Design and Implementation of Traffic Flow Prediction Model Based on Short and Long Time Memory Network

Sheng Liu, Xinyue Li, Ting Cao(✉), and Shuxiao Chang

School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

caoting@xaut.edu.cn

Abstract. Due to the randomness, fuzziness, time variability and uncertainty of traffic flow, it is difficult for traditional forecasting models based on time series or artificial neural networks to accurately reflect the actual traffic situation, etc. This paper takes the demand for short-term traffic flow forecasting of urban rail transit as the research object, and analyzes the implementation methods suitable for short-term traffic flow forecasting. LSTM neural network was used to construct the model for simulation experiment analysis. The results of data analysis show that the LSTM neural network model obtains the minimum average absolute percentage error MAPE value of 10.6% and the highest average accuracy of 89.4%, which has a good prediction effect and can improve the prediction work of short-term traffic flow.

Keywords: Neural network · Integrated learning · LSTM · Short time traffic flow forecast

1 Introduction

In recent years, with the rapid development of our economy, the number of motor vehicles and non-motor vehicles in urban areas is also increasing. The overall number of motor vehicles has been increasing, and the problem of the sharp increase in urban traffic pressure has also followed. In order to solve many problems such as the worsening of urban traffic congestion, in the context of the continuous development of modern social science and technology, the Internet of Things and other emerging technologies have made people put forward more advanced ways to improve traffic conditions, and gradually formed the concept of intelligent transportation system (ITS).

ITS is an efficient management system, which relies on road engineering to reduce traffic congestion and natural pollution and ensure smooth traffic operation. Traffic flow prediction is a very key part of ITS [1], which can quickly help the system to achieve timely, dynamic, accurate and reliable quantitative prediction of vehicle data and future road traffic flow conditions.

V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 573, pp. 99–108, 2024.
https://doi.org/10.1007/978-3-031-65126-7_10

For the problem of traffic flow prediction, domestic and foreign scholars have conducted a lot of research, and built three research models, which are support vector machine model (SVM), deep learning model and neural network model (NN). Because of its own characteristics, SVM has been widely used in short-term traffic flow prediction, solving the problem of too small data and other problems. With the continuous development of deep learning, deep learning has also been studied in traffic flow prediction to a certain extent. The research on deep learning in traffic flow prediction is just at the beginning stage. According to the existing results, deep learning has high accuracy in predicting the results of specific data, but it also has certain limitations, and there are certain problems in the aspects of large calculation amount and long consumption time. Over the years, scholars have developed a variety of neural network models. Compared to previous machine learning, neural networks have more hidden units, using methods to abstract objects at a deeper level and extract hidden features that the data cannot see. Neural network also has the characteristics of strong adaptability.

On the whole, the development direction of short-term traffic prediction has gradually transited from linear model to nonlinear model, and from non-intelligent direction to intelligent prediction direction [2]. Based on the analysis of estimation accuracy, training model difficulty and reliability, the neural network model is usually better than the traditional model in the case of the same data, and is more suitable for short-term traffic flow prediction.

This paper will use the previous traffic data and build a model to predict the short-term traffic flow and analyze the results, so as to provide a helping hand to promote the smooth layout of ITS, reduce the current increasingly congested traffic situation, and ensure the safe and convenient travel of the people and the improvement of road travel experience.

2 Short Time Traffic Flow Forecasting Model

2.1 Model Input

The information used here is based on data from the Traffic Flow Prediction Project database collected in real time from individual detectors in the highway system across California's metropolitan areas. The time span is 1 month, 38 days from April 3 to May 10, 2018 (Table 1).

Table 1. Original data examples

Local Date	TIME	ID	CXDM	HPZL	SYXZ
2018/4/8	00:03:40	13631	K33	02	A
2018/4/28	14:38:29	41256	K31	02	A
2018/5/9	23:19:03	67293	K33	03	D
……	……	……	……	……	……

The collected data includes six data types: Local Date, TIME, ID, CXDM, HPZL, and SYXZ. Table 2 describes the meanings of each data type.

Table 2. Original fields

fieldName	Local Date	TIME	ID	CXDM	HPZL	SYXZ
Meaning	Date	Record timestamp	Vehicle number	Vehicle code	Type of license plate	Nature of vehicle use

Based on the research content, part of the data is selected for extraction, and the data used are two types: Local TIME (date) and TIME (record time).

Based on the past traffic flow of a specific section of the road, the vehicle data of the next moment can be predicted, so that it is possible to train the neural network by constructing a data set based on the past traffic flow value. Before constructing the data set, the traffic flow value of the intersection should be extracted successively in time period. Sort the traffic according to the time sequence, and then use the resample() function in Pandas to summarize the time data in a period of 5 min when the traffic passes through the intersection. The summary results are shown in Table 3.

