Long Short Term Memory based on Differential Evolution in Passenger Flow Forecasting

Minwei Shuai [a], Huixin Tian [b]
The School of Electrical Engineering & Automation, Key Laboratory of Advanced Electrical Engineering and Energy Technology, Tianjin Polytechnic University, Tianjin, China
Email: [a] monologuesmw@163.com, [b] icedewl@163.com

***Abstract*—High-speed railway passenger flow forecasting is an essential component of intelligent transportation system. To enhance the forecasting accuracy, a hybrid model combining long short term memory (LSTM) and differential evolution algorithm (DE) was proposed in this paper. DE-LSTM overcomes the shortcomings of the LSTM based on back propagation through time (BPTT) which is easy to converge to the local optimal solution. The proposed model is applied to the prediction of passenger flow for Guangzhou High-speed railway. By using the real data, the performance of the DE-LSTM was compared with the LSTM. Mean Absolute Percentage Error (MAPE) of DE-LSTM have an extra 2.3988 decreasing over the LSTM. The results show that the DE-LSTM outperforms LSTM.**

***Index Terms*—traffic forecasting; high-speed railway passenger flow; LSTM; DE**

I. INTRODUCTION

High-speed railway (HSR) systems have been developed rapidly in China and various other countries throughout the past decade. Since then, high-speed railway transit has become one of the high-quality transportation systems that deliver fast, comfortable, safe, and cost effective services. As the most popular transportation system, HSR has undertaken a large number of passenger missions, which has played a major role in relieving passenger transport pressure. However, the number of passenger flows will fluctuate differently depending on the date attribute. Therefore, the passenger flow prediction is basis for the design, construction, operation, and adjustment of a HSR system. It is beneficial for improving the transport service, giving early warnings when there is a peak of passenger flow, and making the HSR system smarter, safer and more reliable.

The development of data acquisition technology is applied to the real society to provide us with a more effective way for addressing the issues of passenger flow predict. Continuous-real data of passenger flow contains rich information which makes it possible to explore the underlying laws of passenger flow. On the other hand, machine learning is rapidly developing and widely used which creates more opportunities for intelligent transportation systems to meet future demand.

Many research scholars focus on the study of passenger flow prediction and many successful models of passenger flow prediction have been developed. These can be generally divided into three categories: single machine learning-based models, hybrid models, and deep learning models.

Single machine learning models. Zhang et al developed a Kalman filtering method to predict the short-term passenger flow based on characteristics analysis [1]. Consider time series has a strong autocorrelation of seasonal characteristics, Miloš Milenković et al proposed a Seasonal Auto Regressive Integrated Moving Average (SARIMA) method to predict monthly passenger flows on Serbian railways [2]. Consider impact of influenced factors, Shiao et al predicted MRT passenger flow with random forest [3].

Hybrid models. YanrongHu et al established the regressive model based on the least square support vector machine (LS-SVM) [4].Wei Y et al combines empirical mode decomposition (EMD) and back-propagation neural networks (BPN), proposed a hybrid EMD-BPN method to forecast the short-term metro passenger flow [5]. Sun Y et al proposed a novel hybrid model Wavelet-SVM to predict different kinds of passenger flows [6]. Jia, Y. et al proposed a new combined forecasting model based on the GM and ARMA [7]. Li Y et al proposed a novel multi-scale radial basis function (MSRBF) network for forecasting the irregular fluctuation of subway passenger flows [8]. Combine symbolic regression and Auto regressive Integrated Moving Average Model (ARIMA), Li L et al proposed a SR-ARIMA model [9]. Liu L et al proposed a stacked auto encoders deep neural network (SAE-DNN) model to predict BRT passenger flow [10].

Deep learning models. Koesdwiady A et al proposed a method that it incorporates deep belief networks (DBN) for traffic and weather prediction and decision-level data fusion scheme to enhance prediction accuracy using weather conditions [11].

In this paper, we propose a novel passenger flow prediction model using deep learning method and apply it to predict the daily passenger flow for Guangzhou HSR. First, the fluctuation of passenger flow has a strong time-series dependence on the attributes of the date, and long short term memory (LSTM) can learn the long-term dependence between time series data. Then, the traditional LSTM uses BPTT algorithm to optimize the network parameters, but BPTT algorithm is highly complex and easy to converge to the local optimal solution. Conversely, the differential evolution algorithm (DE) has the characteristics of fast convergence and strong search ability. Therefore, we propose a LSTM model based on DE (DE-

LSTM) and apply it to predict passenger flow. Finally, to prove the accuracy of the forecasting results, we compare the values of the loss function of DE-LSTM and LSTM during training process. Moreover, the errors of the two algorithms on the test data are compared. The results show that the DE-LSTM outperforms LSTM.

