

ORIGINAL RESEARCH

THE EFFICACY OF CHOOSING STRATEGY WITH GENERAL REGRESSION NEURAL NETWORK ON EVOLUTIONARY MARKOV GAMES

Shirin Kordnoori*¹ | Hamidreza Mostafaei² | Mohammadmohsen Ostadrahimi³ | Saeed Agha Banihashem⁴

¹Dept. of Computer Engineering-Artificial Intelligenc, Islamic Azad University, Tehran, Iran

²Dept. of Statistics, Islamic Azad University, Tehran, Iran

³Dept. of Mathematics, Islamic Azad University, Tehran, Iran

⁴School of International Relations , Ministry of Foreign Affairs of the Islamic Republic of Iran, Tehran, Iran

Correspondence

*Shirin Kordnoori, Department of Computer Engineering-Artificial Intelligence, North Tehran Branch, Islamic Azad University, Tehran, Iran Email: sh.kordnoori@iau-tnb.ac.ir

Present Address

Department of Computer Engineering-Artificial Intelligence, North Tehran Branch, Islamic Azad University, Vafadar Blvd., Shahid Sadoughi St., Hakimiyeh Exit, Shahid Babae Highway, Tehran, Iran

Abstract

Nowadays, Evolutionary Game Theory which studies the learning model of players, has attracted more attention than before. These Games can simulate the real situation and dynamic during processing time. This paper creates the Evolutionary Markov Games, which maps players' strategy-choosing to a Markov Decision Processes (MDPs) with payoffs. Boltzmann distribution is used for transition probability and the General Regression Neural Network (GRNN) simulating the strategy-choosing in Evolutionary Markov Games. Prisoner's dilemma is a problem that uses the method and output results showing the overlapping the human strategy-choosing line and GRNN strategy-choosing line after 48 iterations, and they choose the same strategies. Also, the error rate of the GRNN training by Tit for Tat (TFT) strategy is lower than similar work and shows a better result.

KEYWORDS:

Boltzmann distribution, Evolutionary Game Theory, General Regression Neural Network, Neuro-Fuzzy Network, Tit for Tat Strategy

1 | INTRODUCTION

Game Theory (GT) is a theory of the science of logical decision-making in humans and computers, whose actions affect each other. When a situation with two or more players involves known payouts or quantifiable consequences, game theory can help determine the most likely outcomes. A few terms commonly used in the study of game theory are listed as follows:

- **Game:** Any set of conditions that their results depend on decision-makers actions (players).
- **Players:** A strategic decision-maker within the context of the game.
- **Strategy:** The action which the player will take at any stage of the game.
- **Payoff:** When a player arriving at a particular outcome, they receive the payout, which can be in any quantifiable form, from dollars to utility.
- **Information Set:** The facts available at a given point in the game. The term information set is most usually applied when the game has serial parts.
- **Equilibrium:** The phase in a game where both players have made their final decisions, and the game attains an optimal solution.

A game typically includes several players, strategies or actions, orders, and payoffs, similar to the individuals, genetic operators, and fitness involved in evolutionary algorithms. Players usually select a strategy based on the expected payoffs. A Markov process can be represented by the stochastic extension of a finite state automaton. In a Markov process, state transitions are probabilistic, and in contrast to a finite state automaton, there is no input to the system. Furthermore, the system is only in one state at each time step. Supposing the space of state is $I = 0, 1, 2, \dots$ and a set of time is $T = 0, 1, 2, \dots$. If for arbitrary t and state i_0, i_1, \dots, i_n, j , then:

$$P\{X_{t+1} = j | X_t = i, X_{t-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0\} = P\{X_{t+1} = j | X_t = i\} \quad (1)$$

Then a stochastic process $\{X_t, t \in T\}$ is called Markov Process which its most recent values determine future probabilities. The one-step transition probability is a conditional probability that has the following definition:

$$P_{ij} = P\{X_{t+1} = j | X_t = i\} \quad (2)$$

Transition probabilities can be then computed as a power of the transition matrix shown in Equation 3.

