

Team Description of opuCI_2D 2010

Takesuke Uenishi, Yosuke Narimoto, and Tomoharu Nakashima

Department of Computer Science and Intelligent Systems
Graduate School of Engineering, Osaka Prefecture University
Gakuen-cho 1-1, Naka-ku, Sakai, Osaka, 599-8531
`uenishi@ci.cs.osakafu-u.ac.jp`
`narimoto@ci.cs.osakafu-u.ac.jp`
`nakashi@cs.osakafu-u.ac.jp`

Abstract. This paper describes opuCI_2D, our soccer team that has been submitted to the qualification for the competition of the soccer 2D simulation league of RoboCup 2010. The main characteristic feature of this team is to use neural networks for pass prediction. Neural networks are trained so that a pass receiver is successfully predicted from a situation when our team is attacking. First we present the introduction of our team. Then the pass prediction task by neural networks is shown with experimental results.

1 Introduction

Team opuCI_2D started in 2009. The opuCI_2D project follows the previous OPU_hana_2D project in Osaka Prefecture University (OPU). CI in opuCI_2D means computational intelligence, which is the main scientific focus of the team.

So far we have applied various computational intelligence techniques [1] such as fuzzy logic [2], neural networks [3], evolutionary computation [4], and reinforcement learning. They were applied to ball intercept, dribble, and team strategy and team formation. This year we applied neural networks to a pass prediction task.

In this paper, first we explain the fundamental part of our team; that is, the basic information about the team including the source code on which our team is based on. We also show how our team is different from the original base team. Then we explain the scientific part of our team. We show the application of neural networks to the pass prediction task. A series of computational experiments are conducted. The results of the experiments show that the pass prediction is effective.

2 Basic Information about the Team

Our team is based on agent2d-2.1.0 [5]. agent2d-2.1.0 depends on the library librcsc-3.1.2. Although a good strategy is already implemented in agent2d, we have modified the source codes and our team behaves totally differently from

the original agent2d. Since the goal keeper’s move is only basic, we added a ball blocking to the player agent. This behavior appears when the goal keeper thinks that it can reach an opponent kicker within a reasonable number of time steps.

The original team formation is 4-3-3. We slightly modified the team formation to 5-2-3 to increase the number of defenders. Note that one defender out of five is rather offensive. Thus it sometimes goes up near the opponents goal area when the team is attacking deeply.

The behavior of the five defenders are designed to block through-passes, which is not implemented in the original agent2d.

The two mid-fielders and the three forwards act as a sub-group especially when attacking. We call the five agents the attacking group. When any one of the five player agent is ball-kickable, they find their position in the field so that the ball-kickable player agent can easily pass the ball to another player agents. The kickable player agent can also dribble when it decides that dribble is the best way to attack from the current situation. It should be also noted that the five player agents do not use built-in low-level skills such as kick, pass, dribble that are available in librcsc-3.1.2. Shoot behavior is only the exception where a built-in low-level skill is used in the five player agents.

3 Pass Prediction by Neural Networks

3.1 Pass Prediction Task

The task that we focus on for the qualification is a pass prediction task. The objective of the task is to correctly predict the pass receiver before a passer kicks the ball. This task is important because the receiver requires a quick action in order to successfully receive the ball kicked by the passer. Neural networks are used for this task. The back-propagation algorithm is used to train the neural networks. Experimental matches are made in order to generate training data sets for the learning of the neural networks. These procedures are done in an off-line manner; that is, they must be conducted before the qualification tournament.

In the following subsections, we explain neural networks, experimental settings, experimental results, and actual implementation for the qualification tournament.

3.2 Neural Networks

Three-layered neural networks are used for the pass prediction task. The activation function used to calculate the output of each unit is the sigmoid function (see Fig. 1).

