Traffic flow prediction with Long Short-Term Memory Networks (LSTMs)

Hongxin Shao
Interdisciplinary Graduate School
Nanyang Technological University
Email: hshao002@e.ntu.edu.sg

Boon-Hee Soong
School of Electrical and
Electronic Engineering
Nanyang Technological University
Email: EBHSOONG@ntu.edu.sg

***Abstract*—Accurate traffic flow information is crucial for an intelligent transportation system management and deployment. Over the past few years, many existing models have been designed for short-term traffic flow prediction. However, they fail to provide favorable results due to their shallow architectures or incapability to extract the sequence correlations in the data. In this paper, we explore the application of Long Short-Term Memory Networks (LSTMs) in short-term traffic flow prediction. As a deep learning approach, LSTMs are able to learn more abstract representations in the non-linear traffic flow data. The intrinsic feature of capturing long-term dependencies in a sequential data also makes it a suitable choice in traffic prediction. Experiments on real traffic data sets indicate a good performance of our model. The LSTMs architecture is also compared with state-of-the-art models and experiments show that our model achieves desirable results by lowering the MAPE metrics to 5.4%.**

I. Introduction

Smart nation is a program that is raised recently in countries like Singapore. It aims to improve the living standards of citizens by harnessing advanced information and communications technologies (ICT). As one of the strong pillar that ensures the success of a smart nation program, intelligent transport system (ITS) has always been a crucial and indispensable element. It includes smart traffic control and management systems, comprehensive vehicle information systems, and commercial vehicle operations which heavily rely on the real-time traffic flow information. The objective of traffic prediction is to provide such traffic flow information. It could empower the road user to make wise travel decision, reduce the carbon dioxide emission, and alleviate traffic congestion.

Traffic prediction finds its root in time-series prediction problems in which one attempts to predict a variable at current state based on a series of past samples of the variable at regular intervals. Let x_t denote the variable of observed traffic flow at the tth time interval in a transportation network. Given a sequence $\{x_t\}$ of collected traffic flow data, where $t = 1, 2, ..., T$, the task is to predict the traffic flow x_{T+1} at next time interval $(T + 1)$.

Many research efforts have been devoted to traffic flow prediction over the past decades. In general, they can be categorized into parametric and nonparametric approaches. Parametric models include various time-series models. The most widely deployed method is autoregressive integrated moving average (ARIMA) model. Levin and Tsao found that the ARIMA (0,1,1) was the most statistically significant for all forecasting intervals [1]. A variant, seasonal ARIMA model with one-week lag, was proposed by Williams *et al.* [2] to further optimize in the traffic forecasting scenario. Another commonly used parametric method is Kalman filtering models. Guo *et al.* proposed an adaptive Kalman filtering model which has demonstrated improved adaptability when traffic is highly volatile [3]. However, due to the stochastic and nonlinear nature of traffic flow, parametric methods are not able to capture the traffic features well and researchers have paid more attention to non-parametric methods in the traffic flow forecasting field. Chen *et al.* [4] investigated the self-organizing maps (SOM) and k-nearest neighbor (k-NN) algorithms in traffic prediction and achieved slightly better result than other state-of-the-art techniques. Another common non-parametric method is artificial neural networks (ANNs) and its variants. Jiang *et al.* [5] introduced the concept of wavelet frame into ANNs and the model shows good forecasting capability in both short-term and long-term traffic. The models were shown to be more superior and more responsive to dynamic conditions due to their multi-input nature. One of another popular method is support vector regression (SVR). SVR depends only on a subset of the training data and is efficient in implementation. Castro-Neto *et al.* [6] applied an online SVR model to both typical and atypical traffic conditions and achieved slightly better results than others. The comparison of parametric and non-parametric methods was also investigated. Smith *et al.*[7] compared non-parametric regression models with several parametric models. The nonparametric regression model significantly outperformed the other models with lower error.

Recently, deep learning, an emerging machine learning method, has drawn a lot of attention from both academic and industrial field. Compared to a simple neural network structure, it has more depth of layers and aims to learn more abstract representations of the data. Deep learning has advantages such as less computational complexity and better feature extraction etc. [8]. It has been proved successful in various areas such as natural language processing, speech recognition, and image recognition. Some pioneering works applied deep learning approach in traffic flow forecasting and achieved satisfactory results. Lv *et al.* [9] explored one deep learning architecture, stacked autoencoder, to predict the traffic

978-1-5090-2597-8/16/$31.00

flow at different traffic intervals ranging from 15-min to 60-min. The model achieves desirable results except in a low traffic flow rate condition. Another deep learning architecture, deep belief network, is tested by Huang *et al.* [10] on one road segment. A multitask learning using K-means based algorithm is then applied to make use of neighboring road information, and the result is promising.

