RoboCup 2019 - 2D Soccer Simulation League Team Descripiton Ri-one (Japan)

Kansei Ishikawa, Jyunichiro Kawano, Taketo Masumi,
Yu Yamaguchi, Takahiro Iwatsu, Tomonari Ujita,
Takahumi Ogawa, Arata Kato, Ryoto Hattori, Daiki Minobe, and Daiki Miyazaki

Ritsumeikan University, Japan
is0394fv@ed.ritsumei.ac.jp
http://rione.org/hp/

Abstract. In this paper, we will outline methods which we have utilized in preparation for RoboCup 2D Soccer Simulation 2018. In this year, we tried to improve through-pass which had been made in the past and optimised evaluation-value with Deep Reinforcement Learning to make Ri-one stronger.

1 Introduction

Ri-one is the project team which belongs to the Information Science and Engineering department at Ritsumeikan University. The organization has participated in the 2D Soccer Simulation League, Rescue Simulation League, and @Home League from before. We tried to advance Ri-one based on agent 2D base (release 3.1.1) had been made by H. Akiyama[1]. We resulted 10th place in Robocup 2018 in Montreal. In Robocup Japan Open, we won the championship in 2012 and 2015. This paper will include the following sections.

1. Introduction
2. Improvement of through-pass
3. Optimising evaluation-value using Deep Reinforcement Learning
4. Conclusion
5. References

In this year, we thought about a technique for improving attack power.

2 Improvement of through-pass

2.1 Introduction

We adopted past Ri-one ' s theme of TDP[2]. We determined our aim to calculate the target point of the pass route which had high success rate with deep learning[3]. We had placed emphasis on the receiver ' s position but now, we placed emphasis target point of pass. From the previous TDP[2], we reflected on what we had adopted the match data of other teams although we had needed to mount this match data in Ri-one and the amount of data had been small. The input layer is also reviewed.

2.2 Method

First, learning is performed using multilayer perceptron. The input layer at that time is the following nine channels. There are nine channels that explain at this paragraph.

1. x-coordinate of ball when it is kicked
2. y-coordinate of the ball when it is kicked
3. the speed of the ball in x direction when it is kicked
4. the speed of the ball in y direction when it is kicked
   Hereinafter, we describe with interpretation which use explanation of learning and learned model when doing regression.
5. x-coordinate of the ball when the through-pass success or failure / x-coordinate of the receive point
6. y-coordinate of the ball when the through-pass success or failure / y-coordinate of the receive point
7. The coordinates of the ball when the slow pass success or failure is determined and x-coordinate when a through-pass is kicked / the distance is the nearest opponent from target point
8. x-coordinate of player who did through-pass when success or failure is decided / receiver' s x-coordinate
9. y-coordinate of player who did through-pass when success or failure is decided / receiver' s y-coordinate

Additionally, hidden layer is as follows:
The first layer : We use Relu function from 9 nodes to 18 nodes.
The second layer : We use Relu function from 18 nodes to 18 nodes.
The third layer : We use sigmoid function from 18 nodes to 1 node output layer.
The output layer is the result of the through-pass / probability of sucess of the through-pass.
Next, the success probability of the through-pass is predicted with the model created by the preliminary learning. When the probability exceeds 70%, a CooperativeAction of the through-pass is generated.
Finally, we add the probability of success of the through-pass predicted before to the evaluation-value returned by the field evaluator of the chain action.

2.3 Result

We measured 30000 through-pass which we got when we made Ri-one play Alice, Helios, Namira, Razi, Cyrus, MT and Yushan. Table1 shows success rate.

Table 1. success rate

	before	after
frequency of success	4982	4857
frequency of failure	5018	5143
success probability	49.8%	48.6%

Although the accuracy exceeded 70% in test data at the time of learning, unfortunately there was no improvement in the success probability of through-pass after implementation.

2.4 Prospects

We tried to predict the success or failure of high precision of through-pass by increasing the number of parameters actually considered necessary for the success of the through-pass. However, it did not come out as we wanted. We thought that it was caused by failing to grasp the factor of the success of the through-pass. If we can successfully detect it by learning, effective results may be obtained.