Table 3. Extracted data

Time	Volume
……	……
2018/4/8 0:05	130
2018/4/8 0:10	155
2018/4/8 0:15	125
2018/4/8 0:20	124
2018/4/8 0:25	111
2018/4/8 0:30	117
2018/4/8 0:35	91
2018/4/8 0:40	119
……	……

After pre-processing and time series value extraction, the traffic data passed by the intersection has been summarized into a five-minute time span of traffic flow data and stored in the document. All data starts at 2018-04-03 00:00:00 and ends at 2018-05-31 23:55:00. The data span is 5 min.

In this paper, the vehicle flow data of 30 days from midnight of April 3 to evening of May 10 are divided into four conditions according to the actual situation: working day, holiday, rainy day and traffic control. According to the situation, the vehicle flow data of the first three days are constructed and trained as a sample set and retained as a vehicle flow prediction set of the following day to test the accuracy of the model.

In this paper, vehicle data at n time intervals before the road surface is used to estimate vehicle data at a subsequent time unit. Therefore, the sample set is constructed by inputting data at n time from past time into the network, where X represents the input network arrangement, the data at n + 1 time becomes the network output, and Y is taken as the network output arrangement. Neural networks generally associate input and output with information. After training, the network will form another input network and obtain a new output by direct association with specific data. Therefore, network input X and network output Y jointly construct the sample set of the experiment, and the data set is obtained in the form of rolling forward sampling, extracting n + 1 pieces of information at a time. Previously, n pieces of information were input X, and rolling prediction was also made in the prediction to build a larger number of samples. Therefore, in order to process the experimental data sequence into a short-duration memory network which can adapt to the data arrangement, it is necessary to use functions to complete the above requirements. There are a total of 3 * 288 = 864 experimental data in the three days from 4–8 to 4–10. When m + 1 experimental data is set at one time, the sample number constructed when m + 1 experimental data is used as the input of the experiment and m + 1 experimental data is used as the output of the experiment is (864-m) item. A data type in the constructed training sample set is shown in Table 4 (m = 11).

Table 4. Data after sequence transformation

X
……
130,155,125,124,111,117,91,119,79,112,81
155,125,124,111,117,91,119,79,112,81,88
125,124,111,117,91,119,79,112,81,88,71
124,111,117,91,119,79,112,81,88,71,78
111,117,91,119,79,112,81,88,71,78,118
117,91,119,79,112,81,88,71,78,118,82
91,119,79,112,81,88,71,78,118,82,71
119,79,112,81,88,71,78,118,82,71,83
……

Since the sigmoid function is used in the hidden layer of the experimental network, in order to achieve the speed of network convergence and prevent the problem of neuron saturation, it is generally required to normalize the training information. Here, MinMaxScaler() function in numpy library is selected to normalize the data.

$$X_{std} = \frac{X - X.\min(axis = 0)}{X.\max(axis = 0) - X.\min(axis = 0)} \tag{1}$$

$$X_{scaled} = X_{std} * (max - min) + min \tag{2}$$

2.2 Model Output

In this paper, Tensorflow + keras architecture is used to construct the prediction model of long and short term memory network and train the neural network model with data set. As an open library [3], Tensorflow is a symbolic mathematical system based on data stream programming, which has been widely used in the research of machine learning, deep neural networks and other fields. Tensorflow is composed of multi-level institutions, which can use GPU and TPU for numerical calculation, and supports C and Python, which is completed in python language in Pycharm. In the process of using Tensorflow to build a neural network model, the environment should first be built. The version selected here is Tensorflow-GPU-2.6.0, and suitable CUDA should be installed. Secondly, various necessary packages should be imported into Pycharm. Such as keras, matplotlib, MKL sklearn etc., after setting various parameters, in this paper, the model set to 2 layer LSTM model. The process of constructing LSTM neural network using Tensorflow library is as follows: First, build a Sequential model with Sequential() to add layers. When building the neural network model, it is necessary to set the parameters of the model in the program, which has been introduced before. The number of hidden neurons in the first layer LSTM network is 50 and the output dimension is 50. Return_sequences is set to True and only the output of the last state is returned. The output dimension of the second layer is 100, the output dimension of the Dense layer is 1, and the activation function is liner.

After the traffic flow prediction model is built, it is necessary to train the neural network model with the sample set. In this paper, the loss function loss is defined as the mean square error function. RMSprop is used in the optimizer and batch gradient descent algorithm is adopted. According to the above, the training batch size is 8, which means that 8 data are extracted from the training set at a time for model training. During the experiment, the data of the training set was continuously converted into the prediction model, and the prediction results were compared with the real data of the verification set [4]. After the deviation was obtained, the backpropagation algorithm was used to optimize the training network, and the model was continuously trained. In this paper, 300 iterations of training were set as the end condition.

