



Automatic Prediction of Traffic Flow Based on Deep Residual Networks

Rui Zhang^{1,2,3(✉)}, Nuofei Li³, Siyuan Huang³, Peng Xie³, and Hongbo Jiang⁴

¹ Hubei Key Laboratory of Transportation Internet of Things, Wuhan University of Technology,
Wuhan 430072, Hubei, China
zhangrui@whut.edu.cn

² Hubei Key Laboratory of Inland Shipping Technology, Wuhan University of Technology,
Wuhan 430072, Hubei, China

³ School of Computer Science and Technology, Wuhan University of Technology,
Wuhan 430072, Hubei, China
linuofeirz@126.com, yinshu@whut.edu.cn, xiepeng0715@gmail.com

⁴ School of Electronic Information and Communications,
Huazhong University of Science and Technology, Wuhan 430074, Hubei, China
hongbojiang2004@gmail.com

Abstract. Traffic flow often contains massive amounts of information that is related to location and shows some regularity. And the traffic flow analysis based on trajectory data has become one of the most popular research topics in recent years. With the wide application of deep learning and for its higher accuracy than other approaches, methods such as convolution neural network and deep residual network have been introduced in traffic flow research and achieve good results. However, these methods usually require the training of a large number of parameters, which leads to some problems. For example, frequent manual adjustment is needed, and some parameters cannot be dynamically adjusted with the training process. We find that learning rate plays a crucial role in all parameters, which has important influence on the training speed of the residual network. In other words, the soundness of traffic flow predication results depends on the learning rate. Hence, we propose G4 algorithm to automatically determine the learning rate. It can be adjusted automatically in the process of trajectory data mining, and therefore solve the traffic flow prediction problem. Experiments on real data sets show that our method is effective and superior over some traditional optimizing methods of traffic flow analysis.

Keywords: Automatic prediction · Traffic flow · Trajectory
Deep residual network · Fourier series

1 Introduction

Trajectory data contains large amounts of information, has close relations with geographic location or point of interest (POI), and can reflect general regularity. Therefore, analyzing traffic flow based on trajectory data has become a hot research direction in recent years. For example, Masahiro et al. found that the frequency of car travel will

not change with the season through the analysis of the GPS data of Hakodate city, while the frequency of the cycling and hiking is severely affected by the change of season [1]. Through the analysis of MIT trajectory data set of vehicles and pedestrians, Dheeraj et al. successfully found out the representatively abnormal phenomena [2]. At the same time, some deep learning methods are introduced in the field of trajectory data mining and achieve high accuracy and small error. For example, for the same topic of traffic flow and pedestrian flow analysis in the center of the city, Stefan et al. managed to use the convolutional neural network to largely accelerated the speed of training [3]. Xiao et al. even proposed the concept of ensemble learning to better study hybrid transportation modes [4].

However, despite their good performance in the field of trajectory data mining, existing deep learning methods still face some problems, especially a large number of parameters and hyper parameters relying heavily on manual adjustment. Zheng et al. used three convolution neural networks to analyze the GPS data in Beijing, predicted the traffic and pedestrian flow at a certain spot, and achieved good results, but there were over a dozen parameters relying on manual adjustment such as smoothness, periodicity and trend which had a direct bearing on the final results [5]. In addition, certain parameters may need to change with the learning process. Song et al. proposed the model of DeepMob to analyze GPS data to help humans avoid natural disasters [6]. According to our research, whether parameters such as learning rate change with the process will greatly influence traffic flow analysis results, in this case, i.e., will affect the final outcome of disaster analysis. The reason is that the prediction of any flow is indispensable to the learning of the existing data, and the learning rate affects the learning speed, thus affecting the final effect.

To solve these problems, our goal is to develop methods that allow the parameters of some deep learning methods to adapt to the learning process automatically and reduce human intervention. To achieve this goal, we consider the following aspects. First, the parameters we study should be applicable to many different methods, rather than individual, specific parameters in certain methods so that a greater variety of traffic flows can be analyzed. Second, our method should be able to adjust parameters spontaneously, thus reducing human intervention and improving analysis performance. Third, our method should be superior to some un-optimized methods and achieve the task without compromising the effect. Therefore, we propose G4 (Gradient FOURier series) algorithm to automatically determine the learning rate so that it can be adjusted automatically in the process of trajectory data mining and solve traffic flow prediction problems. We were first inspired by the Fourier series of signal processing, then built connections between learning rate adjustment and some parameters in the model through Fourier series, and applied it in the deep residual network, finally to address practical problems such as traffic flow. Our main contributions are as follows:

- We proposed G4 algorithm to automatically determine the learning rate of a series of deep learning methods.
- We integrated the algorithm into the deep residual network model, and reduced human intervention in the process of trajectory data analysis.
- According to experiments on real data sets, our method outperforms some traditional analysis methods.