II. PREDICTION MODEL

A. LSTM Networks

LSTM networks is an improved recurrent neural networks (RNN) and it inherited the characteristics of "underlying topology of inter-neuronal connections" and designed to learn long-term dependencies. In other words, LSTM networks are capable of overcoming the inherent problems of RNN, vanishing and exploding gradients. They were first proposed by Hochreiter and Schmidhuber [12], and refined by Graves [13].

LSTM networks consist of an input layer, one or more hidden layers, and an output layer. As with traditional neural networks, the number of input layer neurons is equal to the number of feature spaces, and the number of output layer neurons reflects the output space. The difference between the LSTM networks and the RNN is mainly contained in the hidden layer consisting of memory cells. Each of the memory cells has three gates which are controlling and adjusting the cell state. In other words, three gates are actually logical unit structure, that is, an input gate (i_t), a forget gate (f_t), and an output gate (o_t) to separately control the information accumulation speed, selectively forgetting the preciously accumulated information, and selectively output the memory information. The structure of a memory cell is illustrated in Figure 1. Where "○" indicates element-wise multiplication. And it can be seen, the inputs of three gates are sequence of the current moment x_t and the output of the memory cell of the previous moment h_{t-1}. The three gates use the sigmoid function to map the input to [0, 1]. Each playing a different role:

- The forget gate defines which information to remove from the memory. It can also be seen from the structure in Figure 1. That it determines the degree of influence of C_{t-1} on C_t.
- The input gate defines which information to add the memory. It can also be seen from the structure that it determines the degree of influence of x_t on C_t.
- The output gate defines which information from the memory. It can also be seen from the structure that it determines the degree of influence of C_t on h_t.

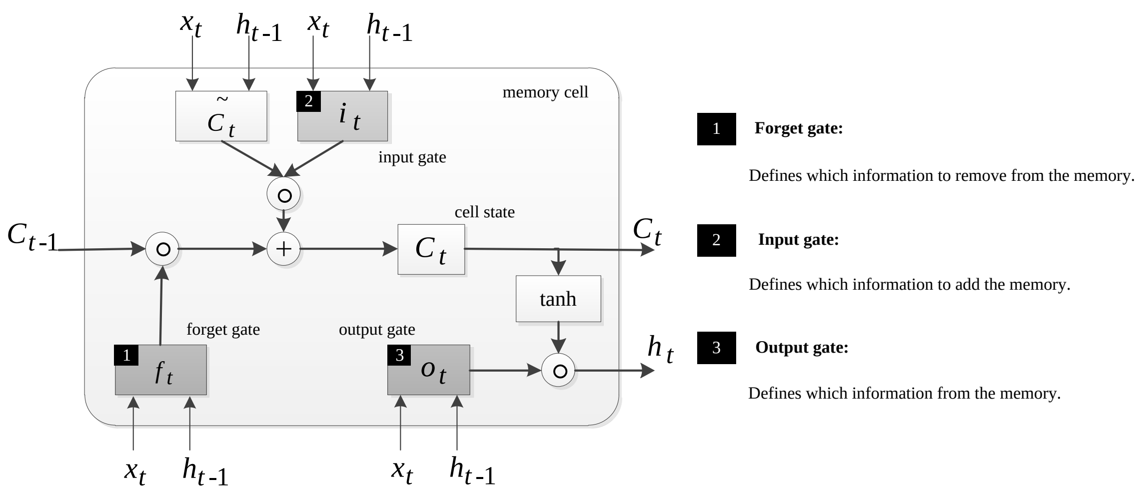

Figure 1. Structure of LSTM memory cell.

