31st Youth Academic Annual Conference of Chinese Association of Automation
Wuhan, China; November 11-13, 2016

Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction

Rui Fu, Zuo Zhang, and Li Li
Department of Automation
Tsinghua University
Beijing, China
li-li@mail.tsinghua.edu.cn

***Abstract*—Accurate and real-time traffic flow prediction is important in Intelligent Transportation System (ITS), especially for traffic control. Existing models such as ARMA, ARIMA are mainly linear models and cannot describe the stochastic and non-linear nature of traffic flow. In recent years, deep-learning-based methods have been applied as novel alternatives for traffic flow prediction. However, which kind of deep neural networks is the most appropriate model for traffic flow prediction remains unsolved. In this paper, we use Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) neural network (NN) methods to predict short-term traffic flow, and experiments demonstrate that Recurrent Neural Network (RNN) based deep learning methods such as LSTM and GRU perform better than auto regressive integrated moving average (ARIMA) model. To the best of our knowledge, this is the first time that GRU is applied to traffic flow prediction.**

Keywords*—*traffic flow prediction; LSTM; GRU; ARIMA

I. Introduction

Over the past few years, ITS has been an important part of smart city and deployed much more than before. As a significant role in ITS, traffic flow information, especially short-term traffic flow information is strongly needed for individual travelers and business companies. However, accurate and real-time traffic flow prediction remains challenging for many decades, due to its stochastic and non-linear nature. Existing methods mainly use linear models and shallow machine learning models to predict the incoming traffic flow and cannot describe the non-linearity and uncertainty well.

In recent years, scholars have been trying to use deep-learning-based NN methods to solve this time series prediction problem and better performance has been reported [1]-[3]. Among these deep-learning-based methods, LSTM NN and stacked autoencoders (SAEs) have reported better performance than some common used traditional prediction models. In traffic speed prediction aspect, Ma et al. applied both LSTM NN and SAEs to the prediction problem and found better performance for LSTM NNs than SAEs [3].

However, which kind of deep neural networks is the most appropriate model for traffic flow prediction remains unsolved. In this paper, we apply LSTM NN and GRU NN methods to traffic flow prediction. For the optimization of this LSTM NN, we use Adam optimizer [4] with adaptive learning rates. We tested the performance of ARIMA, LSTM NN and GRU NN model on the Caltrans Performance Management System (PeMS) dataset and found that LSTM and GRU NNs have better performance than ARIMA, and GRU NNs perform a little better than LSTM NNs and usually converge faster than LSTM.

The rest of this paper is organized as follows. Section II reviews the previous studies on short-term traffic flow prediction. Section III presents the deep-learning-based method with LSTM and GRU NN. Section IV discusses the experiment design and performance of several selected models. Concluding and future envisions are described in Section V.

II. Literature Review

Due to the importance of traffic flow prediction in urban traffic optimization, this subject has been studied since 1970s. With the development of several decades, there are quite many approaches for traffic flow prediction. In general, traffic flow prediction models can be divided into two categories: parameter models and non-parameter models.

A. Parameter Models

Parameter models refer to the models with fixed structure based on some assumptions and that parameters can be computed with empirical data. The most commonly used parameter model, ARIMA model was proposed in the 1970s to predict short-term freeway traffic data [5]. In the next few decades, many time series prediction models were proposed based on ARIMA, such as Kohonen-ARIMA (KARIMA) [6], subset ARIMA [7], seasonal ARIMA [8], etc. These models are based on the assumption of stationary variance and mean of the time series.

Parameter models have several merits. First of all, they are usually simple models and explicit to understand. Secondly, the solution of them is usually easier than non-parameter models and takes less time. However, due to the non-linear and stochastic nature of traffic flow, parameter models may not describe the unique nature well and result in bigger prediction errors than non-parameter models.

