ISSN: 2414-1895 DOI: 10.6919/ICJE.202006_6(6).0035

Research on Network Traffic Forecast Based on Improved LSTM Neural Network

Luming Bai[a], Hui Wang, Zhibo Zhai

Henan University of Science and Technology, Luoyang Henan 471023, China.

[a]740991427@qq.com

Abstract

Accurately predicting network traffic can reduce the frequency of network congestion, prevent network crashes, and ensure network smoothness. In order to solve the problem of long short-term memory (LSTM) model predicting network traffic with large prediction errors and low accuracy, this paper proposes a network traffic prediction model based on the combination of particle swarm optimization (PSO) and LSTM neural network. The improved PSO algorithm is used to optimize the initial parameters of the LSTM neural network, and then the trained IPSO_LSTM model is used to predict the network traffic, and compared with the LSTM model and the PSO_LSTM model. The experimental results show that the IPSO_LSTM model converges faster than the LSTM model, and the prediction error of the IPSO_LSTM model is reduced by 4% compared with the PSO_LSTM model.

Keywords

Network traffic prediction, Particle swarm optimization, Long and short memory, Neural network.

1. Introduction

Network traffic is an important carrier to record and reflect the activities of the network and its users, and network traffic is also one of the important indicators to measure network performance. By analyzing and predicting network traffic, you can grasp the characteristics and changing trends of network traffic in advance, and make corresponding adjustments and controls in a targeted manner to avoid the occurrence of some network congestion. Moreover, analyzing network traffic is also of great significance for service quality assurance and anomaly detection.

Finding accurate and stable network traffic prediction models has always been the focus of researchers. The current network traffic prediction models mainly include linear prediction models and nonlinear prediction models [1].

The linear prediction model is based on the analysis of network traffic data in time series and the prediction models trained, such as the autoregressive moving average model (ARMA), the autoregressive integrated moving average model (ARIMA) and the fractional autoregressive integration moving average model (FARIMA), etc [2, 3]. The linear model algorithm is relatively simple to implement and has good prediction results for small-scale network traffic, but it is not ideal for today's large-scale network traffic prediction. Modern network traffic has non-linear characteristics such as catastrophe and multi-dependence, and the linear prediction model cannot accurately reflect the changing characteristics of network traffic [4].

The neural network model has better prediction results for nonlinear data, and the neural network model is used to predict network traffic. Liu and Wang uses a hybrid network traffic prediction model based on support vector regression. These models can well describe the new characteristics of

ISSN: 2414-1895 DOI: 10.6919/ICJE.202006_6(6).0035

network traffic, and obtain relatively high-precision prediction results. BP neural network is also a widely used forecasting algorithm model [5]. Du uses BP neural network to predict network traffic. The results show that the BP neural network model is effective and feasible for network traffic forecasting good convergence and stability [6]. Jiang and Sun proposed an improved BP neural network network traffic prediction model, which provides high-precision prediction for the network traffic that changes greatly and is unstable [7]. However, the computational overhead required for prediction is large, which makes the working efficiency of the entire model low. But, these algorithms do not consider the time series correlation of time series data.

Recurrent neural network (RNN) introduces the concept of time in training. RNN can extract deep features from samples and is suitable for processing time series [8]. Long short-term memory (LSTM) neural network is a special recurrent neural network model [9]. However, the parameters of the LSTM neural network, such as the number of hidden layer neurons, the learning rate, and the number of iterations, are difficult to determine [10]. The number of hidden layers and the number of hidden layer neurons determine the fitting ability of the training model. Parameters such as learning rate, batch size, number of iterations, and time step affect the training process and final result of the model. In practical applications, these parameters are usually determined empirically and subjectively. In this way, setting the parameters will affect the fitting ability of the model, and the model will not achieve a good prediction effect. Therefore, in this paper, for the problem that the parameters of the LSTM neural network are difficult to determine, this paper uses an improved particle swarm optimization (IPSO) to optimize the LSTM neural network and proposes an IPSO_LSTM network traffic prediction model. Simulation experiments show that the improved model can improve the convergence rate and reduce the prediction error.

