CYRUS 2D Simulation Team Description Paper 2017

Nader Zare[1], Ali Najimi[2], Mahtab Sarvmaili[1],
Aryan Akbarpour[3], Mohsen NaghipourFar[2],
Borna Barahimi[4], Amin Nikanjam[1]

[1]Dept. of Computer Engineering, Khajeh Nasir University of Technology
nader.zare88@gmail.com,
mahtab.sarvmaili@gmail.com,
nikanjam@kntu.ac.ir
[2]Dept. of Computer Engineering, Sharif University of Technology
najimi@ce.sharif.edu,
naghipourfar@ce.sharif.edu
[3]Bellerbys College Brighton
aryan.ak99@yahoo.com
[4]Atomic Energy High-school
bornab1379@gmail.com

Abstract. This paper includes some explanations about algorithms implemented by CYRUS team members. The main objective is to express a brief explanation about intelligent decision making of the agents. For this purpose, first we applied deep neural network for opponents position detection, and also de-noising and agents restricted view removal, and second, another deep neural network were learned to find the opponent team formation. These two neural networks enable our agents to understand the opponents team formation and also improve their performance by de-noising and removing view limitations. The base code used by CYRUS is agent 3.1.0[1].

1 Introduction

"CYRUS" robotic team members created this team six years ago, with the goal of student scientific improvement in *Artificial Intelligence* and multi-agent field. At first, members of this team were bachelors of *Information Technology Dept.* at *Shiraz University of Technology* and now team members are formed of students from *Khajeh Nasir University of Technology*, *Sharif University of Technology*, and *Atomic Energy High-school*.
This team has been qualified to participate in World Competition since 2013 and ranked 8^{th}, 5^{th}, 9^{th}, 12^{th} (CYRUS) and 8^{th} (FURY) sequentially, Also in 2014 achieved First-place of *Iran Open* competition, first-place in *Kordestan* 2013 and first-place in *Fazasazan* 2012. This team has also participated in *Iran Open* 2013, *Iran Open* 2012, *SharifCup* 2012, *Iran Open* 2011, *Sama RoboCup* 2011, etc.
Presently *CYRUS* team members are trying to improve their efficiency by using *Machine Learning*, and *multi-agent* algorithms.

This paper briefly represents *CYRUS* team structure and algorithms, e.g. opponent behavior detection, and opponent team formation detection using deep learning.

2 Opponent Behavior Detection

Some of the important aspects of 2D Soccer Simulation are noise, agents restricted view, and opponents behavior detection in the context of uncertainty. *CYRUS* team uses deep learning from the Robo Cup logs from 2010 to 2016 in order to detect these behaviors.
In this section our purpose is to determine opponents next position by current positions of our team and opponent. At the moment, learning process is only available offline but in the future, online learning process will be added to coach.

2.1 Data-Set

Deep neural networks need large amounts of data for training, to meet this requirement over 15 million data have been gathered from Robo Cup logs since 2010 to train deep neural network.
Each record consists of 144 features which 94 features of them have been considered as inputs to the neural network. Useful information of both teams and ball position of every single cycle (assume as cycle i) are given to the neural network as inputs, and information of the next cycle (assume as cycle $i+1$) are stated as labels.

Inputs		
Ball	Left Agents	Right Agents
	left team id	right team id
ball.pos.x	leftAgent.pos.x	rightAgent.pos.x
ball.pos.y	leftAgent.pos.y	rightAgent.pos.y
ball.vel.x	leftAgent.vel.x	rightAgent.vel.x
ball.vel.y	leftAgent.vel.y	rightAgent.vel.y

Table 1: List of inputs

Labels		
Right Agents	Left Agents	Ball
rightAgent.pos.x	leftAgent.pos.x	ball.pos.x
rightAgent.pos.y	leftAgent.pos.y	ball.pos.y
rightAgent.vel.x	leftAgent.vel.x	ball.vel.x
rightAgent.vel.y	leftAgent.vel.y	ball.vel.y

Table 2: List of labels

2.2 Using Stacked (De-noising) Auto-Encoders[6] to Find The Best Initial Weights

The dominant method for training deep neural networks is gradient descent. Training deep neural network from random initialized parameters, does not work very well, and primary layers don't learn properly. Few algorithms work well for this purpose. Learning layer by layer using Auto-Encoder has been implemented, which weights each layer learns separately to have better initialization for each individual layer so that it may learn useful feature extractors. After building the stack of encoders, the supervised learning can be used to learn for training the deep neural network layers. Auto-encoder sets output label of each layer equal to input of it in order to make distinction between different data and to be noise resistant.

