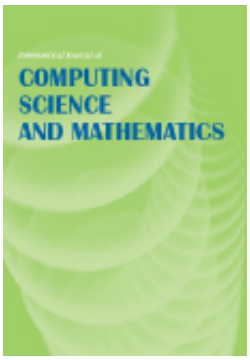
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Computing Science and Mathematics

ISSN online: 1752-5063 - ISSN print: 1752-5055
https://www.inderscience.com/ijcsm

Research on improving Mahjong model based on deep reinforcement learning

Yajie Wang, Zhihao Wei, Shengyu Han, Zhonghui Shi

DOI: 10.1504/IJCSM.2023.10060147

Article History:

Received:	18 July 2022
Last revised:	28 June 2023
Accepted:	26 July 2023
Published online:	22 February 2024

Research on improving Mahjong model based on deep reinforcement learning

Yajie Wang

Engineering Training Center,
Shenyang Aerospace University,
Shenyang City, Liaoning Province, 110000, China
Email: wangyajie@sina.com

Zhihao Wei*, Shengyu Han and Zhonghui Shi

Shenyang Aerospace University,
Shenyang City, Liaoning Province, 110000, China
Email: 15225926807@163.com
Email: hsy_307@163.com
Email: szh13463337193@163.com
*Corresponding author

Abstract: Mahjong is a popular incomplete information game. There are many scholars dedicated to Mahjong research. To improve the game ability of existing Mahjong models. A method based on deep learning and reinforcement learning is proposed. Firstly, a Mahjong program (MPRE) is designed. MPRE is used to generate training data for deep learning and as a comparison program for MPRE_RL, respectively. Secondly, with the feature extraction capability of deep learning, the game ability of MPRE is transformed into a deep learning model. Thirdly, the deep learning model is continuously improved by reinforcement learning. To improve the training speed and stability of reinforcement learning, some improvements are made in the environments and rewards. Finally, the results show that MPRE_RL improved by using the proposed method get a certain enhancement in offensive (27.1% of winning rate) and defensive (19.5% of win by discard rate) aspects compared with MPRE.

Keywords: incomplete information game; Chinese public Mahjong; deep learning; reinforcement learning.

Reference to this paper should be made as follows: Wang, Y., Wei, Z., Han, S. and Shi, Z. (2024) 'Research on improving Mahjong model based on deep reinforcement learning', *Int. J. Computing Science and Mathematics*, Vol. 19, No. 1, pp.83–92.

Biographical notes: Yajie Wang received her PhD in Pattern Recognition and Intelligent Systems from North-eastern University in 2007. Currently, she is a Director of the Chinese Society of Artificial Intelligence, Deputy Director of the Machine Game Committee, and a Member of the Computer Society. Her current research interests include machine games, pattern recognition, image processing and image fusion.

Zhihao Wei received Master of Software Engineering degree from Shenyang Aerospace University. His current research interests include artificial intelligence and applications.

Shengyu Han received his Master of Software Engineering degree from Shenyang Aerospace University. His current research interests include artificial intelligence and applications.

Zhonghui Shi received Academic Master of Computer Technology degree from Shenyang Aerospace University. Her current research interests include computer vision and image understanding.

This paper is a revised and expanded version of a paper entitled 'Research on improving Mahjong model based on deep reinforcement learning' presented at the *6th Asian Conference on Artificial Intelligence Technology [ACAIT 2022]*, Changzhou, China, 9–11 December, 2022.

1 Introduction

Computer game is an important research direction in the field of artificial intelligence (AI). According to the integrity of the game information mastered by the players, computer game can be divided into complete information game and incomplete information game. For complete information game, each player shares the same amount of available information (e.g., features, strategies.), and typical complete information game include Go, Chess, etc.

In recent years, on the one hand, rich results have been achieved by complete information game, such as the AlphaGo (Silver et al., 2016) series of Go, DeepBlue (Campbell et al., 2002) of Chess. On the other hand, in Texas Hold'em, Libratus, an AI program developed by Brown and Tuomas (2018), defeated the world's top human players. In Mahjong, Suphx, developed by Li et al. (2020) through deep reinforcement learning, surpassed 99.99% of players in just three months.