$$x^{(n+1)} = P \cdot x^{(n)} \quad (3)$$

$$x^{(n+1)} = P \cdot x^{(n+1)} = P^2 \cdot x^{(n)} \quad (4)$$

$$x^{(n)} = P^n \cdot x^{(0)} \quad (5)$$

And one-step Markov transition probability matrix, which this paper uses, has a definition as shown in Equation 6.

$$P = \begin{bmatrix} P_{00}(1) & \dots & P_{0n}(1) \\ \dots & \dots & \dots \\ P_{n0}(1) & \dots & P_{nn}(1) \end{bmatrix} \quad (6)$$

In some situations, players would exploit learning methods. Intelligence consists of making the right decisions in a given situation to achieve a certain goal. Game theory provides mathematical models of real-world situations for studying intelligent behavior. Most of the time, effective decision-making in strategic situations (such as competitive situations) requires nonlinear mapping between stimulus and response. This can be provided by artificial neural networks, such as reinforcement learning, belief learning, imitation, neural network, and Fuzzy neural networks to guarantee gains. In this case, the player can decide based on their perception of the actions of other players.

Neural networks are algorithms modeled loosely after the human brain that is designed to recognize patterns. They can be effective in the programming of learning, generalization, and optimization functions^[1]. The General Regression Neural Network

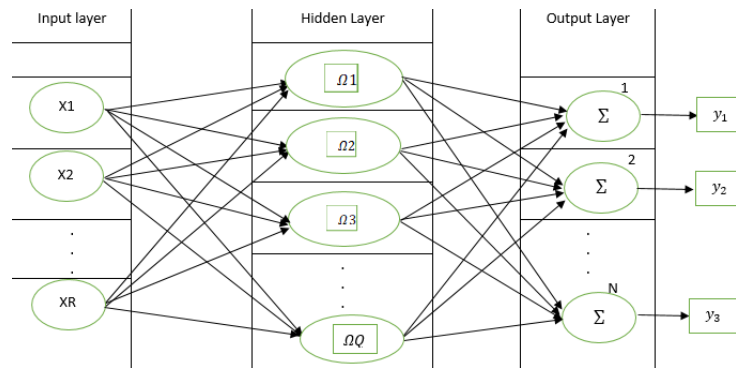


FIGURE 1 The figure of the GRNN neural network.

(GRNN) was introduced by Donald Specht^[2]. The GRNN was introduced as a memory-based neural network that would store all the independent and dependent training data available for a particular mapping. When a new input vector presents, the GRNN uses a distance function to compute the weighted average of the dependent training data whose independent parameters are close to the input vector. Apart from the fact that the GRNN and the Radial Basis Functions (RBFs) were motivated by different principals, their implementation is very similar. Both networks perform identical processing in the hidden layer and linear operations in the output layer. The main difference is that the GRNN output layer performs a weighted average while the RBF performs a weighted sum. Which of these two networks is better depends mainly on the application? The GRNN definition requires that it store the majority of the training data.

When computational constraints are not significant, such as in-stock market prediction, where predictions may only be required once per day, it is probably best to use the memory-based GRNN. However, computational efficiency is more crucial in most other applications, such as embedded control systems, and the RBF network would be preferred. Figure 1 shows a case of a GRNN neural network, this kind of neural network includes: an input layer, pattern layer, summation layer, and output layer.

In this paper, the GRNN neural network simulates the player decision-making for the learning process. In evolutionary Markov Games, the inputs of the neural network are decided by the last actions of players. The output is the next action that the player will choose. Simulation results show that the neural network can successfully simulate evolutionary Markov Games. In continuation of this paper first, the previous researches will be explained. In second, the theory of evolutionary Markov games and GRNN neural networks will present, and in final, the discussion about the output of the simulation will declare.

2 | PREVIOUS RESEARCHES

In biology, game theory is a critical part of a mathematical and computational approach. So the 1970s evolutionary game theory was introduced by Maynard Smith. Ganesan et al.^[3] in 2015 presented another evolution of game-theoretic, name GTDE. GTDE was conducted in the framework of differential evolution (DE) with the strategies and two computational times of DE. Wood et al.^[4] in 2016 studied OPEC and the Seven Sisters. They used a methodological toolkit that involved evolutionary game theory and agent-based modeling. For modeling evolutionary game theory, they used heterogeneous populations, energy-specific variables, and behavioral considerations to capture the essentials of the applied problem. To provide detailed results, an agent-based model was used.

