ISSN 1 746-7233, England, UK
World Journal of Modelling and Simulation
Vol. 6 (2010) No. 1, pp. 63-71

Enhancing agent's learning and decision making in minority game with neural networks*

Taghi Rezvan†

Department of Industrial & Systems Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

(*Received February 14 2009, Accepted August 18 2009*)

Abstract. In original minority game, each agent at every time step chooses the highest-score strategy from its limited number of strategies given arbitrarily in the beginning of the game. In this paper, agents of the minority game are equipped to a multi-layer perceptron neural network that at every time step learns by back propagation error. Simulation results and the comparison of them with the results from original minority game model indicate that the proposed neural network-based learning technique of agents improves the overall performance of the agents.

Keywords: minority game, neural network, learning

1 Introduction

Study of complex adaptive systems has attracted considerable interests across many disciplines[11]. As examples of these systems, we can mention the Internet and World Wide Web, economic markets, supply chain, and human social networks. Agent-based models are used efficiently to analysis such systems[13, 22]. These models are able to predict global behavior of systems comprising from different participant with different goals. In effect each agent in such systems decides based on some global information that is formed by the collective actions of the agents themselves without centralized coordination[13, 22]. The minority game (MG) is a suitable tool for studying many complex adaptive systems like financial markets, and traffic equilibrium problems. This game has introduced by Challet and Zhang[5] and is a simplified and formulized version of the El Farol bar problem, first described by Brian Arthur[2]. In the El Farol bar problem, 100 people decide independently each week in certain night whether to go the bar or stay at home. Since space in the bar is limited, a person decides to go if he/she expects fewer than 60 to show up, or stays home if he/she expects more than 60 to go. Therefore, it is not known that how many will go each week. People have to rely on inductive reasoning to decide on their actions; otherwise they lose. Arthur[2] shows when people are shown by intelligent agents the mean attendance converges always stay in nearly 60 in each week.

The setup of the original minority game is as follows: N agents have to choose at each time step whether to go into room -1 or 1. The agents repeatedly compete to be in a minority group. Therefore the agents who have chosen the less crowded room will win, and the others will lose. They have similar and limited capabilities, but are heterogeneous in the sense that they use different strategies in making decisions by inductive reasoning. A strategy is a mapping from the last M outcomes of the game to an action (1 or -1). At the beginning of the game, each agent picks randomly S strategies, and after each round, each agent assigns one virtual point to each of its strategies which would have predicted the correct outcome. At every time step of the game, agents use strategy that gained the most virtual points. Thus, the agents interact to one another

* The auther is extremely grateful to Prof. Mohammad Reza Zamani and also the referees of World Journal of Modelling and Simulation because of their very helpful comments and suggestions.
† Corresponding author. Tel.: +98-311-39125501, fax: +98-311-3915526. *E-mail address*: taghi_rezvan@in.iut.ac.ir.

through a highest-score strategy selection process with considering previous histories available about the collective decisions made. Major of studies implemented on MG have been presented from theoretical physicists working in statistical mechanics. Among them, the following are worth mentioning.

Johnson et al.[14] introduced the Evolutionary MG and the mixed MG model with variable strategy spaces presented by Li et al.[19] Chow[7] extended MG model with multiple choices. Moelbert and De Los Ria[21] presented the local MG model allowing exchange of local information. Yang and Sun[26] proposed a new model of Incomplete Minority Game, which features incomplete strategies with random bits and players using default hierarchies. Foldy et al.[9] defined and studied a hierarchical extension of the minority game. Chmura and Pitz[6] analyzed the minority game as an elementary traffic scenario. Kim et al.[15] applied the minority game for the patient problem consulted at the department of pediatric cardiology. The MG can been considered as generic model of competing adaptive agents in the economy. Nevertheless, a majority of papers on the MG are motivated by the study of financial markets[3, 4, 20]. Groot and Musters[10] applied MG to study the timing of promotional action at retailers in the fast moving consumer goods on the Dutch market. Kinzel et al.[16] studied a system of interacting perceptrons for the problem of minority game. These perceptrons haven't any hidden layer which is trained on the history of minority decisions. They analyzed the statistical properties of such a system of interacting perceptrons. Also, Abdullah[1] investigated the emergence of heterogeneity in minority-subsequently-majority game where agents interacting to learn through individual neural networks that don't have any hidden layer.