Compared with other incomplete information game, Mahjong has the following thorny problems. First of all, Mahjong is a 4-player game, so there is more information should be considered (e.g., positions, strategies.). Secondly, in the early stage of game, the available information of other players is almost zero. Finally, the action and reward rules of Mahjong are complex and varied, making it difficult to build a game tree to deal with these problems.

The rest of this paper is organised as follows. Section 2 briefly reviews related work of the Mahjong. Section 3 presents the main methods and models implemented in this study. Section 4 describes the experiments and results. Lastly, Section 5 draws the conclusion and discusses future work.

1.1 Our contributions

To improve the agent's game ability, a method that combines deep learning and reinforcement learning is proposed. This method is composed of three parts:

- a Mahjong program based on rules and experience (MPRE) is designed (Section 3.2)

- a large number of game records are generated through MPRE, which are used to train deep learning models (Section 3.3)
- through the exploration and exploitation mechanisms of reinforcement learning, the potential of the model obtained by deep learning is improved (Section 3.4).

2 Related works

In recent years, many achievements have been made in the research of incomplete information game. For example, Texas Hold'em, Libratus designed by Carnegie Mellon University using the counterfactual regret minimisation (CFR) (Zinkevich et al., 2007), beat many of humanity's best players in 2017. Mahjong is also one of the typical representatives of incomplete information game. Researchers have proposed different solutions to deal with Mahjong problems.

In terms of Mahjong rules and experience, Chuang and Wu (2015) designed LongCat based on the idea of quickly winning game and used Monte Carlo Simulation to simulate opponents' private tiles. Lin et al. (2014) designed ThousandWind with the idea of pursuing points and dynamically changing game strategies according to the game process. Handa (2013) proposed the idea of discarding safety tiles, etc.

Kurita and Hoki (2020) proposed to approximate the Mahjong problem as a Markov process. Some researchers use mathematical methods to prove the complexity of tiles' combination, the quality of private tiles (Li and Yan, 2019) and the special win types (K-gates) (Cheng et al., 2017) in Mahjong. After the rise of deep learning, some researchers solved Mahjong problems with the help of deep learning. For example, Mizukami and Tsuruoka (2015), Gao et al. (2019) and Wang et al. (2019) trained an excellent Mahjong program by designing different neural network models.

Most notably, Suphx, a Japanese Mahjong program was developed by Li et al. (2020) through deep reinforcement learning. Suphx was launched on the popular Japanese Mahjong platform "Tenhou", and reached the highest rank of the platform after 5000 games. 99.99% of the platform's human players have been surpassed.

However, there are still improvements that can be made to these Mahjong programs. Firstly, Chuang and Wu (2015) use Monte Carlo to simulate an opponent's private tiles, which lacks the consideration of the opponents' historical information and results in less accurate simulation outcome. Secondly, Handa (2013)'s concept of a safety tiles is only limited to the tiles discarded by other players in this round. Therefore, the scope of the safety tiles is too limited, which should be extended to the whole historical information. Finally, in the reinforcement learning phase of Suphx, the reward is calculated without considering whether the initial private tiles are good or bad. In the proposed method, a weight is set by evaluating the goodness of the initial private tiles, which is added to the reward calculation process.

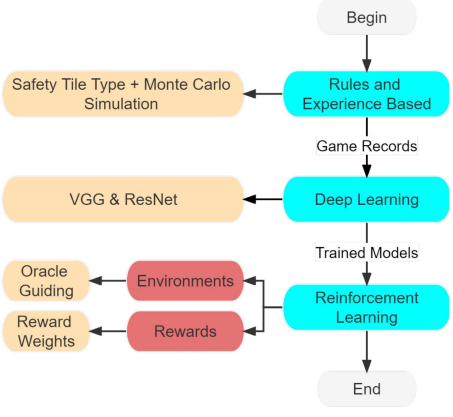
3 Method

3.1 Overall design

The goal of this paper is to improve the game ability of an agent through the proposed method. It can be seen from Figure 1. The overall design of this method mainly consists of

three parts. Firstly, a Mahjong program based on rules and experience is designed (MPRE). Then, to extract the game ability of MPRE into the deep learning model, the game records are generated through MPRE for deep learning training. Finally, the model that perform best in deep learning is continuously improved by reinforcement learning.