3 Related Technology

3.1 The Internal Structure of Long - Term Memory Network and the Calculation Method of Data

Long term memory neural network has completely overcome the shortcomings of ordinary RNNS [5], and is the most widely used RNN at present, which is widely used in many fields such as speech and picture recognition [6], natural language processing [7], emotion recognition [8] and so on. In the LSTM neural network, in addition to the short-term input signal sensitive state h, the cell state c is added and used to store the long-term state.

The LSTM neural network uses two gates to control the cell state c, one is the amnesia gate, whose function is to judge how many cell states exist from the last time c_{t-1} to the time c_t, and the other is the input gate. Its function is to determine how much timely input x_t to the cell state c_t. Another gate is the output gate [9], whose function is to determine how much cell state c_t is output to the LSTM's current output value (Fig. 1).

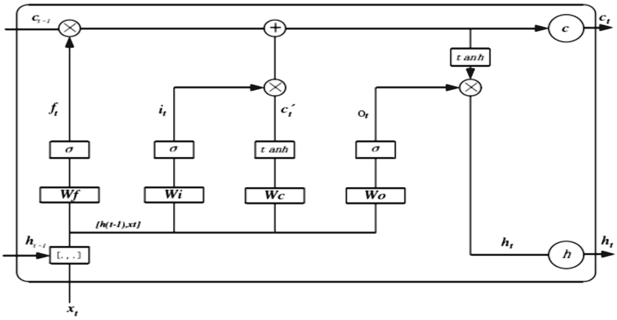

Fig. 1. Internal calculation flow diagram of a short-duration memory neural network unit

In the LSTM network, forward propagation and computation of information are also carried out through neuron transmission, and the network forward computation can be expressed by six formulas [10]. The first is the forgetting door and the calculation of the forgetting door is:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \tag{3}$$

The input gate is calculated as follows:

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{4}$$

The unit state describing the current input is calculated based on the output of the previous moment and the input of the current moment, and its calculation expression is as follows:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t} \tag{5}$$

$$\tilde{c_t} = tanh(W_c \times [h_{t-1}, x_t] + b_c) \tag{6}$$

The number of gates that control the long-term memory acting on the current instantaneous output is output gate o_t, and its calculation formula is (7).

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \tag{7}$$

The output result of the cell is determined by the output gate o_t and the cell state c_t, expressed as follows:

$$h_t = o_t * tanh(c_t) \tag{8}$$

The above formula 8 is the forward calculation expression of the whole long and short time memory network.

3.2 Neural Network Training Steps for Long and Short Time Memory

Based on the above error propagation formula, the training steps of LSTM neural network are as follows:

Firstly, the output value of each neuron is calculated by the forward calculation formula, and the output value of the whole network is calculated according to the output value of each neuron;

The total output is compared with the actual value, the total error value is obtained, and the deviation of each neuron is calculated by the back propagation algorithm; The gradient descent algorithm is used to calculate and update each weight gradient according to the corresponding error value; After the new network weight is obtained, the forward calculation formula is continued to calculate the output of each neuron according to the new input data, and the final output of the network is obtained and then compared with the actual value.

When a certain error accuracy is reached, the network parameter is saved. At this time, when the network training is completed, if the error accuracy is not reached, the iteration is carried out continuously. Until the network output reaches a certain error accuracy.

4 Experimental Results and Analysis

4.1 Software Running Environment and Hardware Configuration

In terms of hardware environment, the Windows OS version is win11, the operating system is 64-bit, the processor model is Intel i5 8300 h, the display adapter is NVIDIA GTX1050ti, and the memory is 16 GB. In terms of software environment, the programming language is python, the python version is 3.6.8, and pycharm is selected as the integrated development environment.

4.2 Operation Results and Analysis

The built training set data was fed into the prediction model, and four different datasets were built using real-time data collected from individual detectors in the highway system across California's metropolitan areas between April 8 and May 10, 2018, classified by four different conditions: weekdays, holidays, rainy days, and traffic control days. Each part of the data is 4-day traffic flow data, and the sample set is built with the number of vehicles passing through high-speed cameras every 5 min. The vehicle flow prediction model of a short-duration memory network is constructed and the trained model is saved. The model input is obtained by the sequential sampling method described above, and then the vehicle flow of the same day is predicted on a rolling basis.

Here, plt.plot() function in Matplotlib is used to plot the prediction results and the real values, and 4 graphs of the traffic flow prediction results under different scenarios are obtained. Figures 2, 3, 4 and 5.

