Traffic flow prediction method based on bidirectional linear recurrent neural network

Ruizhi Xu

N/A

Zhou Shen (✉ shenzhou.research@gmail.com)

N/A

Research Article

Keywords: Traffic flow prediction, recurrent neural network, data-driven approach

Posted Date: June 2nd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-583553/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Traffic flow prediction method based on bidirectional linear recurrent neural network

R. Xu and Z. Shen*

Abstract

In order to improve the rationality and effectiveness of intelligent traffic control and management on urban roads, a bidirectional linear recurrent neural network-based traffic flow prediction method is proposed from the perspective of spatial and temporal characteristics of traffic flow. The method effectively combines the characteristics of fast and accurate bilinear polynomial solution and dynamic calibration of recurrent neural network, and adopts particle swarm algorithm to realize the dynamic pruning process of redundant neurons and weights, which improves the convergence speed and prediction accuracy of the algorithm. The algorithm is trained and experimented with video data, and a comparative analysis is conducted. The results show that the method can achieve accurate prediction of road traffic flow, the traffic flow prediction accuracy reaches more than 90%, meeting the data accuracy requirements of actual traffic management and control, and the convergence speed of the algorithm has also been significantly improved.

1 Introduction

The rapid development of Intelligent transportation system (ITS) has brought great social and economic benefits to the development of society [1]. As the basis for the implementation of ITS control tools, the accuracy of the prediction results directly affects the implementation of traffic control strategies [2, 3].

At present, domestic and foreign scholars have proposed a large number of traffic flow parameter prediction methods from various perspectives, mainly based on regression analysis, intelligent algorithm-based prediction methods, time-series analysis-based prediction methods, and multi-step traffic parameter prediction methods [4–12]. However, due to the variability and complexity of traffic flow on urban roads, all of these methods have certain shortcomings [13]. The prediction methods based on linear regression analysis have low accuracy and tend to ignore the interaction effect between traffic parameters, which cannot describe the nonlinear cause-effect relationship; the prediction methods based on time series analysis have large prediction errors and cannot

*Corresponding. The authors declare no competing interests.

meet the accurate prediction of traffic flow when there are large fluctuations in traffic flow; the methods based on multi-step traffic parameter prediction require the use of equivalent historical data. The traffic parameter-based multi-step forecasting method requires equivalent historical data as the data base, and the lower the correlation, the higher the forecasting error due to the obvious difference between the historical data and the traffic parameter data to be predicted. At the same time, none of the existing traffic flow parameter prediction methods can realize the dynamic self-correction function of the traffic flow parameters obtained from the prediction, i.e., they cannot calculate the corrected parameters based on the traffic flow data obtained at the current time to make dynamic prediction of the predicted values at the next time [14].

Therefore, to address the problems of low prediction accuracy and lack of self-correction function of the above existing traffic flow parameter prediction methods, this paper proposes a Bidirectional linear Recurrent Neural Network (BRNN) algorithm based on the analysis of the spatial and temporal fluctuation characteristics of traffic flow parameters. The algorithm can quickly approximate various nonlinear functions and has self-correction function when predicting traffic flow parameters. At the same time, in order to improve the convergence speed and prediction accuracy of the BRNN model, the particle swarm algorithm is effectively incorporated into the BRNN model to prune the redundant neurons and weights, thus reducing the computational complexity and preserving its generalization ability to improve the computational speed and prediction accuracy of the traffic flow prediction algorithm.

2 Bidirectional Linear Recurrent Neural Network

The bidirectional linear recurrent neural network model effectively integrates bilinear polynomials and recurrent neural networks to approximate multiple nonlinear functions according to the characteristics of the training data without rapidly increasing the complexity of the network, and has the advantage of dynamic recursive calibration, which is easier and faster to implement in applications than higher-order neural networks [15–19].

The bilinear polynomial is a simple periodic nonlinear system model. For a one-dimensional input/output case, the linear polynomial input-output relationship can be expressed as:

$$y[n] = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} b_{ij} y[n-j] x[n-i] + \sum_{i=1}^{N-1} a_i y[n-i] + \sum_{i=1}^{N-1} c_i x[n-i]$$

where $x[i]$ is the input variable, $y[i]$ is the output variable, and N is the recurrence number. When the bilinear recurrent neural network model has N input neurons, M implicit neurons and K polynomials. In the BRNN model, the input variables are:

$$X[n] = [x_1[n], x_2[n], \dots, x_N[n]]^T$$

The network connection weights from the input layer to the implicit layer are output as:

$$net[n] = [\text{net}_1[n], \text{net}_2[n], \dots, \text{net}_N[n]]^T$$

The $M \times K$ dimensional periodic vector is denoted as:

$$Z[n] = [net_1[n-1], \dots, net_M[n-1], \dots, net_1[n-K], \dots, net_M[n-K]]^T$$