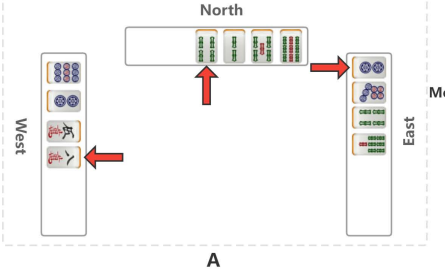
Figure 1 Overall design of the proposed method (see online version for colours)



3.2 Design based on rules and experience

The safety tile type is an improvement on the safety tiles (Handa, 2013). Differences between safety tiles and safety tile type can be better explained by Figure 2. The safety tiles means the tiles discarded by the other three players in the current round. After these tiles are discarded, the other players do not declare any of the three actions of Chow, Pong, and Kong. Therefore, it is considered safer to discard these kinds of safety tiles when you have. The tiles pointed by the red arrow in Figure 2(A) are the safety tiles of the current round. But when you do not have these safety tiles in your private tiles, you may need the help of safety tile type.

Figure 2 Example of safety tile type (see online version for colours)



Monte Carlo Simulation

B

Location\Types	Character	Bamboo	Dots
North	0%	100%	0%
West	50%	0%	50%
East	0%	50%	50%

C

Location\Types	Character	Bamboo	Dots
North	100%	0%	100%
West	50%	100%	50%
East	100%	50%	50%

For example, in Figure 2(A), not only the tiles pointed by the red arrow are considered, but all the tiles discarded by the opponent in the corresponding position will be considered. We calculate the percentage of each type of tiles in the player's discarding history. The table in Figure 2(B) shows the percentage of tiles of each type for the other three players discarded in Figure 2(A). The percentage of each type of tiles is the safety tile type.

As shown in Figure 2, when start simulating the opponent's private tiles by Monte Carlo Simulation. First, the proportion of each type in the discarding history of each player in Figure 2(B) is calculated through Figure 2(A). Then, when combining simulation results with the remaining tiles, the proportion of each type of tile in Figure 2(C) is the probability of taking such tiles as the simulated result. The bigger the probability, the more likely such type of tiles will be a simulated outcome.

3.3 Deep learning

Unlike the methods used by other researchers, this paper does not use existing game records from human experience but the game records generated by MPRE in the competitive mahjong platform we designed. Multiple versions of the program save both the player's hand and the cards played during the game, maintaining a record of the game by playing it multiple times. On the one hand, since the goal of this paper is to enhance the game ability of the existing Mahjong model by the proposed method. More relevant to this goal is to use the data provided by the existing Mahjong model MPRE. On the other hand, given that the rules of the Mahjong game used in this paper are different from those of many online Mahjong games, it is not appropriate to use online human data.

The operation of "selecting actions from different Mahjong game scenes" is approximated as an image classification task. The action selected by MPRE in each Mahjong game scenario is seen as the label for that scene. A $3 \times 27 \times 27$ matrix (as shown in Figure 3) is used to store the information of the Mahjong game. Three channels are used to store information about player's private tiles, discarded tiles which include information about the tiles discarded by all players, and the number of rounds, respectively.

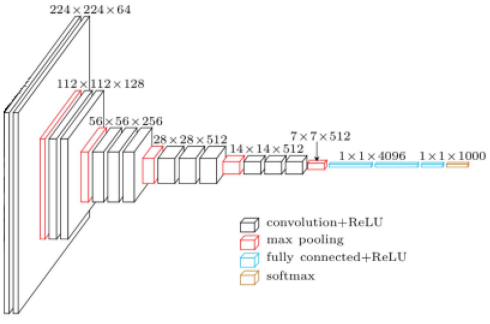
Figure 3 Data structure for information storage (see online version for colours)

As there are 27 unique tiles in Chinese Public Mahjong, a 27×27 multidimensional matrix is used to represent the game states. Since Mahjong is incomplete information game, we need to analyse the state of the player from the information displayed by other players. Therefore, tiles discarded by other players should be saved. Also, it is necessary to save the number of rounds in a game to mark how far the game is progressing.