2. LSTM neural network

2.1 Principle of Recurrent neural network

Recurrent neural network (RNN) is a neural network that introduces cyclic feedback and considers the time series correlation of time series. Figure 1 shows a typical RNN model. The RNN network is memorable, and the memory unit can perceive the information of the last moment, which effectively solves the problem of the correlation between the processing data. Each RNN unit structure includes an input layer X, a hidden layer h, and an output layer O, where U, W, and V are the weight matrix of the corresponding layers.

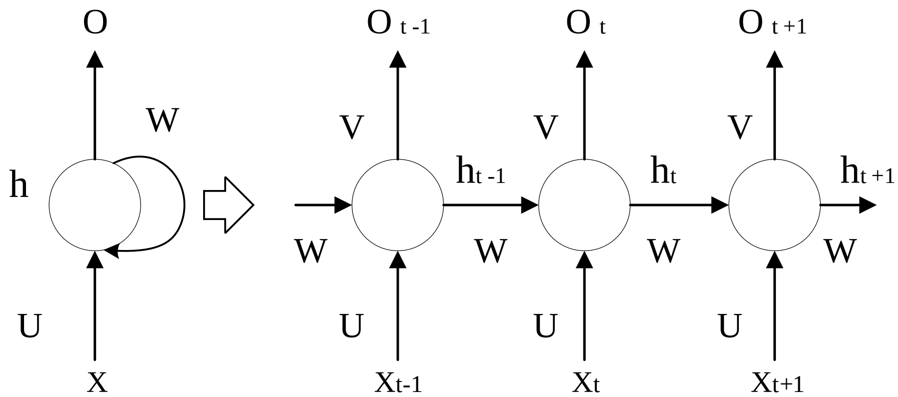

Figure 1. Recurrent neural network model

At time t, Xt represents the input at time t and Ot represents the output. The hidden layer ht of the model is determined by the hidden layer ht-1 and Xt at the previous time, namely:

$$h_t = \phi(Ux_t + Wh_{t-1} + b) \tag{1}$$

Among them, b is the bias term, φ () is the activation function, generally choose the tanh function.

The output at time t is:

$$O_t = Vh_t + c \tag{2}$$

ISSN: 2414-1895
DOI: 10.6919/ICJE.202006_6(6).0035

Where c is the bias term; the model prediction result, the softmax function is generally selected as the activation function σ (), namely:

$$y = \sigma(O_t) \tag{3}$$

The characteristic of RNN is to apply the information of the last moment to the output of the current moment, but as the time interval becomes larger, the general RNN network cannot avoid the problem of gradient disappearance and gradient explosion, and will lose the ability to learn information at a farther distance.

2.2 Principle of LSTM neural network

LSTM is an improved model of RNN, as shown in Figure 2. The difference with RNN is that LSTM adds a gating mechanism and a memory unit. Through the gating unit, the LSTM neural network can not only learn long-distance learning to avoid gradient disappearance during training, but also selectively forget some information to prevent gradient explosion. . The LSTM unit contains three gates: input gate, forget gate and output gate. The use of three control gates can control the amount of information passing through the LSTM unit, which facilitates the further spread of information.

