Neural Network and Evolutionary Game Theory

Linshu Xu[1](✉), Shuchen Zhang[2], Kai Cheng[3], Hai Ci[4], and Ruotong Shen[5]

[1] No. 2 High School of East China Normal University, Shanghai 201203, China
xls2818291092@163.com
[2] University of Science and Technology of China, Hefei 210026, China
[3] Suzhou Industrial Park Xinghai High School, Suzhou 251000, China
[4] Dalian American International School, Dalian 116650, China
[5] Hangzhou High School Qianjiang Campus, Hangzhou 310000, China

Abstract. This thesis proposes the feasibility to advance evolutionary game theory by parameterizing evolutionary game problems with a finite payoff matrix. It eliminates subjectivity while determining the parameters in evolutionary game problems. A neural network trained through learning a large database of the existing examples of similar models outputs the parameters of the model with a few data of the model provided as the inputs. Unlike directly putting the whole adjacency matrix into a training set, in which the complexity to train the algorithm increases quadratically as the amount of parameters increases, the scale of the network is most acceptable and plausible. Therefore, it produces the parameters with both precision and efficiency. With the acquisition of a precise payoff matrix, the evolutionary process modeled by differential equations would be more closely fitted to real datasets than that with a subjectively assumed payoff matrix.

Keywords: neural network · evolutionary game theory · optimization

1 Introduction

Concerns regarding why and how abstract formulations explain the empirical processes of evolution have been raised considering the major role that mathematical models play in contemporary evolutionary theory. Various philosophical viewpoints having been taken on this issue suggests a fresh explanation proposing that causal linkages, rather than merely mathematical presumptions, constitute the basis for evolutionary models. A causal model is presented in this new account as both a "unified nature" that underlies evolutionary induction and an organizing framework that combines mathematical and empirical premises into a coherent network that cooperates to realize the epistemological objectives of evolutionary biology.

1.1 Neural Network

The neural network is a mathematical algorithm that simulates human thinking and is a typical representative of machine learning in data mining. A neural network is

J. Yen et al. (Eds.): ICBIS 2023, AHCS 14, pp. 1504–1512, 2023.
https://doi.org/10.2991/978-94-6463-198-2_156

an abstract computational model of the human brain. Simply put, a neural network is an output model obtained by inputting multiple nonlinear models and weighting the interconnections between different models, where the weighting process is done in the hidden layer. The hidden layer contains a nonlinear function. Artificial neural networks are divided into forwarding neural networks and feedback neural networks. Forward network refers to the direction of propagation from input to output without any feedback; so-called feedback network refers to the direction of propagation from input to output, except for the existence of loop back and feedback [1].

The advantage of neural network applications is first of all the good self-assembly learning ability. Neural networks can constantly modify their behavior according to the changes in external data and have a good classification ability for untrained data patterns. It also has a better ability to discover nonlinear relationships in data and can effectively discover the intrinsic laws of non-linearity [2]. In a practical business environment, there are more possibilities for nonlinear relationships in data than linear relationships. Finally, it has a high tolerance for outliers and noisy data. Regarding the disadvantages and considerations of neural networks, first of all, it requires a longer training time for the model. Secondly, it requires fewer and more accurate variables in the input layer, so the variables need to be selected before building the model. And, it needs to try several different models and then select the most stable model after several validations to ensure that the model has stable results. Also, it is more sensitive to missing values and the tendency to over-fit the data.

1.2 Evolutionary Game Theory

Evolutionary game theory is a theory that combines the analysis of game theory with the analysis of dynamic evolutionary processes. What is the difference between evolutionary game theory and game theory? It differs methodologically from game theory, which emphasizes static equilibrium and comparative static equilibrium, and dynamic equilibrium. Evolutionary game theory has its origins in the theory of biological evolution, which has been quite successful in explaining certain phenomena in the process of biological evolution.

The importance of evolutionary game theory is that today's economists are also achieving impressive results in using evolutionary game theory to analyze the factors that influence the formation of social habits, norms, institutions, or systems and to explain the processes that shape them [3]. Evolutionary game theory is an important analytical tool in evolutionary economics and is gradually developing into a new field of economics.