The main topic of this paper is to introduce a dynamic strategy in minority game with an implementation of reinforcement learning to a multi-layer percepton neural network with using "back propagating reinforcement learning". Since neural networks can approximate the observed bounded rationality that agents appear to demonstrate, they can be used to model the behavior of agents of minority game. Biological neurons in neural networks are connected to each other through junctions similar to synapses. These junctions are assigned connection weights and they can have either excitatory or inhibitory effects depending on the sign of the assigned weight. When a neuron receives a critical amount of excitation, it becomes active and affects the elements with which it is connected[17, 18].

Learning in neural networks is frequently achieved by supervised learning, from examples provided by a knowledgeable external supervisor[24]. In other words, reinforcement learning, as a supervised learning process, appears to be a suitable strategy to mimic agent's behavior in minority game. Reinforcement learning is tantalizing because learning occurs through trial and error experimentation within the environment. Feedback is a scalar payoff, hence no explicit supervisor is required, and little or no prior knowledge is needed[25]. Reinforcement learning is an approach that emphasizes learning by the interaction between an active agent

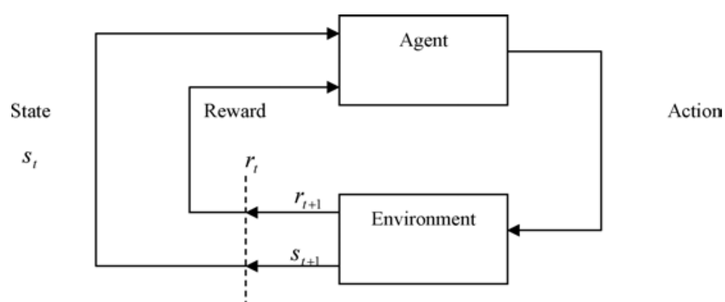

Fig. 1. Cycle of reinforcements learning[24]

and its environment (Fig. 1) Based on this approach, reinforcement learning can be defined as an on-line learning ('learning what to do by doing') framework to find an optimal decision policy, i.e. how to map states or histories to control actions, so as to maximize reward of each agent[24].

Besides, reinforcement learning is the essential adapting tool for agents to take in a specific situation based on feedback obtained from a complex and variable environment to allow the possibility of prompt adjustment. In fact, trial-and-error search and delayed reward are the two most important distinguishing characteristics of reinforcement learning. The secret to the well-known back propagation learning algorithm is that pre-synaptic neurons in the multi-layer neural network change their synaptic weights according to the weighted sum of error signals of the post-synaptic neurons[25]. For instance, if majority of post neurons had positive error, and the synaptic connection from a particular pre-neuron is mostly positive, this particular neuron will change its synaptic weight from its pre-synaptic neurons in order to raise its activity, that is, to say, enhancing excitatory synapses and decreasing inhibitory synapses. Because of this global error gradient descending mechanism, the back propagation method is highly efficient in training multi-layer neural networks[25].

In the standard form of minority game, the learning of agents takes place through the process of virtual rating. This paper proposes a new version of Minority Game, with each of its agents being featured by a neural network instead of strategy tables. Learning of agents is implemented by back propagation error mechanism. Hereinafter, the original minority game is referred to as 'OMG' and the term 'NNMG' is used for the version proposed in this paper. Numerical simulation suggests that the NNMG model can improve the overall performance of the system.

Section 2 of this paper introduces the ideas of this version of MG and the different aspects of this new model. Also, we describe the characteristics of the presented model, including the type of learning and dynamic change of strategies. Section 3 presents the simulation results, and compare them with the results from the OMG model. Section 4 draws conclusions and sketches future study.

2 Basic model

In NNMG, agents are equipped to multi-layer perceptron neural networks and are deprived from strategy tables. Thus each agent selects an action of -1 or 1 by its neural network. This dynamic strategy processes input information (m) based on initial random weights of neural network. Output of neural network is either 1 or -1. In effect if the output is larger than 0, then the agent chooses action 1 and otherwise action -1. The collective decision making of agents determines the wining group. In reality, the determination of this wining group is a response of environment to the executed action by each agent. This response of the environment is communicated to the agent through a scalar reinforcement signal, which indicates whether the action chosen by the agent in the current state is suitable. In other words, each agent computes quantity of its error that is the outcome of the difference between its actual decision and the winning group's decision. The computed error is propagated from the last layer to the first layer. This work modifies weights of the network so that the agents can deicide based on the feedback received. Operations are performed based on back propagation error mechanism. In the process of the game, the input of network is updated. Fig. 2 is based on the strategy mentioned.

