Research on Traffic Flow Forecasting Method based on Optimized BP Neural Network

Jun Ding

Chinese University of Hong Kong, Shenzhen, China

Keywords: BP neural network; particle swarm optimization; mutation operator; traffic flow prediction.

Abstract: In order to improve the prediction accuracy of BP neural network prediction model, a prediction method based on Improved Particle Swarm Optimization BP neural network is proposed. The adaptive mutation operator is introduced to mutate the particles trapped in the local optimum, and the optimization performance of the particle swarm optimization algorithm is improved. The weight and threshold of BP neural network are optimized by improved particle swarm optimization algorithm. Then BP neural network prediction is trained. The model obtains the optimal solution. The forecasting method is applied to the time series of measured traffic flow to validate its validity. The results show that the method has better non-linear fitting ability and higher forecasting accuracy for short-term traffic flow.

1. Introduction

Real-time and accurate traffic flow forecasting is the premise and key of traffic control and traffic guidance, and its research has always been a hot topic in ITS. Over the years, experts and scholars have established many prediction models and methods, such as historical average method, time series method, Kalman filter method, regression analysis method and so on. [1] These methods have matured theoretical basis and many applications. But most of the traditional prediction methods are based on mathematical statistics, and the common feature is to establish the subjective model of data sequence first, and then according to the subjective model. In recent years, many scholars have made a thorough study on the non-linear characteristics of urban traffic flow system [2-3], and established a variety of traffic flow prediction models considering non-linear traffic flow, such as BP neural network model [4-5], RBF neural network model [6], Volterra filter adaptive prediction model [7-8] and so on. Among them, BP neural network model is a relatively successful prediction model. But this model has two obvious shortcomings: one is easy to fall into local minimum; the other is slow convergence speed. To overcome these shortcomings, this paper uses BP neural network prediction model optimized by improved particle swarm optimization (PSO) to apply in short-term traffic flow forecasting.

2. BP Neural Network Prediction Model

The essence of time series prediction is the inverse problem of a dynamic system. The dynamic model F (•) of the system is reconstructed by the state of the dynamic system.

$$F(X_{\mathrm{i}}) = x_i+T \ (\mathrm{T} > 0) \tag{1}$$

In the formula, T is the forward prediction step.

There are numerous methods to construct a non-linear function f (•) to approximate F (•). BP neural network is a good method to construct the non-linear prediction model F (•) of time series.

If the input of a nonlinear discrete dynamic system is $Xi= (x_i, x_{i+T}, \ldots, x_{i+(m-1)T})^{\mathrm{T}}$ and the output is $y_i=x_i+ 1$, a typical three-layer BP neural network is selected. Because BP neural network is used to predict time series, when the number of neurons in the input layer of the neural network is equal to the embedding dimension m of the reconstructed phase space of time series, the prediction effect is better [9]. When the number of inputs of BP neural network is m, the hidden layer is p and the number of outputs is 1, the BP neural network completes the mapping $f: R^m \rightarrow R^1$, and the input of each node in the hidden layer is

$$S_j = \sum_{i=1}^{m} w_{ij} x_i - \theta_j,\ j=1, 2, \ldots, P \tag{2}$$

In the formula, w_{ij} is the connection weight from input layer to hidden layer, and θj is the threshold value of hidden layer nodes.

The transfer function of BP neural network adopts Sigmoid function $f(x) = 1/(1+e^{-x})$, and the output of hidden layer node is

$$B_j = \frac{1}{1 + exp(-\sum_{i=1}^{m} w_{ij} x_i + \theta_j)},\ j=1, 2, \ldots, P \tag{3}$$

Similarly, the input, output and output of the output layer node are:

$$L = \sum_{j=1}^{p} u_j b_j - \gamma \tag{4}$$

$$x_i + 1 = \frac{1}{1+\exp(-\sum_{j=1}^{p} u_j b_j + \gamma)} \tag{5}$$

In formula, u_j is the connection weight from hidden layer to output layer; γ is the threshold of output layer.

The connection weights w_{ij}, u_j and threshold θ_j of BP neural network can be obtained through training of BP neural network, so x_i+1 is predictable. Formula (5) is the prediction model of BP neural network.