In this paper, existing classical models from shallow to deep layers are selected. For example, the VGG family and the ResNet family. The same learning rate (1e-4) and optimiser (Adam) are used by these models. The data generated by MPRE (as shown in

Figure 4) is used to train these models, and the best performing model is then boosted by reinforcement.

Figure 4 The network model of MPRE (see online version for colours)



3.4 Reinforcement learning

Reinforcement learning is mainly composed of agent, environments, states, actions, and rewards. The agent is rewarded by the environmental feedback and keeps trying different actions in the environments to get the optimal action in a certain situation. Since Q-learning (Watkins and Hellaby, 2016) is not suitable for handling scenarios with large state space and DQN (Chung, 2016) suffers from overestimation problem, Double DQN (DDQN) (Van Hasselt et al., 2016) is used in the process of reinforcement learning.

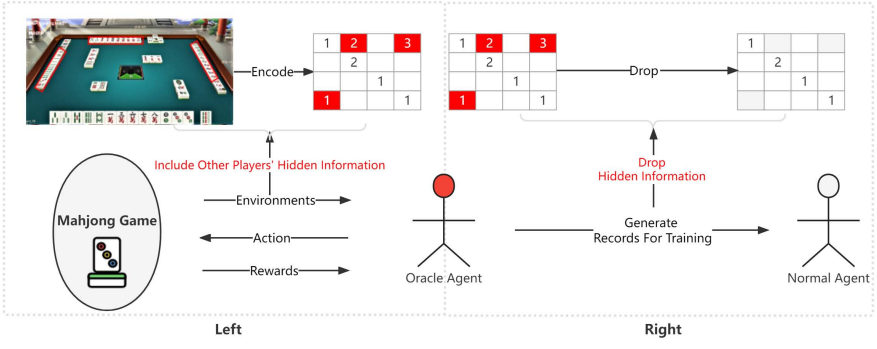
In this paper, both the current Q network and the target Q network in DDQN are replaced with the best performing network in deep learning. In addition, two separate parts are improved. Part one, previous research (Li et al., 2020) has trained a reward predictor by RNN to calculate the reward of the current action. However, because the predictor is trained by human game records, it may have certain limitations of human experience. A reward function based on mahjong rules is designed to break this limitation and to ensure the objectivity of it. The reward rules are shown in Table 1. Different positive and negative rewards are given for different actions according to different feedback of environment.

Table 1 Positive and negative value settings in the reward function

States	*Reward*
Listen num increase	–1
Listen num decrease	+1
Be Chowed	–1
Be Ponged	–2
Be Konged	–3
Win by discard	-6
Others	+0

Part two, A weight by evaluating whether the initial private tiles are good or bad is set. This weight (W) is equal to the number of tiles that are missing from the initial private tiles to make up the winning combination. After the reward function calculates a reward for the action (e.g., –2), product of reward and weight to get the final reward (–2×W).

Figure 5 The process of using and gradually eliminating hidden information from other players (see online version for colours)



In terms of the environments, the private information of the other three players is all opened to the Oracle agent in the early stage of the reinforcement learning (Oracle ability, as shown in Figure 5). By mastering the secrets of other players, the optimal action for the current environment could be quickly discovered by Oracle agent.

However, the condition of knowing the opponent's secret in a real game environments is not possible. Therefore, a way to gradually transform the Oracle agent into a normal agent with some Oracle ability is needed. The way we use is to store the Oracle agent's game records. That is, a game record that contains the secrets is copied to generate a collection of game records that gradually removes the secrets.

4 Experiments

4.1 Deep learning and reinforcement learning experiment results

The experimental environment is: CPU is i7-9700K, GPU is RTX2080, operating system is Windows10, and compilation environment is pycharm2021. For the convenience of description, MPRE_RL is used to describe the improved MPRE model.

