Research on Mathematical Genetic Algorithms Based on Evolutionary Computer Neural Network of Distribution Function Parameters

Yunna Han
Modern College of Northwest University, Xian 710130, China
Hanyunna@xaui.edu.cn

Abstract. Genetic algorithm and neural network are both ideas and algorithm skills. The flexibility of this method provides a stage for many researchers and applications. This paper discusses the neural network model based on the improved genetic algorithm under the parameters of the distribution function and its application in multi-variable, multi-step nonlinear economic forecasting. First of all, this article discusses the method of using improved genetic algorithm to learn neural network connection weights, and proposes a secondary optimization strategy. The main innovations or improvement techniques include: initial population generation strategy, crossover operation secondary optimization strategy, mutation operation Secondary optimization strategy, optimal chromosome inventory strategy, etc. This article uses VB for programming. The developed software has a certain versatility and can be used for prediction problems similar to nonlinear economic data.

Key words. Nonlinear economic forecasting, neural network, genetic algorithm, distributed function parameters.

1. Introduction
In daily life, people often encounter decision-making problems that require multiple goals to be as optimal as possible in a given feasible area. Such as the investment problem, people hope that the investment is the least, the risk is minimized and the investment return is the greatest. This kind of multiple numerical objective optimization problem in a given area is a multi-objective optimization problem (MOP). In reality, almost every important decision-making and prediction problem has to deal with several conflicting goals while considering different constraints. These problems involve the optimization of multiple goals [1]. These goals do not exist independently. They are often coupled in the goals that compete with each other and bind each other together have different physical meanings. One goal must be optimized at the cost of other goals, and the units of each goal are often inconsistent. Therefore, the multi-objective optimization problem is ultimately to find the values of a set of decision variables that not only satisfy the constraints and optimize the overall objective function, in which the elements that make up the overall objective function are sub-objective functions [2]. The traditional multi-objective optimization problem solving method is to aggregate each sub-objective into a weighted
single objective function, the coefficients are determined by the decision maker, and then the single objective optimization algorithm is used to solve the problem.

2. Nonlinear time series model

The various economic indicators describing the same economic problem are often related to a certain degree, so it is necessary to consider them as a whole. For multi-step forecasting, recursive forecasting is mostly used. Due to modelling errors, the forecasting errors are gradually superimposed. To this end, this article considers the following model. Suppose the prediction has r indicators, and the prediction is divided into k steps [3]. The value of the i-th indicator at time t is $y_{i}(t)$, and the corresponding random error term is $\epsilon_{i}(t)$; the excitation function of the i indicator at the j step is $f_{ij}(\cdot)$:

$$f_{ij}(\cdot) = f_{ij}(y_{1}(t), \ldots, y_{r}(t), \ldots, y_{r}(t-p+1))$$  \hspace{1cm} (1)

$f_{ij}(\cdot)$ is a nonlinear function and p is the order. The result of the k step prediction of the first indicator starting at time t corresponds to the equation:

$$\begin{bmatrix}
y_{1}(t+1) \\
y_{1}(t+k)
\end{bmatrix} = \begin{bmatrix}
f_{11}(\cdot) \\
f_{1k}(\cdot)
\end{bmatrix} \begin{bmatrix}
M \\
M
\end{bmatrix} + \begin{bmatrix}
\epsilon_{1}(t+1) \\
\epsilon_{1}(t+k)
\end{bmatrix}$$  \hspace{1cm} (2)

The neural network that predicts r indicators corresponds to the following equation:

$$\begin{bmatrix}
y_{1}(t+1) & y_{1}(t+k) \\
y_{r}(t+1) & y_{r}(t+k)
\end{bmatrix} = \begin{bmatrix}
f_{11}(\cdot) & f_{1k}(\cdot) \\
f_{r1}(\cdot) & f_{rk}(\cdot)
\end{bmatrix} \begin{bmatrix}
M & M \\
M & M
\end{bmatrix} + \begin{bmatrix}
\epsilon_{1}(t+1) & \epsilon_{1}(t+k) \\
\epsilon_{r}(t+1) & \epsilon_{r}(t+k)
\end{bmatrix}$$  \hspace{1cm} (3)

Obviously, equation (3) is the first component equation of the above equations. This article discusses the neural network prediction method based on equation (2).