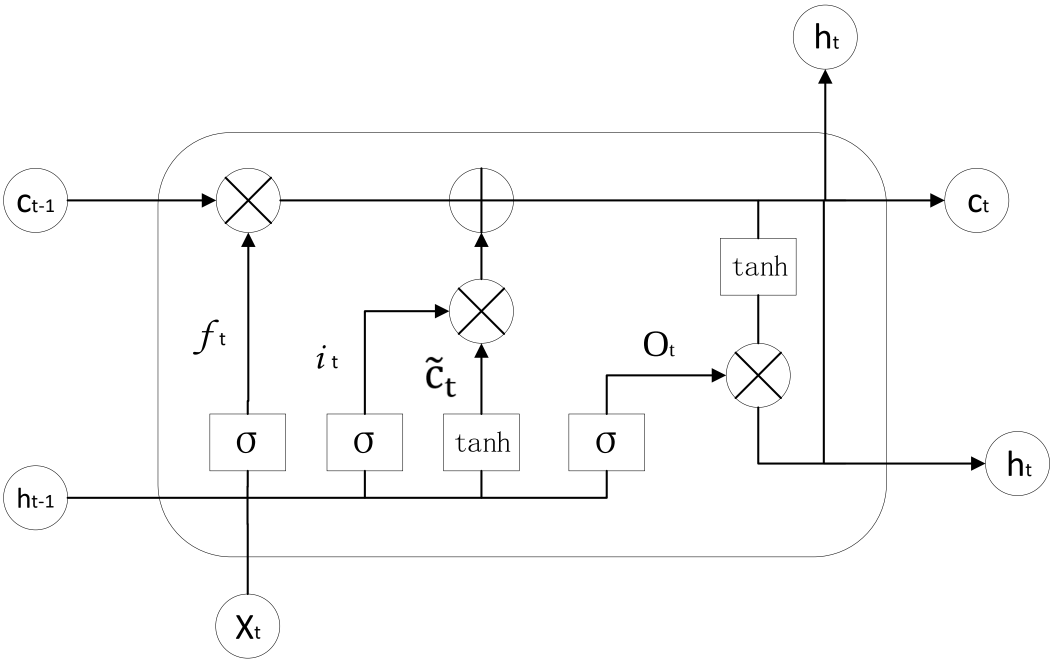

Figure 2. LSTM unit

Among them, f_t is the forget gate, i_t is the input gate, O_t is the output gate, at time t:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_t) \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$

$$O_t = \sigma(Wo \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{7}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{8}$$

$$h_t = O_t \cdot \tanh(C_t) \tag{9}$$

In the formula: σ () activation function is the value of sigmoid function ft is between 0 and 1, when the output value is 1, it means completely reserved, when the output value is 0, it means completely discarded; $\tilde{C}t$ is hidden at the last time The output of the layer and the information in the input at the current time determine the information to be selected by the tanh function; W, b are the respective weights and offset items; Ct is used to update the state of the memory unit; ht is the output of the hidden layer.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202006_6(6).0035

This is the forward calculation process of the LSTM neural network model. the particular structure enables it to learn long-term dependencies and has been widely used in text analysis [11], time-series prediction [10], and other fields.

3. Particle Swarm Optimization improves LSTM

3.1 Particle Swarm Optimization

Particle swarm optimization is a random search algorithm based on swarm intelligence. The algorithm is to find the optimal solution through information sharing in the population and collaborative cooperation among the individuals in the population [13]. Suppose that in the S-dimensional search space, n particles form a population X, X = (X1, X2, ..., Xn), where the position of the i-th particle is Xi = (xi1, xi2, ..., xis) Velocity Vi = (vi1, vi2, ..., vis), in each loop iteration, each particle updates its position by tracking the individual extreme value pbest and the global extreme value gbest. The individual extremum is the optimal solution found by the particle itself, and the global extremum is the optimal solution found by the entire population. The formula is as follows:

$$V_{i,k+1} = w \cdot V_{i,k} + c1 \cdot r \cdot (pbest_i - X_{i,k}) + c2 \cdot r \cdot (gbest_i - X_{i,k}) \tag{10}$$

$$X_{i,k+1} = X_{i,k} + V_{i,k+1} \tag{11}$$

In the formula w is the inertial weight, c1 and c2 are the learning factors, usually c1 = c2 = 2, r is the random number between (0, 1), i represents the i-th particle in the population, and k is the current iteration number.