Inspired by the very few but successful applications of deep learning in traffic flow prediction scene, we proposed a model based on long short-term memory networks (LSTMs). LSTMs is one particular type of recurrent neural networks (RNNs) which are family of neural networks for processing sequential data. RNNs can scale to much longer sequences and this capability make it stand out and be more practical in comparison with other deep learning structures which has no sequence-based specialization.

The rest of this paper is organized as follows. In Section II, we introduce our LSTM model for traffic flow prediction. Section III gives the experimental results of our approach and compares with the state of the art. Analysis with the hyperparameter selection is also presented. Finally, the conclusion is given in Section IV.

II. System Design

A. Introduction of LSTMs

A recurrent neural network is a neural network that is specialized for processing a sequence of values $x^{(1)}, ..., x^{(T)}$. The main idea behind RNN is the built-in loop structure. The loop in RNN allows the information to be passed from one step of the network to the next. The chain-like nature reveals that recurrent neural networks are intimately related to sequential problem, such as speech recognition, language modeling, and image captioning. Also, RNN allows the sharing of parameters for each value of the time index. Statistical strength across different time index is shared at neighboring time indexes and this could enhance the feature extraction process. All these inherent features have endowed RNN the capability to outperform simple multiple layer perceptron, as well as other deep learning architectures (e.g., SAE and DBN). However, the traditional RNN faces problems of vanishing gradients and exploding gradients. The gradient signal progresses along the hidden layer and gets multiplied by the weight matrix of the neurons. If the weight matrix is too small or large, the gradient signal will eventually become either too small that learning stops working, or too large that learning diverges.

To address this problem, long short-term memory (LSTM) was proposed [11]. The key idea of LSTM is a new structure called memory cell. It contains a neuron with a self-recurrent connection and three gates associated with it. The self-recurrent connection allows the state of memory cell to remain constant from one timestep to another without outside interference. Input and output gates can allow the signal to come in or go out of the neuron or prevent it. Forget gate can modulate the memory cell's self-recurrent connection, allowing the cell to remember or forget its previous state.

B. Our model

Suppose in the current timestep, we have historical traffic flow sequence denoted as $x = (x_1, x_2, ..., x_T)$, our task is to predict the traffic flow x_{T+1} at the next timestep. We proposed an encoder-decoder model based on LSTM blocks. The model is illustrated in Figure 1.

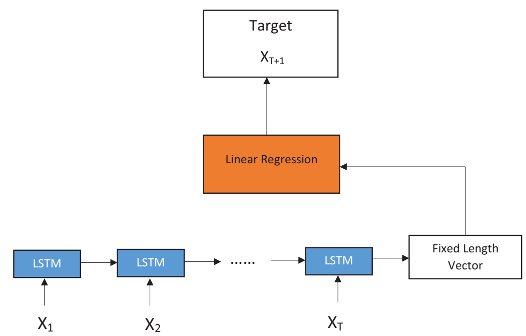

Fig. 1. One-layer LSTM structure

The encoder reads the sequence of traffic flow $x = (x_1, x_2, ..., x_T)$ into a fixed length vector c. Each LSTM block takes the traffic flow at one timestep as input and they are concatenated according to the temporal sequence of the traffic flow. The LSTM block computes the hidden vector h_i from its current input x_i and hidden vector h_{i-1} from previous LSTM block,

$$h_i = f(x_i, h_{i-1}), \tag{1}$$

where f is the activation function of the LSTM block. At the last LSTM block, the fixed length vector c is obtained as

$$c = h_T, \tag{2}$$

where h_T is the hidden state of the last LSTM block. This fixed length vector encodes the temporal correlations up to the last input data and works as a high-level representation of the input sequence. Note that, only one layer of LSTM blocks is drawn in encoder structure for simple illustration. In the real case, multiple layers can be stacked for highly non-linear data and we explore it in section III. The decoder is trained to predict the traffic flow at the next timestep and we adopt a linear regression here as the decoder.