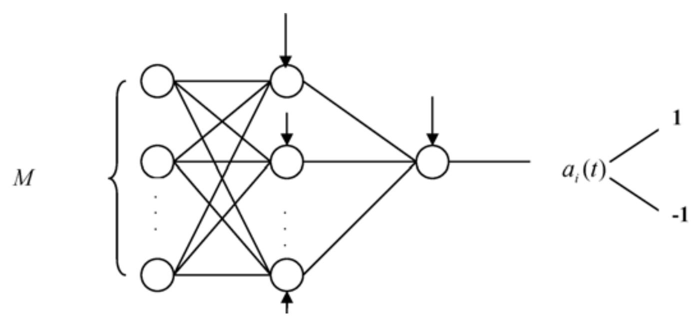

Fig. 2. Sample of neural network that each agent uses for decision making in minority game

With refer to Universal Function Approximation theorem introduced and proofed by Nilson[23], Cybenco[8] and Hornick[12], and equip agents to a 2-layer perceptron neural network to find the best decision they can make. This powerful theorem states that if a mapping exists, then we can find a perceptron neural network with a middle layer which approximates such mapping[8, 23, 25].

We assume that time t is discrete, and consider that each agent $i = 1, \cdots, N$, at each time step t, can do the binary action $a_i(t) \in \{-1, 1\}$ and also, accesses alike to a record of M past periods. This record of information shows which groups have been winner group in the past periods. Since agents have bounded rationality, they are able to use only finite information. The agents are heterogeneous, thus each agent is endowed with the 2-layer perceptron neural network as strategy, which have random weights. Therefore, input vector of each agent's neural network $(X(t))$ is alike and have M dimension with quantities 1 and -1 and its output $(\bar{a}_i(t))$ is scalar quantity that due to being random of weights in networks, for each agent i different from other. $\boldsymbol{W}_i^l(t)$ (The weights between two nodes) and $b_i(t)$ (the arc enter to a node that is called bias) are weights in neural network of agent i at round time t that l is number of layer. a_i is learning rate of agent i and $y_i^{\text{Hidden}}(t)$ is output vector of nodes of hidden layer.

After introducing notations, to illustrate our method further, we present a mathematical algorithm in which each agent processes information $X(t)$ by his/her neural network in forward operation according to:

$$\begin{aligned} y_i^{\text{Hidden}}(t) &= f^1(X(t)\boldsymbol{W}_i^1(t) + b_i^1(t)); \\ \bar{a}_i(t) &= f^2(X(t)\boldsymbol{W}_i^2(t) + b_i^2(t)). \end{aligned} \tag{1}$$

In Eq. (1), $f^i(\cdots), i = 1, 2$ is transfer function which we use tanh $(\cdots)$.

Decision of each agent is $a(t)$ at each time step t which based on below equation:

$$a_i(t) = \begin{cases} 1 & \bar{a}_i(t) \geq 0 \\ -1 & \bar{a}_i(t) < 0 \end{cases} \tag{2}$$

The decisions of all agents are aggregated into the global quantity: $A(t) = \sum_i a_i(t)$. Thus wining group is obtained as: $-\text{sgn}[A(t)]$. All of the agents who belong to the minority gain one point. Based on the recent wining group, information $X(t)$ is updated for the next time step. This updating can be stated on two method; first, it is accomplished to shift information so that the latest past history exists from the record and the recent wining group enters. Another, If we make the correspondence $-1 \to 0$ and $+1 \to 0$. Thus, if the last winning groups were $+1; -1; +1$ the binary representation of $\mu(t)$ is 101 and $\mu(t) = 6$. In other word, $\mu(t) \in \{1, 2, \cdots, P\}$, so $M = \log_2 P$; updating accomplish according to:

$$\mu(t+1) = [2\mu(t) + (1 - \text{sgn}[A(t)])/2] \mod 2^M. \tag{3}$$

Based on output of network and also the wining group, error quantity of an agent is computed as: $e_i(t) = \bar{a}_i(t) - [-\text{sgn}[A(t)]]$. Now reinforcement learning is implemented to apply back propagation error mechanism. This work is accomplished with computing and propagating local gradient according to:

$$\delta_i^2(t) = -2a_i(t)e_i(t); \quad \delta_i^1(t) = f^1(\cdots)\boldsymbol{W}_i^2\delta_i^2(t). \tag{4}$$

Finally weights matrix and bias vector of agent's network is set as:

$$\boldsymbol{W}_i^l(t+1) = \boldsymbol{W}_i^l(t) - a_i\delta_i^l(t)y; b_i^l(t+1) = b_i^l(t) - a_i\delta_i^l(t) \quad (l = 1, 2; i = 1, 2, \cdots, N). \tag{5}$$

In Eq. (5), when $l = 2$, y shows $\bar{a}_i(t)$ and when $l = 1$, y is $y_i^{\text{Hidden}}(t)$. To more understand this algorithm, reader can refer to available pseudo-code in appendix.