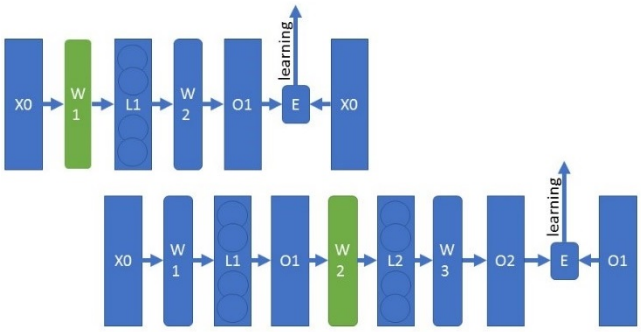


Fig. 1: Comparison between Normal Neural Network and Auto-Encoder

Each layer includes respectively 200, 150, 100, 80, and 48 neurons. Methods of achieving initial weights, will be defined.

2.3 Finding The Best Structure to Learn Neural Network

To achieve the efficient initial weights, *Multi-Layer Perceptron* (MLP) and Rough neural networks have been implemented.

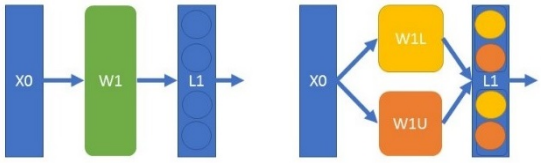


Fig. 2: MLP vs. Rough

Rough neural networks cleave weights into lower bound and upper bound, then from each part of weights output will be averaged.

$$
\begin{aligned}
&MLP: \\
&\quad Net = W_X + b \qquad \longrightarrow \qquad O = f(Net)
\end{aligned}
$$

$$
\begin{aligned}
&Rough: \\
&\quad Net_U = W_U * X + b_U \qquad\qquad Net_L = W_L * X + b_L \\
&\qquad\quad \downarrow \qquad\qquad\qquad\qquad\quad \downarrow \\
&\quad O_U = max(f(Net_U), f(Net_L)) \qquad O_L = min(f(Net_U), f(Net_L))
\end{aligned}
$$

$$W_2 = transpose(W_1) \tag{1}$$

Learn W_1 and gain W_2 from W_1

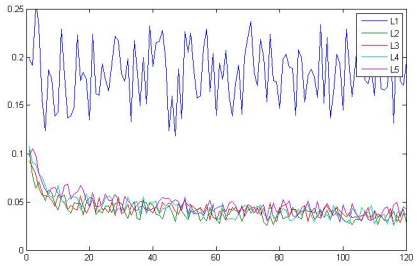
$$W_1 \parallel W_2 \tag{2}$$

Learn W_1 and W_2 simultaneously

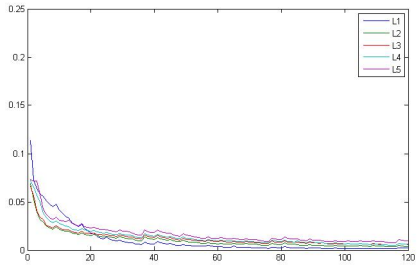
$$W_1 = transpose(W_2) \tag{3}$$

Learn W_2 and gain W_1 from W_2

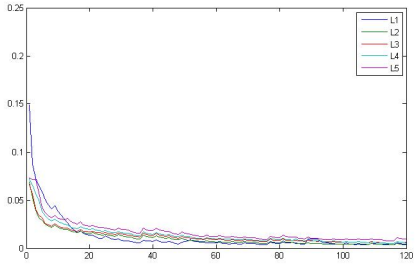
Equations 1 and 2 and 3 are methods to achieve efficient initial weights according to the layer type (MLP or Rough) diagrams are shown in Fig. 3.