III. Experiments

A. Data Description

Our proposed LSTM model was applied on the data collected by Caltrans Performance Measurement System (PeMS). PeMS offers historical database of traffic flow information. The data are collected from over 39,000 individual detectors along the freeway across all major metropolitan areas of the State of California. The collected data are then aggregated into a 5-min or 1-hour interval for each vehicle detection station (VDS) and stored in the online database. The particular VDS we look at in this paper has ID 1201100, and locates at lane

2 of I-405 mainline at Irvine Center Drive. One-month data (from August 1st to Agust 31th 2014) are selected in our experiments. Since weekday and weekend traffic present different patterns, here we only analyze the weekday traffic flow. A common issue with the collected data by roadside detector is data fidelity. Sometimes the detector is not working in normal condition, and the data has to be imputed by recalculation using neighbor roads information. PeMS provides its own test on the data and shows the percentage of data that is observed or imputed. In this paper, the data used has a imputation value of 0%, thus the reliability of the experiments is ensured.

B. Evaluation metrics

Evaluation metrics are required in order to evaluate the validity of our traffic prediction model. Since our model aims to predict the traffic flow at next timestep, the evluation criteria will include accuracy metrics which compare the predicted traffic flow and the real traffic flow. Two commonly used metrics are mean absolute percentage error (MAPE) and root mean square error (RMSE). They are defined by Equation 3 and 4.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|f_i - \hat{f}_i|}{f_i} \tag{3}$$

$$RMSE = [\frac{1}{n}\sum_{i=1}^{n}(|f_i - \hat{f}_i|)^2]^{1/2} \tag{4}$$

where f_i is the real traffic flow observed by the detector and $\hat{f}_i$ is the predicted traffic flow by our model.

C. Results

We compared the performance of our LSTM model with random walk (RW), support vector regression (SVR), wavelet neural network (WNN), and the stacked autoencoder (SAE). RW is a simple method that predicts the traffic flow at next timestep as current value $(i.e., \hat{f}_{t+1} = x_t)$ and works as a baseline here. SVR is an easier and popular method for regression problem. The superiority of neural network has been aforementioned in the literature. As a type of neural network structure, WNN works well in time-series prediction. We also compared with SAE which is a deep learning approach. It stacks several autoencoder layers with one regression layer at the top. In all models, we use the same dataset as described in previous section.

Figure 2 presents the pattern of both real traffic flow and predicted traffic flow within one day. It is observed that, the difference between real and predicted values are small and the predicted traffic flow follows the real traffic flow pattern well.

Table I lists the MAPE and RMSE for different models. The results are all fine-tuned with optimal parameters. For SVR, we use a penalty parameter equals to 2.3 and a 'rbf' kernel. For WNN, the model has one hidden layer with 30 neurons and the input layer takes the size of 6 neurons. SAE model achieves optimal performance with input size 5, 3 stacking layers and 40 neurons for each layer. For our model, a one-layer LSTM with 32 neurons in the hidden layer. The input length is 6. Compared with four state-of-the-art models, our model obtained lowest in both MAPE and RMSE.

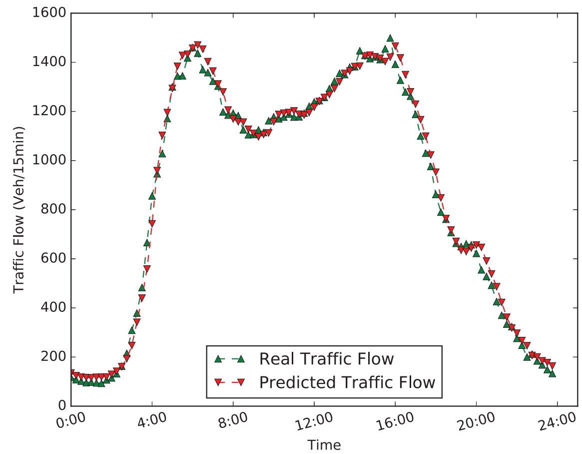

Fig. 2. Predicted traffic flow vs. real traffic flow

TABLE I
COMPARISON OF DIFFERENT MODELS

Metrics	Models				
	RW	SVR	WNN	SAE	LSTM
MAPE(%)	8.11	10.00	9.37	6.57	**5.40**
RMSE	69.27	46.82	53.86	47.32	**40.34**

D. Analysis of hyperparameter

Deep learning approaches inherently involve several hyperparameters in their models. Grid search method is always adopted to select the optimal hyperparameters that will give the best result. We hereby carried out some experiments on effect of different hyperparameters on MAPE and RMSE metrics.

1) number of timesteps: As introduced in section II, LSTM explores the effect of previous tasks to present task in a time-series prediction problem. Therefore, in our experiment, one of the hyperparameter is the number of timesteps before but still affect the present task.