Sharma^[5] in 2016 proposed a Lyapunov theory-based Markov game fuzzy controller, which was both safe and stable. They attempted to optimize a reinforcement learning (RL) based controller using Markov games and hybridizing it with a Lyapunov theory-based control for stability. Yang^[6] in 2017 suggested an algorithm that used an adaptive technique based on cellular automata and named it evolutionary algorithms based on game theory and cellular automata with coalitions (EACO). In most cases, this algorithm exceeds the compared cellular genetic algorithm but for some benchmarks of combinatorial optimization problems is not functional. This method provides game strategies of better quality than genetic algorithms but needs longer running times.

Garay et al.^[7] in 2017 introduced a matrix game under time constraints, where each pairwise interaction had two consequences: both players received a payoff and could not play the next game for a specified time duration. So, their model is defined by two matrices: a payoff matrix and an average time duration matrix. They illustrated the effect of time constraints by the prisoner's dilemma game, where additional time constraints can ensure the existence of unique evolutionary stable strategies (ESS), both pure and mixed, or the coexistence of two pure ESS.

Tosh et al.^[8] in 2018 used evolutionary dynamics to analyze both participation and information sharing games and understood the outcomes of the game, i.e., stable evolutionary strategies (ESS). Various conditional constraints derived from the analysis helped to devise a dynamic cost adaptation algorithm that exploits the participation cost to act as an incentive, in the beginning, to motivate as many firms to participate.

Quan et al.^[9] in 2018 studied the stochastic evolutionary optional public game by considering both continuous noise and the finite size population effects. The evolutionary process of strategies in the population is described as a two-dimensional Markov process. They focused on the SSEs of the system, which was analyzed by the limited distribution of the Markov process. Furthermore, they investigated the influence of parameters on the probabilities of the system to choose different SSEs.

Overton et al.^[10] in 2019 evaluated the fixation probability of invading mutants for given initial conditions by showing how to derive existing pair approximation models and developing node-level approximations to stochastic evolutionary processes on arbitrarily complex structured populations represented by finite graphs. They discussed that the Construction of these models facilitates a systematic analysis and can be valuable for a greater understanding of the influence of population structure on evolutionary processes.

Izquierdo et al.^[11] explained how ABED, free and open-source software for simulating evolutionary game dynamics in finite populations, can simulate a wide range of dynamics considered in the literature and many novel dynamics. They introduced a general model of revisions for dynamic evolutionary models, which decomposes strategy updates into selecting candidate strategies, payoff determination, and choice among candidates.

Jia Wang et al.^[12] constructed the Wright-Fisher evolutionary game dynamic model by introducing different intensities for different strategies. According to their analysis, in finite populations, fixation probability is related to game payoffs and the population size N and is affected by the different selection intensities.

Gu et al.^[13] in 2020 expressed the game payoffs as the normal fuzzy number. In their study, the stochastic evolution game of the Moran process of the finite population was generalized to the fuzzy environment. They also introduced the fuzzy dilemma, according to classified the fuzzy games. Singh et al.^[14] in 2020 considered a dynamic game model of the bitcoin market, where miners or players use mining systems to mine bitcoin by investing electricity into the mining system. This work is motivated by BTC and can be applied to other mining systems similar to BTC. Their work proposed three concepts of dynamic game-theoretic solutions to the model: Social optimum, Nash equilibrium, and myopic Nash equilibrium.

Srivastava et al.^[15] in 2020 presented a novel algorithm to control data security using a hybrid model that used an adaptive encoding technique alongside Chaotic Hopfield Neural Network. The proposed computation upgraded the security of a key shared between any nodes. They showed that the security of transmitted data had better performance than traditional algorithms, and also, the computational time for the proposed algorithm was less than known traditional algorithms.

This group of games has dynamic characteristics; their choosing strategies can change through processing time. One idea to deal with this problem is to map players' strategy-choosing to a Markov decision process with payoffs.

3 | MATERIAL AND METHOD

Human modeling gameplay with a powerful neuro-fuzzy network, like the GRNN network, is the main issue in the context of game theory. The cooperation of two or more players leads to a higher payoff for the cooperating group. To be clear, this research is interested in building a GRNN neural network that converges to optimal results. In continuation of this research, the hope is to illuminate decision-making strategies that may be at work to be applied on this network and compare their results to find the optimal ones which reach the Nash equilibrium.