3.2 Improved Particle Swarm Optimization

For particle swarm optimization, the larger the value of the inertial weight w, the faster the convergence speed of the algorithm and the stronger the global search ability, but the accuracy is lower; otherwise, the stronger the local search ability, the higher the search accuracy, but it is easy to fall into the local optimal Solution [14]. In the standard particle swarm optimization algorithm, linearly decreasing inertia weights are used, and linearly decreasing inertia weights cannot balance the global search ability and local search ability of the algorithm. Therefore, in this paper, adaptive weights are used to make the weight value change with the adaptability of the particles, which can better improve the search ability of the algorithm. The calculation formula of the adaptive weight w is as follows:

$$w = \begin{cases} w_{\min} - (w_{\max} - w_{\min}) \dfrac{f - f_{\min}}{f_{avg} - f_{\min}} & f \le f_{avg} \\ w_{\max} & f > f_{avg} \end{cases} \tag{12}$$

In the formula: wmax and wmin are the maximum and minimum values of w, f is the current fitness value, fmin is the minimum fitness value, and favg is the average fitness value.

The learning factor determines the direction of particle movement and the convergence result. c1 and c2 respectively reflect the ability of particles to learn from themselves and the ability to learn from the optimal particles of the population. It is easy to fall into local optimum [15]. In the usual particle swarm optimization algorithm, c1 = c2 = 2, but this does not meet the requirements of practical applications. Therefore, this paper designs a dynamic learning factor, the formula is as follows:

$$c1 = a - \frac{k(a-b)}{t} \tag{13}$$

$$c2 = b + \frac{k(a-b)}{t} \tag{14}$$

In the formula: a, b are adjustment parameters, t is the maximum number of iterations, and k is the current number of iterations. As the number of iterations increases, c1 decreases and c2 increases, which can quickly find the optimal value in early evolution and converge to the optimal solution in late evolution.

3.3 Particle Swarm Optimization improves LSTM

The particle swarm optimization algorithm is easy to operate and has fast convergence speed. It plays an important role in solving complex optimization problems. Therefore, this paper proposes to use an improved particle swarm optimization algorithm to optimize the LSTM neural network for the problem that the parameters of the LSTM neural network are difficult to determine. According to the root mean square error of the prediction results of the LSTM model corresponding to different parameters, the particle swarm algorithm adaptively adjusts the position and speed of the particles, and finds the optimal parameter combination of the LSTM model. The algorithm flow chart is as follows:

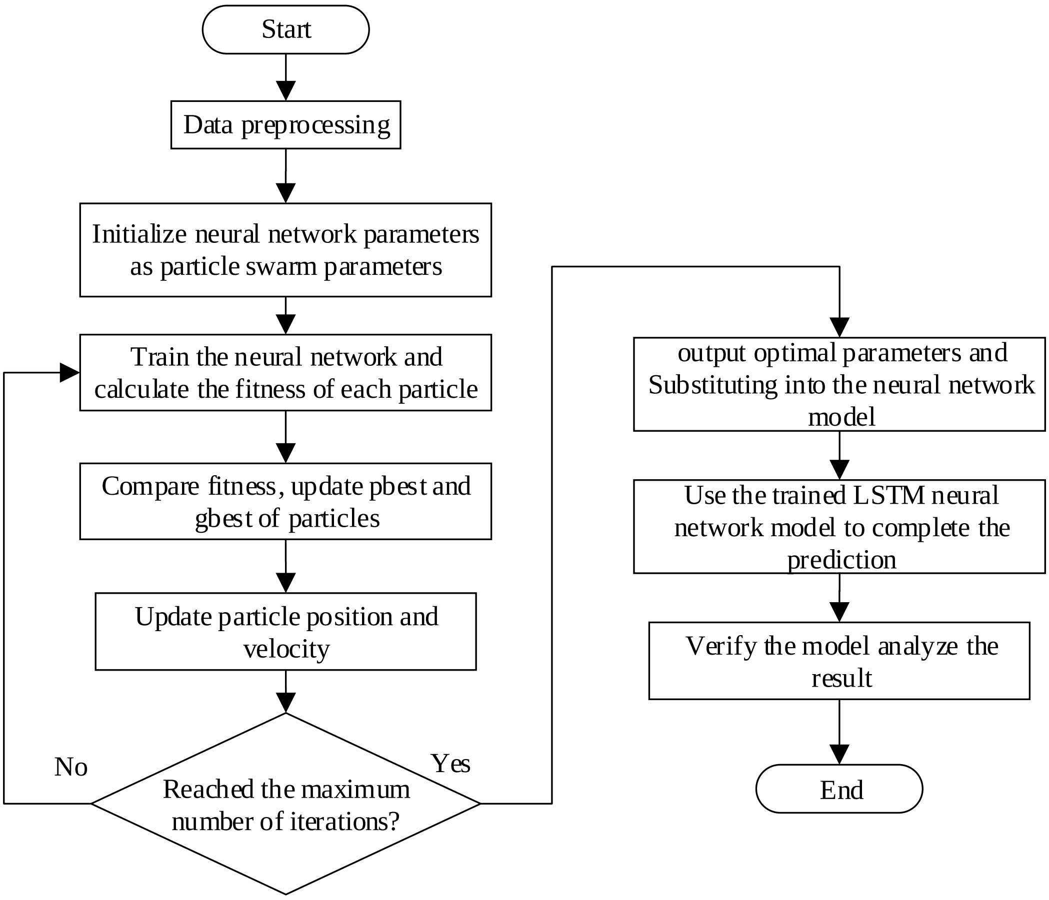

Figure 3. Particle swarm optimization algorithm LSTM algorithm flow chart

And the specific implementation steps are as follows:

Step(1): Normalize the data collected by the experiment.

Step(2): Initialize the parameters, determine the population size, the number of iterations, initialize the position and velocity of the particles, and randomly generate a particle X (node1, node2, look_back), where node1 represents the number of neurons in the hidden layer and node2 represents the second layer The number of neurons, look_back represents the time step.

Step(3): Divide the sample data into training data, verification data and test data, train the neural network, and calculate the fitness of each particle X_i. Set the fitness function *f* of the particle to the mean square error (MSE) of the verification data. The mean square error of the LSTM model verification data obtained after the training times reach the limit. The definition of f is as follows:

$$f = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|y_i^{'} - y_i\right|}{y_i} \tag{15}$$

ISSN: 2414-1895
DOI: 10.6919/ICJE.202006_6(6).0035

In the formula: N is the sample size, y_i' is the predicted value, and y_i is the true value.

Step(4): Compare the fitness value and determine the individual extreme value and global extreme value of each particle.

Step(5): Update the particle speed and position according to formula (10) and (11).

Step(6): Determine whether the maximum number of iterations is satisfied. If not, return to Step3.

Step(7): Bring the output data into the LSTM model.

Step(8): Use the trained model to complete the prediction.

Step(9): Output the predicted value, analyze the result, and the algorithm ends.

4. Experimental Results and Analysis

4.1 Experimental Environment and Data

The experimental environment of this article is in the Windows10 operating system, the CPU is Intel (r) Core (TM) i5-6500@3.20GHz, the RAM is 8.00G, the Keras deep learning framework is used to build a neural network prediction model, and the Python version is 3.6.

The data for this experiment is derived from the egress traffic on a switch on the campus network. The collection time is 500 data per hour between 2019.10.10 and 2019.10.30, as shown in Figure 4:

