Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 10, No. 1, 2023

Research on Improved BP Neural Network Prediction Based on Niche Particle Swarm Optimization Algorithm

Shengjie Jin*, Chunguang Zhang

Dalian Jiaotong University, Dalian 116000, China

* **Corresponding author**: Shengjie Jin (Email: jsj0419@163.com)

Abstract: As for data prediction, a prediction model is proposed based on BP neural network. This method is based on the traditional BP neural network algorithm to improve the problems of slow convergence speed and easy to fall into local optimum. The idea of niche is introduced into the particle swarm optimization algorithm to improve the global search ability and convergence speed, find the global optimum value, and then use LMBP neural network to continue to optimize the parameters to improve the prediction effect.

Keywords: Data prediction; BP neural network; Niche; Particle swarm optimization.

1. Introduction

The field of data prediction can be described as all-encompassing, including weather, earthquake, stock market, price, user psychology, transportation, etc. The time span has existed since ancient times. However, with the development of modernization, the prediction method has been continuously upgraded, and the prediction accuracy has also been continuously improved. Human beings have been trying to break through the limitations and see the future. Now, the development of big data technology makes the future come to us earlier.

The prediction of data has existed since ancient times. Through some factors and signs, we can predict the future development trend. Now, due to the continuous updating of technical means and the accumulation of massive data in various fields, the method of data prediction is also increasingly strengthened.

In the field of data prediction, there have been many achievements in the application of BP neural network algorithm. Researchers at home and abroad have applied BP neural network algorithm to solve data prediction problems in other fields, enriching the solution algorithm of data prediction, and expanding the application field of BP neural network algorithm. However, when the data scale is large, the update speed is fast, and the complexity of the problem increases sharply with the increase of scale and update, it is necessary to apply a more efficient, stable, and easy-to-implement new solution algorithm to the data prediction problem. BP neural network has been studied and verified to have better prediction performance and versatility in prediction problems, and in data prediction problems, some scholars have begun to focus on applying particle swarm algorithm to solve data prediction problems. These studies have verified the feasibility and effectiveness of particle swarm algorithm in this field, and provided a new way to solve such problems. Therefore, this paper applies BP neural network algorithm to solve data prediction problems, In order to improve the algorithm performance and model of BP neural network, this paper proposes a hybrid BP neural network algorithm, which introduces the shared particles in particle swarm to adjust the weight and threshold of adaptive BP neural network.

2. BP neural network

BP neural network (BPNN) is a simplified biological model and one of the most widely used and studied neural network models at present. It is composed of multi-layer neural networks, in which the input layer and output layer are composed of single-layer networks, but the hidden layer can be single-layer or multi-layer networks. Each layer of neural network is composed of multiple neurons, and each neuron is a perceptron, the neurons in each layer are interconnected, and the connection mode is full connection. In general, the structure of BP neural network is that after the input layer gets the stimulation, it will be transmitted to the hidden layer. As for the hidden layer, it will transmit the stimulation to the output layer according to the weight of the neurons' interconnection and the rules. The output layer will compare the results. If it is not correct, it will return the weight value of the neurons' interconnection to be modified, which is the so-called feedforward multilayer network algorithm based on the error back propagation training method.

On the basis of automatic learning, the back-propagation error uses gradient descent optimization algorithm to adjust the weight loss, and then adjust the output. The following figure 2.1 shows the topological structure of BP neural network. At the same time, BP neural network is called a supervised learning algorithm, because the goal of any supervised learning algorithm is to find a function, and the best input gets the most correct output. The back-propagation of BP neural network is to train a multi-level neural network, so that proper learning and adjustment can be carried out internally, so that any input can be mapped to the output, and finally the minimum error can be obtained.

Assume that the input layer has m nodes, the hidden layer has n nodes, and the output layer has a single node. Record the weight matrix from the input layer to the hidden layer as W, the weight matrix from the hidden layer to the output layer as V, the threshold matrix of the hidden layer as A, and the threshold value of the output layer as b, that is, as follows:

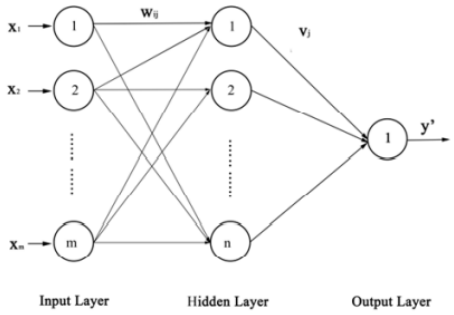


Fig.1 BP neural network topology

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix}$$

$$V = [v_1\ v_2\ \cdots\ v_n]$$

$$A = [a_1\ a_2\ \cdots\ a_n]^T$$

The calculation formula of BP neural network weight is:

$$W_{mn}(n_0 + 1) = W_{mn}(n_0) + W_{mn} \quad (1)$$

The expression of the correction amount of the weight value is:

$$Vw_{jk} = \eta\delta_k^0 y_j = \eta(d_k - o_k)o_k(1 - o_k)y_j \quad (2)$$

$$Vv_{jk} = \eta\delta_j^y x_j = \eta(\sum_{k=1}^{1}\delta_j^y w_{jk})y_j(1 - y_j)x_j \quad (3)$$

In this network model, train a set of sample sets (X, y), which are input values $X(x_1, x_2, \cdots, x_m)$ and output values y. When the function $f(x)$ is activated, the network output value y' obtained in forward propagation is as follows:

$$y' = \sum_{j=1}^{v} f\left(\sum_{i=1}^{u} w_{ij}\, x_i - a_j\right)w_{j1} - b, j = 1,2,\ldots,n \quad (4)$$

In the above formula (2-4), w_{ij} is the weight vector from the i input node to the j output node, v_j is the weight vector from the j hidden layer node to the output layer, a_j is the threshold value of the hidden layer node, and b is the threshold value of the output layer.

If the output layer of the network fails to get the expected output, it will turn to reverse propagation and adjust the weight and threshold of each neuron according to the prediction error. The error value is:

$$e = \frac{1}{2}(y - y')^2 \quad (5)$$

According to the above formula (2-5), the error is actually a function of weight and threshold. The goal of BP algorithm is to obtain the minimum square error value, that is, find the global minimum weight threshold group, and then get the desired output.

The algorithm flow of BP neural network is as follows:

Step 1: normalize the sample data;

Step 2: initialization of BP neural network, setting the initial weight value as a random non-zero value;

Step 3: Provide a matrix of input and output data in the required format for training;

Step 4: calculate the input value and output value of each neuron in each hidden layer;

Step 5: calculate the input value and output value of the output layer;

Step 6: Calculate the error of output layer and hidden layer;

Step 7: Modify the weight of output layer and hidden layer;

Step 8: Return to Step 4 and cycle the above steps until the error meets the set precision.

3. NPSO-LMBP neural network

From the analysis in the previous section, it can be found that BP neural network, to a certain extent, mimics the working mode of neurons from a physiological point of view, and can arbitrarily approach nonlinear complex functions. However, it is easy to fall into over-application, resulting in its slow convergence rate, and thus the performance of the prediction model does not meet expectations. How to accelerate the convergence speed of BP neural network in the training process and avoid falling into the local optimal value when optimizing the weights and thresholds of self-adaptable BP neural network are the problems that need to be solved in the optimization of BP neural network algorithm. The improved particle swarm optimization algorithm has fast convergence speed to avoid falling into the local optimal value. Therefore, this section combines the improved particle swarm optimization algorithm and LM algorithm, and proposes a hybrid adaptive BP neural network algorithm. The improved particle swarm optimization algorithm and LMBP neural network algorithm are mixed by taking advantage of the characteristics of simple implementation and fast convergence speed of particle swarm optimization algorithm. First, the particle of particle swarm optimization is used to represent all the connection weights and thresholds between neurons, and then the improved particle swarm optimization algorithm is used to find the global optimal value, that is, the optimal initial weight and threshold, so as to avoid the disadvantage of the self-adapted BP neural network falling into the local optimal value when optimizing the weights and thresholds, Then use LM algorithm to adjust the weight and threshold of BP neural network, and finally use self-adaptation BP neural network to train the test data to get the final prediction result.

3.1. LMBP neural network

LM (Levenberg-Marquardt) algorithm, also known as damped least square method, is an improved method based on standard numerical optimization. This algorithm is a combination of gradient descent method and Newton method. Using the approximate second derivative information, it has both the global characteristics of gradient descent method and the fast convergence of Gauss-Newton method.

The standard BP algorithm obtains the minimum square error value according to the gradient descent method. The square error value is the objective function that the LM algorithm needs to optimize. Use LM algorithm to adjust the weight and threshold of BP neural network, and the formula is as follows:

$$\Delta W_k = W_{k+1} - W_k = -[J^T(W_k)J(W_k) + \alpha_k I]^{-1}J^T(W_k)e(W_k) \quad (6)$$

In the above formula, W_k is the iterative weight vector; α_k is the constant (adjustment factor) greater than zero in the LM algorithm, which is used to control the iteration of the LM algorithm; I is the unit matrix; $J(W_k)$ is the Jacobian matrix of the weight differential of the error.

