Reinforcement Learning-based Artificial Intelligence for MahjongSoul

Maoxin Ye[1], Zhehua Zhang[2], Zijian Qiao[3], Jianzhi Wang[4]

[1]23020221154137,Information [2]31520221154237,Information [3]30920221154256,AI [4]30920221157444,Information

Abstract

Since the birth of the game, game AI has come into being. Whether it is the classic Mario, Contra, or one of today's popular games, almost all games are full of AI. From the beginning, it was just to increase the game content and satisfy the players' sense of achievement. Now most of them have surpassed the level of top human players, thanks to the continuous enhancement of computing power and the improvement of algorithms. Mahjong is a popular multi player imperfect information game in the world. In our work, we design a data structure to encode the observable states in Mahjong, construct a CNN architecture to solve the decision-making problem in it, and use supervised learning and RL to train AI for Sanma. Sanma is a three-tier variant of Japanese Niichi mahjong with unique features that are more aggressive than the four-tier game. As such, it is challenging and of research interest in its own right. We developed a mahjong AI called Mastermind for Sanma and tested it in ranked mode to surpasses the average human level.

Introduction

In the past two decades, game AI has made great progress. Recent research has gradually developed from relatively simple perfect information or two player games (such as chess and Go) to more complex multiplayer games with imperfect information (such as bridge, Dota, StarCraft II and Texas Poker). Mahjong is a popular tile-based, multi-turn, multi-player, imperfect information game developed in China in the late 19th century. It has hundreds of millions of players worldwide. This is a game of skill, strategy, and calculation. As an incomplete information game, its AI faces the following challenges:

- Mahjong is called incomplete information game. Each player can hold up to cards that are invisible to others. The cards in the card wall are invisible to all players. In addition, there are cards in the center of the table. This part of the card is only visible when played by the player.
- Although the arrangement and combination result of mahjong pieces is much smaller than that of Go, the difficulty lies in the fact that the cards played by the same player twice are mixed with the cards played by the other players and their own touch. In addition, "Chow, Kong and Pong" will make the game change dynamically.
- It has complex games and scoring rules. There are many ways of playing win cards in mahjong games. Therefore, if you want to build a master mah-jong AI, it is not enough to have strong computing pow-er. What's more, you need to let AI have the ability of intuition, prediction, reasoning and modeling decision-making, which is also the difficulty of building a mahjong AI model.which is also the difficulty of building a mahjong AI model.

The differences between Go and Mahjong determine the fundamental differences in their winning strategies. Go is essentially Monte Carlo trees. The winning strategy is to choose or force the opponent to choose a branch. All the outcomes under this branch are his own. However, due to many changes in chess categories, even AlphaGo is far from being able to traverse the whole tree, so AlphaGo will estimate the probability of winning the outcome of a branch through the value network. The winning strategy of Go is to increase their score expectation (EV) as much as possible.

We use RLCard(Zha et al. 2020) realize mahjong AI. RLCard is a toolkit for Reinforcement Learning (RL) in card games. It supports multiple card environments with easy-to-use interfaces for implementing various reinforcement learning and searching algorithms.

The goal of RLCard is to bridge reinforcement learning and imperfect information games. RLCard is developed by DATA Lab at Texas A&M University and community contributors. RLCard allows developers to focus more on algorithm development without wasting development en-ergy on the game itself(Zha et al. 2020).

Related Work

In 2014, UTokyo launched the "explosive hit" AI, which rose to the ninth section and the stable section around 6.5(Naoki et al. 2014).

The excellent performance of AlphaGo in the Go project has caused an upsurge of intensive learning in the academ-ic and industrial. In March 2016, AlphaGo defeated Lee Sedol, a Korean Go master. In May 2017, at the Wuzhen Go Summit in China, AlphaGo Master fought with Ke Jie, the world No. 1 Go champion, and won by a total score of 3 to 0. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play(Silver et al. 2016).