Table 2 MPRE_ RL and MPRE comparison test

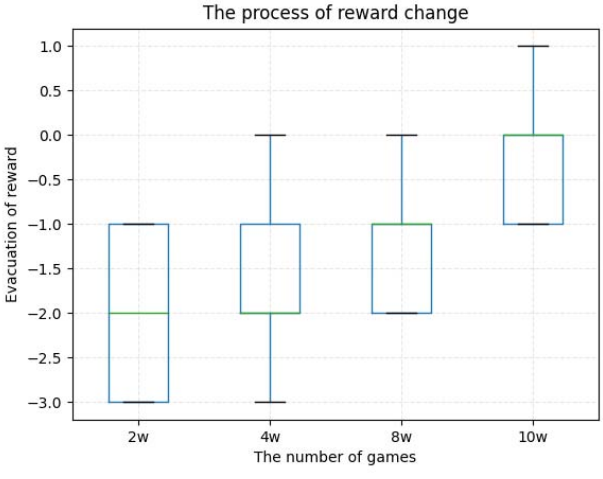
Model	*1st Rank*	*4th Rank*	*Winning rate*	*Win by discard rate*
VGG11	26.8%	22.6%	26.9%	21.7%
VGG16	25.7%	24.1%	26.1%	23.1%
ResNet18	25.3%	23.1%	25.8%	23.6%
ResNet34	24.9%	25.7%	24.2%	24.1%
ResNet50	24.5%	25.3%	24.1%	24.7%
MPRE	27.6%	22.6%	27.3%	20.1%

The experiment is divided into two parts. On the one hand, the performance of each model in the deep learning section on the trained game records is displayed. Reinforcement learning is used to boost the best performing model in deep learning, which is the MPRE_RL (as

shown in Table 2). On the other hand, to prove that the MPRE_RL has better game ability than MPRE, a comparative experiment is held.

During the process of reinforcement learning, reward shows whether an action brings the agent closer to victory or further away. As can be seen in Figure 6, the average reward value in the first 2W rounds is -2.0, which means in most cases the action chosen by the agent keeps the agent away from winning. After 10W games, the average reward value is already greater than zero, that is, in most cases the action chosen by the agent can help the agent move closer to victory. This rising momentum indicates that the agent is continually trade-offs its offensive and defensive ability.

Figure 6 The process of reward change (see online version for colours)



4.2 MPRE vs MPRE_RL

It can be seen from Table 3 that MPRE_RL improved by the proposed method has an improvement of 1.7% in "win by discard" compared to MPRE. As mentioned in the Mahjong rules, "Win by discard" should be avoided as much as possible. Given that this action deducts a significant number of the player's points, so it makes a lot of sense for this to be enhanced. In terms of average winning rate, MPRE_RL outperformed MPRE by 3.2% in the per-game. A good defender in a Mahjong game only ensures lose less points, while to win the game, agent need to rank high or have a high winning rate in most cases. As can be seen from Table 3, MPRE_RL's winning rate and the percentage of top ranking per game are improved compared to MPRE. The effectiveness of the proposed method can be proved by these improvements.

Table 3 MPRE_ RL and MPRE comparison test

Model	*1st Rank*	*4st Rank*	*Winning rate*	*Win by discard rate*
MPRE	24.1%	26.5%	23.9%	21.2%
MPRE_RL	27.3%	23.7%	27.1%	19.5%

5 Conclusion

During the work of this paper, a method to enhance the game ability of existing game model (MPRE) is proposed. For the purpose of providing training data for deep learning, MPRE is designed. In MPRE, the way of an improved safety tile type to incorporate the opponent's historical information into the Monte Carlo Simulation is put forward. Deep learning is used to extract game abilities of MPRE. Reinforcement learning is used to exploit the ability of models trained by deep learning. In the experiments, the comparison data clearly shows that the MPRE_RL obtains a considerable improvement in both offensive (27.1% of winning rate, 27.3% of 1st Rank) and defensive (19.5% of win by discard rate, 23.7% of 4st Rank) capabilities compared to MPRE. These improvements prove the effectiveness of the proposed method.

Several flaws should be noted in further research. It is possible that a better deep learning model can be obtained with higher quality training data. For example, more data like seven pairs, nine gates and other special types of winning combinations.

