A Review of the Application of Artificial Intelligence in Imperfect Information Games Represented by DouDiZhu

Chuan He

[1]Fuzhou University, Fuzhou, China, 350000
*Corresponding author. Email: CHUAN.HE.2020@MUMAIL.IE

ABSTRACT
Recently, deep reinforcement learning has achieved superhuman performance in various games such as Go, chess, and shogi. Compared with Go, DouDiZhu, also known as Chinese competitive poker, belongs to Imperfect Information Games (IIG), including hidden information, randomness, multi-agent cooperation, and competition. It is popular in China that it has become a national game. Imperfect information games, as an important branch of machine game, is closer to the decision-making under uncertainty in the complex real world than the perfect information games with transparent opponent information, and has a deeper research value. In addition, the game of DouDiZhu not only contains asymmetric information game, but also coexists competition and cooperation between players, so this relatively underdeveloped field is more considerable for research. This paper outlined the current mainstream reinforcement learning algorithms applied to this game and analyzed their respective advantages and disadvantages. This paper outlined the current mainstream reinforcement learning algorithms applied to this game and analyzed their respective advantages and disadvantages. At the same time, it prospected the future research directions.

Keywords: *Imperfect Information Game, Monte Carlo Tree Search, Convolutional Neural Network, DouZero, DouDiZhu, Reinforcement Learning*

1. INTRODUCTION

In recent years, with the development of reinforcement learning, it has achieved remarkable results in Perfect Information Games (PIG). In March 2016, Google's AlphaGo [1] agent developed through deep learning and reinforcement learning algorithms successfully defeated the world champion of Go, Li Shishi, this further proves the power of artificial intelligence. However, this algorithm relies on a large number of high-quality training samples, which requires a lot of time and hardware resources. In the field of IIG, in January 2017, Libratus [2] Intelligence of Carnegie Mellon University won the final victory against four of the world's top Texas Hold'em players, but this method uses a lot of game theory thinking, so it can only be used for The game rules of a specific game are developed in a targeted manner, and the problem of simultaneous multiplayer gaming has not yet been solved. This paper stated the research results of multiplayer imperfect information game represented by DouDiZhu, and look forward to the future research direction in this area. It is essentially a zero-sum game, and it is also regarded as a challenging benchmark for imperfect information games.

2. GAME RULES

The rules of playing cards in the "DouDiZhu" game.

1) Licensing: 3 players each deal 17 cards, leaving 3 cards as landlord cards. These 3 cards are revealed after the landlord is determined, and then belong to the landlord, that is, the current landlord has 20 cards, and the peasants each have 17 cards.

Determine the landlord: The system randomly selects a player and can choose to be the landlord first. When one player chooses the landlord, the bet is calculated by 3 times the bottom bet. At this time, the next player can choose to grab the landlord or not to grab the landlord. If he grabs the landlord, he becomes the landlord, and the bet is doubled at this time; if he does not grab the landlord, he gives up being the landlord, and then the next player chooses whether to be the landlord until one player

grabs the landlord and the rest of the players do not choose to double the landlord. (or when all other players give up grabbing the landlord), he becomes the final landlord.

2) Playing cards: The landlord first plays the cards, and then in the counterclockwise order, the next house needs to play a bigger card than the previous house, and except for the bomb, the type of cards should be the same as the previous house. If there is no card, choose "pass", and then the next player will play the card. If no other player plays a bigger card after a player's cards are played, the player will get another chance to play cards at will. When a player's hand number is 0, the side wins and the game ends.

3) Card size: the double jokers is the larges; the bomb is smaller than the double jokers and larger than other cards, and the same bomb is compared according to the number of the card; single consecutive cards start from 5 cards, and two pairs start from 3 cards Starting with 3 of a kind, 2 starting with 3 of a kind (you can choose to bring a single card or a pair); except for double jokers and bombs, other card types must be of the same card type and the same number of cards to compare the size, and the size of the single card is from the largest The order from the smallest is: Red Joker > Black Joker > 2 > A > K > Q > J > 10 > 9 > 8 > 7 > 6 > 5 > 4 > 3, regardless of suit.