The respect of this paper is organized as follows. Section 2 describes the related work. The third section gives the definition of problem and its mathematical description. The fourth section presents the framework and detailed implementation of the method. Section 5 evaluates our method through experiments. Section 6 concludes the paper.

2 Related Work

In this section, we explain some other work related to our research, including a brief introduction to Fourier series and some improvements achieved by other scholars in traffic flow analysis.

2.1 Fourier Series

In electronic technology, Fourier series is used for signal transformation, so that it can be restored with some simple signals. The single entry form of the Fourier series is:

$$f(x) = c_n e^{i \frac{n\pi x}{l}} \quad (1)$$

where c_n , x , l , i represent coefficient, time (signal changes over time), half period, and imaginary unit, respectively. Note that if a certain time and a certain semi-cycle are given, the size of a signal can be determined. In practice, signals produced by electronic devices are often very complicated and cannot be described with simple mathematical laws. The Fourier series describes a way that can transform any form of signal into a summary of several simple periodic functions.

2.2 Other Work

Many scholars have analyzed traffic flow by improving the learning rate. For example, Sun et al. learnt human walking trajectories using RMSProp, and then predicted human trajectory [7]. Gang et al. even made clear that the deep learning model has greatly improved the analysis of group movement behaviors with the use of RMSProp. However, although methods such as RMSProp and Adam have performed very well, yet they require manual adjustment of the decay rate. When using Adam, the user must adjust two different decay rates. These parameters must be configured manually. In other words, this type of methods actually replaces the adjustment of learning rate with the adjustment other parameters, and has not solved the problem fundamentally. Even the number of parameters that need to be adjusted may increase rather than decrease. On the other hand, although some approaches used by researchers do not increase the number of parameters requiring manual adjustment, these methods often achieve general improvements for learning rate, rather than for the unique model or approach for the analysis of traffic flow. In other words, these methods may fail to take into account the characteristics of trajectory itself. For example, Tong et al. used Adagrad to optimize the simple linear model, and then directly realized the prediction of taxi route [8]. However, this optimization ignored some features of trajectory. For example, will the

number of taxis on this route between adjacent intervals (for example, half an hour or an hour) affect the number of current intervals? Could there be a time interval yesterday affect the same interval today? None of these questions can be answered by such an optimization. Therefore, our approach should strive to avoid these problems.

3 Proposed Method

In this section, we give relevant definitions and mathematical descriptions of our methods, and then explain some of the concepts applied to traffic flow analysis.

3.1 Gradient Fourier Series

Considering the time of the signal, as each neural unit can exist independently, and almost all properties of the unit in the learning process keep changing, we set up a parameter as time in Fourier transformation. For a single unit, the gradient of its weight plays a key role in its learning process, and it changes with the number of iterations, which is similar to the structure of time. Therefore, we have the following definitions:

Definition 1 (Gradient Instant). The gradient of the weight which connecting two neural units at any instant is called a gradient instant.

Definition 2 (Gradient Time). For each individual neural unit, the summary of the gradient of the weight connecting it to any other unit is called gradient time. Each gradient time consists of multiple gradient instants.

Considering the half period. The half period describes the time degree of harmonic transformation. In other words, this parameter determines the duration of the change. It is obvious that the number of iterations in the trajectory data mining determines the length of learning time. (We normally do not consider the scenario where the iteration is terminated when the loss function is lower than the threshold). Therefore, we have the following definition:

Definition 3 (Period). The number of iterations is the period of the current harmonic transformation.

Considering the coefficient. As the initial learning rate never changes, and the change only happens in the process of learning, so the initial learning rate can be seen as a coefficient which remains the same despite the change of gradient instant and gradient time. We have the following definition:

Definition 4 (Coefficient). Initial learning rate is the coefficient.