3. Neural Network Model

The research of NN belongs to the category of artificial intelligence. NN is a simulation of the brain structure and the study of intelligent behaviour from a micro level. Therefore, NN is usually a large-scale network structure formed by a large number of artificial neuron cells connected to each other. Each artificial neuron cell (also called processing unit or network node) has only simple nonlinear processing capabilities, but through their interconnection and interaction, they can exhibit complex intelligent processing functions or nonlinearities Processing power [4]. The neural network is composed of simple units and is a nonlinear model structure for large-scale parallel information processing. The three-layer neural network can realize most nonlinear mapping. Figure 1 shows the neural network model.

![Figure 1. Neural network model](image-url)
Here, a three-layer feedforward neural network is used for system modelling. Let m, q, and k be the number of nodes in the input layer, hidden layer and output layer respectively, then the node outputs of the hidden layer and output layer are respectively

\[ u_s = f \left( \sum_{j=1}^{m} w_{sj} x_j + \theta_s \right) \quad s = 1, \Lambda , q \]  

(4)

\[ z_i = \sum_{s=1}^{q} v_{is} u_s + \varphi_i \quad i = 1, \Lambda , k \]

The excitation function is usually required to satisfy a monotonic increase and be continuous. The following non-linear function forms can be used:

- **Hard limit function:**
  \[ f(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \]  
  (5)

- **Linear limiting function:**
  \[ f(x) = \begin{cases} 0 & x < 0 \\ x & 0 \leq x < \beta \\ 1 & x \geq \beta \end{cases} \]  
  (6)

- **Sigmoid function**
  \[ f(x) = \frac{1}{1 + e^{-x}} \]  
  (7)

- **Hyperbolic tangent function (symmetrical Sigmoid function)**
  \[ f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \]  
  (8)

We chose the Sigmoid function, namely \( f(x) = 1/(1 + \exp(-x)) \).

### 4. Improved genetic algorithm for network weight learning

Here we discuss the learning method of network weights in the neural network model (4). Let us first discuss the problem of determining the weights at a given time assuming r, p, q, and k. Suppose the observed data is \((y_1(t),...,y_r(t)), t=1,...,n\). The sliding window can be used to select learning samples, that is, for time t, the input of model (4) is

\[ x_{(i-1)p+j} = y_i(t-j) \quad i = 1, \cdots, r \quad j = 0, \cdots, p-1 \]  

(9)

The expected output is

\[ z_i = y_i(t+i) \quad i = 1, \cdots, k \]  

(10)

We assume that the corresponding output when the weight is determined is \( \tilde{z}_i, i = 1, \cdots, k \), then the square error at this time is

\[ SE(t) = \sum_{i=1}^{k} (\tilde{z}_i - z_i)^2 \]  

(11)

Where \( z_i, \ i = 1, \cdots, k \) is determined by (4). Therefore, the weight learning problem can be reduced to the following optimization problem:

\[ \min_{w,j,\theta,\varphi} \sum_{t=p+1}^{n} SE(t) \]  

(12)

Among them
\[ W = \begin{bmatrix} w_{11} & \cdots & w_{1m} \\ M & \cdots & M \\ \vdots & \ddots & \vdots \\ w_{q1} & \cdots & w_{qm} \end{bmatrix} \quad V = \begin{bmatrix} v_{11} & \cdots & v_{1q} \\ M & \cdots & M \\ \vdots & \ddots & \vdots \\ v_{k1} & \cdots & v_{kq} \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_q \end{bmatrix} \quad \varphi = \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_k \end{bmatrix} \] (13)