3. ALGORITHM APPLICATIONS

3.1. Monte Carlo tree search

3.1.1. Basic algorithm

For ordinary game problems, the general method is to use the game tree search. Perform analysis to find out the optimal strategy that game players should take. However, game tree search has great limitations, that is, in game tree search, the tree depth should not be too large at present, because with the increase of game tree depth, game tree The number of nodes that need to be processed increases exponentially, which leads to an exponential increase in the time overhead of processing all the nodes [3].

Monte Carlo Tree Search [4] (MCTS) is a method for making optimal decisions in artificial intelligence problems, generally in the form of action planning in combinatorial games (See Figure 1). It combines the generality of stochastic simulation with the accuracy of tree search. This method can keep the approximate optimality of the solution while reducing the scale of the problem [5]. Among them, the Monte Carlo method uses the empirical average to replace the expectation of random variables. For example, in the state s in the game, the expected value is (s), it is difficult to find this value by calculation, but a series of gains (s), (s), ..., (s) can be obtained by the Monte Carlo method. According to the law of large numbers[6], when n tends to infinity, The mean of the sampling returns is close to the expected value. Define v(s) as the mean of the series returns, n is:

$$v(s)=\frac{G_1(s\ )+G_1(s)+\ldots+G_n(s)}{n}$$

$$v(s) \to v_\pi\ (\ s\ )\ \text{as}\quad n \to +\infty \tag{1}$$

MCTS simulates the real game many times. When the number of simulations is enough, the node with the best profit after the simulation (using the law of large numbers) is close to the theoretical real best profit node. Then the action contained in this node is the best choice in the current state. When it comes to DouDiZhu, MCTS is to frame the optional actions under the current situation of each hand, in line with DouDiZhu's rules and card-playing routines, conduct n simulated games and execute each game. Not exactly the same actions, record and update the benefits of each action, and finally choose the action with the best benefits (play, call, or no).

Each MCTS tree search is divided into 4 steps:

(1) Selection:

Select nodes that are not fully expanded from non-leaf nodes to expand; if all leaf nodes have been expanded, select the node with the highest UCT[7].

(2) Extension:

Choose the first untried action. Create a new MCTS node with this action. The parent node is the current node, the game state is the game state after the action is executed, and the action is the action triggered by the node.

Add the expanded new node to the MCTS tree. Return the new node.

(3) Simulation:

Deduce the game process, and finally return the game result information (generally including game score, winner, etc.).

(4) Backpropagation:

Backpropagating the game results, updating the q and n of each node. Repeat the above 4 steps enough times to select the node with the highest UCT as the next action.

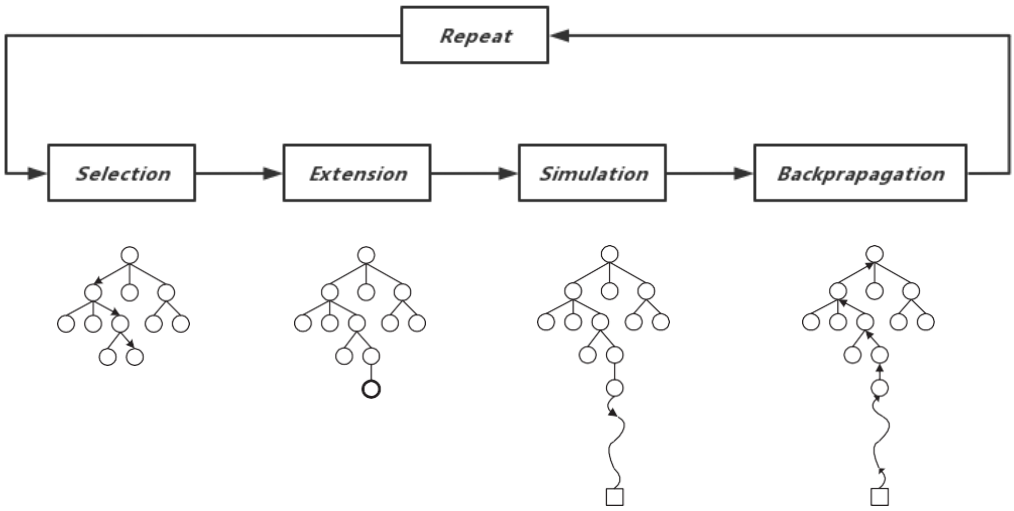

Figure 1. Flow chart of Monte Carlo Tree Search

3.1.2. Advantages and Disadvantages

Experiments show that Monte Carlo tree search can better solve the problem of imperfect information game in the "DouDiZhu" game, but to ensure the accuracy, there are also some shortcomings:

(1) Due to the nature of incomplete information, the number of required nodes is greatly increased, which also directly leads to the huge energy consumption required by this AI agent compared to games such as Go that are applied to games with complete information.