In the process of network training, with the increasing number of iterations, when α_k closes to zero, the LM algorithm is close to the Gauss-Newton method. Because of its second-order convergence speed, it converges faster than the BP algorithm based on the gradient descent method. And it has the advantages of faster calculation speed and higher accuracy when the error is closer to the minimum value.

3.2. Niche particle swarm optimization

The particle swarm optimization algorithm simulates the behavior of bird predation. Each bird represents a particle, and the food represents the optimal solution. The process of bird predation is the process of finding the optimal solution. In the process of bird predation, information is shared through the sharing mechanism, and the next update of position information and speed information is determined through fitness sharing. The fitness value of the current particle position is calculated by the objective function, The fitness value can be abstractly understood as the distance from "food". In each particle movement, the particles refer to their own historical position, but also according to the optimal particle position in the population, so that we can know how the next movement tends to the optimal position. The specific speed and position formula is as follows:

$$v_{id}(t+1) = w * v_{td}(t) + c_1 * rand() * (P_{id}(t) - X_{id}(t)) + c_2 * rand() * (P_{id}(t) - X_{id}(t)) \quad (7)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), 1 \le i \le N, 1 \le d \le M \quad (8)$$

In formulas (3-2) and (3-3), w is inertia weight; c_1 and c_2 are the learning factors, which are generally taken as a random number around 2.5; $rand$ is a random number between [0,1]; $P_i = (P_{i1}, P_{i2}, K, P_{id})$ is the historical optimal position $pbest$ of the i particle; All the optimal individuals $P_i = (i = 1,2,K,\cdots,N)$ are recorded as p_g, the global optimal positions $gbest$.

Particle Swarm Optimization (PSO) has become a research hotspot of many scholars at home and abroad because of its simple concept and easy implementation. However, there are still some shortcomings in PSO at present, such as weak theoretical basis of the algorithm and insufficient exploration ability in the later stage. In view of the above problems, we use niche technology for reference and introduce sharing mechanism into PSO.

The niche particle swarm algorithm is to divide the whole particle swarm into several types of niche particle swarm. Each particle can find its own niche according to the location information. The information between particles in each niche is shared, but different from the particles in other niches, the local optimal value is shared between each niche through the sharing mechanism, and the position and speed of particles are updated by comparison until the global optimal value is found at the end of the cycle, The following is the population division, sharing function and applicable value update function of niche particle swarm optimization.

(1) Division of niche groups

The niche population is divided by calculating the distance between particles $X_i = (x_{i1}, x_{i2}, \cdots, x_{im}), i = 1,2,\cdots,N.$

$$d_{ik} = \|X_i - X_k\|, k = 1,2,\cdots,N \quad (9)$$

For a given boundary partition parameter σ_0, if $d_{ik} < \sigma_0 (k = 1,2,\cdots,N)$, the individual joins the niche population X_{pi}, and the distance between particles is generally taken as Euclidean distance. Euclidean distance is also used in this chapter.

(2) Sharing function between particles

The sharing function between particles i and j is:

$$sh(d_i) = \begin{cases} 1 - (\frac{d_{ij}}{\sigma_{share}})^{\alpha}, d_{ij} < \sigma_{share} \\ 0, \quad \text{others} \end{cases} \quad (10)$$

In the above formula, α is the parameter that controls the shape of the shared function.

(3) Update of fitness function

Update the fitness of particles according to the sharing mechanism function:

$$f_{share}(i) = \frac{f_{original(i)}}{m} \quad (11)$$

$$m = \sum_{j=1}^{N} sh(d_{ij}), j = i, i+1, \cdots, i+p-1 \quad (12)$$

The flow of niche particle swarm optimization algorithm is as follows:

Step1: Initialize the particle population as $X = \{X_1, X_2, \cdots, X_N\}$, where each individual particle is $X_i = (x_i, x_2, \cdots, x_m), i = 1,2,\cdots,N.$

Step 2: Determine the individuals meeting the niche population according to the following steps:

1. Calculate the Euclidean distance of two particle individuals according to formula (2-4);

2. According to $d_{ij} < \sigma_{share}$, the niche particle group is determined as x_{pi}, P_i is the number of elements x_{pi}. σ_{share} depends on the actual situation, and is initially set as 1/16 of the limit.