In 2017, AI Libratus defeated four top Texas poker players in the game of 120000 hands. Libratus of Carnegie Mellon University adopts a combat strategy called ”Nash Equilibrium”. In this strategy, as long as the strategies of other players remain unchanged, a single player cannot benefit from changing strategies. What Libratus needs to do is to identify the hopeless strategies, so as to find the Nash equilibrium point more quickly. After repeated train-ing, Libratus has been able to ignore those bad paths(Brown and Sandholm 2018).

In 2018, Dwango Company launched NAGA25, which has been upgraded to the ninth section and the stable section at about 6.7.

In 2019, OpenAI Five developed by OpenAI on Dota2 can reach the top level of mankind (Berner et al. 2019), and AlphaStar devel-oped by Deepmind on StartCraft can also reach the level of human professional players(Vinyals et al. 2019).And soon after, the Mahjong AI system Suphx(Li et al. 2020), developed by Microsoft Research Asia, became the first AI system to be promoted to ten sections on the internationally renowned profession-al mahjong platform ”tenhou”, which is the best perfor-mance of the AI system in the mahjong field at present, and its strength exceeds the average level of top human players in the open room of the platform.

In 2022, Tencent AI Lab announced a new breakthrough in the chess and card AI ”FineArt”, and defeated the professional champion in 1-on-1 Mahjong benchmark. Under the framework of large-scale reinforcement learning algorithm, the team proposed a new strategy optimization algorithm Actor Critical Hedge (ACH), which partially solved the problem that the self game of large-scale deep reinforcement learning could not converge to the optimal solution of Nash equilibrium(Fu et al. 2021).

Three Player Mahjong Rules and Data Encoding Form

The source of our mahjong matchmaking data is the Tenhou website[1].So we chose the more popular Three-person mahjong on Tenhou as the type of game we experimented with. Three player mahjong, as the name implies, is a type of mahjong played by Three players, which usually does not allow for Chii, but only Pon or Kan.

Players can form an open sequential group, that is, a sequence (3 consecutive cards of the same suit), by calling out ”Chii” in order with the cards discarded by the player on their left. Players place the cards face up on the table, usually on the right side of their hand, with the discard side placed on the far left side of the deck to indicate which card was taken from the left discard pile. A player may form an open identical group, a triplet (3 cards of the same suit, or 3 identical honor cards), by calling ”Pon” with cards discarded by any other player. The player places the cards face up on the table, with one of the cards placed sideways to indicate who the discarder is. Players may splice with 4 identical cards of the same suit or 4 identical honor cards. After the foursome

[1]https://tenhou.net/

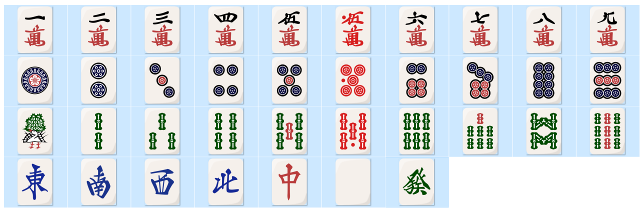
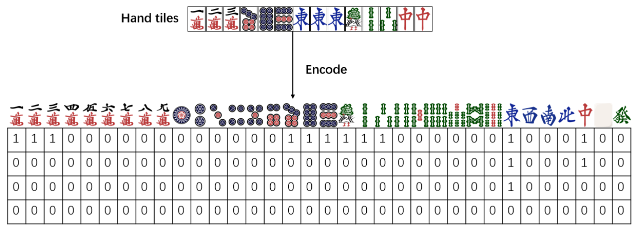


Figure 1: Patterns of Japanese Mahjong



1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2: Encoding of an example hand tiles using a four-row matrix

is called, the next adjacent Dora indicator tile is turned over and the player needs to draw a supplementary tile from the end of the dead wall. Depending on the rules, the number of tiles in the dead wall is kept at 14 by keeping the last available tile of the dead wall, or the number is reduced at that time. There are three types of quads. Players have to shout ”kan” for all quads.