Assume that the initial learning rate, the weight matrix, the loss function and the current iteration are α , w , $E(w)$, t respectively. According to the definition, gradient instant is $\partial E(w)$, therefore gradient time is $\sum_{i=1}^t \partial E(w)$, and period is t . In addition, because i is an imaginary unit, in trajectory data mining we convert it back to the real unit. Therefore, we get the equation of harmonic transformation of learning rate:

$$\alpha = \alpha e \sqrt{\frac{\sum_{i=1}^t \partial E(w)}{t}} \quad (2)$$

where α , $\sum_{i=1}^t \partial E(w)$, t are c_n , x and l , respectively. The reason of introducing the square is to make the transformation smoother. The harmonic transformation of learning rate is applied to the learning process of the deep residual network to optimize learning rate.

3.2 Trajectory Deep Residual Networks

Considering other features of traffic flow, we need to handle some other settings. Referring to the settings used by Zheng et al. on their research of traffic flow [5], we have following definitions.

Definition 5 (Interval). Trajectory data may undergo a very long time. The basic unit we study is called an interval. Usually, the interval can be one hour, half an hour, etc.

Definition 6 (Closeness). If the adjacent n intervals ($n \geq 1$, similarly hereinafter) have an effect on the current interval of the trajectory, then this effect is called closeness.

Definition 7 (Cycle). If the same intervals in the adjacent n days have an effect on the current interval, then the effect is called cycle.

Definition 8 (Trend). If the same intervals in the same week m , ($m = Mon., Tue., \dots, Sun.$) among the adjacent n weeks have an effect on the current interval, then the effect is called trend.

With these characteristics, we can better analyze trajectory data by catering to trajectory patterns.

4 Harmonic Transformation

The framework of our method is shown in Fig. 1. Firstly, we initialize the learning rate which can be set manually or randomly. Randomly generate the weight matrix of the deep residual network (DRN). Then, set the characteristics associated with traffic flow data. After the initialization is completed, the flow data is taken into the DRN, and the learning rate is adjusted dynamically during the training process. Then, train the flow data and their residual according the learning rate, and feed return the learning results back to the DRN for iterative training. When the training is complete, the test set is brought into the network for further adjustment. Finally, the results are compared against other methods. We emphasize that we analyze traffic flow model for the DRN, which combines the weights during activation function of neural units with flow itself to learn flow rules and separate from other network methods.

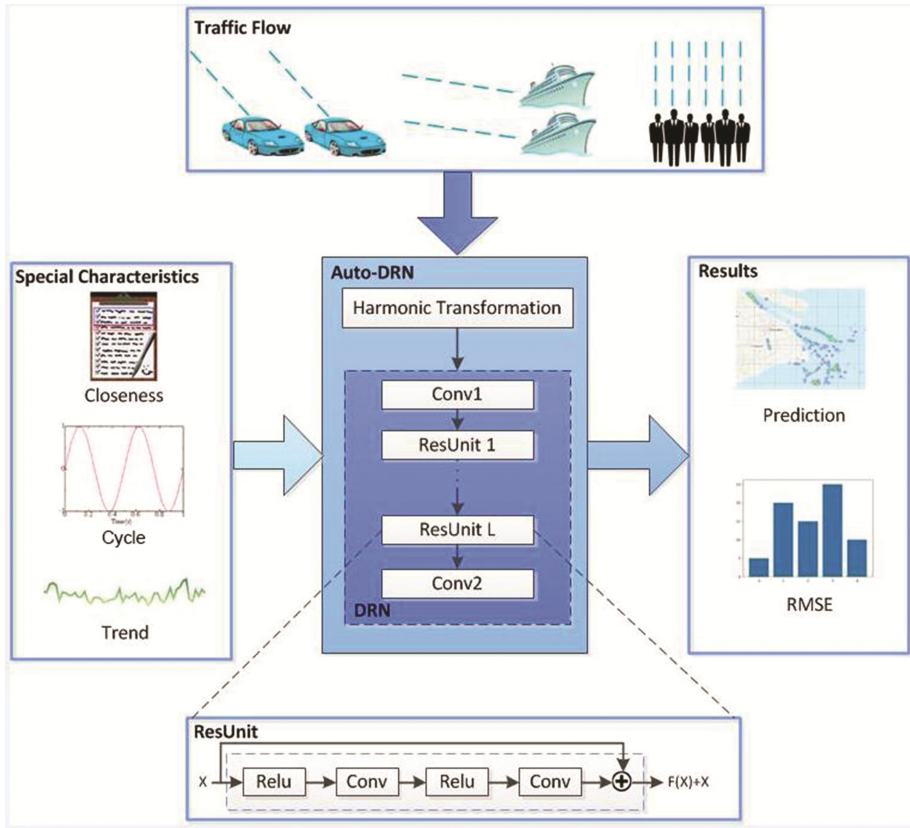


Fig. 1. Framework of G4.