978-1-5090-4423-8/16/$31.00 ©2016 IEEE

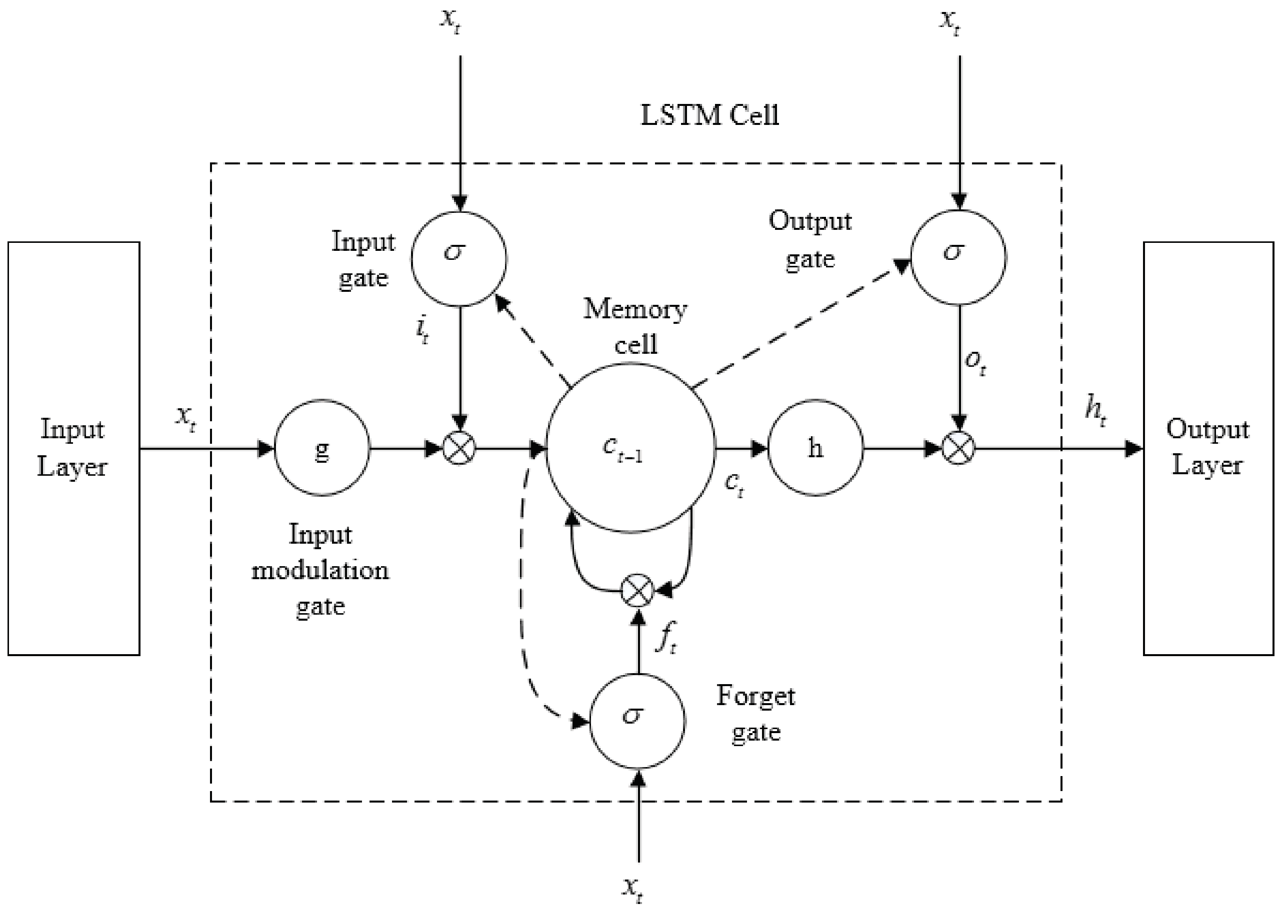

Fig. 1. Structure of LSTM NN cells.

B. *Non-parameter Models*

Non-parameter models refer to the models with no fixed structure and no fixed parameters. Popular non-parameter models include KNN, artificial neural network(ANN), etc.

Non-parameter models such as neural network models may have superior expressive abilities and could fit nearly all the functions to arbitrary precision. However, non-parameter model optimization problem often falls into non-convex optimization problem and may be easy to encounter local minimum. The optimization of non-parameter models may be hard and overfitting is really a tough problem to be solved.

With deep learning reporting huge success by Hinton [9], more and more researchers have been trying to apply deep NN methods to traffic related predictions. Lv *et al.* [1], first applied SAES to traffic flow prediction and reported better performance than SVM, feed forward neural network (FFNN), etc. After that, Ma et al. used LSTM NN for traffic speed prediction and reported success in comparison with most non-parameter models as well as ARIMA (2,1,1) model [3]. Tian et al. proposed LSTM NN for traffic flow prediction and proved that LSTM NN do have better performance than most of the non-parameter models [2]. Due to the excellent ability to memorize long-term dependencies, LSTM NNs have special advantages for traffic flow prediction.

LSTM was initially proposed in 1997 for language models and was well-known for its excellent ability to memorize long-term dependencies [10]. However, due to its complex structure, the solution to LSTM NNs usually takes long time. To speed up training, in 2014, GRU was proposed as a modification for LSTM to do machine translation, for its simpler structure and convenience to solve [11]. However, GRU NNs have not been used for traffic flow prediction yet.