Step 3: update the position and speed of particles in the divided niche groups;

1. Set the acceleration factor c_1、c_2, and the value of the maximum number of iterations. The current number of iterations is set to 1. Set the optimal particle size as the current particle initialization position, and complete the speed initialization of each particle at the same time.

2. Update the position and velocity of particles according to formula (3-2) and formula (3-3). The population optimum is the niche individual optimum, not the whole population optimum;

3. Check the conditions. For the updated individual x'_{pi}, update the speed and position of the j particle in the formula (3-5), and update the individual fitness.

Step 4: Keep the optimal fitness value and individual, check whether the optimization conditions are met, and if the error accuracy is reached, the end. Otherwise, then enter the niche population of the next particle for optimization;

Step 5: When the optimal value of each niche group is still not found after optimization, concentrate the optimal value of each niche group into a new space, and continue to search from Step 2.

To sum up, the particle swarm is divided into multiple niche groups by the distance between particles, so that each particle can find its own niche. Then the particle swarm algorithm is applied to each niche group to update the position and speed of each particle. Finally, the applicable value of each updated niche group is updated using the sharing function.

3.3. NPSO-LMBP neural network

The advantage of particle swarm optimization algorithm is that particles have the ability to expand the search space. The update of solution depends on particle random walk. Its random walk has learning factors and acceleration coefficients. It can coordinate cognitive and social information and accelerate the convergence speed by adjusting parameters. The improved particle swarm optimization algorithm introduces the idea of niche to share the local optimal value among particles and find the global optimal solution, thus avoiding falling into the local optimal.

In this section, combining the excellent mechanism of niche particle swarm optimization algorithm and LMBP neural network algorithm, a hybrid BP neural network algorithm (NPSO-LMBP) is proposed to improve the global search ability and convergence speed. In this algorithm, the topological structure of the neural network is determined

according to the input and output variables of the neural network and the number of neurons in the hidden layer. The particles of particle swarm are used to represent all the connection weights and thresholds between neurons. The niche particle swarm algorithm is used to find the optimal value, that is, the optimal initial weight and threshold, so as to avoid the defect of BP neural network falling into local optimum when optimizing the weights and thresholds, Then the LM algorithm is used to continue to optimize the parameters of BP neural network, and the final result is obtained.

The calculation process of NPSO-LMBP prediction model is as follows:

Step 1: Determine the network structure of BP neural network. The topological structure of the neural network is determined according to the input and output variables of the neural network and the number of neurons in the hidden layer. The particles of the particle swarm are used to represent all the connection weights and thresholds between neurons. The number of network weights is determined to be $n*m+m*q$ and thresholds is determined to be $m+q$, so the total number of particles is $n*m+m*q+m+q$.

Step 3: Initialize particle swarm speed, position, inertia factor w, learning factor c_1、c_2, maximum number of iterations, neural network learning rate η, and target accuracy g.

Step 4: Calculate particle fitness value. Generally, the mean square error is selected as the criterion of fitness evaluation, and the initial weight and threshold are brought in. The fitness function obtained is as follows:

$$f(R_i(t)) = \frac{1}{RMSE(w_{ij},v_j,a_i,b)} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(y_i-y_i')^2} \quad (13)$$

In formula (3-8), n is the number of training samples, y_i is the expected output value, and y_i' is the output value of the network. Find a group (W,V,A,b) in the training process to make $RMSE$ minimum.

Step 5: Find the global optimum of the whole group. According to the flow of niche particle swarm optimization algorithm, the global optimization is obtained.

Step 6: Output the best particles. The particles that meet the requirements are fed into the LMBP neural network model, and the LMBP algorithm continues to adjust the weights and thresholds internally to get the best results.

Step 7: Train NPSO-LMBP neural network and predict wheel set size data.

Step 8: Output the prediction results, compare them with the actual values, and evaluate the prediction effect.

4. Experimental analysis of data prediction based on NPSO-LMBP algorithm

4.1. Prediction result evaluation index

In this paper, mean absolute error (MAE), mean square error (MSE), mean absolute percentage error (MAPE) and correlation coefficient R are selected as model evaluation indicators.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - y_i'| \quad (14)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - y_i')^2 \quad (15)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} |(y_i - y_i')/y_i| \quad (16)$$

$$R = \frac{\sum_{i=1}^{n}(y_i-\overline{y})(y_i'-\overline{y'})}{\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2 \sum_{i=1}^{n}(y_i'-\overline{y'})^2}} \quad (17)$$

In the above formula, y_i is the expected output value, y_i' is the network output value, $\overline{y}$ and $\overline{y'}$ is the average of the real value and the network output value respectively, and n is the number of prediction samples.