The algorithm is shown in Algorithm 1. The calculation of the time complexity of the algorithm is very simple. Assume through m iterations end training, through n iterations end testing. Because of harmonic transformation occurred and only occurred once in each iteration, in terms of G4 algorithm, the time complexity must be $O(m + n)$. Note here that the time complexity we're talking about is only for our algorithms, not include the time complexity for the structures of convolutional neural network and deep residual network respectively.

Algorithm 1 (G4: automatically adjust the learning rate and analyze the traffic flow)

Input: Traffic flow data, DRN

Output: Automatic learned DRN model

- (01) initialize learning rate;
 - (02) initialize weight matrix, set the characteristics of **traffic flow data**;
 - (03) $t = 1$;
 - (04) **while** $t < iterations$ or $E(w) > 0$:
 - (05) calculate the flow and its MSE and import them into the **DRN**;
 - (06) get the gradient corresponding to each of the current weights;
 - (07) get the gradient time through accumulator;
 - (08) perform harmonic transformation, see equation (2);
 - (09) feedback to DRN based on residual and **traffic flow data** itself;
 - (10) $t++$;
 - (11) **end while**
 - (12) import the test set into **DRN** and follow the steps (04)-(10);
 - (13) **return DRN model**;
-

5 Experiments

In this section, we conduct some experiments based on real data sets to evaluate our method. First, we describe the data sets, then explain the parameters settings of some models, and finally present the results.

5.1 Datasets

We use AIS data to validate our approach. The AIS data records the location information and other information of the ship over time. We select the AIS data recorded from March 2, 2015 to June 30, 2015 in Zhoushan port, China. Since we are forecasting regional activities or traffic flow of ships, we adopt the following methods to carry out the experiment. We divide the research area into 16×8 grids and use *interval* as the basic unit to count the number of signals emitted by ships in each region as the basis for predicting traffic flow. The schematic diagram is shown in Fig. 2. For a specific grid, the existence of ship signal in a grid in an *interval* indicates that the ship is located in this grid in that interval. If in the next *interval* this ship signal is not in the grid, but in an adjacent grid, it means that the ship has moved to the next grid from the current grid, so that the AIS trajectory data can be converted to the grid's data format, which can be imported to the DRN.



Fig. 2. Trajectory data can be transformed into the grid's data format, which can be imported to the DRN.

5.2 Parameters Settings

Next, we describe some parameter settings. The number of iterations for the validation set and the test set are set to 50 and 100, respectively. The number of iterations for the validation set can be set to be smaller, because the validation set comes from the training set. Therefore, its training speed will be faster than the test set which is not from the training set. *Interval* is set to half an hour, i.e. 48 *intervals* a day. Closeness is set to 3, that is, considering a total of three *intervals* from (*interval* - 3) to (*interval* - 1) have an impact on the current *interval*. Both cycle and trend are set to 1, which means that the same *interval* yesterday and the same *interval* last week have an impact on the current *interval*. The residual units are set to 2, that is, two DRNs analyze the flow simultaneously. Special emphasis, there are two identical matrices of flow, but the data of them are different, meaning that one of them saves how much flow for each grid in each interval more than the former interval, and vice versa.

5.3 Results

For the ease of comparison, we use the traditional stochastic gradient descent (SGD) and our methods to predict the traffic flow. Figure 3 demonstrates a comparative experiment, where the initial learning rate of best-SGD is set to the best, i.e. the learning rate has the best performance after we choose from manual debugging, and the initial learning rate of rand-SGD method and our method are randomly set. The x-coordinate shows the number of iterations, and the y-coordinate represents the loss function, which is set to mean squared error (MSE). It can be seen that even for an appropriate learning rate that has been fixed for a long time, our method still outperforms best-SGD from the beginning

to the end. On the other hand, in terms of prediction accuracy, we set up multiple initial learning rates to start together, but all the RMSE of G4 is lower than SGD,. In other words, the accuracy of flow prediction is higher. Some comparisons are shown in Fig. 4. The horizontal coordinate represents different initial learning rates, and the vertical coordinate represents RMSE.