Before training, the connection weights and thresholds of each layer are randomly initialized to the values between [0,1]. This kind of non-optimized random initialization will slow the convergence speed of BP neural network and make the final result non-optimal. The initial weights and thresholds can be optimized by using improved particle swarm optimization algorithm. The optimized initial weights and thresholds can make the BP neural network converge faster and the final result closer to the optimal solution.

3. Improvements of PSO Algorithms

Particle swarm optimization (PSO) [10] is a global random search algorithm based on swarm intelligence, which was proposed by Kennedy and Eberhart in 1995, inspired by the research results of artificial life. It simulates the migration and clustering behavior of birds in the process of foraging. In this algorithm, all candidate solutions of optimization problems are the states of a particle in the search space, and each particle corresponds to a routing item. The fitness value determined by the scalar function and the speed of particles determine the direction and distance of their flight. Particles are dynamically adjusted according to their own flight experience and that of their companions, i.e. the optimal solution found by the particles themselves and the optimal solution currently found by the whole population. Thus, they search continuously for the solution space until they meet the requirements.

3.1 Basic PSO Algorithms

In a S-dimensional search space, the population $W = (W_1, W_2, \ldots, W_n)$ consisting of n particles. The i^{th} particle is represented as a S-dimensional vector $W_i=(w_{i1}, w_{i2}, \ldots, w_{iS})^T$, which represents the position of the i^{th} particle in the S-dimensional search space and represents a potential solution of a problem. The fitness value corresponding to position of each particle can be calculated according to the objective function. The velocity of the i^{th} particle is recorded as $V_i = \{V_{i1}, V_{i2}, \ldots, V_{iS})^T$, whose individual extremum is $P_i=(P_{i1}, P_{i2}, \ldots, P_{iS})^T$, and the global extremum of population is $P_g=(P_{g1}, P_{g2}, \ldots, P_{gs})^T$.

In each iteration, the particle updates its velocity and position through individual and global extremums, and then the model is updated.

$$V_{id}^{k+1} = \omega V_{id}^{k} + c_1 r_1 (P_{id}^{k} - W_{id}^{k}) + c_2 r_2 (P_{gd}^{k} - W_{gd}^{k}) \tag{6}$$

$$W_{id}^{k+1} = W_{id}^{k} + V_{id}^{k+1} \tag{7}$$

In the formula, ω is the inertia weight; d = 1,2, ..., S; i = 1,2, ..., n; k is the current iteration number; V_{id} is the velocity of particles; c_1 and c_2 are non-negative constants, called acceleration factors; r_1 and r_2 are random numbers distributed between [0.1].

3.2 Improvement of PSO Algorithms

Because the particles in PSO aggregate to their own historical best position and neighborhood or group historical best position, the rapid convergence effect of the particle population is formed, and it is prone to fall into local extreme value, premature convergence or stagnation [11]. In order to overcome the above shortcomings, this paper introduces adaptive mutation operator in the PSO algorithm, that is to say, some variables are reappeared with certain probability by using the mutation idea of genetic algorithm. The mutation operation expands the searching space of the population which is shrinking continuously in the iteration, so that the particle can jump out of the optimal position previously searched and search in a larger space. At the same time, it keeps the diversity of the population and improves the possibility of finding a better value by the optimization of the algorithm.

4. Optimizing BP Neural Network Prediction Algorithms by Improving PSO Algorithms

The basic steps of improving PSO algorithm to optimize BP neural network prediction algorithm are as follows.

Step 1: Initialize parameters including population size, number of iterations, learning factor and the limited interval of position and velocity. Assign random values to particle position and velocity.

Step 2: Construct BP neural network topological structure according to the number of input and output parameters of time series, and randomly generate a population particle $W_i=(w_{i1},w_{i2},\ldots,w_{iS})^T$, i = 1, 2, ..., n, which represents the initial value of BP neural network.

$$S = RS_1+S_1S_2 + S_1+S_2 \tag{8}$$

In the formula, R, S_1 and S_2 are the input layer node, hidden layer node and output layer node of BP neural network respectively.