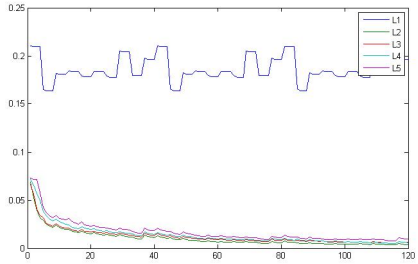
(a) Learn W_1 MLP & Auto-Encoder



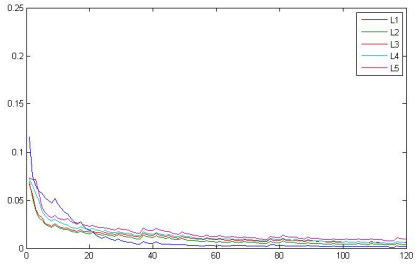
(b) Learn W_1 & W_2 MLP & Auto-Encoder



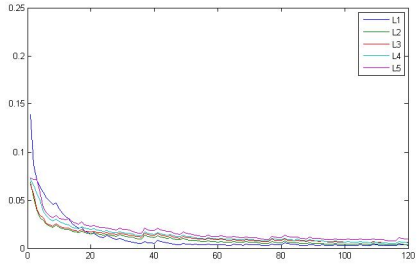
(c) Learn W_2 MLP & Auto-Encoder



(d) Learn W_1 Rough & Auto-Encoder



(e) Learn W_1 & W_2 Rough & Auto-Encoder

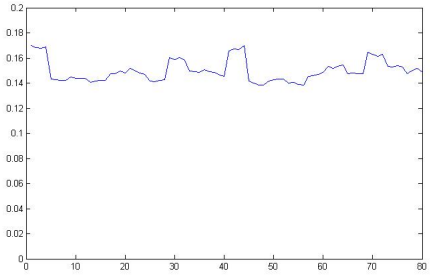


(f) Learn W_2 Rough & Auto-Encoder

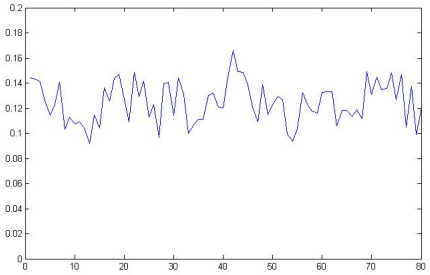
Fig. 3: 6 methods of auto-encoder to gain initial weights using MLP & Rough

2.4 Learn Deep Neural Network Using Mined Initial Weights

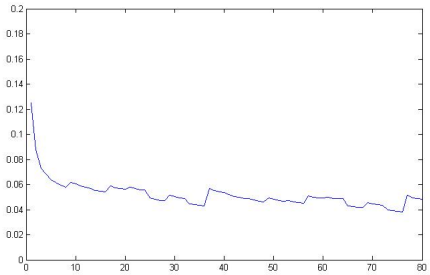
To detect opponents next cycle behavior, a 7-layers deep neural network with 200, 150, 100, 80, 48, 48, 48 neurons sequentially were used in this algorithm that initial weights can be assigned randomly or by using mined weights from above methods, are shown separately in Fig. 4, and *CYRUS* team uses the best result.