3 Numerical simulations

Without using any special package and just by using Visual Basic programming language, we implemented the entire algorithm and incorporated neural networks in a simulation environment. Over one thousand lines of code, including comments, express the scope of programming task performed. To understand the investigation of the agents' behavior, we studied long-time behavior (after 1000 time steps) of the agents. These studies include different states of game by changing their parameters. In addition to number of periods which their information history (M) are available, the agents' learning rate, and the number of hidden layer node of agents' neural network are suitable for study and analysis. In principle, the agent's behavior depends on these parameters. We investigated the agent's attendance fluctuations of game and actual point obtained by them.

3.1 Experiment 1: changing M

Having the parameter M, we will discuss agents' coordination and actual points obtained by agents in the following opposite states: learning rate of each agent is random versus it is certain and alike for all agents.

First we assume that the learning rate is random and simulate 1001 agents who use 2-layer perceptron neural network with 3 nodes in hidden layer and random learning rate by $M = 3, 5, 7$. When the parameter M is considered large, the agents are more likely to make a better choice and increase the coordination of population. If we plot global quantity of population for these time series in Fig. 3, it can be seen that fluctuations will indeed decrease when intelligent agents are increased. Note that fluctuations also decrease when agents' memory (M) is increased in OMG and this decreasing is significantly further according to paper Challet and Zhang[4] compared with NNMG. The variation of actual points of agents can be seen as a function