Evolutionary game theory has many advantages. The first one is that evolutionary game theory abandons the assumption of complete rationality, based on Darwin's theory of biological evolution and Lamarck's theory of genetics and genetics, and takes the adjustment process of group behavior as a dynamic system from system theory, and the behavior of individuals and their relationship with the group is carved out separately. Game theory assumes that the actor has a perfectly rational mind, which is an unimaginably infinite reasoning process, and is a very strict assumption as far as the actor's ability to understand the real world is concerned. The real world is usually not guaranteed by this assumption. But evolutionary game theory avoids this very well. Second, time plays a very important role in evolutionary game theory. The game theory ignores the time problem, emphasizes the equilibrium of instantaneous problems of actors, and considers time as symmetric or reversible even when it is considered. Finally, in game theory, when there are multiple Nash equilibria, the refinement of Nash equilibrium can be achieved using backward induction, but this approach presupposes that the participants need to satisfy a stronger rationality assumption than perfect rationality - serial rationality [4]. This is not possible to achieve in reality. In evolutionary game theory, the refinement of equilibrium is achieved by the forward induction method, i.e., the participants choose their future behavioral strategies based on the history of the game, which is a dynamic selection and adjustment process. Thus, although the participants are finitely rational, the dynamic selection mechanism will make it possible to reach one of the Nash equilibriums and achieve the refinement of the Nash equilibrium in the presence of multiple Nash equilibriums.

Evolutionary game theory uses the assumption of limited rationality for the actors, more relevant to reality, so these individuals do not have the "omniscience" of the actors in game theory and cannot instantaneously obtain optimal results in economic activities.

2 Application of Neural Networks

A straightforward 2-player evolutionary game example can be used to illustrate the disadvantages of the traditional mathematical method. Consider the situation where a population of animals competes for a limited supply of resources in the game and each individual's fitness increases with the number of resources it acquires. Say there are two groups, the hawks and the doves, competing for food. Hawks belong to a capital group, while doves belong to a small company. Resulting of the disparity in their physical performances, hawks are aggressive, using whatever means necessary to defend the food, while doves are noncompetitive and prefer to share resources rather than compete for them. The terms "hawk" and "dove" refer to the behavior of the same species of animals.

In this case, three types of paired competitions are possible:

1. Hawk vs Hawk

Two hawks competing for the face-off in a 50:50 fight over the resource. The resource's whole worth belongs to the winner in this winner-takes-all scenario. The injured loser pays a premium and suffers some fitness loss.

2. Hawk vs Dove

When a hawk approaches a dove, the bird will promptly retreat. The dove leaves empty-handed while the hawk takes home the whole value of the resource. But they incur no expense.

3. Dove vs Dove

Two doves who cross paths decide to divide the available resource equally. No one is harmed [5].

The mathematical solution determines whether various tactics can coexist or if one of them prevails. The problem is how the payoff matrix for hawks and doves can be determined. Usually, the payoff matrix is subjectively set up. Say the payoff of a hawk is 1 when 2 opposite hawks compete and the payoff of it is 6 when it approaches a dove. But where do the 1 and the 6 in the payoff matrix come up? A valid arrangement of the payoff matrix considers far beyond the distribution of food they receive as the behavior is also associated with numerous aspects of their life, including energy consumption, the relationship between the species, and other aspects impossible to be all parameterized [6]. To unravel all these uncertainties in the reality, which the mathematical solution cannot do, a neural network is introduced to compare the model with numerous existing models that are previously trained and learned. Computers make it possible to consider many independent variables with fair precision.

Our purpose is to find a proper sequence of parameters to describe an evolutionary game model. Start with the simplest evolutionary game model: finite-times-played 2-player symmetric game. Its playoff matrix is a necessary parameter in describing. Additionally, we can easily see that the initial action in the first round of players and the number of rounds played should be included. After adding up 7 parameters are required to describe this model. In more complex cases we may need other variables. A complex network or small-world network might be introduced to conclude their relationships if we intend to analyze a multiplayer game. A direct way is to put the whole adjacency matrix into a training set. The complexity of the training algorithm increases quadratically, while all information can be considered while the scale is acceptable for a neural network. Additionally, when extra examples are added, neural networks have advantages in maintaining the data set.

Briefly, a neural network regards the evolutionary process as a "black box" simply represented by a set of parameters. After learning a large database of the existing examples of similar models, it can output the parameters of the model with but a few data of the model provided as the inputs. It does not mean that the evolutionary process is disregarded. Instead, with the acquisition of a precise payoff matrix, the evolutionary process modeled by the differential equations would be more closely fitted to reality than that with a subjectively assumed payoff matrix.

3 Method and Algorithm

Solving the eventual strategies of players in a typical evolutionary game model deductively requires a sequence of computations including playing the game and changing strategies at each stage, in which the way of learning and network structure of complex networks are taken into consideration. However, errors are introduced when trying to describe the complex behavior of natural individuals by a single learning method. As massive examples of similar games in practice exist, naturally we want to predict an evolutionary game model through induction. We can easily connect this problem with a common fitting problem: If we can use a bunch of parameters to describe an evolutionary game model, we can extract these data from examples that existed in reality and train a neural network accordingly so that it can identify new models of similar structure and produce the result depending on the results of fitting, in case that simply fitting by interpolation would consume an unacceptable amount of time with so many features. In other words, we are viewing the whole game system as a "black box" and simulate it by a neural network trained with its inputs and outputs, as the mind map in Fig. 1 shows.