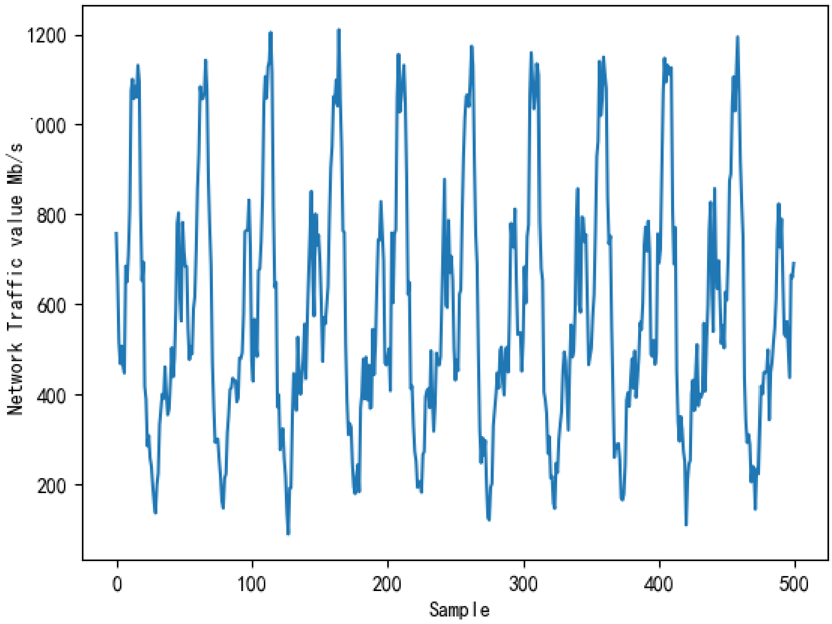

Figure 4. Sample data set

4.2 Data preprocessing and parameter selection

The sample data of the network traffic is subjected to Min-Max normalization through formula (16), so that the sample data is between [0, 1].

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{16}$$

Among them: $X_{\max}$ and $X_{\min}$ represent the maximum and minimum values in the number of samples, respectively, and after processing, the last 80 samples of the data are used as the test set, and the rest are used as the training set.

The experiment uses mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) as the evaluation criteria of the model, namely:

ISSN: 2414-1895
DOI: 10.6919/ICJE.202006_6(6).0035

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - y_i'\right| \tag{17}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2} \tag{18}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|y_i - y_i'\right|}{y_i}\cdot 100 \tag{19}$$

In the formula: y_i and y_i' are the real value and the predicted value, respectively, N is the number of predicted samples.

Set the number of particle swarm algorithm iterations to 20, respectively take wmax = 0.9, wmin = 0.4, a = 2.5, b = 0.5; at the same time use dropout to prevent training overfitting [16], set nerve The network parameters dropout = 0.1, validation_split = 0.2.

4.3 Result Analysis

In order to make the IPSO_LSTM algorithm have better contrast, this article uses LSTM, PSO_LSTM as the comparison model. The training samples are input into the training model for learning, and the prediction results are output after the model training is completed. The prediction comparison results of each model are shown in Figures 5-7.