3.1 | Evolutionary Markov Games

This paper considers the repeated games, which look at iteration times of games as a set of time $t=1,2,\dots$ in the Markov process. Strategy space was mapped to the state space I of the Markov process. In repeated games, players by their payoffs update strategies, and the payoffs in repeated games were represented as rewards for state transition. This paper discusses the iterated prisoner's dilemma, which uses a binary choice for the players, either cooperation (C) or defection (D).

In general, if at time t (the t -th repeated game) the player is in state- i ($i = C, D$), then at time $t + 1$, the probability that the player chooses strategy C or D can be obtained by Markov transition probability matrix, which shows in Figure 2. So the iterated prisoner's dilemma is a Markov process with the two-state transition.

$$\begin{array}{c} X_{t+1} \\ D \quad C \\ X_t \begin{array}{c} D \\ C \end{array} \begin{bmatrix} P_{DD} & P_{DC} \\ P_{CD} & P_{CC} \end{bmatrix} \end{array}$$

FIGURE 2 Markov Process for iterated prisoner's dilemma.

Consider a Markov process with N states. If the state transfers from i to j , then u_{ij} can be the payoff and a series of rewards for state transition, which are stochastic variables. The Markov transition probability matrix determines its probability distribution.

For more definitions, consider 2-player symmetric repeated games. The player's strategy is i , the total expected payoff after n times repeated games can calculate by using the Markov process. If $v_i(t)$ would be the total expected payoff after t times transition from state i , then the expected reward after the one-time transition from state i can be calculated by Equation 7.

$$q_i = \sum_{j=1}^N p_{ij} u_{ij}, \quad (i = 1, 2, \dots, N) \quad (7)$$

And also the following equation can define:

$$v_j(t) = \left[\sum p_{ij} u_{ij} + \sum p_{ij} v_j(t-1) \right], \quad (i = 1, 2, \dots, N; t = 1, 2, 3, \dots) \quad (8)$$

By using q_i in Equation 1, we can define Equation 9.

$$v_i(t) = q_i + \sum_{j=1}^N p_{ij} v_j(t-1) \quad (9)$$

Equation 2 will be defined as the following by using the form of vectors, where $V(t)$ is the column vector of $v_i(t)$, Q is the column vector of q_i , and P is the Markov transition probability matrix:

$$V(t) = Q + P V(t-1), \quad (n = 1, 2, 3, \dots) \quad (10)$$

This paper uses the Boltzmann distribution for calculating p_{ij} . Equation 11 calculates the probability of state transition from state i to j at time t :

$$p_{ij} = \frac{e^{\frac{u_j(t)}{\lambda}}}{\sum_k e^{\frac{u_k(t)}{\lambda}}}, \quad (k = 1, 2, \dots, N) \quad (11)$$

In Equation 11, $u_j(t)$ is the payoff when the player chooses strategy j . For the learning process, the parameter λ has an important role in choosing the optimal strategy more accurately. By increasing λ , the randomness of decisions can increase and upside down.

3.2 | General Regression Neural Network (GRNN)

Artificial Neural Networks (ANNs) have two main types: The Feed Forward ANNs (FFANNs), in which the input will only flow to the output layer in the forward direction, and the Recurrent ANNs (RANNs) in which data flow can be in any direction. Generalized Regression Neural Networks (GRNN) is a single-pass associative memory feed-forward type Artificial Neural Networks and uses normalized Gaussian kernels in the hidden layer as activation functions. GRNN is made of input, hidden, summation, division layer, and output layers.

When GRNN is trained, it memorizes every unique pattern. This is the reason why it is a single-pass network and does not require any backpropagation algorithm. After training GRNN with good training patterns, it will be able to generalize for new inputs. The output of GRNN can be calculated using Equation 12 and Equation 13.

$$D_i = (X - X_i)^T (X - X_i) \quad (12)$$

$$\hat{Y} = \frac{\sum_{i=1}^N Y e^{\left(\frac{-D_i}{2\sigma^2}\right)}}{\sum_{i=1}^N e^{\left(\frac{-D_i}{2\sigma^2}\right)}} \quad (13)$$

Where D_i is the Euclidean distance between the input X_i and the training sample input X , Y is the training sample output, σ is the smoothing parameter of GRNN.