Without losing generality, we still take a 2-player symmetric evolutionary game model as an example (In fact, more complex models can be solved with exactly the

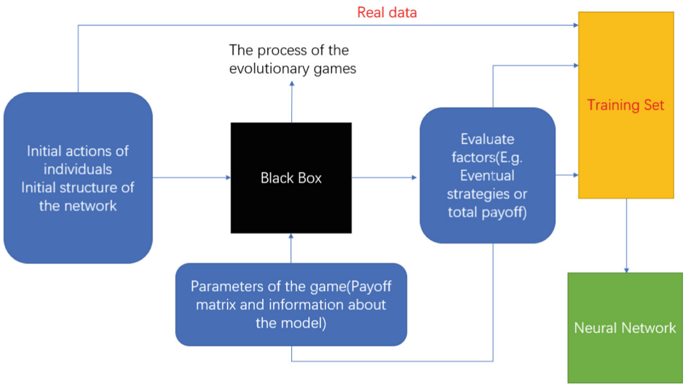

Fig. 1. Mind Map of the Algorithm [Owner-drawn]

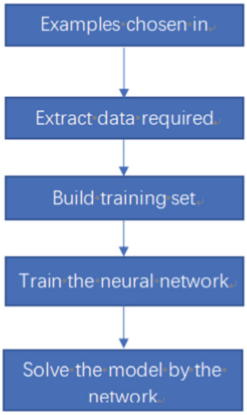

Fig. 2. Flow Chart of the Algorithm [Owner-drawn]

same thought). Notice that examples to extract data from should have a reasonable and convincing connection with the model to predict. Now we represent their initial strategy by x_1^0, x_2^0, meaning their probability to choose strategy 1 at the beginning, t to be the number of rounds played, a_{11}, a_{12}, a_{21}, a_{22} to be the parameters in the payoff matrix. The flow chart of the main part of the algorithm is shown in Fig. 2.

4 Applications

4.1 Two-Party Evolutionary Game Problems

Consider the case where the competent department of a public rented house chooses to exercise the supervision function to the lessee or not. And the lessee can choose the active exit strategy or the delayed exit strategy. All the parameters to be used are listed in Table 1.

Table 1. List of Parameters [7].

C_1	The public rental housing authority supervises the cost
C_2	Tenant delay exit behavior is not found to cause the waste of public resources
R_1	Housing costs saved by renters choosing to defer exit
R_2	The director of the public rental housing department is rewarded by the higher authority when he or she detects delayed withdrawal behavior
R_3	Lessee voluntarily withdraw from the reward received
F_1	The tenant's penalty for delaying withdrawal
P	The competent department of public rental housing supervises efficiently
F_g	The public rental authority found that delaying withdrawal gained prestige
F_d	The authority of public rental housing loses prestige when the delay is not detected

For competent authorities, the expected income when the competent department of public rented house does exercise the supervision function and the income when it does not is, respectively:

$$U_{Gx} = y(-C_1 - R_3) + (1-y)\left[p\left(-C_1 + R_2 + F_1 + F_g\right) + (1-p)(-C_1 - C_2 - F_d)\right] \tag{1}$$

$$U_{G1-x} = y(-R) + (1-y)(-C_2 - F_d) \tag{2}$$

The average expected revenue of the public rental housing authority is therefore:

$$\overline{U_G} = xU_{Gx} + (1-x)U_{G1-x} \tag{3}$$

The probability change rate of supervision and inspection conducted by the public rental housing authority is d_x/d_t. The evolutionary dynamic equation of the payoff of public rental housing authority group can be written and simplified as follow:

$$F_G(x) = \frac{d_x}{d_t} = x(U_{Gx} - \overline{U_G}) = x(1-x)(U_{Gx} - U_{G1-x}) \tag{4}$$

On the other hand, the expected income for the lessee when it chooses the active exit strategy and the delayed exit strategy is, respectively:

$$U_{Ty} = xR_3 + (1-x)R_3 \tag{5}$$

$$U_{T1-y} = x[p(-F_1) + (1-p)R_1] + (1-x)R_1 \tag{6}$$

The average expected revenue of the lessee group can then be obtained and simplified as:

$$\overline{U_T} = yU_{Ty} + (1-y)U_{T1-y} \tag{7}$$

Table 2. Public rental housing authority [7].