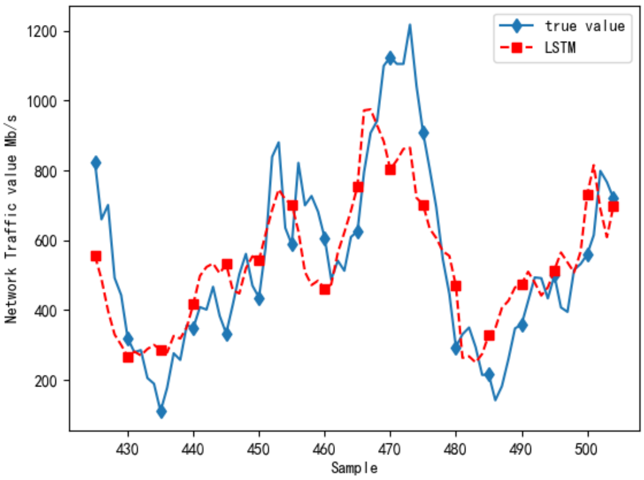

Figure 5. Prediction result of the LSTM model

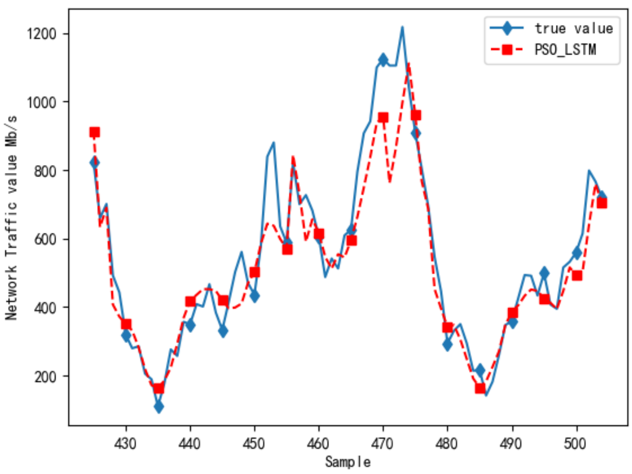

Figure 6. Prediction result of the PSO_LSTM model

ISSN: 2414-1895 DOI: 10.6919/ICJE.202006_6(6).0035

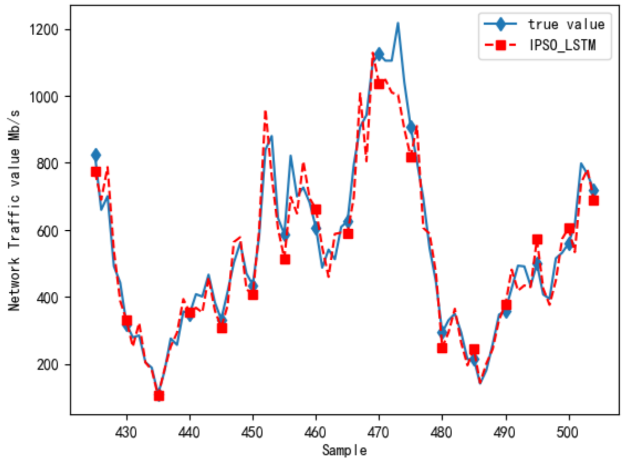

Figure 7. Prediction result of the IPSO_LSTM model

It can be seen from the figure that the prediction curve of the LSTM model can reflect the overall change trend of network traffic, but the error is large; and the fitting degree of the PSO_LSTM model and the IPSO_LSTM model is better than the LSMT model, which can track the network traffic better The trend of change has realized a more accurate forecast. Comparing the results of the PSO_LSTM model and the IPSO_LSTM model, it can be seen that the prediction error of IPSO_LSTM is smaller, especially at the moment when some burst traffic occurs.

The changes of the training loss function of the three models are shown in Figure 8. The loss of the LSTM model optimized by particle swarm optimization is smaller than that of the unoptimized model, and the IPSO_LSTM model can converge earlier than the PSO_LSTM model, and the loss value is relatively small.

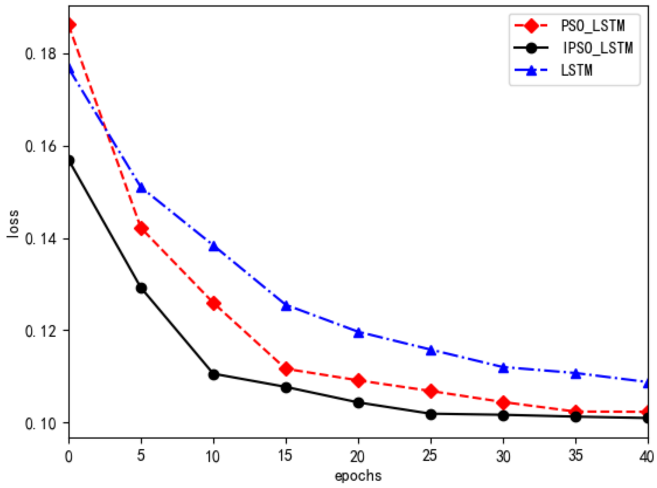

Figure 8. Changes in the training loss function of each model

The calculated error values of each model are shown in Table 1.

Table 1. Error of each model

	LSTM	PSO_LSTM	IPSO_LSTM
RMSE	89.76	67.36	62.60
MAE	62.56	50.87	49.35
MAPE	11.53	9.01	8.64

ISSN: 2414-1895
DOI: 10.6919/ICJE.202006_6(6).0035

It can be seen from the table that comparing the prediction errors of the IPSO_LSTM model and the standard LSTM model, it is found that the evaluation index value of the IPSO_LSTM model is less than that of the LSTM model. Effectively improve prediction performance. Moreover, the RMSE, MAE, and MAPE of the IPSO_LSTM model are all less than PSO_LSTM, and the prediction errors are reduced by 7%, 3%, and 4%, respectively. The results show that after the improved particle swarm optimization algorithm is used to optimize the LSTM model, the model not only converges faster but also has lower prediction error.