GRNN's advantages include its quick training approach and its accuracy. On the other hand, one of the obstacle ideas of GRNN is the growth of the hidden layer size. However, this issue can be solved by implementing a special algorithm that reduces the growth of the hidden layer by storing only the most relevant patterns.

4 | RESULTS AND DISCUSSION

The prisoner's dilemma is a classic model in Game Theory; This game is a non-cooperative, non-zero-sum game and shows why two completely rational individuals might not cooperate, even if it appears that it is in their best interests to do so. The prisoner's dilemma has illustrated with the payoff matrix. Figure 3 shows the calculation of payoff players with the following descriptions: Two members of a criminal gang are arrested and imprisoned. Each prisoner is in solitary confinement with no means of communicating with the other. The prosecutors lack sufficient evidence to convict the pair on the principal charge, but they have enough to convict both on a lesser charge. Simultaneously, the prosecutors offer each prisoner a bargain. Each prisoner is given the opportunity either to betray the other by testifying that the other committed the crime or to cooperate with the other by remaining silent. The possible outcomes are:

- If A and B each betray the other, they both receive reward three for cooperating.
- If A betrays B, but B remains silent, A will be set free, and B will serve five years in prison.
- If A remains silent but B betrays A, A will serve five years in prison, and B will be set free.
- If A and B both remain silent, both of them will serve only one year in prison (on the lesser charge).

So each player has two choices; either cooperation or defection. Figure 3 shows the calculation of payoff players.

It can be understood from Figure 3 that defection is a dominant action, and any rational player will choose defection no matter what others choose. But if they cooperate, they would get more.

The iterated prisoner's dilemma repeats the conventional game numerous times without knowing the number of repetitions of both players. By using this iteration, the hope of cooperation would be increased. This method is widely used to model systems

		Player 2	
		Cooperation	Defection
Player 1	Cooperation	3 3	0 5
	Defection	5 0	1 1

FIGURE 3 Payoffs in prisoner's dilemma.

in biology, sociology, psychology, and economics. The situation of this game is dynamic and can be modeled as an evolutionary Markov game with payoff and transition probability is made by Boltzmann distribution.

The GRNN neural networks match pair to play iterated prisoner's dilemma games with a human. The GRNN neural network comprises an input layer, one hidden layer, and one output layer. The input layer includes six input units, which is his and the opponent's last actions, and the hidden layer is composed of 20 units. Table 1 shows the simulation parameters:

TABLE 1 Simulation parameters and their definition.

Simulation Parameters	Definition
Action information	1 → cooperation 0 → defection
Maximal iteration times	5000
Minimal error of mean square	0.01
Number of input units at the neural network	6
Number of hidden units at neural network	20
Number of output units at the neural network	1

The evolutionary curve of human and neural networks has shown in Figure 4. It shows that after 70 iterations, human and GRNN Network cooperate and get the result that it has better performance than other similar work with MLP neural network. This figure shows that after 48 iterations, humans and GRNN choose the same strategy.

The most successful strategy for playing the prisoner's dilemma is Tit for Tat (TFT) strategy submitted by Anatol Rapoport. The start of this game is with a move to cooperation but repeats the last move made by its opponent. GRNN neural network uses to play with TFT strategy, and Figure 5 is the curve of epoch and error by GRNN neural network. It can be understood from the results that after 40 times iterated games, the error rate would start to decrease. The best error rate would be 0.0059 in 256 epochs.

Although MLP networks have better performance than GRNN networks in some problems, in this case, GRNN shows better results than Weibing et al.^[16]. But in their work, after 217 iterations, the MLP neural network and humans successfully choose the same strategy. Still, by using the GRNN network, the iterations decrease to 48, and also the error rate, which was sum square error, in their works in playing with TFT strategy was 10^{-2} but mean square error has decreased by GRNN neural networks and was less than that, which showed in Figure 5.