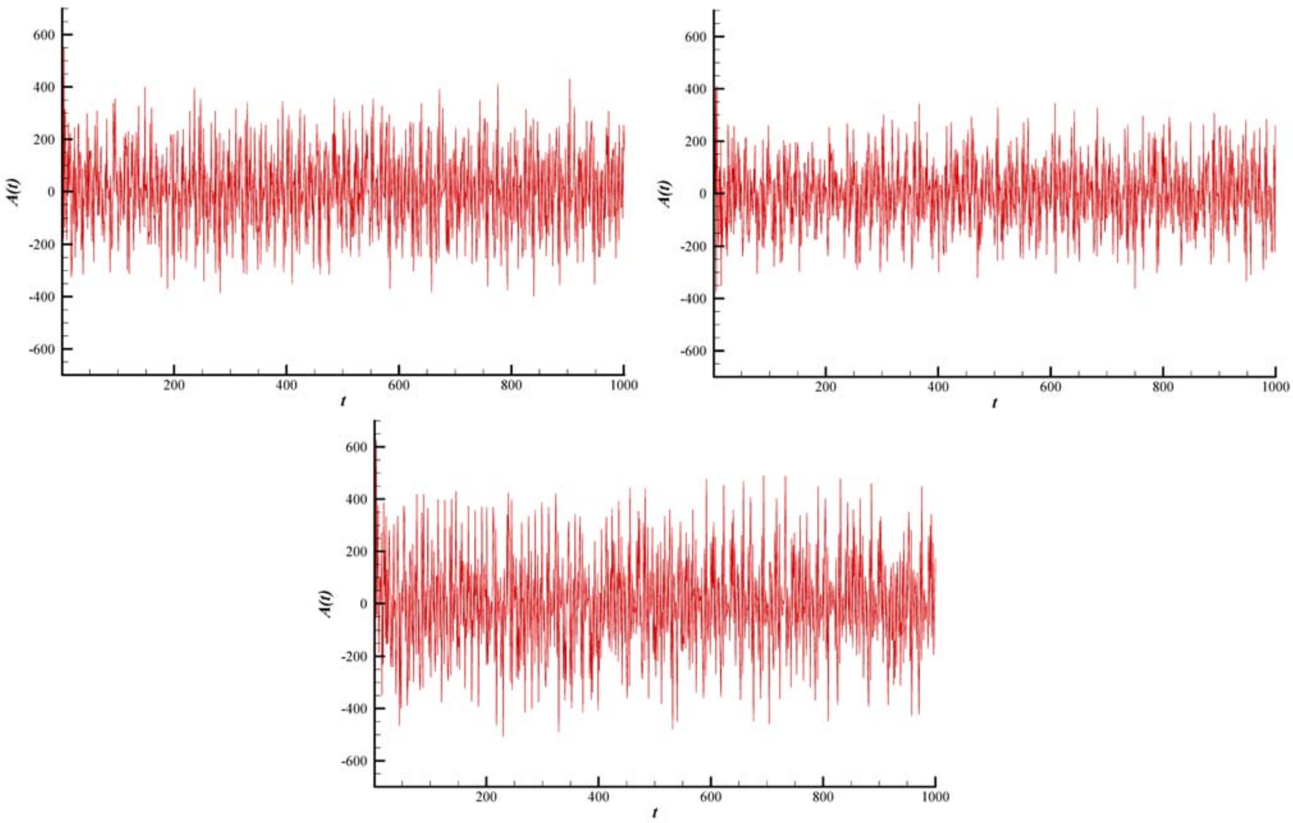

Fig. 3. Global quantity of the population for $N = 1001$, $M = 3, 5, 7$. Strategy is a 2-layer perceptron neural network with 3 nodes in hidden layer

of learning rate in Fig. 4 and Fig. 5. One can easily realize that agents with high learning rate have obtained more actual points. However, agents with low learning rate have much diversity on actual points. To increase the nodes of hidden layer, the agents perform approximately identical.

We first let the learning rate to be alike and certain. In real word, the agents aren't homogeneity and haven't identical learning rate; however we simulated the population of agents with mentioned assumption. Later we will remove this relaxation and let the agents to have different rates of learning. This study showed that the fluctuations of attendance decrease again with increasing memory (M). In effect, the fluctuations decrease of population with enhancing memory compared to the OMG is very trivial in this situation.

Expeiment1 shows that the agent's information process power has increased by neural network and therefore by enhancing memory, the fluctuations have reduced noticeably.

3.2 Experiment 2: changing learning rate

In this experiment, all of parameters are fixed expect learning rat and we simulate a population with 101 agents which use neural networks. This can be observed in Fig. 7 that the population with larger learning rate has better performance and thus fluctuations decrease and subsequently average points of agents increase accordingly. Fig. 8 and Fig. 9 depicts the situation. In high learning rates, population behavior is similar to evolutionary agents at the last step of simulation. This situation is preferable since it has the lowest fluctuations. Note that coordination and cooperation happen perfectly. The above results obtained for interacting neural networks, by 5 input of information and/or number of 9 nodes in hidden layer; but in other deterministic amount of this two parameters, these results are repeated; although, curves of Fig. 6 and Fig. 7 shift to right or left side.

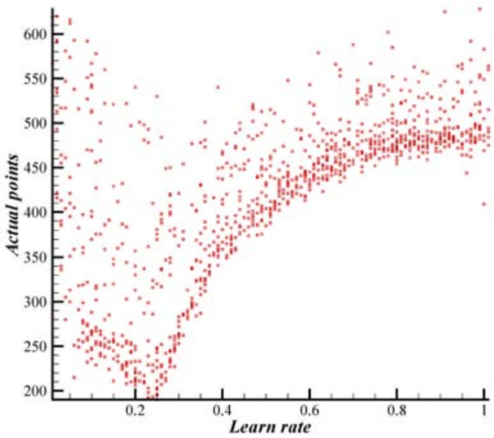

Fig. 4. Actual points of agents versus learning rate: agents with neural network 3-3-1

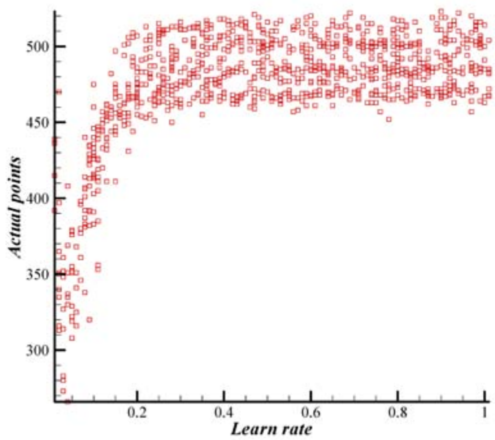

Fig. 5. Actual points of agents versus learning rate: agents with neural network 3-15-1

3.3 Experiment 3: changing the number node in hidden layer

For investigating the effect of variation of the number of nodes in hidden layer, we simulated population of agents for the cases $N = 1001$, $M = 3$ that they use 2-layer perceptron neural network with random learning rate. Plotting global quantity of population in Fig. 8 and Fig. 9, shows that the fluctuations are indeed decreasing when the number nodes in hidden layer is increased. In this state, the agents also behave similar to evolutionary agents.

