

Team Description of OPU_hana_2D 2008

Tomoharu Nakashima, Yukio Shoji, Seiya Fujii, and Takesuke Uenishi

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

Abstract. This paper describes OPU_hana_2D, our soccer team that has been submitted to the qualification for the competition in the simulation league of RoboCup 2008. The basic skills such as dribble, pass, and shot are improved by modifying the source code of the base team. The main feature of OPU_hana_2D is the introduction of neural networks for dribbling. We call this *neuro-dribble*. In the neuro-dribble, a three-layered neural network is used to determine the dribble direction. A training data set for the learning of the neural network is obtained from the observation of the behaviour of other teams.

1 Introduction

Team ‘hana’ has been participating in the RoboCup world competitions since 2002. Our first trial in 2002 ended in losing in all matches without marking any scores. In the second trial in 2003, we survived the first elimination match before losing in the second elimination. We also could proceed to the second round in 2004. As a base team, we used YowAI team in 2002, and have used UvA Trilearn basic [1] since 2002. The team name was changed from ‘hana’ to ‘OPU_hana_2D’ in the 2005 competition in order to clearly show our affiliation (Osaka Prefecture University).

In this team description paper, we describe our main features of development. That is, the development of a neural network-based dribble skill is explained in this team description paper.

2 Neuro-Dribble

2.1 Neural Network

In this section we describe neural networks for mimicking the behavior of a target agent. We specifically focus on the dribble skill of the target agent. Thus the task

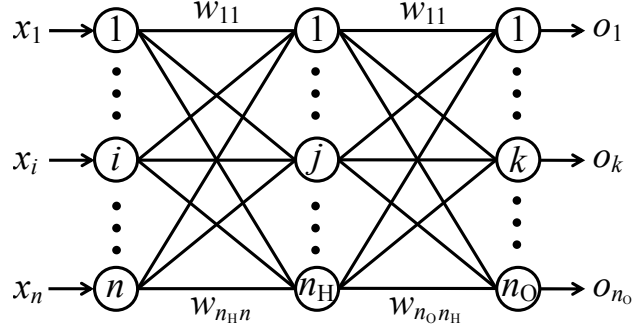


Fig. 1. Feed forward neural network.

of the neural networks is to learn the sensor-action mapping of the target agent. We use a standard three-layered feed-forward neural network as shown in Fig. 1.

In Fig. 1, there are n units in the input layer, n_H units in the hidden layer, and n_O units in the output layer. The symbols $w_{n_H n}$ and $w_{n_O n_H}$ show the connection weight between the n_H -th hidden unit and the n -th input unit and between the n_O -th output unit and the n_H -th hidden unit, respectively. It is also assumed that each unit in the hidden layer and the output layer has a threshold value (θ_j for the hidden layer and θ_k for the output layer). The input-output mapping of the neural network with an n -dimensional input vector in Fig. 1 is shown as follows:

Input layer:

$$o_i = x_i, \quad i = 1, 2, \dots, n, \quad (1)$$

Hidden layer:

$$o_j = f(\text{net}_j) = \frac{1}{1 + \exp(\text{net}_j)}, \quad j = 1, 2, \dots, n_H, \quad (2)$$

$$\text{net}_j = \sum_i^n w_{ji} o_i + \theta_j, \quad (3)$$

Output layer:

$$o_k = f(\text{net}_k) = \frac{1}{1 + \exp(\text{net}_k)}, \quad k = 1, 2, \dots, n_O, \quad (4)$$

$$\text{net}_k = \sum_j^{n_H} w_{kj} o_j + \theta_k, \quad (5)$$

Error back-propagation algorithm is used to learn the weights of the neural network. The next subsection describes the extraction of a set of training data for the learning of the neural network.

2.2 Generating Training Data

The task of the neural network is to learn the sensor-action mapping of a target agent. We use log files of soccer games that record the position and the velocity of all objects in the soccer field. Soccer games are performed several times using the target agent in order to obtain log files. Since the task in this paper is to mimic the dribble behavior of the target agent, we extract necessary parts from the whole log files so that the target agent is dribbling in the extracted parts of the log files. This process is manually conducted. From the extracted part, we generate a set of training patterns for the learning of the neural network.