The period vector of the p th implicit layer node is denoted as:

$$net_p[n] = w_p + A_p^T Z[n] + Z[n] B_p X[n] + C_p^T X[n]$$

where w_p is the neuron bias value, A_p is the periodic weight vector, B_p is the bilinear polynomial weight matrix, C_p is the feedforward weight vector, and T denotes the transpose matrix. When the implicit layer neuron activation function is $f(g)$, the output variables of the implied layer neuron are:

$$O_p[n] = f(net_p[n])$$

The classical BRNN is a simple pre-feedback neural network from the implicit layer to the output layer. If $O_p[n]$ is the output variable of the l -th neuron at time n , then the period vector of the p -th implicit layer node and the implicit layer output variable are:

$$net_l[n] = \sum_{p=0}^{N_k} w_{lp} O_p[n]$$

$$O_l[n] = f(net_l[n])$$

where w_{lp} is the weight factor from the implicit layer to the output layer. In summary, the output variables of the output layer are:

$$y_l[n] = f\left(vl + \sum_{p=0}^{N_k-1} w_{lp} O_p[n]\right)$$

3 Optimization of BRNN based on particle swarm algorithm

3.1 BRNN optimization model

The presence of a large number of neurons and weight parameters in the classical bilinear recurrent neural network makes the network structure complex and the convergence speed of the algorithm slow, and the convergence speed of the algorithm is related to the dynamic performance of the prediction model [20–24]. Therefore, in order to improve the convergence speed of the BRNN model

and enhance the reliability of traffic flow prediction results, the particle swarm algorithm is integrated with the bilinear recurrent neural network algorithm, so as to prune the redundant neurons and weights in the classical BRNN model to reduce the complexity of the network, prevent it from falling into local optimum and improve the computational speed [25–31].

The bilinear recurrent neural network structure is input layer—hidden layer—output layer [32–35]. When the number of output layers N_0 is 1, the multiplication term in the first equation is denoted as:

$$N_c = (N_i + N_f + N_i \cdot N_f) \cdot N_h + N_h$$

where N_i is the number of input layer neurons, N_f is the number of feedback nodes, and N_h is the number of hidden layer neurons.

To further improve the prediction accuracy of the BRNN model, a portion of the pre-feedback input variables are added to the linear feedback layer, and the output variables of the optimized BRNN model are converted from the first equation to:

$$y_p[n] = d_p + \sum_{i=0}^{N_i-1} \sum_{j=S}^E b_{pi} y_p[n-j] \Bigg) x[n-i] + \sum_{i=0}^{N_f-1} a_{pi} y_p[n-i] \\ + \sum_{i=0}^{N_i-1} c_{pi} x[n-i]$$

where $E = p + (N_i - 1)/2$ and $S = (N_i - 1)/2 - p$.

Therefore, the number of multiplication terms in the output layer of the BRNN model is changed from N_c to N_S , and N_S is expressed as:

$$N_S = N_i \cdot N_f + (2N_i + N_f) \cdot N_h + N_h$$

The reduction in the number of multiplicative terms ΔN in the formula for calculating the output variables of the output layer of the optimized bilinear recurrent neural network can be expressed as:

$$\Delta N = N_c - N_S = N_h \cdot N_f \cdot N_i - N_h \cdot N_i - N_f \cdot N_i$$

In order to further reduce the computational load of the BRNN model, a particle swarm algorithm is used to dynamically prune the BRNN model. In order to further reduce the computational load of the BRNN model, a particle swarm algorithm is used to dynamically prune the BRNN model by dynamically removing weights and neurons from the network while ensuring the reliability of the output prediction results.

