5T-02

A behavior analysis of sequential and off-table information in the game of Mahjong via deep convolutional neural networks

Shiqi Gao† Fuminori Okuya† Yoshihiro Kawahara† Yoshimasa Tsuruoka†
Graduate School of Information Science and Technology, The University of Tokyo†

1 Introduction

The evaluation function is a very important factor for policy decision but always hard to define exactly especially in imperfect information games. Most of the studies on the game of Mahjong use concrete game rules and use traditional AI methods with artificially well-designed function blocks. For the benchmark of agreement rate on tile discard, traditional baseline is 62.1% [1]. Our past proposal designed a new model with deep convolutional neural networks and raised this result by 6.7% which is much higher than before. However, the model is still not perfect and missed some information. In this paper, we make a comparison of the behavior on past rounds' information learning, and a comparison of behavior on including off-table messages which are missing in our previous models.

2 Past research

Traditional mahjong AI usually has two function blocks, the offense part and the defense part. For the offense part, the tile efficiency is the main factor considered in order for faster winning and higher winning scores, no matter what other players are doing. For the defense part, how to avoid ron (which means defeated) by others is the only thing. The training is to achieve a good balance between offense and defense and decide output strategies in different situations. The state-of-art agreement rate accuracy for haifu learning during test is 62.1% [1].

Deep learning has been a hot research field these several years and is featured for automatic feature extraction ability which needs no artificial extraction. However, how to design the data structure and the network is still a difficult task. Tsukiji [2] designed a 5 by 34 by 5 data structure like an image to contain the tile information, 5 planes for own hand tiles and four players' discard tiles, and the 34 by 5 structure for each plane, achieving a test accuracy at 53.98%. In our previous paper [3], the agreement rate for discard tile learning reached 68.8%, which is 6.7% higher than baseline. However, there is still some part of information not contained in, which are not shown in tiles but can be known by all players, which we call off-table information. Different approaches of containing these information were tried previously, however they did not work well.

{gaosq0604, okuya23, kawahara}@akg.t.u-tokyo.ac.jp, { tsuruoka}@logos.t.u-tokyo.ac.jp

3 Sequential model behavior

Table 1: Last model's input features

Feature	# of planes
Own hand tiles	1
Discarded tiles	4
Stealing tiles	4
Dora indicators	1
Richi players	3
Round wind	1
Own wind	1
Past 1 situation	13
Past 2 situation	9
Past 3 situation	9
Past 4 situation	9

Table 2: Sequential model data structure

Feature	# of planes
Own hand tiles	1
Discarded tiles	4
Stealing tiles	4
Dora indicators	1
Richi players	3
Aka five sign	1
Round wind	1
Own wind	1

The structure of our last model [3] is shown in Table 1. Each plane is a 34 by 4 data structure representing one part of information on desk. In that model, we included information about the last four rounds, because past decisions have great influence on present choice making. However, for real games, only four past rounds are not enough. In this section, we rearrange the input structure and adopt a sequential model to see its performance.

Our new input features are shown in Table 2. The new structure has 16 planes. As we can see, it is almost the same as the previous one's first part. The only difference is that we add another plane in order to mark whether there are aka five tiles in hand, since aka five tiles usually will not be discarded. We convert this plane in a similar way as others, but here we fill all the four elements in the corresponding row with 1s instead of one 1 and three 0s as before. In this way, we manage to maintain the rule that in tile information planes, where each type of tile can have at most four on table.

Here we adopt a GRU model for this task. After going through all game records. the largest round length in one subgame is 22, so for invariant time-step training, the time-step is set to 22. For those which are shorter than 22 rounds, we adopt 0 paddings to fill, complement to the length of 22. Therefore, the input structure for one game becomes a four-dimension matrix of (22, 16, 34, 4). We make study on whole game records in year of 2015 of Houou Table from Tenhou. We got in all 193696 subgames. Each subgame is set to 22 time-steps, but about half of them are 0 paddings because of filling. We actually have 2135331 valid game rounds. The network structure is shown in Figure 1. We use TimeDistributed wrapper for time-steps and results of return sequences of GRU layer for further training into softmax of 34. We adopt filters of 5 by 2 in kernel and 128 in height for convolutions. The hidden state elements for GRU is set to 256.

During training, since we have about half 0-padding data, the accuracy on validation set is useless even it reaches over 80% easily. For test data picked in year of 2014, we picked 50000 valid data, and achieve an agreement rate accuracy of 68.14%. Although it's better than other researchers' result, it's worth than our previous model.