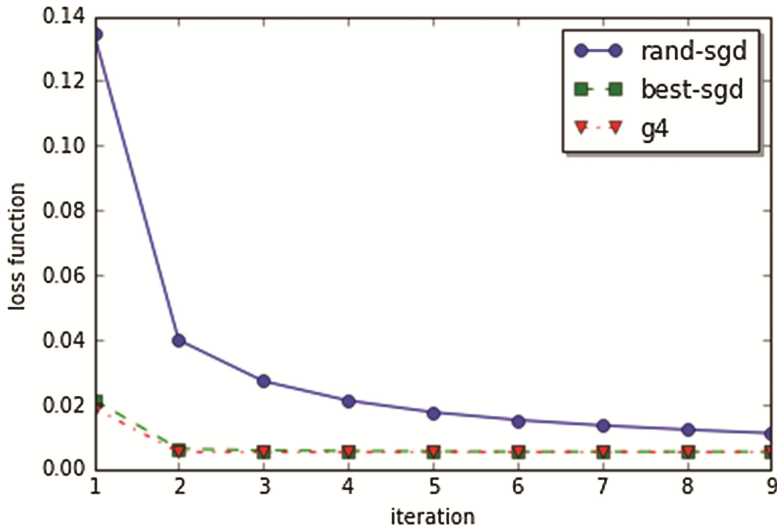


Fig. 3. The loss function changes with the iterations.

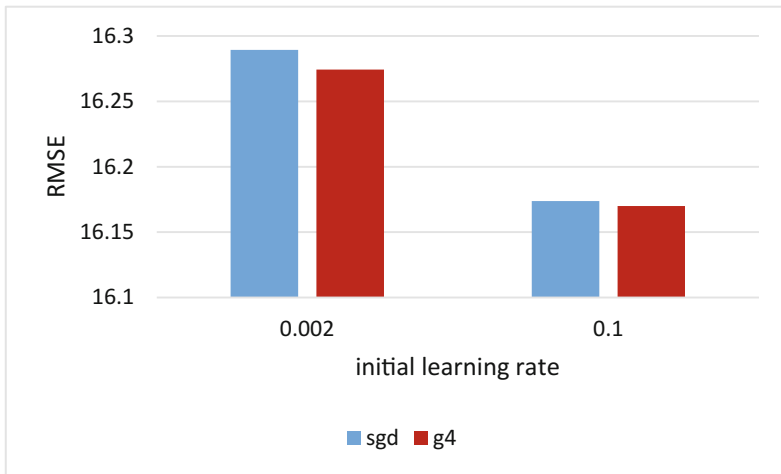


Fig. 4. The loss function changes with the iterations.

6 Conclusions

In this paper, G4 algorithm is proposed to automatically determine the learning rate and predict the traffic flow. Experiments on real data sets show that our algorithm reduces the tedious manual adjustment of parameters, and outperforms some traditional methods. Even the classic method with the optimal parameter settings is still slower than our approach in training. Future work will also include automation research and applications in the field of trajectory data for other parameters of DRN or other deep learning methods.

References

1. Araki, M., Kanamori, R., Gong, L., Morikawa, T.: Impacts of seasonal factors on travel behavior: basic analysis of GPS trajectory data for 8 months. In: Sawatani, Y., Spohrer, J., Kwan, S., Takenaka, T. (eds.) *Serviceology for Smart Service System*, pp. 377–384. Springer, Tokyo (2017). https://doi.org/10.1007/978-4-431-56074-6_41
2. Kumar, D., et al.: A visual-numeric approach to clustering and anomaly detection for trajectory data. *Vis. Comput.* **33**(3), 265–281 (2017)
3. Hoermann, S., Bach, M., Dietmayer, K.: Dynamic Occupancy Grid Prediction for Urban Autonomous Driving: A Deep Learning Approach with Fully Automatic Labeling. arXiv preprint [arXiv:1705.08781](https://arxiv.org/abs/1705.08781) (2017)
4. Xiao, Z., et al.: Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS Int. J. Geo-Inf.* **6**(2), 57 (2017)
5. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: *AAAI* (2017)
6. Song, X., et al.: DeepMob: learning deep knowledge of human emergency behavior and mobility from big and heterogeneous data. *ACM Trans. Inf. Syst. (TOIS)* **35**(4), 41 (2017)
7. Sun, L., et al.: 3DOF Pedestrian Trajectory Prediction Learned from Long-Term Autonomous Mobile Robot Deployment Data. arXiv preprint [arXiv:1710.00126](https://arxiv.org/abs/1710.00126) (2017)
8. Tong, Y., et al.: The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM (2017)