3 Optimising evaluation-value using Deep Reinforcement Learning

We considered that it was impossible that people found the best evaluation-value about behavior controlled by the situation of the agents holding ball. Therefor, we made a hypothesis that agents could spontaneously learn the best evaluation-value by Deep-Q-learning[4]. However, we found problems. For example, we thought about Deep Reinforcement Learning with multilayer perceptron which inputs the position of all agent in the beginning, but not all agents are recognizing agents of everyone else. Therefore, the input layer could not be fixed. In order to improve these, we tried to fix the input by treating the soccer field like an image. We came up with a screenshot of the football field of Ri-one's TDP[5] last year to deduce the behavior of the ball holding agent. Next, although CNN[6] was used, the problem that it did not move had occurred because the calculation amount increased. Therefore, we focused on using CNN of Transfer-learning as a feature quantity extractor. We perform pre-leaning at CNN with the state of the field as a vector and the action taken actually as a label. We got A 35-dimensional feature quantity vector of the state of the field by using the CNN model which did pre-leaning. We conducted Deep Reinforcement Learning with a multilayer perceptron using a 38-dimensional vector which consist of the above-mentioned 35-dimensional vector and a number representing the type of action coordinates of the target point as input layer. Since the state of the field is fixed in one cycle, the convolution calculation can be performed once in each cycle, and the problem of the calculation amount is improved.

3.1 Method Pre-learning

We did Pre-learning to use the Convolution-neural-network as the Feature-amount-extractor. The structure of the Convolution-neural-network is shown below.
Input-layer has 4 channels below.

1. Location of the teammates
2. Location of the opponents
3. Our location
4. Location of the ball

Size of each channel was 34×55. We substituted 1 for the position of coordinates to corresponding to soccer field. In each layer, by using zero padding in range 1, we prevented size of each channels from getting smaller at the time of convolution.

First layer: Convolution layer which has 16 types of filter of size of 3×3, Relu function, maxpooling layer of size of 2

Second layer: Convolution layer which has 8 types of filter of size of 3×3, Relu function, maxpooling layer of size of 2

Third layer: Convolution layer which has 2 types of filter of size of 3×3, Relu function, maxpooling layer of size of 2

Softmax function from the total coupling layer outputs from 35 nodes to 6 nodes of the third layer. The output 6 nodes correspond to Hold, Dribble, Pass, Shoot, Clear, Move.

We played 50 games each with 9 teams that participated in Robocup 2018, Yushan, Razi, Alice, Helios, Namira, FCP_GPR, MT, Cyrus, we got data on actions corresponding to each situation. Data was learned by using CNN with the above structure. The outline of the Convolution-neural-network configuration is shown in Fig.1.

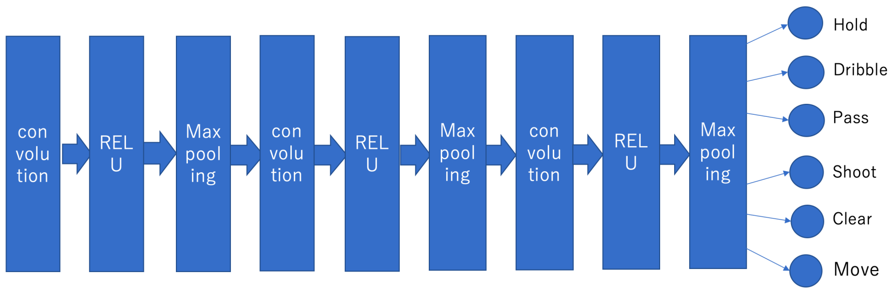

Fig. 1. process

3.2 Method during the game

We used 35 nodes of fully connected layer in CNN which is stated Fig.1 as feature value of situation of 2D soccer. As shown in Fig.2, we added the kind of action and predicted x-coordinate and y-coordinate of ball to that 35 nodes. Then, it became 38 nodes. We substituted them neural network is composed by 38 input layers, 1 output layer and 2 hidden layers. The value of output layers made evaluation value and it was evaluation value of generated action. We played each 9 teams which had participated in Robocup in 2018, Yushan, Razi, Alice, Helios, Namira, FCP_GPR, MT, Cyrus, with this NN. When our team scored a goal, we gave one point as reward to a series of the action. On the other hand, we gave minus one point to that when opponent scored a goal. In this way, it was learned. We made 45 steps with this game and learning series as 1 step.