Layer (type)	Output Shape	Param #
conv1 (TimeDistributed)	(None, 22, 128, 30, 3)	20608
bn1 (TimeDistributed)	(None, 22, 128, 30, 3)	12
drop1 (Dropout)	(None, 22, 128, 30, 3)	0
conv2 (TimeDistributed)	(None, 22, 128, 26, 2)	163968
bn2 (TimeDistributed)	(None, 22, 128, 26, 2)	8
drop2 (Dropout)	(None, 22, 128, 26, 2)	0
conv3 (TimeDistributed)	(None, 22, 128, 22, 1)	163968
bn3 (TimeDistributed)	(None, 22, 128, 22, 1)	4
drop3 (Dropout)	(None, 22, 128, 22, 1)	0
flatten (TimeDistributed)	(None, 22, 2816)	0
gru (GRU)	(None, 22, 256)	2360064
drop_gru (Dropout)	(None, 22, 256)	0
softmax (Dense)	(None, 22, 34)	8738

Total params: 2,717,370
Trainable params: 2,717,358
Non-trainable params: 12

Figure 1: Sequential model network structure

4 Off-table information behavior

Table 3: Off-table information model data structure

Feature	# of planes
Own hand tiles	1
Discarded tiles	4
Stealing tiles	4
Dora indicators	1
Richi players	3
Rank	4
Honba	8
Aka five sign	1
Round wind	1
Own wind	1
Past 1 situation	13
Past 2 situation	9
Past 3 situation	9
Past 4 situation	9

In this task, we add off-table information of rank and honba into the model compared with our previous paper. For rank information, the player may be from the top player to the 4th one, so we use four black/white channels to represent. Similarly, in East-South games, the honba number can reach to at most eight (West 1 and South 4 are both last round and regarded as same: 8th honba), so we have eight channels. The network structure is shown in Figure 2. We have filters of kernel in 5 by 2 and height of 100 same as our previous paper. We make training on whole records of the year 2015, which in all have 2135331 rounds for the player. We achieve a validation accuracy of 70.67%. For test, on 50000 data picked from year of 2014, we get a test accuracy of 70.336%, which is higher than our previous model.

Actually, we also make another training with 86 planes in all. Compared with this model, we add another two past situation of past 5 and past 6, nine planes each. We achieve a test accuracy of 70.438%, which is a little bit higher, and this is our best result so far.

Layer (type)	Output Shape	Param #
main_input (InputLayer)	(None, 68, 34, 4)	0
conv1 (Conv2D)	(None, 100, 30, 3)	68100
bn1 (BatchNormalization)	(None, 100, 30, 3)	12
dr1 (Dropout)	(None, 100, 30, 3)	0
conv2 (Conv2D)	(None, 100, 26, 2)	100100
bn2 (BatchNormalization)	(None, 100, 26, 2)	8
dr2 (Dropout)	(None, 100, 26, 2)	0
conv3 (Conv2D)	(None, 100, 22, 1)	100100
bn3 (BatchNormalization)	(None, 100, 22, 1)	4
dr3 (Dropout)	(None, 100, 22, 1)	0
flatten (Flatten)	(None, 2200)	0
softmax (Dense)	(None, 34)	74834

Total params: 343,158
Trainable params: 343,146
Non-trainable params: 12

Figure 2: Off-table information model network structure

5 Conclusion

In this paper, we make several trials of improvement beyond our last model. We make a trial of using a GRU model, although it cannot reach the best accuracy ever, this way has a promising potential. We also make a trial of containing some other information in this task, and achieve a good result.

Acknowledgement

This work was supported by JST ERATO Grant Number JPMJER1501, Japan.

Reference

[1] Naoki, M. and Yoshimasa, T.: Building a computer Mahjong player based on Monte Carlo simulation and opponent models, IEEE Conference on Computational Intelligence and Games (CIG), pp.275-283(2015).

[2] 築地毅, 柴原一友: CNN 麻雀-麻雀向け CNN 構成の有効性, ゲームプログラミングワークショップ 2017 論文集, pp.163-170(2017).

[3] Gao S., Okuya F., et al.: Supervised Learning of Imperfect Information Data in the Game of Mahjong via Deep Convolutional Neural Networks, ゲームプログラミングワークショップ 2018 論文集, pp.43-50(2018).

[4] Gao S., Okuya F., et al.: Gao S, Okuya F, Mizukami N, et al. Improved Data Structure and Deep Convolutional Network Design for Haifu Data Learning in the Game of Mahjong, 情報処理学会第 80 回全国大会.