(2) Because it is a multi-player game with imperfect information, its ability to judge the cards of other players is weaker than that of human professional players.

3.2. Convolutional Neural Network

3.2.1. Introduction to Algorithms

Convolutional Neural Networks (CNNs) represent feedforward neural networks consisting of various combinations of convolutional layers, max-pooling layers, and fully connected layers, and exploit spatial local correlation by implementing local connection patterns between neurons in adjacent layers. sex. Convolutional layers alternate with maximum aggregation layers, simulating the properties of complex and simple cells in the mammalian visual cortex [8]. A CNN consists of one or more pairs of convolutional and max-pooling layers and ends up as a fully connected neural network. A typical convolutional network structure is shown in the Figure 2 below [9].

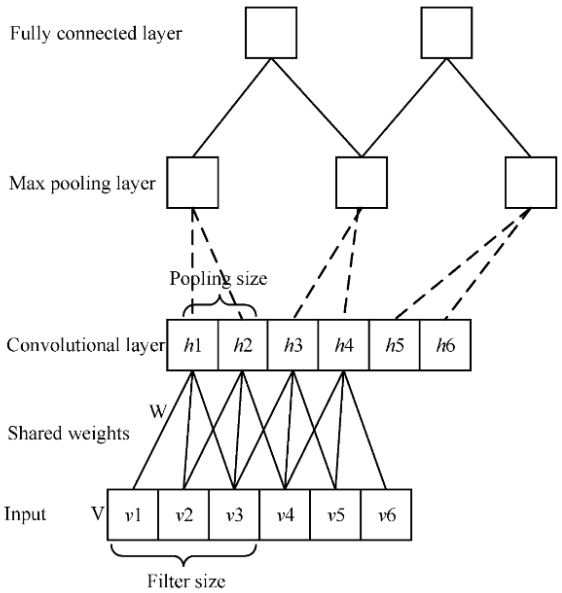

Figure 2. Convolutional Neural Network Architecture

3.2.2. A combat-type agent

In the second session of the Tencent Game Developers Conference, a research team of this company designed a model that arranges the channels of CNN into a 15*4 matrix, representing 3, 4, 5, 6, 7, 8, 9, all the way to jokers respectively and four different suits. In this way, CNN can encode the spatial features of straight, pair, or 3 with 2 through convolution encoding. The output of this model is directly what kind of action is performed. There are 27,472 kinds of action space in DouDiZhu. With this basic model, the researchers carried out the initial training of the first step, but the effect of the initial training was not particularly ideal, including the classification accuracy and AUC were not particularly ideal, so the researchers optimized the entire model:

First of all, they have done a relatively large sorting and optimization of the output of the model. Among all the possible action spaces of DouDiZhu's cards, most of them are concentrated in 4 with 2 or plane with wings, because they can bring arbitrary two, various combinations which means it will generate a large amount of data. Thus they split the original single model into a hierarchical model. The advantage of it is to dismantle the 4 with 2 and the plane with wings on the first layer to the model of the second layer. The first layer is only to identify the possibility of such a situation, and if it exists, it is placed in the second layer model for prediction and classification. In this case, the whole model is more similar to the aircraft and 4 with 2, like wings, so we call this model a layered wing model. In this way, in the second-layer model, the original number of more than 10,000 possible combinations has been reduced to the level of hundreds, and the efficiency has been greatly improved. Secondly, they artificially made some key features, which is of great benefit to the overall accuracy of the model. The researchers also binarized the entire model. Because for convolutional neural networks, binarization usually achieves better results.