III. LSTM AND GRU NEURAL NETWORK FOR PREDICTION

RNNs were initially used for language models, due to its ability to memorize long-term dependencies. However, with time lags increasing, gradients of RNNs may vanish through unfolding RNNs into very deep feed forward neural networks. In order to solve gradient vanishing problem, certain structure of RNNs such as LSTM and GRU were propose with forget units, which was designed to give the memory cells ability to determine when to forget certain information, thus determining the optimal time lags. LSTM was proposed for language models in 1997 and was utilized for traffic flow prediction since 2015, while GRU NN has not been used in this aspect until now.

The rest of this section presents the structure of LSTM and GRU NN.

A. LSTM

The typical structure of LSTM NN cells is in Fig. 1. A typical LSTM NN cell is configured mainly by four gates: input gate, input modulation gate, forget gate and output gate. Input gate takes a new input point from outside and process newly coming data. Memory cell input gate takes input from the output of the LSTM NN cell in the last iteration. Forget gate decides when to forget the output results and thus selects the optimal time lag for the input sequence. Output gate takes all results calculated and generate output for the LSTM NN cell. In language models, usually a soft-max layer is added to determine the final output of the NN. In our traffic flow prediction model, a linear regression layer is applied on the output layer of the LSTM cell.

Let us denote the input time series as $X=(x_1,x_2,...,x_n)$, hidden state of memory cells as $H=(h_1,h_2,...,h_n)$, output time series as $Y=(y_1,y_2,...,y_n)$. LSTM NNs do the computation as follows:

$$h_t = H(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

$$p_t = W_{hy}y_{t-1} + b_y \tag{2}$$

where weight matrices are denoted as W, and bias vectors denoted as b. The hidden state of memory cells is computed in the following formulas:

$$i_t = \sigma\left(W_{ix}x_t + W_{hh}h_{t-1} + W_{ic}c_{t-1} + b_i\right) \tag{3}$$

$$f_t = \sigma\left(W_{fx}x_t + W_{hh}h_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{4}$$

$$c_t = f_t * c_{t-1} + i_t * g(W_{cx}x_t + W_{hh}h_{t-1} + W_{cc}c_{t-1} + b_c) \tag{5}$$

$$o_t = \sigma\left(W_{ox}x_t + W_{hh}h_{t-1} + W_{oc}c_{t-1} + b_o\right) \tag{6}$$

$$h_t = o_t * h(c_t) \tag{7}$$

where σ stands for the standard sigmoid function defined in Eq. (8), * stands for the scalar product of two vectors or matrices, g and h are the extends of stand sigmoid function with the range changing to [-2, 2] and [-1, 1].

$$\sigma(x) = \frac{1}{1+e^x} \tag{8}$$

For objective function we use square loss function given by the following formula:

$$e = \sum_{t=1}^{n}(y_t - p_t)^2 \tag{9}$$

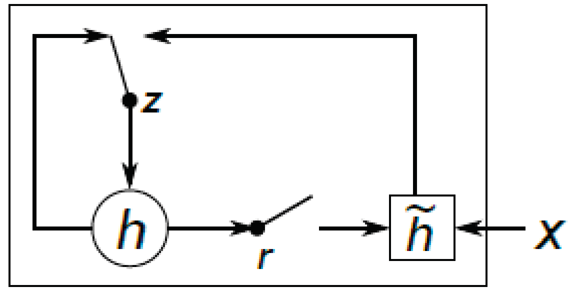

Fig. 2. Structure of GRU cells.

where y represents the real output and p represents the predicted traffic flow. In order to minimize training error and meanwhile avoid local minimal points, Adam optimizer, a modification of stochastic gradient descent (SGD) optimizer with adaptive learning rates, is applied for back propagation through time (BPTT). Neural networks have been known for its tremendous expressive abilities, and are especially easy to overfitting. Neural network training has been a difficult problem for long, and a large quantity of regularization methods have been proposed to reduce overfitting. In 2012, dropout [12] was proposed as a very efficient method to train neural networks in order to gain better features of images. However, due to the recurrent property of RNNs, dropout has been difficult to apply to language models of RNNs. Until 2014, dropout methods have been reported to be successfully applied to RNNs [13].

B. GRU

GRU was proposed by Cho *et al*. [11] in 2014, similar to LSTM, but simpler to compute and implement. The typical structure of GRU cells is shown in Fig. 2.