It is a parameter to be determined. If the slow change of the structure is taken into account, it can be multiplied by the forgetting factor \( \lambda \), then the formula (13) can be reduced to

\[
\max_{W,V,\theta,\varphi} \mathbf{f}(W,V,\theta,\varphi) = -\sum_{t=0}^{T} \mathbf{SE}(t)
\] (14)

Where \( 0<\lambda<1 \). Next consider the genetic algorithm of optimization problem (14). The learning of neural network connection weights has the characteristics of too many parameters. In order to solve the problem of too long encoding caused by the high dimension of variables, we consider improving the genetic algorithm, that is, not binary encoding the variables, but directly using the parallel search mechanism of the genetic algorithm [5]. The algorithm steps are as follows:

A. Coding method. It can be formed by cascading various parameters (\( W, V, \theta, \varphi \)), and its total length is \( r p q + q k + q + k \).

B. Parameter setting. Determine the algorithm parameters, such as population size \( N \), crossover probability \( P_c \), mutation probability \( P_m \), genetic size \( 2d \) (\( 2d<N \)).

C. Initial group selection. \( t=0 \), select the initial population \( G_0 = \{g_1, \ldots, g_N\} \).

D. Set the \( t \)-th generation group \( G_t = \{g_1, \ldots, g_N\} \), calculate and adjust the fitness value as follows:

\[
F_i = f(g_i) - f_{\min} + (f_{\max} - f_{\min})/N
\]

\[
f_{\min} = \min \{f(g_j) \mid j=1, \ldots, N\} \quad f_{\max} = \max \{f(g_j) \mid j=1, \ldots, N\}
\] (15)

E. Define the following probability distribution:

\[
P(g_i) = F_i / \sum_{i=1}^{N} F_i \quad i=1, \ldots, N
\] (16)

F. Genetic algorithm. \( t=t+1 \), generate a new group according to the following steps:

(1) Copy operation. From the probability distribution (16), randomly select a sample vector with a capacity of \( 2d \), which may be recorded as \( g_1, \ldots, g_{2d} \).

(2) Cross operation. With probability \( P_c \), crossover operations are performed on \( g_{2k-1} \) and \( g_{2k} \). The crossover operations we propose are \( \alpha \cdot g_{2k-1} + (1-\alpha) \cdot g_{2k} \) and \( (1-\alpha) \cdot g_{2k-1} + \alpha \cdot g_{2k} \), where \( \alpha \) Take the random number in \( (0,1) \). The result after the crossover is still recorded as \( g_1, \ldots, g_{2d} \).

(3) Mutation operation. Use probability \( P_d \) to mutate \( g_k \) (\( k=1, \ldots, 2d \)) with respect to each component. If more than one mutation result is produced, select the optimal one among these vectors.

(4) The \( 2d \) new-generation individuals generated by the three operations together with the \( N-2d \) individuals with the best performance in the previous generation form a new group \( G_t = \{g_1, \ldots, g_N\} \).

G. As in step 4, calculate the fitness value of the group.

H. Whether the termination conditions are met, if not, return to step 5. If satisfied, end.

Initial group generation strategy. Each time, 6 individuals are completely randomly generated in the feasible solution domain, and the one with the largest fitness value is selected to join the initial population. This strategy not only improves the quality of the initial population, but also ensures the randomness of the initial population. Determination of model order. For (1), \( m=rp \). Therefore, for a specific model, the four parameters \( r, p, q, \) and \( m \) should be determined. If \( r \) and \( p \) are taken, then \( q \) takes \( 2rp+1 \). Therefore, \( r \) and \( p \) need to be determined. Cross operation strategy. In order to improve the convergence speed, 5 groups of 10 crossover results are randomly generated, and 2 optimal values are selected as the result. Stop conditions and optimal chromosome inventory [6]. Each time you set the number of learning times according to your available time, and then save the best learning results in a text file, that is, the best chromosome inventory. Next time you can choose to restart learning or continue learning with the last result. When replacing the content in the best chromosome inventory, the fitness value is automatically compared to avoid the loss of the best result.
5. Application examples

The thesis uses the price index to conduct the overall analysis of the real estate market and has many years of experience in my country. The price index is composed of data from market surveys. These data come from real estates in different locations [7]. They record the trajectory of market fluctuations at all times, forming a dynamic picture of market observations. Quantitatively studying the trajectory of the price index and making accurate descriptions and predictions are extremely important for studying the real estate market.