In summary, the forgetting gate and the input gate act on the adjustment of the state of the memory cell C_t, and the adjusted memory cell state and the output gate together determine the output of the hidden layer. The specific operation process is as follows:

In the first step, the forget gate f_t determines which information should be remove from the cell states C_{t-1}. Proportion of memory is computed based on the current input x_t, the outputs h_{t-1} of the memory cells at the previous moment, and the bias term b_f of the forget gates. The sigmoid function finally maps the result of the calculation into the range between 0 (completely forget) and 1 (completely remember):

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \quad (1)$$

In the second step, the input gate determines which information should be added to the cell state C_t. This procedure contains two operations: first, the intermediate variable $\tilde{C}_t$ that could be added to the cell state should be calculated, and then the output of the input gate should be calculated:

$$\tilde{C}_t = \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \quad (2)$$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \quad (3)$$

And then, the new cell state C_t is calculated based on the result of $\tilde{C}_t$ and i_t :

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \quad (4)$$

In the last step, the output of the output gate o_t and the output of the memory cell h_t as follows:

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \quad (5)$$

$$h_t = o_t \circ \tanh(C_t) \quad (6)$$

Finally, the calculation from the hidden layer to the output layer is as follows:

$$y_t = \sigma(W_{hy}h_t + b_y) \quad (7)$$

The forward calculation is performed according to the above formula, and the calculation result is the output of the model, and then the BPTT algorithm is used for training, and 14 parameters can be determined.

B. Differential Evolution Algorithm

DE is a heuristic random search algorithm based on group difference [14], its principle is very similar to genetic algorithm (GA). The framework of DE consists of three parts: mutation operation, cross operation and selection operation.

$$v_i^{k+1} = X_{i_3}^k + F(X_{i_1}^k - X_{i_2}^k) \quad (8)$$

Where F is the scaling factor, k is current evolution generation. $(X_{i1}^k - X_{i2}^k)$ is a random difference vector, and X_{i3}^k is base vectors.

The cross operation is generating a new individual by crossing the parent and the mutated individuals according to the crossover probability CR. The cross operation is formulated as below:

$$u_{ij}^{k+1} = \begin{cases} v_{ij}^{k+1} & rand < CR \mid j = randn(i) \\ X_{ij}^k & otherwise \end{cases} \quad (9)$$

The selection operation is a greedy choice mode.

III. Research Method

The determination of the parameters of LSTM is iterated during the training process by BPTT algorithm. However, the BPTT algorithm has high computational complexity and may converge to local optimal values. DE has the characteristics of fast convergence, simple structure, low space complexity and easy implementation. Therefore, we propose a LSTM model based on DE.

The network structure used in the model is the same as in Figure 1. It can be seen from (1)-(6) that there are 14 parameters in LSTM, which are W_{fh}, W_{fx}, b_f, W_{ih}, W_{ix}, b_i, W_{oh}, W_{ox}, b_o, W_{ch}, W_{cx}, b_c, W_{hy}, b_{hy}. Their respective dimensions are as follows: $W_{fh} \in R^{hidden \times hidden}$, $W_{fx} \in R^{hidden \times lookback}$, $b_f \in R^{hidden \times 1}$, $W_{ih} \in R^{hidden \times hidden}$, $W_{ix} \in R^{hidden \times lookback}$, $b_i \in R^{hidden \times 1}$, $W_{oh} \in R^{hidden \times hidden}$, $W_{ox} \in R^{hidden \times lookback}$, $b_o \in R^{hidden \times 1}$, $W_{ch} \in R^{hidden \times hidden}$, $W_{cx} \in R^{hidden \times lookback}$, $b_c \in R^{hidden \times 1}$, $W_{hy} \in R^{hidden \times predstep}$, $b_c \in R^{predstep \times 1}$. Where $hidden$ is number of neurons in the hidden layer, $lookback$ is the number of neurons in the input layer, $predstep$ is the number of neurons in the output layer.

Therefore, when the parameters are optimized using the DE, the code length of the individual in the population can be calculated as follows:

$$\begin{aligned} &4 \times (hidden \times hidden) + 4 \times (hidden \times lookback) + \\ &4 \times (hidden \times 1) + prestep \times (hidden + 1) \end{aligned} \quad (10)$$

The value of the loss function is usually used to describe the effect of the LSTM in the training data. When using the DE algorithm for network parameters optimization, it is necessary to judge the merits of each individual by fitness function value. Therefore, the fitness function can be calculated as follows:

$$loss = \sum_{i=1}^{n} (\overline{y_i} - y_i)^2 \quad (11)$$

where n is the number of training samples, $\overline{y_i}$ is the predicted value, y_i is the true value.