A typical GRU cell is consisted of two gates: reset gate r and update gate z. Similar to LSTM cell, hidden state output at time t is computed using the hidden state of time t-1 and the input time series value at time t, which is presented in Eq. (11).

$$h_t = f(h_{t-1}, x_t) \tag{10}$$

The function of reset gates is similar to forget gates of LSTM. As GRU NNs have several similarities to LSTM NNs, we will not go deep into the detailed formula. Readers who are interested in this are referred to Cho *et al*. [11]. Regression part and optimization method we use in this paper for GRU NNs is the same as the LSTM NNs.

IV. EXPERIMENTS

A. Data Description and Experiment Design

The data used in this paper is collected from PeMS dataset, which has over 15,000 sensors deployed statewide in California. The experimental traffic network that we analyze is

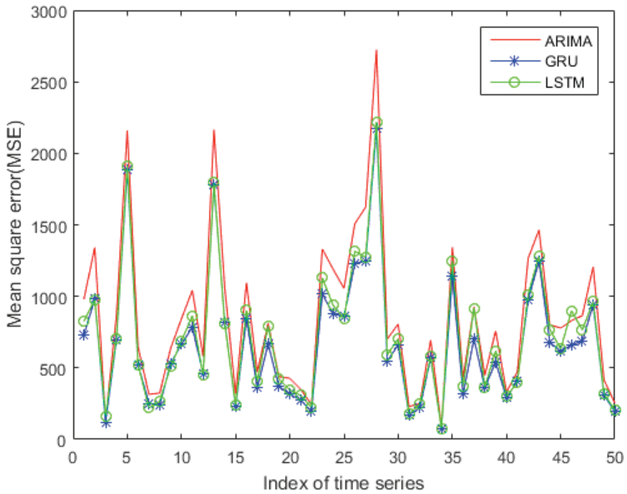

Fig. 3. MSE over 50 selected time series.

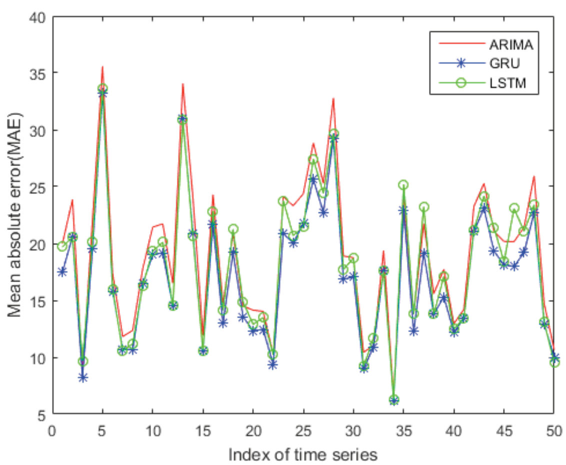

Fig. 4. MAE over 50 selected time series.

in the fourth district of PeMS dataset, which lays in the Bay Area, Alameda, Oakland of the U.S.The original data was collected every 30 seconds and some data points are missed. According to previous studies [14], 5-minute traffic flow is more suitable and predictable for our study. Because the missing data only occupies a small part of the whole dataset, we impute the missing data points using historical average value. The following experiments are based on the imputed dataset.

In our experiment, we select the traffic flow of the past 30 minutes, which actually is a time sequence of 6 data points, to predict the coming traffic flow in the next 5 minutes. Data is divided into two parts. The first three weeks are used to train our model, and data of the last week is used to test our model prediction accuracy. Since each time series may have a different traffic flow pattern and no pattern could fit all the traffic flow series, we create a unique model for each of the traffic flow series collected by a single sensor.

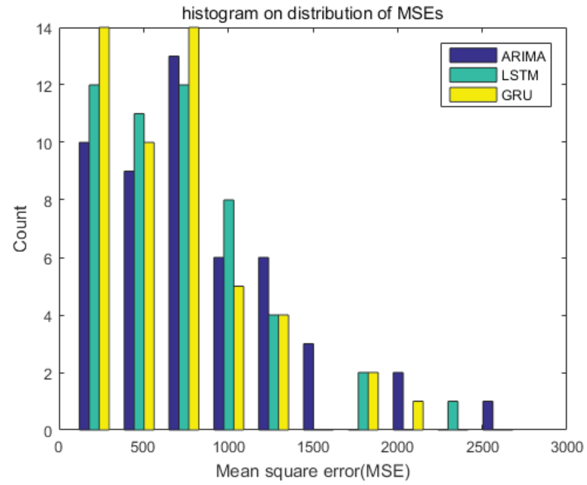

Fig. 5. Histogram on distribution of MSEs.