Three player mahjong retains all the mahjong tiles until the last one is touched, which is a relatively common rule, characterized by easy mahjong, but the playing time is similar to four player mahjong. Therefore, some of the rules take the following measures in order to make the number of touches close to that of four-player mahjong. Three player mahjong has only East, South and West. If you keep North Wind it may be used as a special tile with some special notes.

Three player mahjong has 108 blocks of 27 different types of tiles in Figure1, each with four identical Each type of tile has four identical tile faces.

To facilitate training and storage, we use a matrix with 4 rows and 34 columns to store the entire mahjong situation. 14 hands correspond to a matrix in which only 14 positions are 1 and the remaining positions are 0. If there are multiple identical hands, then there are multiple 1 in the same column,in Figure2.

Using matrices makes it relatively easy to put data into a network structure such as a CNN for training, by changing the simulated matchmaking in the RL card to matrix information for training. An Agent can get the information of all the cards that have been played in the whole game, and the information of the other 2 players, Pon, Chii, and Kan, and there are 4 matrices in total.

Propoal Method

Meowjong moves include discard, Pon, Kan, Kita, and Richi (upright). We develop a Meowjong strategy for each move, using a model to parameterize each Meowjong move, called a action model. The action model uses a CNN with 4 convolutional layers followed by a fully-connected layer. The first three layers have 64 filters and the last layer has 32 filters, each of the same size, and the hyperparameters are adjusted separately for each individual action, with 256 hidden nodes in the fully connected layer. A normalization layer and a discard layer with a discard rate of 0.5 are added after each convolution layer and fully connected layer to prevent overfitting. The action model has a similar structure, the size of filters and output dimensions are not the same, where the discard output dimension is 34 and the others are 2. Meowjong's card problem can be seen as a multiclass problem, so we define the loss as:

$$L(w) = -\sum_{i=0}^{m}\sum_{k=1}^{k} {y_k}^{(i)}.log\hat{y}_k^{(i)} \tag{1}$$

m is examples, and K is classes.

The problem of playing cards can be understood as a classification problem, can be divided into 34 class, because there are a total of 34 different types of cards, however, this may appear discard model to choose the hand does not exist, however, in our experiment there is no such situation, for other actions, the player can choose to take or not, so it can be regarded as a dichotomous problem, we set a two-dimensional output for each action. For Meowjong's RL training, we use the Monte Carlo policy gradient method to update the weights of the discarded model w:

$$\grave{w} = w + \eta\gamma^t G_{tw} log\pi(A_t|S_t, w) \tag{2}$$

where S_t denotes the round in which the game is played, A_t denotes the action taken in round t, G_t denotes the accumulated prizes, and finally the score of the round. A detailed description of our RL algorithm can be found in [5]. After hyperparameter tuning, we use $\eta = 10^{-3}$ and $\gamma = 0.99$ as the optimal hyperparameter settings.

Experiment

After completing the model modification, we tested the modified model on the dataset, including offline and online tests.We empirically demonstrate our model's effectiveness on MahjongSoul and achieve results beyond the average hunman level.We train the model on MoPaaS AI with one NVIDIA RTX 3090 GPU and 8GB memory in total.

Experiment Settings

We download dateset from tenhou platform which is an internationally renowned professional janpanese mahjong platform.Unlike Chinese mahjong rules rules are variable, Japanese mahjong rules are fixed, and there are more players around the world.We selected 50000 rounds of 2020 year's game logs from the tenhou.We used the Houou's date for our model training because the houou is only open for the top ranked players,so it represents the highest level of Japanese mahjong.In addition, we selected the 2021 year's game data logs as the test dataset to test the generalizability of our trained model.The downloaded game datasets contains log ates which are processed to generate five different states of training data to train different models.The sizes of the datesets are shown in Table 1.We used TensorFlow to implement our model with about 30 hours and selected test accuracy and win rate as the evaluation metric.We used Gao and Suphx models as baselines and applied to compare to our model.