4 OMG compared with NNMG

The common parameters of tables' strategy and the proposed one in this paper are: number of agents (N) and memory (M). Thus, these parameters are identical in both of version; but quantities of other parameter are selected to make the lowest fluctuations of attendance. For comparison of the agents' performance two versions of game, we simulated them.

By plotting global quantity of agents for time step 1000 in the population 1001 with memory 5, for two versions of game in Fig. 10 and Fig. 11 show fluctuations of attendance in NNMG and in OMG, respectively. The agents' attendance fluctuations in Fig. 10 are significantly smaller than those of Fig. 11. The proposed model shows that agents with neural network are more adaptive and dynamic and therefore can be considered to be more intelligent. Our simulation results show that in our model, agents can produce more efficient cooperation. If we address agents of NNMG and agents of OMG as group 1 and 2, respectively, Fig. 12 shows a comparison between the performances of the two groups. It can be seen that group 1 has better performance and diversity of agents' points of group 1 is significantly higher than those of the other group.

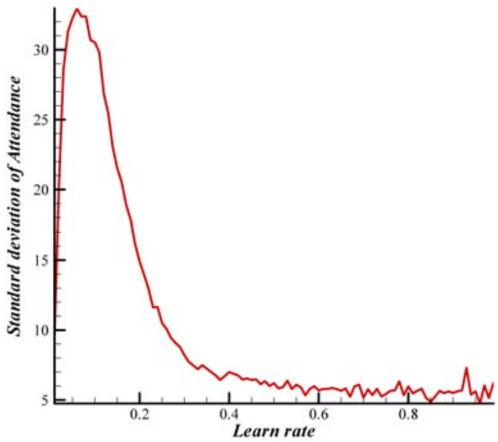

Fig. 6. Standard deviation of attendance versus learning rate for 1000 time step

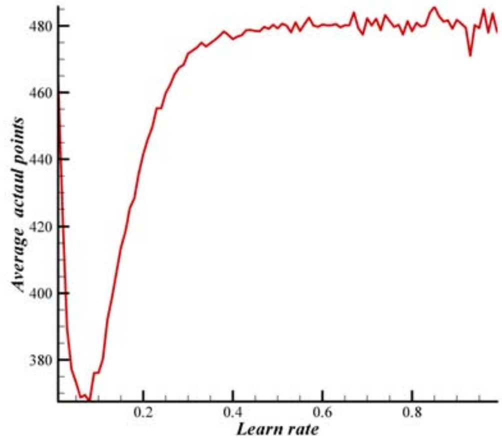

Fig. 7. Average actual points versus learning rate for 1000 time step

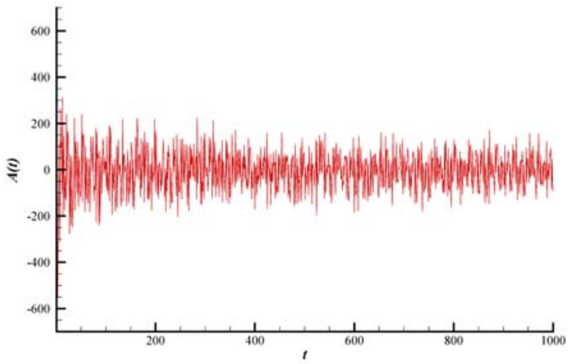

Fig. 8. Total action of the population for $N = 1001$, $M = 3$: 2-layer perceptron neural network with 10 nodes in hidden layer

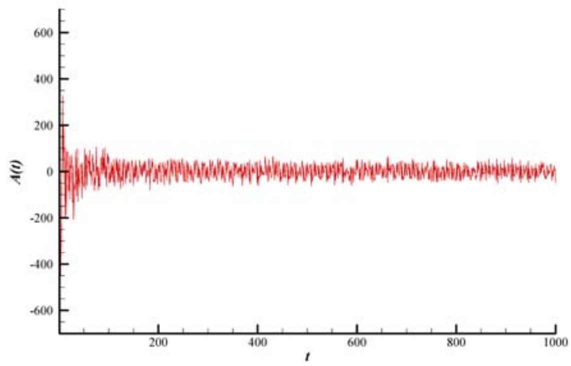

Fig. 9. Total action of the population for $N = 1001$, $M = 3$: 2-layer perceptron neural network with 30 nodes in hidden layer