Step 3: Determine the evaluation function of the particle. Given an evolutionary parameter of BP neural network, the weights and thresholds of BP neural network are assigned by the particle W_i obtained in Step 2, and the training samples are input for the training of BP neural network. The output value $\hat{y}_i$ of network training is obtained if it lies within the accuracy range. Then the fitness fit_i of individual W_i in population W is defined as follows:

$$fit_i = \sum_{J=1}^{M-1} (\hat{y}_j - y_j)^2, \; i = 1,2, ..., n \tag{9}$$

In the formula, $\hat{y}_j$ is the training output value; y_j is the expected value of the training output; M is the phase number in the reconstructed phase space; n is the population size.

Step 4: Calculate the fitness value corresponding to each particle position W_i according to the input and output samples. Determine the individual extreme value and the group extreme value according to the initial particle fitness value, and take the best position of each particle as its historical optimal position.

Step 5: Update the velocity and position of the particle itself by formula (6) and formula (7). Introduce a simple adaptive mutation operator, reinitialize the particle with a certain probability after each update. Calculate the fitness value of the new particle. Update the individual extreme value and the population extreme value according to the fitness value of the new population particle.

Step 6: Satisfy the maximum number of iterations. The optimal particle obtained by the improved Particle Swarm Optimization algorithm is assigned to the weights and thresholds of the BP neural network connection. After training, the BP neural network prediction model predicts the optimal solution output by time series.

5. Optimizing BP Neural Network Prediction Algorithms by Improving PSO Algorithms

5.1 Simulation Conditions

In order to illustrate the effectiveness of the proposed algorithm, under the environment of MATLAB 2009b, the algorithm calculation program is written in MATLAB language, and three prediction models are constructed by using MATLAB neural network toolbox. They are: improved PSO algorithm to optimize BP neural network prediction model (MPSOBP model), PSO algorithm to optimize BP neural network prediction model (PSOBP model) and general BP neural network prediction model (BP model). In the same measured traffic flow time series, the comparative experiment of traffic volume prediction is carried out.

In the experiment of traffic flow time series data according to the formula (10) as the mean of 0, amplitude is normalized time sequence 1, and to reconstruct the phase space of the normalized time series.

$$y_i = \frac{x_i - \frac{1}{n}\sum_{i=1}^{n} x_i}{max(x_i) - min(x_i)} \tag{10}$$

In the formula, $\{x_i\}$ is the original time series and $\{y_i\}$ is the normalized time series.

The error evaluation system of the experiment adopts absolute error err, average absolute error MAE and relative error perr, that is to say, absolute error err, average absolute error MAE and relative error perr.

$$err = xi - \hat{x}_i \tag{11}$$

$$MAE = \frac{1}{N_p}\sum_{i=1}^{N_p}|x_i - \hat{x}_i| \tag{12}$$

$$Perr = \frac{\sum_{i=1}^{Np}(x_i - \hat{x}_i)^2}{\sum_{n=1}^{Np} x_i^2} \tag{13}$$

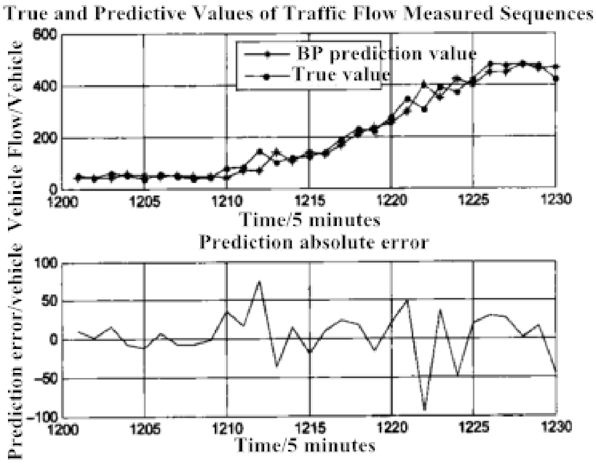
In the formula, x_i and $\hat{x}_i$ are the real value and the predicted value N_p are the predicted samples.

The experiment adopts m, 2m+1, 1, three-layer BP neural network structure, m is the embedding dimension of phase space reconstruction of traffic flow time series 5; BP neural network parameters are set as follows: training times are 100, training objectives are 1.0e-005, learning rate is 0.01; particle swarm optimization parameters are set as follows: population size is 30; evolutionary algebra is 100; acceleration factor is $c_1 = c_2 = 1.49445$; mutation probability of adaptive mutation operator $p_m \in [0.01, 0.05]$; the intervals of particle position and velocity are [-5,5] and [-1,1].