5. Conclusion

In order to solve the problem that the parameters of LSTM neural network are difficult to determine, this paper designs and implements a network traffic prediction model (IPSO_LSTM) based on improved particle swarm algorithm optimization LSTM neural network. The model uses an improved particle swarm algorithm to optimize the parameters of LSTM neural network Select, and then use the trained model to predict network traffic. The simulation results show that the PSO_LSTM measurement model has better prediction accuracy than the LSTM neural network, and the IPSO_LSTM model using the improved particle swarm algorithm also has good convergence efficiency, the prediction error is relatively small, and the prediction accuracy rate is obtained. improve.

Acknowledgments

Key research project of Henan Provincial Department of Science and Technology (182102210293), scientific research project of Henan Provincial Department of Education (18A520007),

CERNET Next Generation Internet Technology Innovation Project (NGII20180107).

References

[1] Jing Q , Jing-Bo X , Ji-Xiang W U . Research and development of network traffic prediction model[J]. computer engineering and design, 2012, 33(3):865-869.

[2] Chen G J, Liang P, Wang K . Research on network traffic prediction model [J] Information and Communications, 2017 (08): 191-194.

[3] Huafeng C, Jianing L . Modeling and forecast of wireless network traffic based on combinatorial optimization theory[J]. Modern Electronics Technique, 2016,39(23):43-46.

[4] Aldhyani T H H, Joshi M R . Integration of time series models with soft clustering to enhance network traffic forecasting[C]// International Conference on Research in Computational Intelligence & Communication Networks. IEEE, 2017:212-214.

[5] Liu Y, Wang R X . Study on network traffic forecast model of SVR optimized by GAFSA[J]. Chaos Solitons & Fractals, 2015, 89(3):153-159.

[6] Du Z N. Network Flow Predicting Model Based onCuckoo Search Algorithm Optimizing Neural Network[J]. Application of Electronic Technology,2015,41(03):82-85.

[7] Jiang L Q, Sun W S . Network traffic prediction model based on improved BP neural network [J]. Communication Technology, 2017, 50 (01): 68-73.

[8] Yang Y Y, Fu Q , Wan D S . Time series prediction model based on deep recurrent neural network [J]. Computer Technology and Development, 2017,27 (03): 35-38 + 43.

[9] Li X, Peng L , Yao X , et al. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation[J]. Environmental Pollution, 2017, 231:997-1004.

[10] Kong W, Dong Z Y , Jia Y , et al. Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network[J]. IEEE Transactions on Smart Grid, 2017:1-1.

[11] Song M, Zhao X , Liu Y , et al. Text Sentiment Analysis Based on Convolutional Neural Network and Bidirectional LSTM Model[J]. in Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators, 2018, vol. 902:55–68.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202006_6(6).0035

[12] Choi J Y, Lee B . Combining LSTM Network Ensemble via Adaptive Weighting for Improved Time Series Forecasting[J]. Mathematical Problems in Engineering, 2018, 2018(PT.9):2470171.1-2470171.8.

[13] Kennedy J, Eberhart R C. Particle swarm optimization[C].Proceedings of the 1995 IEEE International Conference on Neural Networks,1995:1942-1948.

[14] Pi Q Y, Ye H T . A particle swarm optimization algorithm with dynamic adjustment of inertia weight [J]. Journal of Guangxi University of Science and Technology, 2016,27 (03): 26-32.

[15] Sun X B, Liu Q S . BP network optimization of improved chaotic PSO algorithm [J].Microcomputer Information, 2011,27 (05): 157-159.

[16] SRIVASTAVA N，HINTON G，KRIZHEVSKY A，etal．Dropout：a simple way to prevent neural networks from overfitting[J]．The Journal of Machine Learning Research，2014，15(1)：1929-1958．