5 Conclusion

In this paper, we developed a version of minority game. It is a competitive mechanism for determining how to take advantage of limited resources. In the original form the agents in minority game make decisions using strategy tables. These tables are developed based on the history of the past. In the original form of minority game, the learning of agents takes place through the process of virtual rating. We equipped the agent to two layer perceptron neural networks instead of strategy tables. Back propagation algorithm is used as the method of learning of the neural network.

Our simulation results showed that to increase intelligence of agent's fluctuation of attendance decreases in proposed version similar to OMG. Effective parameters in NNMG are learning rate and node number of hidden layer. Also, the agents with small memory, high learning rate, and low node number of hidden layer have behaviour equal to agents with small memory, low learning rate, and high node number of hidden layer.

Eventually an analogy survey of the agents' performance is carried out between the previous and the new approaches. Results demonstrate that applying two layer perceptron neural networks yields better decisions according to reduced agents' attendance so it enhances performance averages.

The adaptive complex system is a meaningful and interesting field. There are still many questions that need to be answered. This modelling can be extended to the other case of such systems. Also, one can extend this model for forecasting time series in financial markets as well.

References

[1] W. Abdullah. Emergence of heterogeneity in an agent-based model. *Physics A*, 2003, (324): 311–316.

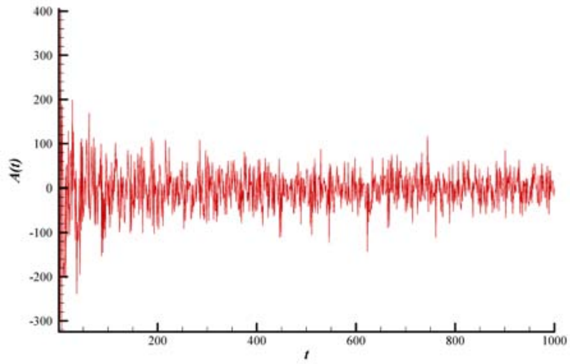

Fig. 10. Fluctuations of attendance with interacting neural network 5-15-1

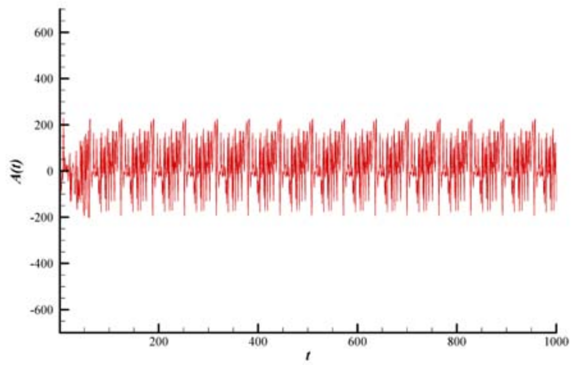

Fig. 11. Fluctuations of attendance in OMG with $M = 5$, $S = 2$

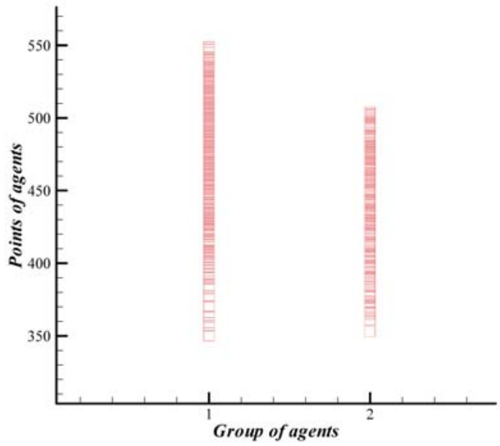

Fig. 12. Comparison of agents' performance of two groups

[2] W. Arthur. Inductive Reasoning and Bounded Rationality. *Am Econ*, 1994, **84**: 406–411.
[3] D. Challet, A. Chessa, et al. From Minority Games to Real Markets. *Quantitive Finance*, 2001, **1**: 168–176.
[4] D. Challet, M. Marsili, Y. Zhang. Stylized Facts of Financial Markets and Markets Crashes Minority Games. *Physica A Statistical Mechanics and its Applications*, 2001, **294**(3-4): 514–524.
[5] D. Challet, Y. Zhang. Emergence of Cooperation and Organization in an Evolutionary Game. *Physics A*, 1997, (246): 407–418.
[6] T. Chmura, T. Pitz. Successful Strategies in Repeated Minority Games. *Physics A*, 2006, (363): 477–480.
[7] F. Chow, H. Chau. Multiple choice minority game. *Physics A*, 2003, (319): 601–615.
[8] G. Cybenco. Approximation by Superpositions of Sigmoid Function. *Mathematics of Control Signals and systems*, 1989, **2**: 303–314.