Table 1: SIZE OF DATASETS.

Model	Dataset Size		
	Training	Validation	Test
Discard	675,185	88,588	147,444
Pon	151,348	16,817	16,887
Kan	34,319	3,814	3,548
Kita	136,924	15,214	15,498
Riichi	109,804	12,201	11,944

Table 2: ACCURACIES FOR THE ACTION MODELS IN SUPERVISED LEARNING.

Model	Test Accuracy		
	Mastermind	Gao et al	Suphx
Discard	63.81%	68.8%	76.7%
Pon	70.95%	88.2%	91.9%
Kan	92.45%	–	91.9%
Kita	89.25%	–	–
Riichi	62.63%	–	85.7%

Main Results

We decomposed the training task into five sub-tasks for each model, which can greatly reduce the size of our training task because of the limitations of equipment.We choosed Adam as optimizer with learning rate 0.001 after Hyperparameter adjustment.Table 2 shows the test accuracy of our model and two baselinses.The test accuracy of Master has been greatly improved, but there is still a gap between it and other models.This is because the number of CNN convolution layers in our model is too small compared to the suphx model, and their model uses a large data set and training time, and we cannot reach their training cost.After completing supervised learning, we will further improve the performance of the model through self-play reinforcement learning.As Figure 3 shows, our model has developed its own mahjong style after 300 rounds of sparring training and has continued to improve its win rate, but there is still a gap compared to human expert players.This is due to the size of our small model and the low training cost.

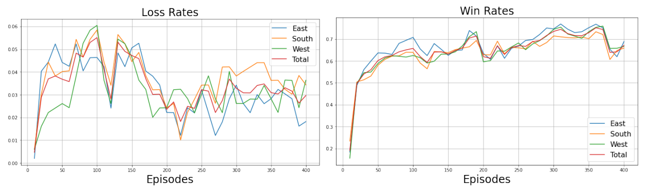


Figure 3: Win and loss rates of Mastermind in training

Conclusion

In this paper, we design a data structure to encode the observable states in Sanma, construct a CNN architecture to solve the decision-making problem in Sanma, and use supervised learning and RL to train several agents. Through supervised learning, all of our action models achieve test nice accuracy and receive significant further enhancements from RL.

One possible future work is to increase the flexibility of this model from a gameplay perspective. For example, if the model encounters a stronger opponent, it could use a more optimal strategy, but if it evaluates the opponent's strength as not being particularly strong, it could hold back some of its own strength in order to make the game more challenging for the player. This could potentially contribute positively to the player's experience.

References

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Brown, N.; and Sandholm, T. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374): 418–424.

Fu, H.; Liu, W.; Wu, S.; Wang, Y.; Yang, T.; Li, K.; Xing, J.; Li, B.; Ma, B.; Fu, Q.; et al. 2021. Actor-Critic Policy Optimization in a Large-Scale Imperfect-Information Game. In *International Conference on Learning Representations*.

Li, J.; Koyamada, S.; Ye, Q.; Liu, G.; Wang, C.; Yang, R.; Zhao, L.; Qin, T.; Liu, T.-Y.; and Hon, H.-W. 2020. Suphx: Mastering mahjong with deep reinforcement learning. *arXiv preprint arXiv:2003.13590*.

Naoki, M.; Ryotaro, N.; Akira, U.; Miwa, M.; Yoshimasa, T.; and Takashi, C. 2014. Realizing a Four-Player Computer Mahjong Program by Supervised Learning with Isolated Multi-Player Aspects.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Zha, D.; Lai, K.-H.; Huang, S.; Cao, Y.; Reddy, K.; Vargas, J.; Nguyen, A.; Wei, R.; Guo, J.; and Hu, X. 2020. RLCard: A Platform for Reinforcement Learning in Card Games. In *IJCAI*.