Figure 3 illustrates the RMSE and MAPE with different values of timesteps. The timesteps take the range in $[1, 2, ..., 10]$ and each timesteps value k exploits the prediction of traffic flow by using previous k timesteps traffic flow. Experiment shows that, with timesteps at 6, both RMSE and MAPE can achieve the optimal value. Also, MAPE and RMSE show the same error trend and it could be explained by the model complexity. Timesteps are the input size to the model and determines number of LSTM blocks in each level. When the input size increases from 1, more previous traffic flow information is used and the model is able to provide a better result. Therefore, the error is decreasing. After timesteps value of 6, the model gets more and more complex, overfitting problems occurs and is reflected in the increase of generalization error.

2) number of layers: In deep learning, desirable result is always obtained by stacking several layers of the same

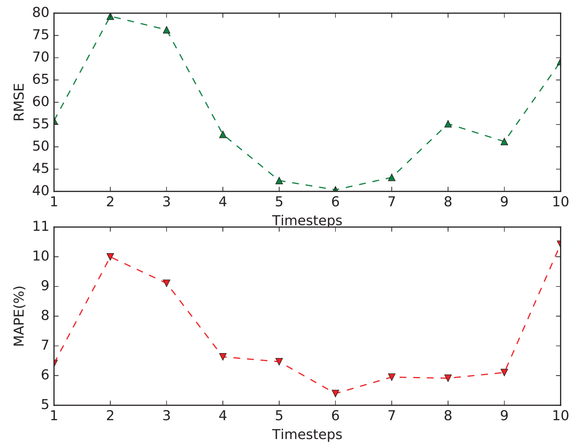

Fig. 3. RMSE and MAPE with different timesteps values

network, such as SAE and DBN. Thus, we exploit the effect of stacking multiple layers of LSTMs in our experiment. Figure 4 shows that both RMSE and MAPE increase when more LSTM layers are added to our model. One possible reason might be due to the moderate size of our data set. With such size, overfitting occurs easily when we try to increase the number of layers. Thus, the optimal layer in our case would be one-layer LSTM.

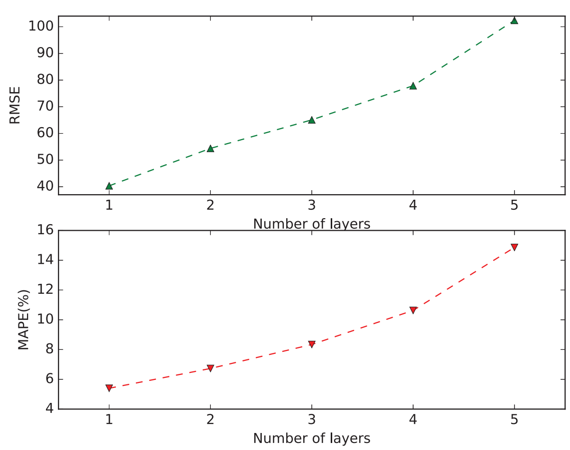

Fig. 4. RMSE and MAPE with different layers

IV. Conclusion

We propose a LSTM model to predict the short-term traffic flow. LSTM is able to exploit the long-term dependency in the traffic flow data sequence. As one of the deep learning approach, LSTM is able to discover the latent feature representations hidden in the traffic flow. We evaluated the performance of our model on PeMS data and compared with RW, SVR, WNN, and SAE. Results show that the proposed model is superior to other methods. In our paper, the model mainly considers the temporal series data. However, the traffic network is large and complex, and many interconnections exist. It will be interesting to consider the neighboring roads traffic and use it to enhance on the current model.

Acknowledgements

This research is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore. Also, the authors would like to thank Caltrans Performance Measurement System (PeMS) for their invaluable sharing of data resources.

References

[1] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)," *Transportation Research Record*, no. 773, 1980.
[2] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.
[3] 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.
[4] Y. Chen, Y. Zhang, and J. Hu, "Multi-dimensional traffic flow time series analysis with self-organizing maps," *Tsinghua Science & Technology*, vol. 13, no. 2, pp. 220–228, 2008.
[5] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *Journal of transportation engineering*, vol. 131, no. 10, pp. 771–779, 2005.
[6] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert systems with applications*, vol. 36, no. 3, pp. 6164–6173, 2009.
[7] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," *Journal of transportation engineering*, vol. 123, no. 4, pp. 261–266, 1997.
[8] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
[9] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 2, pp. 865–873, 2015.
[10] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 5, pp. 2191–2201, 2014.
[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.