3.2 BRNN Optimization Process

Particle Swarm Optimization (PSO) seeks the global optimal solution through iteration from the perspective of randomized solution, with the advantages of

high robustness, fast convergence and simple implementation. The integration of particle swarm algorithm into the BRNN optimization process can effectively improve the convergence speed and learning ability of BRNN. According to the number of weight and bias values in the BRNN model, the search space dimension S_V of the particle swarm algorithm is determined, which can be expressed as:

$$S_V = N_i \cdot N_h + N_h \cdot N_f + N_i + N_h$$

In the S_V dimensional search space, the population $S = (S_1, S_2, \dots, S_m)$ consisting of m particles, the q -th particle represents an S_V dimensional vector, which is the spatial position of the q -th particle in the S_V dimensional space. The q -th particle represents an S_V -dimensional vector, i.e., the spatial position of the q -th particle in S_V -dimensional space. The velocity of the q th particle is denoted by $Vq = (Vq_1, Vq_2, \dots, Vq_{s_v})^T$, and the individual and global extremes are denoted as $Pq = (Pq_1, Pq_2, \dots, Pq_{s_v})^T$ and $Pgs_v = (Pq_1, Pq_2, \dots, Pq_{s_v})^T$. During each iteration of the particle swarm algorithm, the particles will update their iterative velocity and position according to the individual and global extremes. The particle velocity and position update models are:

$$\begin{aligned} V_{q_{s_v}}^{k+1} &= \omega V_{q_{s_v}}^k + c_1 r_1 (P_{q_{s_v}}^k - S_{q_{s_v}}^k) + c_2 r_2 (P_{gs_v}^k - S_{gs_v}^k) \\ S_{q_{s_v}}^{k+1} &= S_{q_{s_v}}^k + V_{q_{s_v}}^{k+1} \end{aligned}$$

where ω is the inertia weight, c_1 and c_2 are learning factors, and t_1 and t_2 are random numbers distributed between $(0, 1)$.

The specific process of optimizing the BRNN model is as follows:

1. Network initialization. The dimensionality S_V , learning factors c_1 and c_2 (generally equals, with values from 0 to 4), and the maximum number of cycles of the particle swarm algorithm are initialized. The initialization of particles is mainly to assign random values to the initial velocity and spatial position of particles.
2. The basic structure of the network topology of the BBRN optimization model is determined according to the traffic flow parameters.
3. Evaluation of the fitness function. Random population particles are generated for the optimized BRNN model is trained to output the fitness values that meet the accuracy requirements.

$$f(net) = \sum_{i=1}^m \sum_{j=1}^{N_o-1} |y_{ij} - \hat{y}_{ij}|$$

where y_{ij} and $\hat{y}_{ij}$ are the true and predicted values of the test set, respectively.

4. Start the particle swarm algorithm. The individual velocities and positions of the particles are updated according to the previous equations, and the particle population is initialized with a certain probability, updated, and the particle fitness.
5. Pruning operation. Compare the size of the fitness index before and after the update. When the updated fitness index is smaller than the original fitness index, the original particle population is deleted, the population is updated, and the invalid implicit layer or input layer neurons and their corresponding weights are deleted from the BRNN network.
6. Loop steps (4) and (5) in turn, search for the minimum fitness index and its corresponding particle population within the set maximum number of loop steps, and output the calculation results.
7. The link weights and thresholds are dynamically set for the BRNN optimization network according to the optimal particles, and the traffic flow prediction results are output after the BRNN optimization network prediction model is trained.

4 Case Studies

4.1 Configurations

In order to verify the validity of the optimized BRNN model proposed in this paper, two prediction models were constructed using MATLAB programming software: the classical BRNN model and the optimized BRNN network model using particle swarm algorithm. The prediction results are validated by comparative analysis using the actual traffic parameters collected, the prediction data obtained by the classical BRNN model, and the prediction data obtained by the BRNN optimization model proposed in this paper.

The experimental real data are obtained from the video data of two main urban road sections in Shanghai, labeled as Section 1 and Section 2. The model calibration data are the traffic flow parameters for 14 consecutive days from April 15 to 29, 2014. The sampling time interval is 5 minutes. The traffic flow data for the first 13 days were selected as the training data, and the data collected on the 29th day were used as the validation data. Section 1 is a 4-lane roadway and Section 2 is a 3-lane roadway. The experimental neural network structure is 4-3-1 (2), the number of training sessions is 100, the learning efficiency is 0.1, the training target is 0.05, the learning factor and the value of 1.7923. The particle swarm algorithm is initialized, the population size is 1000, and the number of iterations is 200. In order to evaluate the robustness of the model for long-term prediction, the relative and absolute error functions are used to check the prediction effect of the model.

4.2 Results

In order to check the reliability of the traffic flow prediction model, the traffic flow parameters of Road Section 1 and Road Section 2 during peak and flat hours were used as model input data and calibration data.