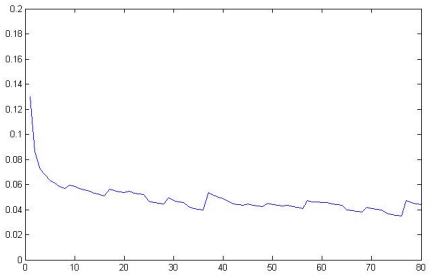
(a) Deep MLP with Random Weights



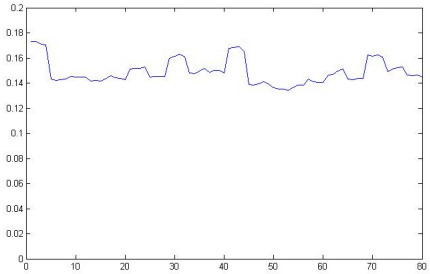
(b) Deep MLP with W_1



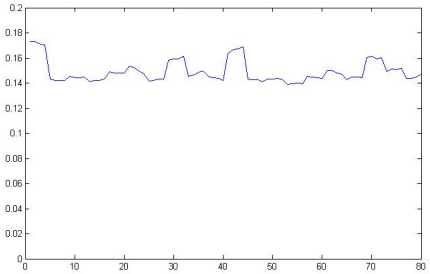
(c) Deep MLP with W_2



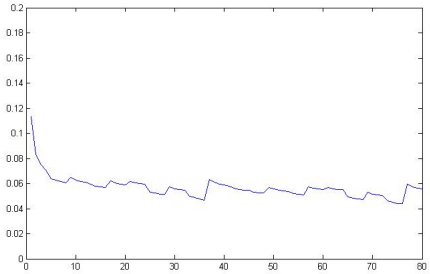
(d) Deep MLP with W_1 and W_2



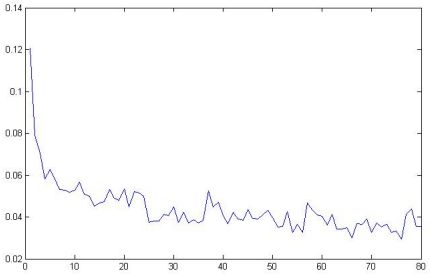
(e) Deep Rough with Random Weights



(f) Deep Rough with W_1



(g) Deep Rough with W_2



(h) Deep Rough with W_1 and W_2

Fig. 4: Error charts based on total epochs

2.5 Using Deep Learning to Detect Opponents Next Cycle Behavior

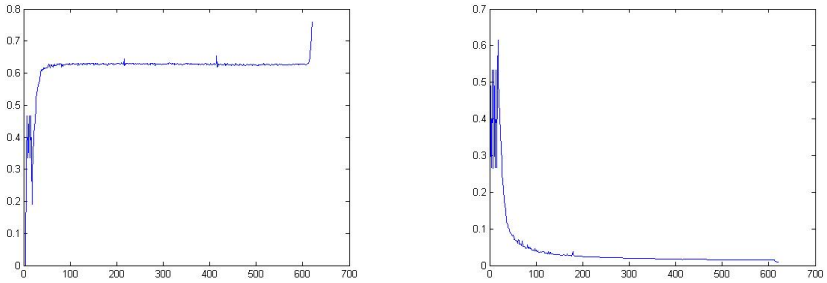
Currently, *CYRUS* team can detect opponents next cycle behavior using defined deep neural network, and in the future, it will detect next cycles by adding self positions. For instance, Cycle i information will be given to the neural network and cycle *i+1* will be our result, then cycle *i+1* information will be given to the neural network and cycle *i+2* will be our result. This operation will be continued until the desired information is achieved. Table 3 shows the result of behavior detection.

Team	Without NN		With NN	
	Goals Scored	Goals Lost	Goals Scored	Goals Lost
Agent2D	3.18	1.1	5.1	0.8
Random Team	2.12	1.91	2.44	1.79

Table 3: Average scores with and without neural network

3 Formation Detection

Formation is the most distinctive feature of every 2D Soccer Simulation team and it consists of different lineups e.g. defense, midfield, and offense. *CYRUS* team will form its best formation against different teams and formations using opponents formation detection and *Artificial Intelligence* methods, that will be defined, before Robo Cup.
In order to determine opponents formation, first a great data-set of 80 different formations were created using fedit[2]. Each and every data from data-set includes: ball position, positions of all the agents, angle and direction between ball and every agent as inputs of the neural network and number of agents in each line as labels of inputs. The learning process uses a 6-layer neural network which contains 25, 80, 50, 30, 15, and 5 neurons sequentially. The initial weights are calculated from auto-encoder method. Fig. 5 shows total error and accuracy according to the epochs in the learning process.



