol 23. Springer, Cham.

[3] Bowling, Michael, Fürnkranz, Johannes, Graepel Thore, Musick Ron. 2006. J. Machine learning and games. Machine Learning. 63. 211-215. 10.1007/s10994-006-8919-x.

[4] Yuan W, Luo J, Lu L, Chen Jg, Zhang W, Chen J. 2022. J Methods in Adversarial Intelligent Game:A Holistic Comparative Analysis from Perspective of Game Theory and Reinforcement Learning. Computer Science, 49(8): 191-204.

[5] Zhou Y, Kantarcioglu M, Xi B. 2018. J. A survey of game theoretic approach for adversarial machine learning.

[6] N Justesen, P Bontrager, J Togelius, S Risi. 2020. J. Deep Learning for Video Game Playing. IEEE Transactions on Games, 12(1):1-20, March 2020, doi: 10.1109/TG.2019.2896986.

[7] Silver D, Huang A, Maddison C, et al. 2016. J. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489 (2016).

[8] Silver D, Schrittwieser J, Simonyan K, et al. 2017. J. Mastering the game of Go without human knowledge. Nature 550, 354–359 (2017).

DOI: 10.54254/2755-2721/8/20230255

[9] Silver D, Hubert T, Schrittwieser J, et al. 2017. J. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815, 2017.

[10] Lanctot M, Zambaldi V, Gruslys A, et al. 2017. J. A unified game-theoretic approach to multiagent reinforcement learning. Advances in neural information processing systems, 2017, 30.

[11] Li J, Koyamada S, Ye Q, et al. 2020. J. Suphx: Mastering mahjong with deep reinforcement learning. arXiv preprint arXiv:2003.13590, 2020.

[12] Heinrich J, Silver D. 2016. J. Deep reinforcement learning from self-play in imperfect-information games[J]. arXiv preprint arXiv:1603.01121, 2016.

[13] Zhao E, Yan R, Li J, Li K, Xing J. 2022. Pro of Int conf. AlphaHoldem: High-Performance Artificial Intelligence for Heads-Up No-Limit Poker via End-to-End Reinforcement Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 36(4), 4689-4697.

[14] Sun Y, Yuan B, Xiang Q, et al. 2022. J. Intelligent Decision-Making and Human Language Communication Based on Deep Reinforcement Learning in a Wargame Environment. IEEE Transactions on Human-Machine Systems, 2022.

[15] Zha D, Xie J, Ma W, et al. 2021. Pro of Int conf. Douzero: Mastering doudizhu with self-play deep reinforcement learning//International Conference on Machine Learning. PMLR, 2021: 12333-12344.

[16] Risi S, Togelius J. 2019. J. Procedural content generation: From automatically generating game levels to increasing generality in machine learning. arXiv preprint arXiv:1911.13071, 2019: 11.

[17] A Sarkar, S Cooper. 2021. Pro of Int conf. Towards Game Design via Creative Machine Learning (GDCML). IEEE Conference on Games (CoG), Osaka, Japan, 2020, 744-751, doi: 10.1109/CoG47356.2020.9231927.

[18] Kopel M, Hajas T. 2018. J. Implementing AI for Non-player Characters in 3D Video Games. In: Nguyen, N., Hoang, D., Hong, TP., Pham, H., Trawiński, B. (eds) Intelligent Information and Database Systems. ACIIDS 2018. Lecture Notes in Computer Science, vol 10751. Springer, Cham.