2.3 Implementation

In this implementation, we use a single neural network (see Fig. 2). As we can see from this figure, the ball information is sent to the neural network. The neural network receives the ball position and outputs the direction to turn the body.

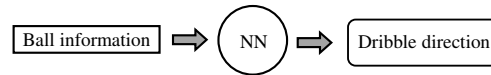


Fig. 2. Implementation.

It is assumed in this implementation that which action is taken according to the agent's situation. We show the procedure of implementation as follows:

[Procedure]

- Step 1: Stop the ball by applying a negative force to the ball.
- Step 2: Determine the dribble direction by the neural network. Send a turn command with the specified direction to the soccer server.
- Step 3: If the ball is in front of the ball, determine the dash power by a hand-coded algorithm and send a dash command with the specified power to the soccer server. Otherwise determine the kick power and the kick direction with the specified kick power and direction to the soccer server.
- Step 4: Stop the dribble procedure if a pre-specified termination condition is satisfied. Otherwise go to Step 3.

2.4 Experimental Settings

This section shows the computational experiments of the neuro-dribble. In the computational experiments the task of neural networks is to capture the dribbling behavior of a target agent by using neural networks. As a target agent UvA trilearn 2005 is used. UvA trilearn won the RoboCup world competition in

2003. The main characteristic feature of UvA trilearn is the implementation of their sophisticated basic skills. Sophisticated dribble skills are also implemented.

We specify the learning rate of the neural networks as 0.1. Although we can cancel the noise effect in the soccer simulation, we did not do so. This is because training data under practical conditions are useful in training neural networks. We collected log files of UvA trilearn playing soccer against various teams.

Figure 3 shows the extracted dribble trajectories by UvA trilearn in a game. These extracted dribble trajectories are used to generate training patterns for neural networks.

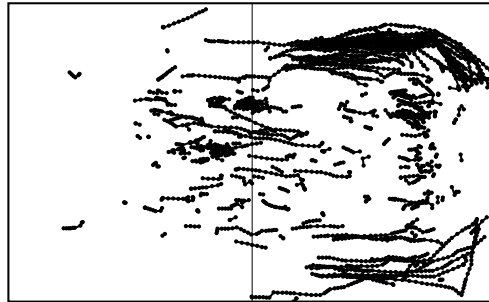


Fig. 3. Extracted dribble trajectories by UvA trilearn.

The mimicking agent starts the dribble behavior if the ball is kickable and the consequent action of the applied action rule is to dribble. We also specified five termination conditions for the dribble as follows:

- (1) There are any opponent agents that are very close to the mimicking agent,
- (2) The ball is taken by an opponent player,
- (3) The remaining stamina value of the mimicking agent is less than a pre-specified value,
- (4) The position of the ball is in pre-specified subareas (e.g., in the penalty area), and
- (5) The mimicking agent has been away from the ball for more than 15 time steps.

Figure 4 shows the dribble trajectory performed by our neural network. From the comparison between Fig. 3 and Fig. 4, we can see that the neuro-dribble successfully learns the dribble trajectories of UvA trilearn's dribble trajectories.

3 Conclusions

This paper described the development of this year's team OPU_hana_2D . The main feature of the team is the dribble skill of soccer agents. The standard type

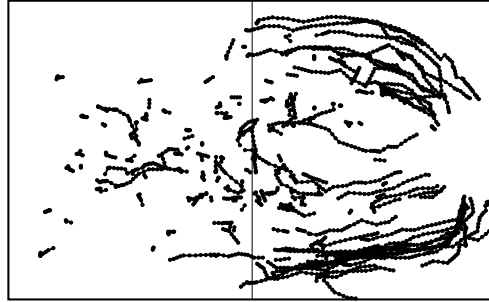


Fig. 4. Dribble trajectories performed by neural networks.

of neural networks are used for this purpose. Training patterns for the learning of the neural networks are automatically generated from the extracted dribble trajectory of a target team.

Future developments include the sophistication of the neuro-dribble skills. Although only one neural network is used in the current development of the neuro-dribble skills, multiple neural networks can be used. On-line learning of the dribble trajectory of the opponent team is also an interesting topic as our future development.

References

1. UvA Trilearn, URL at http://staff.science.uva.nl/~jellekok/robocup/index_en.html