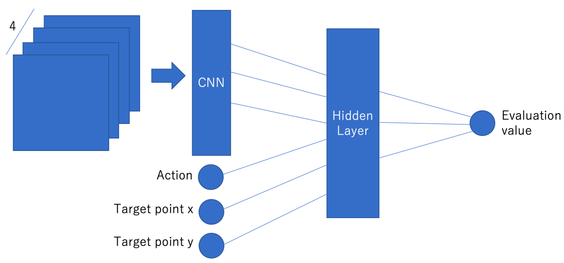

Fig. 2. CNN model

3.3 Result

By using this way, all we have to do came to call CNN only once per cycle and we could solve problem of calculation amount. The result of this experiment is shown in Fig.3. X-axis shows the sum of goal difference per frequency of learning. Y-axis show the number of learning.

According to the Fig. 3, we could not improve goal difference. However, goal difference in last half in the game became lower than first half by set, so it looked like stable. We gave rewards to a series of the action so we thought the effective action was hard to be recognized and it could not do effective learning. Furthermore, losing a point were larger than getting a points so we thought the action to prevent goal of opponent was increased.

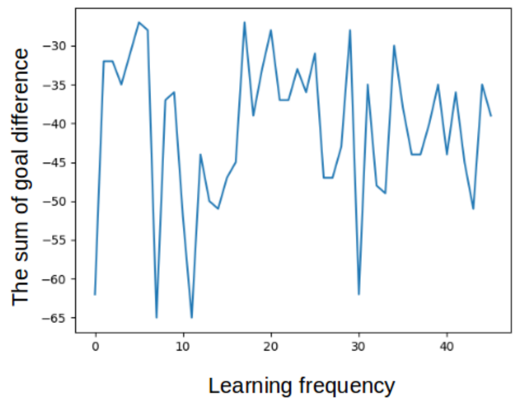

Fig. 3. a line graph of result

3.4 Prospects

This experiment had fallen short of my expectations. However, by using CNN as Feature-amount-extractor, we could utilize CNN in 2D Soccer simulation. Utilizing this experience, we will try to optimize evaluation value through experiment which is not in all field but in small area like penalty-area.

4 Conclusion

We found from the result that we examined each method. We aimed at strengthening the team by using machine learning but we could not strengthen the team. We think it is difficult to strengthen the 2D Soccer simulation team by machine learning only. However, we think that by devising ways to use either method, it will lead to strengthening the team by using it for local judgment.

References

1. RoboCup tools /soccer simulation wiki. http://rctools.sourceforge.jp
2. Asai, K., Katsumata, Y., Shibayama, T., Nomura, H., Kondo, R., Tanaka, H., Uchinishi, K., Mizumoto, M., Fujimitsu, T., Sei, M., Tani, Y., Kubo, S., Matsushita, Y. : RoboCup 2016 - 2D Soccer Simulation League Team Description (2016)
3. Saitoh, K.(2016). *To create the Deep Learning from scratch.* O'Reilly Japan.
4. Tomoshi, Otsuki.(2017). AlphaGo's demolition book. http://home.q00.itscom.net/otsuki/alphaZero.pdf#search=%27 アルファ碁+解説%27
5. Kono, S., Asai, H., Asahi, K., Inoue, S., Oe, H., Kawakami, Y., Kitamura, T., Kitamura, N., Kobayashi, K., Takeda, S., Naito, T., Hosomi, Y., Aoki, E., Ikuta, T., Ishikawa, K., Kawano, J., Masumi, T., Nomura, R., Yamaguchi, Y. : RoboCup 2017 - 2D Soccer Simulation League Team Description (2017)
6. Nakayama.H. Extraction of features of image by CNN and Transfer-learning. http://www.nlab.ci.i.u-tokyo.ac.jp/pdf/CNN_survey.pdf