References

Brown, N. and Tuomas, S. (2018) 'Superhuman AI for heads-up no-limit poker: Libratus beats top professionals', *Science*, Vol. 2018, pp.418–424.

Campbell, M., Hoane Jr., A.J. and Hsu, F. (2002) 'Deep blue', *Artificial Intelligence*, Vol. 2002, pp.57–83.

Cheng, Y, Li, C-K. and Li, S.H. (2017) *Mathematical Aspect of the Combinatorial Game "Mahjong"*, arXiv preprint. Vol. 2017, arXiv:1707.07345.

Chuang, L.K. and Wu, I.C. (2015) 'A study of Mahjong program design', *National Chiao Tung University*, Vol. 2015, pp.136–151.

Chung, J. (2013) 'Playing Atari with deep reinforcement learning', *Comput*, Vol. 2016, pp.351–362.

Gao, S., Okuya, F., Kawahara, Y. and Tsuruoka, Y. (2018) *Supervised Learning of Imperfect Information Data in the Game of Mahjong via Deep Convolutional Neural Networks*, Information Processing Society of Japan, Vol. 2018.

Gao, S., Okuya, F., Kawahara, Y. and Tsuruoka, Y. (2019) *Building a Computer Mahjong Player via Deep Convolutional Neural Networks*, arXiv preprint, Vol. 2019, arXiv:1906.02146.

Handa, H. (2013) 'Evolution of the weight vectors in Mahjong non-player characters', *2013 World Congress on Nature and Biologically Inspired Computing*, Vol. 2013, pp.147–152, doi: 10.1109/NaBIC.2013.6617853.

He, K., Zhang, X., Ren, S., Sun, J. (2016) 'Deep residual learning for image recognition', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2016, pp.770–778

Kurita, M. and Hoki, K. (2020) 'Method for constructing artificial intelligence player with abstractions to markov decision processes in multiplayer Game of Mahjong', *IEEE Transactions on Games*, Vol. 2017, pp.99–110, doi:10.1109/TG.2020.3036471.

Li, S. and Yan, X. (2019) *Let's Play Mahjong!*, arXiv preprint, Vol. 2019, arXiv:1903.03294.

Li, J., Koyamada, S., Ye, Q., Liu, G. and Hon, H.W. (2020) *Suphx: Mastering Mahjong with Deep Reinforcement Learning*, arXiv preprint, Vol. 2020 arXiv:2003.13590.

Lin, S., Chen, C.H., Hsu, S.C., Wu, I.C., Yen, S.J. and Chen, J.C. (2014) 'TCGA 2014 computer game tournament', *ICGA Journal*, Vol. 37, No. 4, pp.226–229.

Mizukami, N. and Tsuruoka, Y. (2015) 'Building a computer Mahjong player based on Monte Carlo simulation and opponent models', *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, Vol. 2015, pp.275–283, doi: 10.1109/CIG.2015.7317929

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D., (2016) 'Mastering the game of Go with deep neural networks and tree search', *Nature*, Vol.2016, p.nature16961, doi:10.1038/nature16961.

Simonyan, K. and Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv preprint, Vol. 2014, arXiv:1409.1556.

Van Hasselt, H., Guez, A. and Silver, D. (2016) 'Deep reinforcement learning with double q-learning', *Deep Reinforcement Learning with Double Q-learning*, Vol. 2016, abs/1509.06461, pp.2094–2100.

Wang, M., Yan, T., Luo, M. and Huang, W. (2019) 'A novel deep residual network-based incomplete information competition strategy for four-players Mahjong games', *Multimedia Tools and Applications*, Vol. 78, No. 16, pp.23443–23467.

Watkins, C. and Hellaby, J.C. (1989) 'Learning from delayed rewards', *Robotics Autonomous Systems*, Vol. 19, No. 4, pp.233–235, doi:10.1016/0921-8890(95)00026-C.

Zinkevich, M., Johanson, M., Bowling, M. and Piccione, C. (2007) 'Regret minimization in games with incomplete information', *Oldbooks.nips.cc*, Vol. 2017, pp.1729–1736.