After the optimization of the above method, the accuracy of the entire model is improved by about 15%, and then the model is trained by using the data of about 4 million high-magnification games of the DouDiZhu game developed by Tencent, and the model converges to a comparison. Satisfying result. Experimental data shows that AI agents are already at the level of skilled human players in the vast majority of cases, but there are also downsides. For example, the researchers found some problems through a large number of observations, and these problems were particularly prominent at the end of the game. This is because the sample data is extracted from the actual playing process of human players. Because people make mistakes, they will introduce many mistakes made by human beings. Through the previous principles of imitation learning or supervised learning, the robot does not understand what the DouDiZhu is all about. It is just imitating the actions of humans on how to play cards under different boards or imitating the actions of humans under different boards. Therefore, it is not feasible for such a huge amount of data to manually screen out high-quality games for AI agents to learn. At present, researchers have proposed a coping strategy: by imitating the human master's card counting and guessing, it plays an auxiliary role in predicting the way of playing cards by winning percentage, that is, by recording the cards and rules of the other two players, when the opponent's number of cards is small, this auxiliary algorithm will be activated, but the more cards remaining, the more difficult the prediction will be, and the greater the energy consumption will be. The algorithm to achieve the optimization of this problem has become a future research direction.

3.3. DouZero

3.3.1. Theoretical basis

What is more interesting is that the core algorithm of DouZero is extremely simple. The design of bucket zero is inspired by the Monte Carlo method. Specifically, the goal of the algorithm is to learn a value network. The input to the network is the current state and an action, and the output is the expected payoff (such as winning rate) of doing this action in the current state. In simple terms, the value network calculates at each step which hand has the highest probability of winning and then chooses the hand with the highest probability of winning. The Monte Carlo method continuously repeats the following steps to optimize the value network:

Generate a game with the value network. Record all states, actions, and final benefits (win rate) of the match.

Take each pair of state and action as the network input, and the revenue as the network output and use gradient descent to update the value network once.

In fact, the so-called Monte Carlo method is a kind of stochastic simulation, that is, the real value is estimated through repeated experiments. For example, estimating probabilities with frequencies is typical of Monte Carlo methods. The above is a simple application of Monte Carlo methods in reinforcement learning. However, Monte Carlo methods are ignored by most researchers in the field of reinforcement learning. The academic community generally believes that Monte Carlo methods have two shortcomings:

(1) Monte Carlo methods cannot handle imperfect state sequences.

(2) The Monte Carlo method has a large variance, resulting in low sampling efficiency.

However, the authors were surprised to find that the Monte Carlo method works very well. First of all, DouDiZhu can easily generate a complete game, so there

is no imperfect state sequence. Secondly, the authors found that the efficiency of the Monte Carlo method is not very low. Because Monte Carlo methods are extremely simple to implement, we can easily parallelize to collect a large number of samples to reduce variance. In contrast, many state-of-the-art reinforcement learning algorithms have better sampling efficiency, but the algorithms themselves are complex and therefore require a lot of computing resources. Taken together, the wall-clock time of the Monte Carlo method is not necessarily inferior to the state-of-the-art methods. In addition, the author believes that the Monte Carlo method has the following advantages:

(1) Actions are easy to code. There is an internal connection between the actions of DouDiZhu and the actions before. Take three with one as an example: if the agent plays KKK with 3 and is rewarded for bringing the card well, then the value of other cards, such as JJJ with 3, can also be improved to a certain extent. This is because the neural network predicts similar outputs for similar inputs. Action coding is very helpful for dealing with DouDiZhu's huge and complex action space. Even if the agent has not seen action, it can estimate the value from other actions.

(2) Not subject to over-estimation. The most commonly used value-based reinforcement learning method is DQN [10]. But DQN is known to suffer from overestimation, i.e. DQN tends to overestimate values, and this problem is especially pronounced when the action space is large. Unlike DQN, Monte Carlo methods estimate value directly and are therefore immune to overestimation. This is very applicable in the huge action space of DouDiZhu.

(3) Monte Carlo methods may have some advantages in the case of sparse rewards. In DouDiZhu, the rewards are sparse, and players need to play the entire game to know whether to win or lose. The DQN method estimates the value of the current state by the value of the next state. This means that the reward needs to be propagated forward from the last state bit by bit, which can cause the DQN to converge more slowly. In contrast, Monte Carlo methods directly predict the reward for the last state, unaffected by sparse rewards.