5 | CONCLUSION

In recent year's evolutionary game theory, especially, the study on the learning model of players has attracted more attention than before because they can simulate the real situation and dynamic during processing time. This paper map players' strategy-choosing to a Markov decision process with payoffs to create the evolutionary Markov process. The transition probability is made by Boltzmann distribution and also used the GRNN neural network to simulate the strategy-choosing in evolutionary Markov games. This process applied to iterated prisoner's dilemma, and output results showed that after 48 iterations, the human

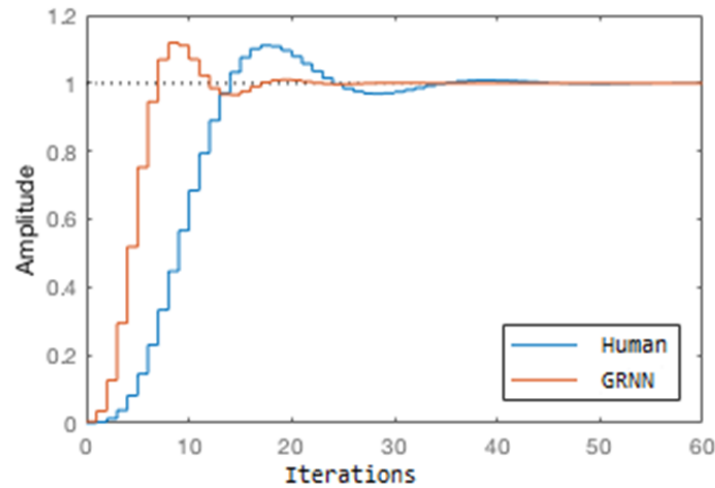


FIGURE 4 The evolutionary curve of human and GRNN neural network.

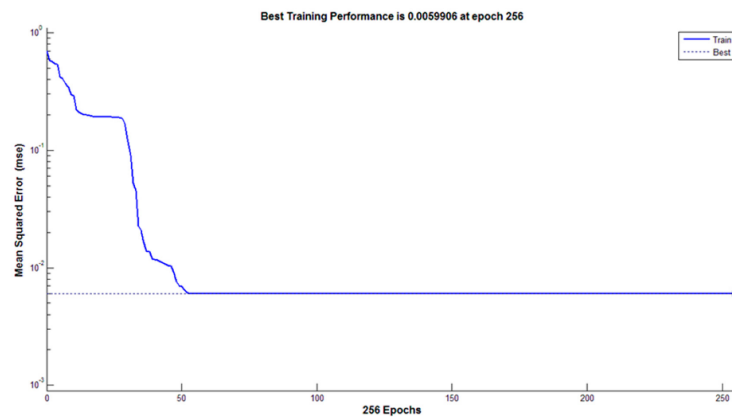


FIGURE 5 The curve of epoch-error.

strategy-choosing line and GRNN strategy-choosing line overlapped, and they choose the same strategies. Also, GRNN trained by TFT strategy, the error rate was less than similar work and showed improvement in result. For future work, it's useful to use other kinds of neural networks in unsupervised groups to get better performance and decrease the number of iterations in choosing a strategy. It would lower the computation. On the other hand, it's better to make tournament strategies to find fitter one in reaching Nash equilibrium, such as Defector, the handshaking CollectiveStrategy, and Aggravater appearing, which are good competitors for TFT strategy.

CREDIT

Shirin Kordnoori: Investigation, Software, Conceptualization, Writing - Original Draft. **Hamidreza Mostafaei:** Formal analysis, Supervision, Writing - Review and Editing. **Mohammadmohsen Ostadrahimi:** Data Curation, Writing - Review and Editing. **Saeed Seyed Agha Banihashemi:** Validation, Writing - Review and Editing.