Detailed explanations of DE algorithm optimize LSTM network parameter procedures are given below:

Step 1: Initialize the relevant parameters in LSTM network, including $hidden$, $lookback$, $predstep$ and epoch $Iter$.

Step 2: Initialize the relevant parameters of the DE, including population size $popsize$, the dimensions of individual D, scaling factor F, and crossover possibility CR.

Step 3: Initialize the population $X_{i,j}$, $i \in [1, popsize]$, $j \in [1, D]$, and $X_{i,j} \in [l_j, u_j]$. Where l_j and u_j represent the range of values for each dimension of the individual.

Step 4: Each individual is divided for the network parameters of the LSTM and calculated the fitness value $loss$ for each individual.

Step 5: The best fitness value of the initial population is the output of the first iteration of the LSTM, and the output of the second iteration is the best fitness value of the initial population after a DE iteration.

Step 6: Repeat step 5 until the number of iterations reaches $Iter$.

Step 7: Return the final LSTM network parameters.

IV. Experiments

In this chapter, the proposed DE-LSTM is tested in the passenger flow forecast. The 438 daily passenger flow dataset was extracted from Guangzhou high-speed railway from January 1, 2015 to March 14, 2016. To prove the accuracy of the forecasting results, we compare the values of the loss function of DE-LSTM and LSTM during training iteration. Then, the model is applied to the test data, and the error of the test result is compared.

A. *Structure and Key Parameter Specifications*

The parameters of the DE-LSTM algorithm were set to the following: population size $popsize = 30$, scaling factor $F = 0.7$, crossover possibility $CR = 0.4$; in additional, $lookback = 3$, $hidden = 10$, $prestep = 1$, and $Iter = 2000$.

B. *Prediction Results*

Based on the parameters selection, the models were built with 384 train data and 50 test data. Then, the error curves of DE-LSTM and LSTM on training data sets could be obtained which was shown in Figure 2. It can be seen that the convergence speed of LSTM is very fast, and it has converged when the number of iteration is 80.

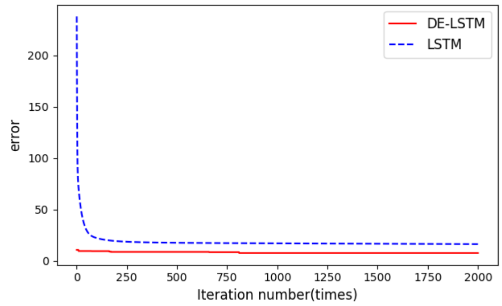

Figure 2. Error curves of DE-LSTM and LSTM on training data sets.

However, as the number of iterations increases, the convergence speed is slow. Conversely, the convergence of DE-LSTM is very good and always better than LSTM.

Figure 3 shows the predicted passenger flow based on LSTM and DE-LSTM. Figure 4 shows the prediction errors (Absolute Relative Error, ARE) of different models, and the errors by the LSTM are larger. Tab.1 shows the performance of the prediction based on LSTM and DE-STM. This result means that the DE-LSTM has better accuracy and generalization performance.

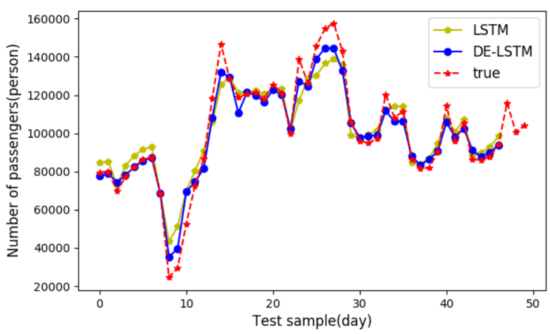

Figure 3. The result of passenger flow forecast.

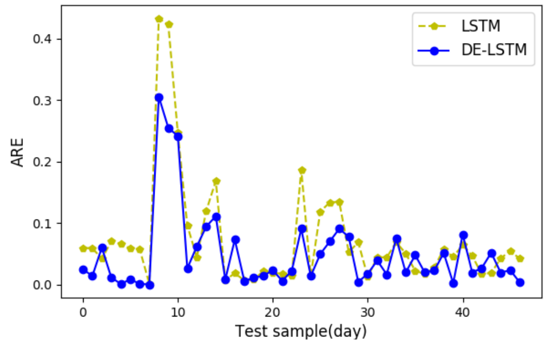

Figure 4. The result of prediction errors.