As shown in the figure above, the traffic flow of the following day is predicted based on the traffic flow data of the previous three days, and the total number of samples is 1152 each time. The 864 samples of the first three quarters are used as training sets,

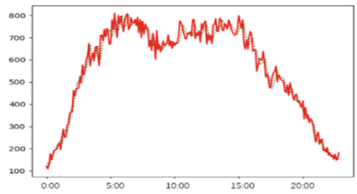

Fig. 2. Forecast of weekday traffic flow

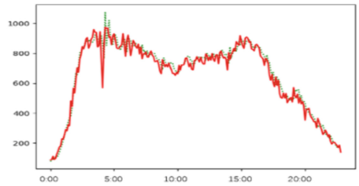

Fig. 3. Forecast of holiday traffic flow

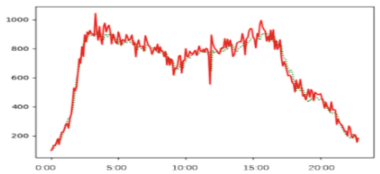

Fig. 4. Forecast of vehicle flow in rainy days

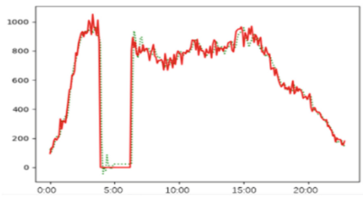

Fig. 5. Flow prediction under traffic control

which are brought into the model for training, and the prediction results are obtained. Use the plotting function to represent the prediction results. Among them, the real value is set as the red solid line, and the prediction result is the green dotted line. It can be seen from the figure that the short-term vehicle flow prediction model based on LSTM has a better vehicle flow prediction result for working days.

In general, the relative error of the sequence value of traffic flow on the predicted date can be obtained from the figure. To a certain extent, the main reason for the error in the analysis is that the model only considers the time characteristics of vehicle flow while other influencing factors are not taken into account, and there are not enough data vehicles in the training set. Although the relative error of a few predicted values is slightly larger, most of the results can meet the requirements.

This paper also introduces the mean absolute percentage error (MAPE), which indicates the accuracy of the model, and from the numerical values in the MAPE, how accurate the model is (Table 5).

Table 5. Lists the data.

Map	MAPE value
Working day	0.083704
Holidays and festivals	0.106437
Rainy day	0.126403
Traffic control day	0.107493

The average MAPE value of the calculated scene is 0.106, indicating that the average error is about 10.6% and the prediction accuracy is 89.4%, indicating that the model has a high precision forecast of short-term traffic flow.

5 Conclusion

As an important part of ITS, traffic flow forecasting system based on LSTM provides great help to people's travel and traffic management department's work. The system uses python language, uses Tensortflow + keras architecture to establish LSTM prediction model, and processes the acquired data set, which is divided into training set and test set. The prediction results of the training set are compared with the test set, and a good prediction accuracy is obtained. The model can be used to predict the future traffic flow, which provides help for people's travel and the work of relevant departments.

Acknowledgment. This work was supported in part by the Special Project of Technological Innovation and Guidance in Shaanxi Province under Grant 2022QFY01-03, in part by the Natural Science Foundation in Shaanxi Province under Grant 2022JQ-476, and in part by the Natural Science Foundation of Deduction Department in Shaanxi Province under Grant 2022JK0474, and by Science and Technology Program in Xi'an city under Grant 21XJZZ0055.

References

1. Wei, C.: Research on Taxi Shelter Design based on Service Design Concept. Xi 'an Polytechnic University (2017)
2. Liu, C.: Research on Urban Macro Travel Speed Prediction based on Classical Model and LSTM Model. Beijing Jiaotong University (2019)
3. Bonnin, R.: TensorFlow Machine Learning Project. Yao Pengpeng, trans. Beijing: People's Posts and Telecommunications Press (2017)

4. Zhi-Liang, D., Yi-Qun, P., Jiantong, X., et al.: Application of reinforcement learning algorithm in operation optimization of air conditioning system . Build. Energy Conserv. **7**, 7 (2020)
5. Wang, X., Wu, J., Liu, C., Yang, H., Du, Y., Niu, W.: Fault time series prediction based on LSTM recurrent neural network. J. Beijing Univ. Aeronaut. Astronsut. **44**(4), 13 (2018)
6. Zhao, S., Dong, X.: Research on speech recognition based on improved LSTM deep neural networks. J. Zhengzhou Univ. Eng. Ed. **39**(5), 5 (2018). (in Chinese)
7. Ren Zhihui, X., Haoyu, F.S., et al.: Chinese lexical segmentation for sequence annotation based on LSTM network. Appl. Res. Comput. **34**(5), 5 (2017)
8. Rao, Q.: Research on Dimension Emotion Recognition based on Context. Jiangsu University (2017)
9. Lei, X.: Research on Short-time Vehicle Flow Prediction Model based on Integrated LSTM. Chongqing University of Posts and Telecommunications (2019)
10. Zhang, X.W., Li, Y.Y., Huang, S., et al.: Population situation prediction method of COVID-19 based on LSTM. CN111798991A (2020)