Here, the method of this paper is used to model the data of China Real Estate Composite Index and Second-hand Housing Index from January 2015 to October 2017. First of all, we consider the issue of data stabilization and take the ring-on-month increase rate [8]. Then do normalization in the software. Table 1 shows the actual predicted value of the China Housing Composite Index. It can be seen from Table 1 that the actual forecast error rates for November and December 2017 were 0.94%, 0.28%, and 0.19%, respectively, and the effect was good.

Table 1. Composite index and second-hand housing index forecast results in November and December 2017

|                                | 2017  | 2018  |
|--------------------------------|-------|-------|
| China Housing Composite Index  | 1859  | 1925  |
| One-step forecast of the rising rate of the composite index | 4.85% | 3.55% |
| China Real Estate Composite Index One Step Forecast | 1841.44 |
| China Housing Composite Index One-step Forecast Error Rate | 0.94% |
| Two-step forecast of the rising rate of the composite index | 4.56% | 3.75% |
| Two-step forecast of the rising rate of the composite index | 1853.85 | 1928.71 |
| Two-step prediction error rate of China Housing Composite Index | 0.28% | 0.19% |

6. Conclusion

Multi-objective optimization has always been a big problem in the optimization world. The research on this problem has always been challenging and attractive. However, the study of multi-objective optimization has always lacked an efficient and practical solution method. The BP neural network algorithm based on genetic algorithm proposed in this paper not only utilizes the characteristics of BP neural network to complete arbitrary air-to-sound mapping through sample learning, but also makes use of genetic algorithm to achieve globalization for complex, nonlinear and non-differentiable functions. The advantages of search have achieved good results in solving multi-objective optimization problems.

References

[1] Ding, S., Su, C., & Yu, J. An optimizing BP neural network algorithm based on genetic algorithm. Artificial intelligence review, 36(2) (2011) 153-162.
[2] Che, Z. G., Chiang, T. A., & Che, Z. H. Feed-forward neural networks training: a comparison between genetic algorithm and back-propagation learning algorithm. International journal of innovative computing, information and control, 7(10) (2011) 5839-5850.
[3] Oreski, S., Oreski, D., & Oreski, G. Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. Expert systems with applications, 39(16) (2012) 12605-12617.
[4] Irani, R., & Nasimi, R. Evolving neural network using real coded genetic algorithm for permeability estimation of the reservoir. Expert Systems with Applications, 38(8) (2011) 9862-9866.
[5] Ahmadi, M. A., Zendehboudi, S., Lohi, A., Elkamel, A., & Chatzis, I. Reservoir permeability prediction by neural networks combined with hybrid genetic algorithm and particle swarm optimization. Geophysical Prospecting, 61(3) (2013) 582-598.
[6] Huang, D. Z., Gong, R. X., & Gong, S. Prediction of wind power by chaos and BP artificial neural
networks approach based on genetic algorithm. Journal of electrical engineering and technology, 10(1) (2015) 41-46.

[7] Fernandez, M., Caballero, J., Fernandez, L., & Sarai, A. Genetic algorithm optimization in drug design QSAR: Bayesian-regularized genetic neural networks (BRGNN) and genetic algorithm-optimized support vectors machines (GA-SVM). Molecular diversity, 15(1) (2011) 269-289.

[8] Qdais, H. A., Hani, K. B., & Shatnawi, N. Modeling and optimization of biogas production from a waste digester using artificial neural network and genetic algorithm. Resources, Conservation and Recycling, 54(6) (2010) 359-363.