Generally speaking, the smaller the values of MAE, RMSE and MAPE, the higher the accuracy of the model prediction; The closer the value of R is to 1, the higher the prediction accuracy of the model is.

4.2. Simulation data prediction experiment analysis

In order to test the prediction performance of NPSO-LMBP model, a simulation experiment is designed. In this section, noise is added to a function to realize numerical randomization, and 300 groups of data are selected, of which 270 groups are used as training sets and 30 groups are used as test sets for algorithm experiments. Compare the results of this algorithm with those of traditional BP algorithm. The experiment was carried out based on the MATLAB 2020 version and was completed in the AMD Ryzen5 4600U processor, 16Gb memory and 64-bit Windows10 environment. The experimental results are shown in the figure below.

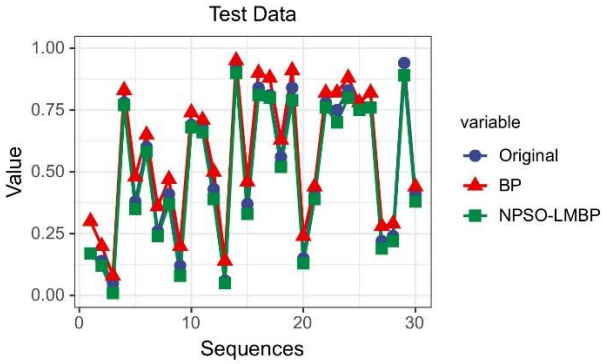


Fig.2 Simulation results of prediction algorithm

It can be seen from the figure that the prediction data of BP neural network fluctuates greatly, with poor prediction effect and low accuracy. The improved NPSO-LMBP algorithm has a good prediction effect and is more consistent with the real data.

The evaluation index values are shown in the following table:

Table 1. Prediction model evaluation index

	Algorithm	MSE	MAE	MAPE	R
Data	BP	0.235	0.323	0.212	0.897
	NPSO-LMBP	0.098	0.134	0.113	0.954

As shown in Table 4.1, the four indicators of the NPSO-LMBP prediction model are better than the traditional BP model, which verifies the good prediction performance of the model.

5. Conclusions

Aiming at the problem of data prediction algorithm, this paper analyzes the characteristics and shortcomings of the traditional BP neural network algorithm, proposes an improvement to the shortcomings of the algorithm, and proposes a hybrid BP neural network algorithm, which uses the particles of particle swarm to represent all the connection weights and threshold values between neurons, and optimizes the weights and thresholds to improve the convergence speed

and avoid falling into local optimization, The improved particle swarm optimization algorithm and LMBP neural network are applied to the data prediction model, and the reliability of the algorithm model is verified by simulation experiments, and the algorithm has better performance than the traditional algorithm.

References

[1] Wang S , Zhu H , Wu M , et al. Active Disturbance Rejection Decoupling Control for Three-Degree-of- Freedom Six-Pole Active Magnetic Bearing Based on BP Neural Network[J]. IEEE Transactions on Applied Superconductivity, 30.

[2] Liu X , Zhou Y , Wang Z , et al. A BP Neural Network-based Communication Blind Signal Detection Mefthod with Cyber-Physical-Social Systems[J]. IEEE Access, 2018:1-1.

[3] Yang A , Zhuansun Y , Liu C , et al. Design of Intrusion Detection System for Internet of Things Based on Improved BP Neural Network[J]. IEEE Access, 2019, 7:106043-106052.

[4] Xu L , Wang H , Lin W , et al. GWO-BP Neural Network Based OP Performance Prediction for Mobile Multiuser Communication Networks[J]. IEEE Access, 2019, 7:1-1.

[5] Pan X , Zhou W , Lu Y , et al. Prediction of Network Traffic of Smart Cities Based on DE-BP Neural Network[J]. IEEE Access, 2019, 7:55807-55816.

[6] Liu F , Huo W , Han Y , et al. Study on Network Security Based on PCA and BP Neural Network under Green Communication[J]. IEEE Access, 2020, PP(99):1-1.

[7] Li H X , Zhao X H , Chi H Y , et al. Prediction and analysis of land subsidence based on improved BP neural network model[J]. Journal of Tianjin University, 2009, 42(1):60-64.

[8] Zhang Y , Bing C , Yuan Z , et al. Wind Speed Prediction of IPSO-BP Neural Network Based on Lorenz Disturbance[J]. IEEE Access, 2018, PP(99):1-1.