[9] C. Foldy, Z. Somogyvaria, P. Erdi. Hierarchically Organized Minority Games. *Physics A*, 2003, (323): 735–742.
[10] R. Groot, P. Musters. Minority Game of price promotions in fast, moving consumer goods markets. *Physics A*, 2005, (350): 533–547.
[11] J. Holland. Complex Adaptive Systems. *Daedalus Boston*, 1992, **121**(1): 17–30.
[12] G. Hornik, M. Stinchcombe, H. White. Multilayer Feedforward Neural Networks are Universal Approximation. *Neural Networks*, 1989, **2**(5): 359–366.
[13] N. Jennings, K. Sycara, M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1998, **1**: 275–306.
[14] N. Johnson, P. Hui, et al. Self-organized segregation within an evolving population. *Physical Review Letters*, 1999, **82**(16): 3360–3362.
[15] K. Kim, S. Yoon, M. Yum. Dynamics of the minority game for patients. *Physics A*, 2004, (344): 30–35.
[16] W. Kinzel, R. Metzler, I. Kanter. Dynamics of Interacting Neural Networks. *Physics A: Mathematical General*, 2000.
[17] B. Krose, P. Smagt. *An introduction to neural network*, 8th edn. The University of Amsterdam, 1996.
[18] J. Lawrence. *Introduction to Neural Networks*. California Scientific Software Press, 1993.

[19] Y. Li, R. Riolo, R. Savit. Evolution in minority games (II): Games with variable strategy spaces. *Physics A*, 2000, (276): 265–283.
[20] M. Marsili. Market Mechanism and Expections in Minority and Mijority Games. *Physics A*, 2001, (299): 93–103.
[21] S. Moelbert, D. Rios. The local minority game. *Physics A*, 2002, (303): 217–225.
[22] J. Muller. Multi-agent Systems and Simulation of Complex System. *LAFMI*, 2003. http://lafmi.Lania.mx/escuelas/-esd03/Ponencia/Muller.pdf.
[23] R. Nielson. Kolomogrov's Mapping Neural Network Existence Theorem. *IEEE First International Conferenceon Neural Networks*, 1987, 11–14.
[24] R. Sutton, A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1988.
[25] S. Whitehead, L. Lin. Reinforcement learning of non-Markov decision processes. *Artificial Intelligence*, 1995, 271–306.
[26] C. Yang, S. Sun. The minority game with incomplete strategies. *Physics A*, 2007, (397): 645–653.

Appendix

Computerized pseudo-code minority game with Neural Networks.

C *Agents, choice*

Do $i = 1, N$
Neural Network (i)
    $y(i, \text{Hidden}) = f^1(X(i) * W(i,1) + b(i,1))$
    $\bar{a}(i) = f^2(y(i, \text{hidden}) * W(i,2) + b(i,2))$
    If $(\bar{a} > 0)$ then
        $a(i) = 1$
    Else
        $a(i) = -1$
    End if
End do

C *Reinforcment learning*

Do $i = 1, N$
Neural Network (i)
    $e(i) = a(i) - \text{Winside}$
    Sigma $(i,1) = -2 * a(i) * e(i)$
    Sigma $(i,2) = f^1(\cdots) * W(i,2) * \text{Sigma}(i,1)$
    Do $l = 1, 2$
        $W(i,1) = W(i,1) + \text{LearnRate} * \text{Sigma}(i,1) * y$
        $b(i,1) = b(i,1) + \text{LearnRate} * \text{Sigma}(i,1)$
    End do
End do

C *Information update*

Shift Information
Or $\text{Mu} = \text{mod}(2 * \text{Mu} + \text{Winside}, 2\,\hat{}M)$