3.3.2 How is the "DouZero" system implemented?

The implementation of the bucket-zero system is not complicated, and it mainly consists of three parts: action/state coding, neural network, and parallel training.

(1) *Action/State Coding*

As shown in the Figure 3 below, DouZero encodes all card types into a 15x4 matrix of 0/1. Each column represents a type of card, and each row represents the number of corresponding cards. For example, for four 10s, column 8 has 1 in each row; for a 4, only the last row is 1 in the first row. This coding method can be applied to all card types in DouDiZhu.

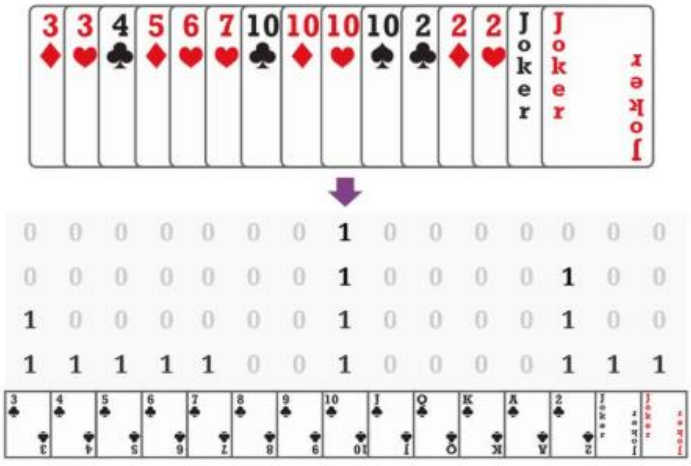

Figure 3. Storage example of a deck of cards

DouZero extracts multiple such matrices to represent the state, including the current hand, the sum of other players' hands, and so on. At the same time, Dou Zero extracts some other 0/1 vectors to encode the number of other players' cards in hand, and the number of bombs currently played. Actions can be coded in the same way.

(2) *Neural Networks*

As shown in the Figure 4 below, DouZero adopts a value neural network whose input is state and action, and the output is value. First, past plays are encoded with an LSTM[11] neural network. Then the output of the LSTM and other representations are fed into a 6-layer fully connected network, which finally outputs the value.

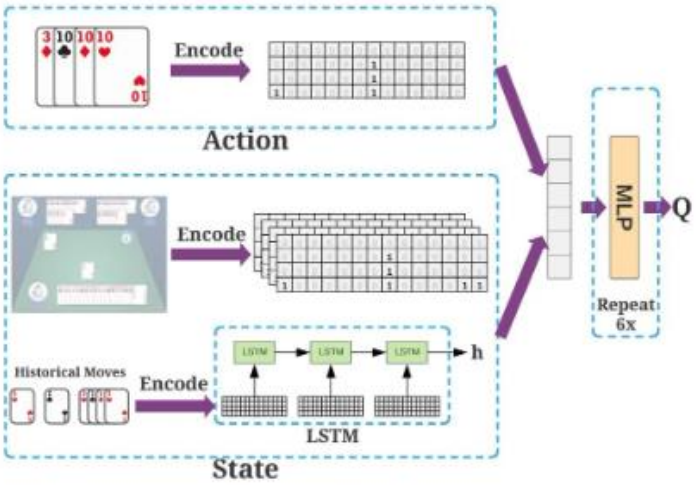

Figure 4. The Q-network of DouZero consists of an LSTM

(3) Parallel Training

The main bottleneck in system training is the generation of simulated data because each card move requires a forward pass to the neural network. Dou Zero adopts a multi-actor architecture. On a single GPU server, 45 actors are used to generate data at the same time, and the final data is collected into a central trainer for training. What is more interesting is that DouZero does not require too many computing resources, and only needs an ordinary four-card GPU server to achieve good results. This makes it easy for most labs to do more experiments based on the author's code.