While the output range of the neural networks is from 0 to 1 due to the nature of the sigmoid function, we modified the output values from the neural networks so that the minimum and the maximum value is 0.1 and 0.9 as follows:

$$o'_{NN} = \max\{0.1, \min(0.9, o_{NN})\}, \quad (1)$$

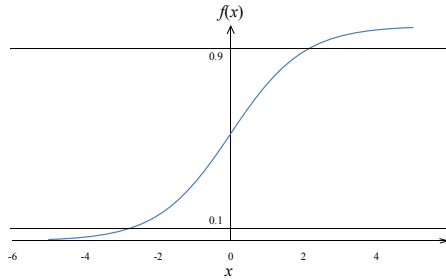


Fig. 1. Sigmoid function (Cutting lines at $f(x) = 0.1, 0.9$).

where o_{NN} is the output value from a neural network and o'_{NN} is the modified output. The reason for this modification is that we want the neural network to learn efficiently. It is obvious that the derivative of the activation function is not very sensitive around its extreme values (i.e., around the output values 0.0 and 1.0). This modification is simple but powerful for efficient learning for the valid output interval (i.e., from 0.1 to 0.9).

3.3 Experimental Settings

In this subsection, we evaluate the performance of the trained neural networks for the pass prediction task. We show several methods to generate training data sets and investigate which method produces the better neural networks.

As the configuration of the neural networks, we specified the number of input units as six, hidden units as 15, and output units as three. We represent the current situation of the soccer field by the estimated time steps that the three forwards reach the receiving points and the most probable opponent player that might intercept the pass to each forward. Thus the number of time steps that are used as the input to the neural networks is six (three for the three forward and the other three for the corresponding opponent players). We illustrate the input to the neural networks in the case of two receivers in Fig. 2. In this figure, the ball-kickable agent has two receiving points (i.e., the current position of the receivers). Two opponents that blocks the pass are also shown in this figure. We estimate the time steps in which the four player agents (two receivers and two opponents). This information is used as the input to neural networks.

The output units represent which forward is the pass receiver.

Through experimental matches, we generate training data for neural networks. The experimental matches are made in order to accumulate knowledge about pass prediction. That is, we compare the output of the neural networks with the kicker’s action when one of the five attackers is ball-kickable. Then if the output of neural networks does not coincide with the kicker’s action, we record the field information such as the estimated time step of the three forward to the receiving point and the most probable opponent that blocks the

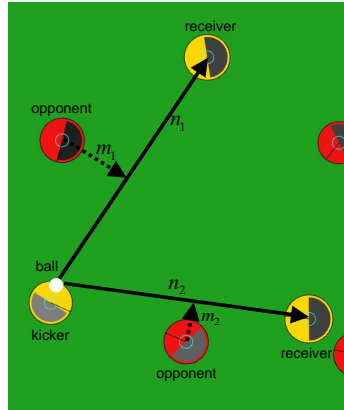


Fig. 2. Neural network inputs (in the case of two receivers).

receivers to generate a training pattern. We conducted ten experimental matches to accumulate training data.

Since the performance of neural networks are influenced by training data, we investigate four methods for generating training data. We call them NN1, NN2, NN3, and NN4. All the four methods generate training data through ten experimental matches. The main difference is how neural networks are used in the experimental matches and how they are trained.

[NN1]

In the first method of training data generation, During the first experimental match, we use an initialized neural networks where all connection weights are set to a random value between $[-0.5, 0.5]$. Then from the second match to the tenth match, we used neural networks that are trained from the generated training data in the previous experimental matches. As the number of experimental matches increase, we accumulate the generated training data, so we do not remove the previously generated patterns. For example, in the third experimental match, neural networks that are trained using the data set generated by the first and the second experimental matches.

[NN2]

In the second method, we initialized neural networks just once. That is, first we initialized neural networks with random connection weights. Then using the neural networks, we conducted ten experimental matches. Finally, the neural networks are trained using the training data sets that are generated through the ten runs of the experimental matches.

[NN3]

The third method is an incremental learning. After neural networks are initial-

ized, they are used to generate training data sets from the first experimental match. The neural networks are trained by the generated training data sets. The trained neural networks are used in the second experimental matches to generate the second training data set. The neural networks are not initialized but trained with the already learned connection weights. The training patterns that are generated so far are used to train neural networks at each iteration.