5.2 Empirical Analysis of Measured Traffic Flow Time Series

The short-term traffic flow data in the simulation experiment comes from the traffic detector data of a city. The data are recorded every five minutes, and 1302 data are generated.

In order to test the accuracy of the forecasting method, different numbers of training samples are taken for experiments. The forecasting results of three forecasting models when the training sample is 1200 and the forecasting sample is 30 are given in Fig. 1, Fig. 2 and Fig. 3 respectively. The conditions of three forecasting models in different numbers of training samples are given in Table 1. The average absolute error MAE and relative error perR of the next 30 prediction samples.



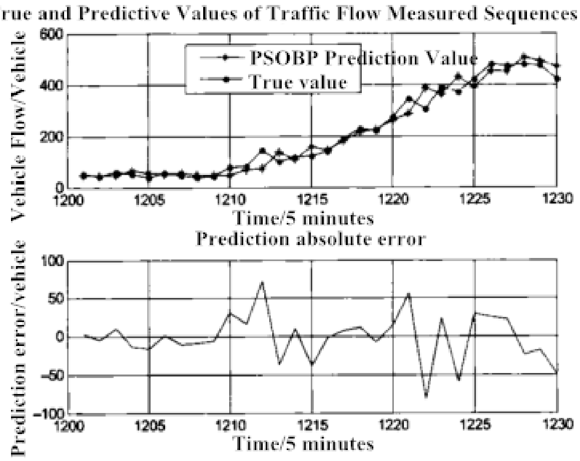


Fig. 1 BP neural network prediction results   Fig. 2 PSOBP neural network model prediction results

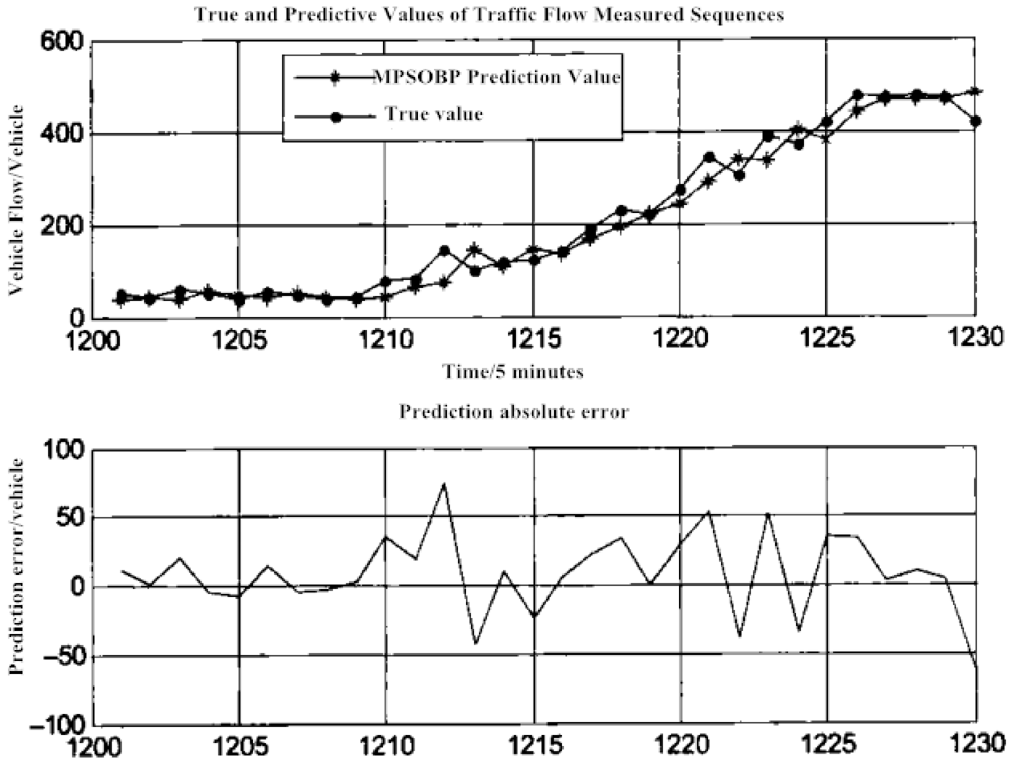


Fig. 3 Prediction results of MSOBP neural network model

It can be seen from Fig. 1, Fig. 2 and Fig. 3 that the forecasting results of the three forecasting models can well reflect the trend and regularity of traffic flow change, which shows that the three forecasting models can better meet the requirements of short-term traffic flow forecasting.