The success of DouZero shows that a simple Monte Carlo algorithm can have a very good effect on the complex DouDiZhu environment with some enhancements (neural network and action coding). The authors hope that this result will inspire future research in reinforcement learning, especially in tasks with sparse rewards, complex action spaces. The Monte Carlo algorithm has been neglected in the field of reinforcement learning. The author also hopes that the successful function of Dou Ling will inspire other researchers to do more in-depth research on the Monte Carlo method and better understand under what circumstances the Monte Carlo method is applicable. Under what circumstances does not apply.

In order to promote follow-up research, the author has open-sourced the simulation environment of DouDiZhu and all the training codes. It is worth mentioning that Dou Zero can be trained on ordinary servers and does not require cloud computing support. The author also open-sourced an online demonstration platform and an analysis platform to help researchers and DouDiZhu fans better understand and analyze AI's playing behavior. Given that the current algorithm is extremely simple, the author believes that there is still a lot of room for improvement in the future, such as introducing an experience replay mechanism to improve efficiency, explicitly modeling the cooperative relationship between farmers, and so on. The author also hopes to apply DouZero's technology to other poker games and more complex problems in the future.

4. CONCLUSION

Based on previous research results, this paper summarized the current mainstream algorithms used by artificial intelligence agents in Doudizhu games. In general, the AI agents they trained were already excellent and showed strong learning ability and infinite potential in the actual games with human players. However, some aspects of the performance have not met the expectations of the researchers. For example, the variance in the early learning stage is high, the model convergence speed is relatively slow. In many cases, it is very dependent on high-quality training samples, and the training energy consumption is large. Therefore, learning efficiency, learning stability, and even complete self-learning ability will become the focus of future research.

AUTHORS' CONTRIBUTIONS

This paper is independently completed by Chuan He.

ACKNOWLEDGMENTS

Throughout the writing of this dissertation, I have received a great deal of support and assistance.

I would like to thank my supervisor, Professor Pietro Lio', for sparking my interest in Deep Reinforcement Learning. You gave me a lot of academic help during the course study.

I would also like to thank my tutor, Mr. Ye, for his valuable guidance throughout my project studies. You provided me with the useful tools that I needed to choose the right direction and complete my paper.

Finally, I would like to thank my parents for the material and spiritual help they gave me, for giving me wise advice when I was at a loss. You are always there for me.

REFERENCES

[1] D. Silver, A. Huang, C. J. Madduson, et al. Mastering the game of Go with deep neural networks and tree search [J]. Nature, 2016, 529(7587): 484-489.

[2] N. Brown, T. Sandholm, Superhuman AI for heads-up no-limit poker: Libratus beats top professionals [J]. Science, 2018, 359(6374): 418-424.

[3] Q. Peng, Y. Wang, X. Yu, et al. Monte Carlo Tree Search for “Doudizhu” based on hand splitting [J]. Journal of Nanjing Normal University (Natural Science Edition), 2019, 42(3):107-114.

[4] R’emi Coulom. Effificient selectivity and backup operators in monte-carlo tree search. In 5th International Conference on Computer and Games, 2006.

[5] J. Zhang, Research on Risk and Opponent Model in Incomplete Information Machine Games [D]. Harbin: Harbin Institute of Technology, 2015.

[6] H. Ji, A. Zhong, G. Wu, Interpretation of the Theorem of Large Numbers [J]. Heilongjiang Science, 2019,10(2):21-23.

[7] Y. Wang, A. Ding, B. Qi, et al. Texas Hold'em Algorithm Based on Expected Return Strategy and UCT [J]. Journal of Chongqing University of Technology (Natural Science Edition), 2021, 35(3):166-173.

[8] F. Xu, K. Wei, Y. Wang, et al. “DouDiZhu” Strategy Based on Convolutional Neural Networks [J]. Computer and Modernization, 2020(11): 28-32.

[9] T. N. Sainath, A. Mohamed, Kingsbury B, et al. Deep convolutional neural networks for LVCS [C]//2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013: 8614-861.

[10] J. Lei, J. Wang, H. Ren, et al. Incomplete information game algorithm based on Expectimax search and Double DQN [J]. Computer Engineering, 2021, 47(3): 304-310, 320.

[11] HOCHREITERS, SCHMIDHUBER J. Long short-term memory [J]. Springer Berlin Heidelberg, 2012, 8(8): 1735-1780.