Towards traffic matrix prediction with LSTM recurrent neural networks

Jianlong Zhao✉, Hua Qu, Jihong Zhao and Dingchao Jiang

This Letter investigates traffic matrix (TM) prediction that is widely used in various network management tasks. To fastly and accurately attain timely TM estimation in large-scale networks, the authors propose a deep architecture based on LSTM recurrent neural networks (RNNs) to model the spatio-temporal features of network traffic and then propose a novel TM prediction approach based on deep LSTM RNNs and a linear regression model. By training and validating it on real-world data from Abilene network, the authors show that the proposed TM prediction approach can achieve state-of-the-art TM prediction performance.

Introduction: Most of the typical network management functions, including traffic engineering or network planning, all depend on the traffic information between all pairs of origin-destination (OD) nodes (e.g. prefixes, links or routers) in a network, which is usually denoted as a traffic matrix (TM) [1, 2]. Generally, TM information can be revealed by direct measurements in real time (i.e. TM monitoring) or by computational estimation of historical traffic data (i.e. TM prediction). However, the TM monitoring techniques suffer from some technical and mercantile issues, especially in large-scale networks. Therefore, the network operators incline to predict and estimate TM from the previously achieved network traffic data instead of measuring it directly.

Several types of research have been devoted to achieving TM prediction and estimation. At first, simple statistical models like Gaussian or Poisson distributions were widely used to solve TM prediction and estimation problem [3]. However, these methods do not consider the spatial or temporal features of traffic flow evolution, and as a result, cannot be applied to large-scale timely TM prediction. Afterwards, the most popular prior methods consist of the TomoGravity [4] and the principal component analysis (PCA) [3]. Although these methods can reduce the sensitivity to prior assumptions to a certain extent, the time-varying features of OD flows are still not taken into account. As time goes on, the prediction error of these prior methods will increase. The neural network (NN) has also been successfully applied to TM prediction and estimation problem. The authors of [2] have successfully adopted a deep belief network to model the TM prediction and estimation problem. Back-propagation NN (BPNN), a common simple but practical NN, was also employed in the TM estimation [5]. Unlike the information in BPNN that can only be passed forward, recurrent NN (RNN) has cyclic connections over time and can capture spatial and temporal features of network traffic. As an alternative RNN framework, long short-term memory (LSTM) has gained more attention in the sequence modelling tasks, and it has also been demonstrated to be more effective than RNN [6]. Stemmed from the temporal evolution of traffic flows and the superior performance of LSTM in the long-term time-series prediction [7], we employ LSTM as a key component to fast and accurately achieve TM prediction in large-scale networks.

In this Letter, a novel deep architecture based on LSTM RNNs is proposed to deal with the TM prediction problem, which is termed as DLTMP. DLTMP first use a deep LSTM network, which is composed of stacked LSTM layers, to model the spatio-temporal features of network traffic. Additionally, a linear regression model is employed for calculating the final predicted TM. Finally, by training and validating DLTMP on real-world data from Abilene network, we demonstrate that our prediction approach can achieve state-of-the-art TM prediction performance.

DLTMP: The general TM prediction aims to solve the problem that obtains the predictor of future TM by the previously achieved network traffic data. In this Letter, the network topology is modelled as an undirected graph $G = (V, E)$, where V represents the set of network nodes and E models the set of links that connect two network nodes. The TM can be represented by a $n \times n$ matrix $\boldsymbol{S} = [s_{ij}]$, where n denotes the size of V and each element s_{ij} denotes the traffic volume transmitted from source node i to destination node j. Since the traffic evolves during the time, we define a dynamic TM $\boldsymbol{S}_t = [s_{ij}^t]$, where t denotes the time-slot of traffic generation. We assume that time-slot is discretised and collect T samples of the TM (i.e. $t = 1, \ldots, T$). To simplify the notation and facilitate the training of our prediction model, we transform each TM into a $n^2 \times 1$ traffic vector $\boldsymbol{D}_t = [d_k^t]$, where each element d_k^t corresponds to an element s_{ij}^t of the TM using the relation $k = i \times n + j$. Intuitively, a traffic vector $\boldsymbol{D}_t$ represents the traffic volume of n^2 OD flows at the particular time-slot t. When observing the historical traffic vectors $(\boldsymbol{D}_{t-1}, \boldsymbol{D}_{t-2}, \ldots, \boldsymbol{D}_1)$ to time $t-1$, the general TM prediction can be summarised as solving the predictor of $\boldsymbol{D}_t$ (denoted by $\hat{\boldsymbol{D}}_t$) from these observations and formulated as

$$\hat{\boldsymbol{D}}_t = g(\boldsymbol{D}_{t-1}, \boldsymbol{D}_{t-2}, \ldots, \boldsymbol{D}_1), \tag{1}$$

where $g(\cdot)$ represents the predicting function to be solved that can model the complex non-linear correlations between the observed network traffic data and the desired future network traffic data. Herein, the main challenge of TM prediction is to model the inherent correlations (i.e. spatio-temporal features) among OD flows, and then build the predicting function $g(\cdot)$.

Fig. 1 illustrates our proposed DLTMP architecture. Specifically, it consists of two parts, one is a deep LSTM network, and the other is a linear regression model. The deep LSTM network is composed of m stacked LSTM layers, which are used to model the spatial-temporal features of network traffic. The output layer adopts a linear regression model to calculate the future TM. The precise implementation of DLTMP architecture is described as follows.