From Table 1, we can see that when the training sample is 800, the PSO algorithm forms the fast convergence effect of the particle population and the neural network falls into the local extremum, so the prediction accuracy of the PSOBP model is lower than that of the BP model. The MPSOBP model optimized by the PSO algorithm with the adaptive mutation operator in this paper avoids the particle from reaching its own historical optimal position or the best population history. The fast convergence effect of population formed by location aggregation has higher prediction accuracy than PSOBP model and BP model, which shows that MPSOBP prediction model is effective in predicting traffic flow.

Table 1. Prediction Errors of Different Training Samples in Measured Traffic Flow Time Series

Number of training samples		1200	1000	800	600	400
Predicted sample size		30	30	30	30	30
MAE	BP	24.1353	25.5804	34.272	31.1454	37.6787
	PSOBP	25.4632	26.5499	29.0602	30.528	33.0644
	MPSOBP	22.7096	23.5779	26.2551	23.4968	28.1463
perr	BP	0.015	0.0158	0.0365	0.0263	0.0364
	PSOBP	0.0144	0.0158	0.0216	0.0258	0.0305
	MPSOBP	0.0134	0.0136	0.0164	0.013	0.0204

From Table 1, we can also see that the less training samples, the more accurate the MPSOBP model is than the PSOBP model and BP model, which is of great significance for short-term traffic flow to achieve small sample forecasting.

6. Conclusion

Aiming at the local minimum defect and slow convergence speed of BP neural network prediction, an adaptive mutation operator is introduced into PSO algorithm, and a time series prediction method based on improved PSO algorithm to optimize BP neural network is proposed. It is applied to microscopic traffic flow prediction, and compared with PSOBP prediction model and BP model, the results show that this method greatly reduces the chance of being trapped in local optimum. Compared with PSOBP prediction model and BP prediction model, this method has better non-linear fitting ability and higher prediction accuracy for measured traffic flow. Moreover, even the MPSOBP can be further improved. For instance, it is possible to also use PSO to optimize some coefficients set by default in BP Neural Network, such as the number of hidden layers.

References

[1] He G G, Li Y, Ma S F. Discussion on short-term traffic flow forecasting methods based on mathematical models[J]. Systems Engineering -Theory & Practice, 2014, 20(12): 51-56.

[2] Wang D S, He G G. Summary and prospects of the study on traffic chaos[J]. China Civil Engineering Journal, 2016(1): 68-73.

[3] Li S, He G G, Zhang J. Relationship between distance headway and chaos in traffic flow on expressway [J]. Journal of Southwest Jiaotong University, 2017, 42(3): 304-308.

[4] Dong C J, Liu Z Y. Multi-layer neural network Involving chaos neurons and Its application to traffic-flow prediction [J]. Journal of System Simulation, 2017, 19(19): 4450-4453.

[5] Wu H. Based on traffic flow and chaos theory of traffic flow chaos identification and prediction research[D]. Xi’an: Chang’an University, 2019.

[6] Zhang Y M, Qu S R, Wen K G. A short-term traffic flow forecasting method based on chaos and RBF neural network [J]. Systems Engineering, 2017, 25(11): 30-34.

[7] Li S, Liu L J, Guo H L. Comparative of the predictive method of chaos in short-term traffic flow[J]. Systems Engineering, 2018, 27(9): 60-64.

[8] Zhang Y M, Qu S R. Adaptive Volterra prediction model for traffic flow based on chaos [J]. Application Research of Computers, 2017, 27(12): 4486-4488.

[9] Zang L. Application of neural network to analysis and treatment of chaotic observation time series [J]. Journal of Data Acquisition & Processing, 2018,16(4): 98-101.

[10] Kennedy J, Eberhart R C. Particle swarm optimization[C]// Proceedings of the 4th IEEE International Conference on Neural Networks, Piscataway: IEEE Service Center, 2015, 4: 1942-1948.

[11] Ling S H, Iu H, Leung F H F, et al. Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging [J]. IEEE Transactions on Industrial Electronics, 2018, 55(9): 3447-3460.