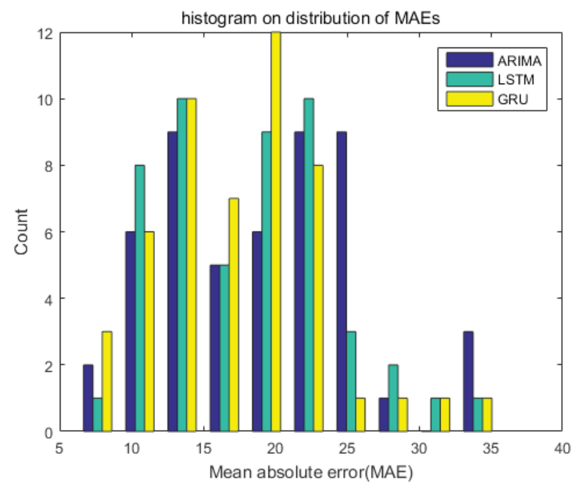

Fig. 6. Histogram on distribution of MAEs.

In our experiment, ARIMA, LSTM NN and GRU NN models are compared. To test our model prediction accuracy better, we use both mean square error (MSE) and mean absolute percentage error (MAPE) defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i)^2 \tag{11}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}| y_i - y_i | \tag{12}$$

B. Experimental Results

We randomly selected 50 sensors and tested model prediction accuracy on the selected sensors. In this experiment, LSTM and GRU NNs are both trained after a specified steps and performance is measured after that. MSEs over the 50 sensors are plotted in Fig. 3 and MAEs are plotted in Fig. 4. Actually, the performance of these three method does not differ much. According to the figures, LSTM NN and GRU NN is a

TABLE I. MSE AND MAPE FOR THREE TESTED MODELS

	ARIMA	LSTM NN	GRU NN
MSE	841.0065	710.0502	668.9304
MAE	19.1753	18.127758	17.2116

bit better than ARIMA model which selects the best parameter (p, d, q) pair automatically, and the performance of the two RNNs are quite near, as shown in Table I. The distribution of MSEs and MAEs over the selected 50 traffic flow series is plotted in Figs. 5 and 6. From Fig. 5 we could find that most MSEs are located in (0, 1200) and for GRU and LSTM NN, the peak of distribution appears earlier than ARIMA, which means MSEs of the two RNNs tend to be smaller. The distribution of MAEs seem to be more center distributed, and the same property as MAEs is also suitable for distribution on MAEs.

Experimental results show that LSTM and GRU NNs have better performance than ARIMA model and GRU have a little better performance than LSTM NN. Meanwhile, on 84% of the total time series, GRU NNs have better performance than LSTM NNs.

V. CONCLUSION

In this paper, a model called GRU NN is proposed for traffic flow prediction. We compared the performance of prediction on ARIMA, LSTM and GRU models and found that LSTM NNs and GRU NNs outperform ARIMA model in our experiment. In average, GRU NNs have reduced MAE at about 10% level than ARIMA model, and 5% than LSTM NN model. In future work, RNNs with more hidden states are to be tested and the changeable length of time sequence inputs may be a help for RNNs to determine the optimal time lags automatically.

REFERENCES

[1] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865-873, Apr. 2015.

[2] Y. Tian and L. Pan, "Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network,"in *2015 IEEE International Conference on Smart City*, Chengdu, pp. 153-158, 2015.

[3] X. Ma, Z. Tao, Y. Wang, H. Yu and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C*, *Emerg. Technol.*, vol. 54, pp. 187-197, May 2015.

[4] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv: 1412.6980*,* 2014.

[5] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box–Jenkins techniques," *Transp. Res. Rec.*, no. 722, pp. 1–9, 1979.

[6] M. vanderVoort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow, " *Transp. Res. C*, *Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, Oct. 1996.

[7] S. Lee and D. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transp. Res. Rec.*, vol. 1678, pp. 179–188, 1999.

[8] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, 2003.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.

[11] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, D. Bahdanau, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv: 1406.1078, 2014.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[13] W. Zaremba, I. Sutskever and O. Vinyals, "Recurrent neural network regularization," arXiv preprint arXiv:1409.2329, 2014.

[14] C. Chen, Y. Wang, L. Li, J. Hu, and Z. Zhang, "The retrieval of intra-day trend and its influence on traffic prediction," *Transp. Res. C*, *Emerg. Technol.*, vol. 22, pp. 103–118, Jun. 2012.