References

1. Liu WB, Wang XJ. An evolutionary game based particle swarm optimization algorithm. *Journal of Computational and Applied Mathematics* 2008;214(1):30–35. <https://www.sciencedirect.com/science/article/pii/S0377042707000799>.
2. Tomandl D, Schober A. A Modified General Regression Neural Network (MGRNN) with new, efficient training algorithms as a robust ‘black box’-tool for data analysis. *Neural Networks* 2001;14(8):1023–1034. <https://www.sciencedirect.com/science/article/pii/S089360800100051X>.
3. Ganesan T, Elamvazuthi I, Vasant P. Multiobjective design optimization of a nano-CMOS voltage-controlled oscillator using game theoretic-differential evolution. *Applied Soft Computing* 2015;32:293–299. <https://www.sciencedirect.com/science/article/pii/S1568494615001726>.
4. Wood AD, Mason CF, Finnoff D. OPEC, the Seven Sisters, and oil market dominance: An evolutionary game theory and agent-based modeling approach. *Journal of Economic Behavior & Organization* 2016;132:66–78. <https://www.sciencedirect.com/science/article/pii/S0167268116301202>, thresholds, Tipping Points, and Random Events in Dynamic Economic Systems.
5. Sharma R. Lyapunov Theory Based Stable Markov Game Fuzzy Control for Non-Linear Systems. *Eng Appl Artif Intell* 2016 Oct;55(C):119–127. <https://doi.org/10.1016/j.engappai.2016.06.008>.
6. Yang G. Game Theory-Inspired Evolutionary Algorithm for Global Optimization. *Algorithms* 2017;10(4). <https://www.mdpi.com/1999-4893/10/4/111>.
7. Garay J, Csiszár V, Móri TF. Evolutionary stability for matrix games under time constraints. *Journal of Theoretical Biology* 2017;415:1–12. <https://www.sciencedirect.com/science/article/pii/S002251931630409X>.
8. Tosh D, Sengupta S, Kamhoua CA, Kwiat KA. Establishing evolutionary game models for CYBer security information EXchange (CYBEX). *Journal of Computer and System Sciences* 2018;98:27–52. <https://www.sciencedirect.com/science/article/pii/S002200001630085X>.
9. Quan J, Liu W, Chu Y, Wang X. Stochastic dynamics and stable equilibrium of evolutionary optional public goods game in finite populations. *Physica A: Statistical Mechanics and its Applications* 2018;502:123–134. <https://www.sciencedirect.com/science/article/pii/S0378437118301882>.
10. Overton CE, Broom M, Hadjichrysanthou C, Sharkey KJ. Methods for approximating stochastic evolutionary dynamics on graphs. *Journal of Theoretical Biology* 2019;468:45–59. <https://www.sciencedirect.com/science/article/pii/S0022519319300724>.
11. Izquierdo LR, Izquierdo SS, Sandholm WH. An introduction to ABED: Agent-based simulation of evolutionary game dynamics. *Games and Economic Behavior* 2019;118:434–462. <https://www.sciencedirect.com/science/article/pii/S0899825619301459>.
12. jia Wang X, ling Gu C, Quan J. Evolutionary game dynamics of the Wright-Fisher process with different selection intensities. *Journal of Theoretical Biology* 2019;465:17–26. <https://www.sciencedirect.com/science/article/pii/S0022519319300062>.
13. Gu C, Wang X, Zhao J, Ding R, He Q. Evolutionary game dynamics of Moran process with fuzzy payoffs and its application. *Applied Mathematics and Computation* 2020;378:125227. <https://www.sciencedirect.com/science/article/pii/S009630032030196X>.
14. Singh R, Dwivedi AD, Srivastava G, Wiszniewska-Matyszkiewicz A, Cheng X. A game theoretic analysis of resource mining in blockchain. *Cluster Computing* 2020 23:3 2020 jan;23(3):2035–2046. <https://link.springer.com/article/10.1007/s10586-020-03046-w>.
15. Srivastava G, Vinoth Kumar CNS, Kavitha V, Parthiban N, Venkataraman R. Two-stage data encryption using chaotic neural networks. *Journal of Intelligent & Fuzzy Systems* 2020 jan;38(3):2561–2568.

16. Weibing L, Xianjia W, Binbin H. Evolutionary Markov Games Based on Neural Network. In: Yu W, He H, Zhang N, editors. *Advances in Neural Networks – ISNN 2009 Berlin, Heidelberg*: Springer Berlin Heidelberg; 2009. p. 109–115.

How to cite this article: Kordnoori S., Mostafaei H., Ostadrahimi M., Banihashemi S.A., (2021), The Efficacy of Choosing Strategy with General Regression Neural Network on Evolutionary Markov Games, *IPTEK The Journal of Technology and Science*, 32(1):13-22.