[NN4]

The fourth method uses exactly the same training data sets as NN3. The difference between NN3 and NN4 is that NN4 uses all training data sets through ten experimental matches according to the NN3 method. The connection weights of NN4 neural networks are initialized to random values between $[-0.5, 0.5]$. Thus the timing of initialization is different between NN3 and NN4 methods.

3.4 Computational Experiments

We first show the number of successful predictions and failure predictions for the four data generation methods in Table 1.

Table 1. The number of successful and failure predictions.

	Success	Failure	Total predictions	Rate (%)
NN1	843	429	1272	66.3%
NN2	772	426	1198	64.4%
NN3	851	308	1159	71.0%
NN4	888	488	1376	64.5%

From this table, we can see that the results by NN3 are the best among the four method in terms of the success rate.

Next, we examined the ball possession during the evaluation matches. The ball possession is defined as the number of time steps during which a team has the control of the ball. For example, the ball is in the possession of a team when the ball is kickable by any player agent of the team. The ball is also in the possession of a team even when the ball is not kickable if the kicker and the receiver is from the same team. Table 2 shows the ball possession for the four data generation methods as well as when no neural networks are used.

From this table, we can see that NN3 is again the best among the compared methods. From the experimental results shown in Tables 1 and 2, it is found that NN3 is the best generation method of training data for the pass prediction task. The reason would be that the training data are generated using the already trained neural networks. Also based on the generated training data, neural network are incrementally improved.

Table 2. Ball possession.

	Our team	Our attackers (A)	Opponent (B)	Rate (A/B)
NN1	13415	7598	3480	2.18
NN2	11829	6679	3828	1.74
NN3	13288	8278	3216	2.57
NN4	12564	7850	3714	2.11
None	11963	6817	3767	1.81

3.5 Implementation in the Submitted Team

The trained neural networks that showed the best performance (i.e., NN3) is implemented in the team submitted to the qualification. The neural networks are already trained before the qualification tournament and never changes during the tournament. The five player agents (two mid-fielders and three forward) have the same trained neural network. First, they make an attacking formation when one of them is ball-kickable. Then non-kickable player agents predict to whom the ball-kickable player agent passes the ball. The five players have the common knowledge about the pass receiving points. The predicted player agent dashes to the receiving point so that the pass becomes successful.

4 Conclusions

This paper presented the development of this year’s team opuCL2D. The scientific focus of the team is the use of neural networks for the pass prediction task. The standard three-layered neural networks are used for this purpose. We evaluated the influence of training data generation methods on the performance of the neural networks. We investigated four methods for generating training data. The best method was used to train the neural networks for the submitted team to the qualification tournament.

References

1. T. Nakashima and H. Ishibuchi, “Computational Intelligence in RoboCup Soccer Simulation”, *Computational Intelligence: Principles and Practice*, G. Y. Yen and D. B. Fogel (eds.), IEEE Computational Intelligence Society, pp. 217–236, 2006.
2. T. Nakashima, M. Udo, and H. Ishibuchi, “A fuzzy reinforcement learning for a ball interception problem,” *Proc. of RoboCup 2003 Symposium*, in CD-ROM (8 pages), Padova, Italy, July 10-11, 2003.
3. T. Nakashima, and H. Ishibuchi, “Mimicking Dribble Trajectories by Neural Networks for RoboCup Soccer Simulation,” *Proc. of 2007 IEEE Multi-conference on Systems and Control*, pp.658-663, Singapore, 2007.
4. T. Nakashima, M. Takatani, M. Udo, H. Ishibuchi, and M. Nii, “Performance evaluation of an evolutionary method for RoboCup soccer strategies,” *Proc. of RoboCup 2005 International Symposium*, in CR-ROM (8pages), Japan, 2005.
5. H. Akiyama, rctools front page, <http://rctools.sourceforge.jp/pukiwiki/>