Table 1. The performance of passenger flow forecast based on different methods.

Method	MAPE	Max ARE	ARE
LSTM	7.3429	0.4324	0.0734
DE-LSTM	4.9441	0.3052	0.0494

V. CONCLUSION

The main contribution of this paper is to present a DE-LSTM model to predict the daily passenger flow. This model can be used not only for the prediction of passenger flow, but also for the prediction of time series. We present two major conclusions:

(1) The LSTM model based on BPTT algorithm has high complexity and unstable convergence effect in the training process. In this paper, we propose a DE-LSTM, which uses DE instead of BPTT in the training process and achieves good results.

(2) The DE-LSTM algorithm takes a long time in the training process, but DE converges quickly, so the epoch can be reduced, thereby reducing the training time.

ACKNOWLEDGMENT

This paper was supported by the National Natural Science Foundation of China under Grant 61403277 and Grant 71602143, Tianjin Natural Science Foundation (18JCYBJC22000), Tianjin science and technology correspondent project, Tianjin high school innovation team training pro-gram (TD13-5036). Tianjin Science and Technology Correspondent Project (18JCTPJC62600).

REFERENCES

[1] C. H. Zhang, R. Song, Y. Sun, “Kalman filter-based short-term passenger flow forecasting on bus stop”. Transport. Syst. Eng. Inform. Technol. 2011, 11 (4): 154-159.

[2] Miloš Milenković, Libor Švadlenka, V. Melichar, et al., “SARIMA modeling approach for railway passenger flow forecasting,” Transport, 2015: 1-8.

[3] Y. C. Shiao, L. Liu, Q. Zhao, et al., “Predicting passenger flow using different influence factors for Taipei MRT system,” IEEE international conference on adaptive science technology, 2017.

[4] Yanrong Hu, Chong Wu, Hongjiu Liu, "Prediction of Passenger Flow on the Highway Based on the Least Square Suppoert Vector Machine," Transport, 2011, 26 (2): 7.
[5] Y. Wei and M. C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," Transportation Research Part C Emerging Technologies, 2012, 21 (1): 148-162.
[6] Y. Sun, B. Leng, W. Guan, "A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system," Neurocomputing, 2015, 166 (C): 109-121.
[7] Y. Jia, P. He, S. Liu, L. Cao, "A combined forecasting model for passenger flow based on GM and ARMA," Hybrid Inform. Technol. 2016, 9 (2), 215–226.
[8] Y. Li, X. Wang, S. Sun, et al., "Forecasting Short-term Subway Passenger Flow under Special Events Scenarios using Multiscale Radial Basis Function Networks," Transportation Research Part C Emerging Technologies, 2017, 77: 306-328.
[9] L. Li, Y. Wang, G. Zhong, et al., "Short-to-medium term passenger flow forecasting for metro stations using a hybrid model," KSCE Journal of Civil Engineering, 2017.
[10] L. Liu and R. C. Chen, "A novel passenger flow prediction model using deep learning methods," Transportation Research Part C: Emerging Technologies, 2017, 84: 74-91.
[11] A. Koesdwiady, R. Soua, F. Karray, "Improving Traffic Flow Prediction with Weather Information in Connected Cars: A Deep Learning Approach," IEEE Transactions on Vehicular Technology, 2016, 65 (12): 9508-9517.
[12] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," Neural Computation, 1997, 9 (8): 1735-1780.
[13] A. Graves. "Long Short-Term Memory" Supervised Sequence Labelling with Recurrent Neural Networks. Springer Berlin Heidelberg, 2012, 1735-1780.
[14] K. Price and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," Kluwer Academic Publishers, 1997.

Huixin Tian received the Ph.D. degree in 2009 from Information Science and Engineering, Northeastern University, Shenyang, China.
Now she is an associate professor and a master's supervisor in School of Electrical Engineering & Automation and Key Laboratory of Advanced Electrical Engineering and Energy Technology, Tianjin Polytechnic University. Her main research interests are modeling, control and optimization in complex industrial system.

Minwei Shuai is currently working towards the master's degree in control theory and control engineering at Tianjin Polytechnic University. His research interests include modeling and optimization in complex industrial process and soft sensor technology.