The predicted traffic flow on Road Section 1 during peak hours is analyzed with the experimental results. The traffic flow on urban roads during peak hours is high and the traffic flow curve is more volatile, which makes it more difficult to predict the traffic flow. When the classical BRNN model is used to predict the peak hour traffic flow, the maximum absolute error between the predicted and real values reaches 5.899 veh/min and the average absolute error is 2.862 veh/min. If the optimized BRNN model is used to predict the peak hour traffic flow parameters, the maximum absolute error and the average absolute error are both significantly reduced to 4.301 and 4.862 respectively. The maximum absolute error and average absolute error are significantly reduced to 4.301veh/min and 2.102 veh/min, respectively. As shown in Table 1, the prediction accuracy of the classical BRNN model is 89%, while the accuracy of the BRNN optimized model is 93%, based on the traffic count of Section 1. It can be seen that the BRNN model is more effective than the classical BRNN model in predicting the traffic flow of the four-lane roadway during the peak hours of the roadway 1.

According to the experimental results, the optimized BRNN model can accurately predict the traffic flow of Section 2 during the peak hours. Compared with the classical BRNN model, the predicted traffic flow obtained by the optimized BRNN model has a better fit with the real traffic flow obtained from the video data. At the same time, there is a significant difference between the absolute error values of traffic flow prediction of the classical BRNN model and the optimized BRNN model, i.e., the absolute error of traffic flow prediction of the optimized BRNN model is smaller than that of the classical BRNN model, and the prediction accuracy of traffic flow parameters reaches 90.8%. It can be seen that the optimized BRNN model is able to predict the traffic flow parameters of type 2 road sections during peak hours effectively compared to the classical BRNN model.

At the same time, the fluctuation of traffic flow on urban roads during the peak hours is relatively stable, and the predicted traffic flow of the optimized BRNN model is less different from the actual value. The overall mean absolute error of the traffic flow prediction of the optimized BRNN model is 1.15 veh/min lower than that of the classical BRNN model. As shown in Table 1, the BRNN optimization model achieves 95.2% accuracy in predicting peak traffic. In the experimental programming of the three sets of traffic flow forecasts, the runtime analysis of the two models in MATLAB software shows that the BRNN optimization model has an average 36% lower runtime compared to the classical BRNN model.

Model	Peak road 1	Peak road 2	Off-peak road 3
BRNN	0.108	0.131	0.115
Optimized BRNN	0.070	0.042	0.048

Table 1: BRNN and Optimized BRNN Prediction Relative Error

5 Conclusions

In this paper, we propose to use particle swarm algorithm and bilinear recurrent neural network to predict the traffic flow of urban roads. The method uses particle swarm algorithm to prune the bilinear recurrent neural network to improve the convergence speed and prediction accuracy of the algorithm, and the traffic flow prediction model based on the optimized bilinear recurrent neural network is calibrated using video data. The results show that the optimized bilinear recurrent neural network can predict the traffic flow of urban roads during peak hours and flat hours with a prediction accuracy of more than 90%, which can meet the actual traffic control prediction needs and provide a more accurate data base for the implementation of intelligent traffic control and guidance of urban roads.

References

- [1] A. Perallos, U. Hernandez-Jayo, I. J. G. Zuazola, and E. Onieva, *Intelligent transport systems: technologies and applications*. John Wiley & Sons, 2015.
- [2] J. J. Q. Yu, “Citywide traffic speed prediction: A geometric deep learning approach,” *Knowledge-Based Systems*, p. 106592, Nov. 2020.
- [3] Y. Kamarianakis and P. Prastacos, “Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches,” *Transportation Research Record*, vol. 1857, no. 1, pp. 74–84, 2003.
- [4] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, “Privacy-preserving traffic flow prediction: A federated learning approach,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, Aug. 2020.
- [5] X. Fei, C.-C. Lu, and K. Liu, “A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1306–1318, 2011.
- [6] G. Lu, J. Li, J. Chen, A. Chen, J. Gu, and R. Pang, “A long-term highway traffic flow prediction method for holiday,” in *Proc. International Conference on. Future Information Technology*, Salerno, Italy, Apr. 2018.
- [7] J. J. Q. Yu and J. Gu, “Real-time traffic speed estimation with graph convolutional generative autoencoder,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3940–3951, Oct. 2019.
- [8] J. J. Q. Yu, “Sybil attack identification for crowdsourced navigation: A self-supervised deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, 2020, in press.