(a) Accuracy According to Epochs (b) Error According to Epochs

Fig. 5: Total Error and Accuracy According to epochs

A teams formation may be different in viewpoint of person to person, it is not possible to compare with reality, Therefor coach gathers 500 cycles of information then gives it to neural network in two different ways:
(1) each cycles information will be given to neural network and final result will be averaged.
(2) all information will be averaged then will be given to neural network.

	1	2
Accuracy	64%	73%

Table 4: Accuracy of each method for agent2d formations

4 Future

CYRUS team members plan to enhance offense and defense system using deep neural networks, and deep reinforcement learning. In order to achieve this goal, *CYRUS* team will reprogram its defense real-time decision making using *actor-arctic* deep neural network and multi-agent reinforcement learning, goalie decision making in penalty kicks using actor-arcitic multi-agent deep learning and deep reinforcement learning, online opponents high level behavior detection using *PGM* and deep learning with coach.

References

1. *"agent2D-3.1.0 RoboCup tools - OSDN."* [Online]. Available: agent2D-3.1.0 . [Accessed: 22-Jan-2016].
2. *"fedit2-0.0.0 RoboCup tools - OSDN."* [Online]. Available: fedit2-0.0.0

3. Teshneh Lab, Mohammad, and Pouria Jafari. *Neural Networks and Advanced Neuro-Controlers, A Rough Neural Network Approach.* Khajeh Nasir University of Technology, 2015.
4. Haykin S., *"Neural Networks: A Comprehensive Foundation (2 ed.)"*, Prentice Hall, 1998.
5. H. Akiyama, T. Nakashima, and S. Mifune, *"HELIOS2015 Team Description Paper,"* pp. 16, 2015.
6. Vincent, Pascal, et al. *"Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion."* Journal of Machine Learning Research 11.Dec (2010): 3371-3408.
7. Slotine, Jean-Jacques E., and Weiping Li. *Applied nonlinear control.* Vol. 60. Englewood Cliffs, NJ: Prentice-Hall, 1991.
8. Akiyama, H., Noda, I.: *Multi-agent positioning mechanism in the dynamic environment* . In: Visser, U., Ribeiro,F., Ohashi, T., Dellaert, F., eds.: RoboCup 2007: Robot Soccer World Cup XI, Lecture Notes in Artificial Intelligence. Volume 5001., Springer (2008) 377-C384.
9. N.Zare, A.Keshavarzi, E.Beheshtian, H.Mowla, H.Jafari, *"CYRUS 2D Simulation Team Description Paper 2016"*, The 20th annual RoboCup International Symposium, Germany, Leipzig, 2016.
10. N.Zare, M.Karimi, A.Keshavarzi, E.Asali, H.Alipour, A.Aminian, E.Beheshtian, H.Mowla, H.Jafari, M.J.Khademian, *"CYRUS 2D Simulation Team Description Paper 2015"*, The 19th annual RoboCup International Symposium, China, Hefei, 2015.
11. R.Khayami, N.Zare, M.karimi, P.Mahor, F.Tekara, E.Asali, A.Keshavarzi, A.Afshar, M.Asadi, M.Najafi, *"CYRUS 2D Simulation Team Description Paper 2014"*, The 18th annual RoboCup International Symposium, Brazil, Joao Pessoa, 2014.
12. M. Dezfoulian, N. Kaviani, A. Nikanjam, M. Rafaee, *Training a Simulated Soccer Agent how to Shoot Using Artificial Neural Network*, 13th Multi-disciplinary Iranian Researchers Conference in Europe (IRCE), Leeds, UK, 2005.
13. M. Prokopenko, P. Wang, and O. Obst, *"Gliders2015: Opponent avoidance with bio-inspired flocking behaviour,"* pp. 15, 2015.
14. H. Zhang, M. Jiang, H. Dai, A. Bai, and X. Chen, *"WrightEagle 2D Soccer Simulation Team Description 2015,"* Wrighteagle.Org, pp. 38, 2015.
15. S. Marian, D. Luca, B. Sarac, and O. Cotarlea, *"OXSY 2015 Team Description,"* 2015.