	Supervision (x)	Slack Supervision (1-x)
Voluntary Withdrawal (y)	$R_{3'}$ - C_1 - R_3	$R_{3'}$ - R_3
Delay Out Of (1-y)	p(- F1) + (1 - p)R1 p(R2 + F1 + Fg - C1) + (1-p) (- C1 - C2 - Fd)	$R_{1'}$ - C_2 - F_d

The rate of change of the probability that the lessee chooses to quit actively is d_y/d_t, and the evolutionary dynamic equation of the lessee group can be obtained as follows:

$$F_T(y) = \frac{d_y}{d_t} = y(U_{Ty} - \overline{U_T}) = y(1-y)(U_{Ty} - U_{T1-y}) \tag{8}$$

The total payoff can be represented in the Table 2:

We can then implement a trained neural network to acquire the value of the parameters C_1, C_2, R_1, R_2, R_3, F_1, P, F_g, F_d by inputting the results of all four possibilities of the decisions, and model the evolutionary process with the precise parameters.

4.2 Extended Application on Public Goods Game Problems

In the reality, there are not only games between two parties, but games among multiple players. A neural network can be implemented as long as the parameters are finite and there is enough database for the network to learn. The public goods game is a model of a multiplayer evolutionary game. Say there is a group of people (n) and a pool in the middle of the group. Each individual in the group has the option to put one unit of currency into the pool. They have two choices: they can choose to donate money, which means to cooperate (C), or they can choose not to donate money, which means to betray (D). No matter what choice they make, eventually, all the currencies in the pool will be multiplied by a coefficient of coordination greater than one and divided equally among all the individuals in the group. If n is used to represent the number of cooperating individuals, then their payoff formula is as follows:

$$\pi_c = \frac{rn_c}{n} - 1 \tag{9}$$

$$\pi_d = \frac{rn_c}{n} \tag{10}$$

Where π_c and π_d represent the payoffs of cooperators and betrayers, respectively. As can be seen from the above formula, The defector can gain the most money, so the defection strategy is the optimal strategy. It is saying that after a long period of time, the possibility for any play to choose to betray turns to be 1, where the evolutionary stability is reached. Using a trained neural network, we can derive a payoff matrix that considers the consequences of betrayal, the responses of other people, and all the possible influences, instead of mere consideration of the empirical value of the currency. It eventually leads to a precise analysis of the evolutionary process.

5 Conclusion

Our model is still theoretical because we have not found and used enough databases to train and test it. The thesis only proposes the feasibility of parameterization of evolutionary game problems. To validate our proposal, further adjustment should be made by testing and analyzing the sensitivity of the networks. So far, we are still adopting the most commonly used neural network in fitting large amount of figures: the FeedForward Network (FFN). One particular benefit is that FFN already has fully-developed and high-efficiency toolbox on major platforms, such as Matlab. In most cases, it produces result of enough accuracy. However, FFN occasionally needs to manually adjust in case the algorithm couldn't convergent. Solutions include using the LM training method, which might cost unacceptable space when conducted on large scale [8, 9]. Considering the fundamental thoughts of the evolutionary game it can be inferred that the Generative Adversarial Networks (GAN) which comes from confrontation game would be more appropriate [10]. Besides, interface should be added in further works to allow convenient edition of the training set. And a better way might be changing the structure of the neural network similar to the complex network of the game so that fewer parameters are required. With a mature network trained, we can further apply it into the cases mentioned in the application part. It can then come up with a result that requires further analysis and comparison with the result acquired via the conventional mathematical means to evaluate our model.

References

1. Picton, Phil, (1994) What is a neural network?. Macmillan Education UK.
2. Lawrence, Jeannette. (1993) Introduction to neural networks. California Scientific Software.
3. Friedman, Daniel. (1998) "On economic applications of evolutionary game theory." Journal of evolutionary economics 8: 15-43.
4. Weibull, Jörgen W. (1997) Evolutionary game theory. MIT press.
5. Sigmund, Karl, and Martin A. (1999) Nowak. "Evolutionary game theory." Current Biology 9.14.
6. Tanimoto, Jun. (2015) Fundamentals of evolutionary game theory and its applications. Springer Japan.
7. Siyu Liu. (2020) Research on The Supervision of Public Rental Housing Exit Problem Based on Evolutionary Game Theory. Chongqing University.
8. Sazli, Murat H. (2006) "A brief review of feed-forward neural networks." Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering.
9. Sanger, Terence D. (1989) "Optimal unsupervised learning in a single-layer linear feedforward neural network." Neural networks 2.6: 459-473.
10. Wang, Kunfeng. (2017) "Generative adversarial networks: introduction and outlook." IEEE/CAA Journal of Automatica Sinica 4.4: 588-598.