- [9] —, “Semi-supervised deep ensemble learning for travel mode identification,” *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 120–135, Mar. 2020.
- [10] J. Guo, W. Huang, and B. M. Williams, “Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.
- [11] J. J. Q. Yu, W. Yu, and J. Gu, “Online vehicle routing with neural combinatorial optimization and deep reinforcement learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3806–3817, Oct. 2019.
- [12] C. Zhang, J. J. Q. Yu, and Y. Liu, “Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting,” *IEEE Access*, vol. 7, pp. 166 246–166 256, 2019.
- [13] D. Park and L. R. Rilett, “Forecasting freeway link travel times with a multilayer feedforward neural network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 5, pp. 357–367, 1999.
- [14] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, “A comprehensive survey on traffic prediction,” 2020, [arXiv:2004.08555 \[eess.SP\]](#).
- [15] J. J. Q. Yu, D. J. Hill, V. O. K. Li, and Y. Hou, “Synchrophasor recovery and prediction: A graph-based deep learning approach,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7348–7359, Oct. 2019.
- [16] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, “Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 644–654, Jun. 2012.
- [17] A. P. Marugán, F. P. G. Márquez, J. M. P. Perez, and D. Ruiz-Hernández, “A survey of artificial neural network in wind energy systems,” *Applied energy*, vol. 228, pp. 1822–1836, 2018.
- [18] J. J. Q. Yu, D. J. Hill, and A. Y. S. Lam, “Delay aware transient stability assessment with synchrophasor recovery and prediction framework,” *Neurocomputing*, vol. 322, pp. 187–194, Dec. 2018.
- [19] J. J. Q. Yu, Y. Hou, A. Y. S. Lam, and V. O. K. Li, “Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks,” *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1694–1703, Mar. 2019.
- [20] J. J. Q. Yu, A. Y. S. Lam, D. J. Hill, and V. O. K. Li, “A unified framework for wide area measurement system planning,” *International Journal of Electrical Power & Energy Systems*, vol. 96, pp. 43–51, Mar. 2018.
- [21] L. Zhu and N. Laptev, “Deep and confident prediction for time series at Uber,” in *Proc. IEEE International Conference on Data Mining*, New Orleans, LA, Nov. 2017.
- [22] J. J. Q. Yu and V. O. K. Li, “A social spider algorithm for global optimization,” *Applied Soft Computing*, vol. 30, pp. 614–627, May 2015.
- [23] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for city-wide crowd flows prediction,” in *Proc. AAAI Conference on Artificial Intelligence*, 2017.

- [24] J. J. Q. Yu and V. O. K. Li, “A social spider algorithm for solving the non-convex economic load dispatch problem,” *Neurocomputing*, vol. 171, pp. 955–965, Jan. 2016.
- [25] J. J. Q. Yu and A. Y. S. Lam, “Core-selecting auctions for autonomous vehicle public transportation system,” *IEEE Systems Journal*, vol. 13, no. 2, pp. 2046–2056, Jun. 2019.
- [26] J. J. Q. Yu, “Two-stage request scheduling for autonomous vehicle logistic system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1917–1929, May 2019.
- [27] R. A. Vasco and R. Morabito, “The dynamic vehicle allocation problem with application in trucking companies in brazil,” *Computers & Operations Research*, vol. 76, pp. 118–133, 2016.
- [28] J. J. Q. Yu and A. Y. S. Lam, “Autonomous vehicle logistic system: Joint routing and charging strategy,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2175–2187, Jul. 2018.
- [29] P. Duan, G. Mao, J. Kang, and B. Huang, “Estimation of link travel time distribution with limited traffic detectors,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3730–3743, 2020.
- [30] K. Zhang, N. Jia, L. Zheng, and Z. Liu, “A novel generative adversarial network for estimation of trip travel time distribution with trajectory data,” *Transportation Research Part C: Emerging Technologies*, vol. 108, pp. 223–244, 2019.
- [31] J. J. Q. Yu, A. Y. S. Lam, and Z. Lu, “Double auction-based pricing mechanism for autonomous vehicle public transportation system,” *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 151–162, Jun. 2018.
- [32] H. Liu and I. Lee, “End-to-end trajectory transportation mode classification using Bi-LSTM recurrent neural network,” in *Proc. 12th International Conference on Intelligent Systems and Knowledge Engineering*, Nanjing, China, Nov. 2017, pp. 1–5.
- [33] Z. Cui, R. Ke, Z. Pu, X. Ma, and Y. Wang, “Learning traffic as a graph: a gated graph wavelet recurrent neural network for network-scale traffic prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 115, p. 102620, 2020.
- [34] J. J. Q. Yu, D. J. Hill, A. Y. S. Lam, J. Gu, and V. O. K. Li, “Intelligent time-adaptive transient stability assessment system,” *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1049–1058, Jan. 2018.
- [35] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2019.