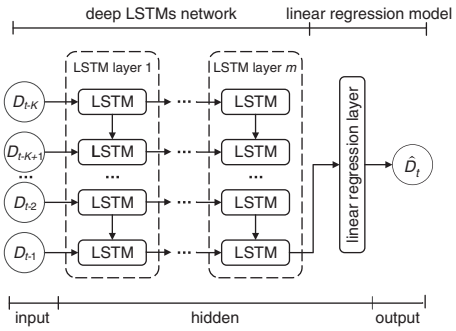

Fig. 1 *DLTMP architecture*

For convenience, we use $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ to denote the input and output traffic vector. When training DLTMP, an input traffic vector $\boldsymbol{X}_t$ is an input to the 0th layer (input layer), and then a supervised learning mechanism is used to characterise the non-linear correlations between $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$. The specific iterations are executed by forwarding pass from $t = 1$ to T

$$\begin{cases} \boldsymbol{h}_t^0 = \boldsymbol{X}_t, \\ \boldsymbol{h}_t^p = \Gamma_t^p(\boldsymbol{h}_t^{p-1}, \boldsymbol{h}_{t-1}^p), \\ \boldsymbol{Y}_t = \Lambda\left(\boldsymbol{W}^{m,m+1}\boldsymbol{h}_t^m + \boldsymbol{b}^{m+1}\right), \end{cases} \tag{2}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ denote output weight matrices and output bias vectors, respectively (with superscripts denoting the layer indices). $\boldsymbol{h}_t^p$ denotes the outputs of the pth $(p = 1, \ldots, m)$ hidden layer at time t. Λ represents the ReLU activation function in the linear regression model. Γ_t^p denotes a composite function used in the deep LSTM network instead of a simple sigmoid activation function that belongs to a traditional shallow LSTM network [7]. Notably, the purpose of this composite function is to represent $\boldsymbol{h}_t^p$ by jointly scaling the recurrent connection $\boldsymbol{h}_{t-1}^p$ and feedforward connection $\boldsymbol{h}_t^{p-1}$, which also means that the spatial and temporal features of the traffic flow evolution are taken into consideration simultaneously. Each hidden layer consists of several LSTM memory blocks, each of which has a memory cell that is used to store and remember the temporal states (state variables) of the unit in the network. During the iterative process, the state variables $\boldsymbol{c}_t^p$ will be scaled by the activation of the output gates $\boldsymbol{o}_t^p$

$$\begin{cases} \boldsymbol{h}_t^p = \boldsymbol{o}_t^p \otimes \tanh\left(\boldsymbol{c}_t^p\right), \\ \boldsymbol{c}_t^p = \boldsymbol{f}_t^p \otimes \boldsymbol{c}_{t-1}^p + \boldsymbol{i}_t^p \otimes \boldsymbol{j}_t^p, \\ \boldsymbol{j}_t^p = tan\, h\left(\boldsymbol{W}_c^{p-1,p}\boldsymbol{h}_t^{p-1} + \boldsymbol{W}_c^{p,p}\boldsymbol{h}_{t-1}^p + \boldsymbol{b}_c^p\right), \end{cases} \tag{3}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ denote weight matrices and bias vectors of the memory cells, respectively (with superscripts denoting layer indices). $\otimes$ indicates the Hadamard product operator and tan$h$ denotes the standard hyperbolic tangent activation function. The $\boldsymbol{c}_{t-1}^{p}$ and $\boldsymbol{c}_{t}^{p}$ are two consecutive cell states. Additionally, the forget gate $\boldsymbol{f}_{t}^{p}$ decides whether to forget the cell state $\boldsymbol{c}_{t-1}^{p}$, and the activation of the input gate $\boldsymbol{i}_{t}^{p}$ controls the input from the recurrent and feedforward connections. The concrete activations of the input, output and forget gates are calculated as

$$\begin{cases} \boldsymbol{i}_{t}^{p} = \sigma\left(\boldsymbol{W}_{i}^{p-1,p}\boldsymbol{h}_{t}^{p-1} + \boldsymbol{W}_{i}^{p,p}\boldsymbol{h}_{t-1}^{p} + \boldsymbol{b}_{i}^{p}\right), \\ \boldsymbol{o}_{t}^{p} = \sigma\left(\boldsymbol{W}_{o}^{p-1,p}\boldsymbol{h}_{t}^{p-1} + \boldsymbol{W}_{o}^{p,p}\boldsymbol{h}_{t-1}^{p} + \boldsymbol{b}_{f}^{p}\right), \\ \boldsymbol{f}_{t}^{p} = \sigma\left(\boldsymbol{W}_{f}^{p-1,p}\boldsymbol{h}_{t}^{p-1} + \boldsymbol{W}_{f}^{p,p}\boldsymbol{h}_{t-1}^{p} + \boldsymbol{b}_{o}^{p}\right), \end{cases} \tag{4}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ denote weight matrices and bias vectors of gates, respectively (with superscripts denoting the layer indices and subscripts denoting specific gates). σ is the sigmoid activation function.

During DLTMP training, DLTMP is supervised to learn the data by representing the training data $\boldsymbol{X}_t$ at the input layer and dynamically adjusting the above described parameters to obtain the desired output value $\boldsymbol{Y}_t$. In addition, on the basis of the moving horizon scheme [8], we also introduce an input horizon parameter K, which denotes the fixed but relevant number of previous time-slots to learn from in order to accurately predict the future traffic data. Thus, combined with (1), we can acquire $T-K$ training data pairs. Specifically, $\boldsymbol{X}_t = [\boldsymbol{D}_{t-1}, \ldots, \boldsymbol{D}_{t-K}]$ and $\boldsymbol{Y}_t = [\boldsymbol{D}_t]$. By continuously feeding these training data pairs into DLTMP (as shown in Fig. 1), we can gradually obtain an accurate prediction model, which could capture the spatio-temporal features among OD flows faithfully, and finally, build the precise predicting function $g(\cdot)$. After that, when the obtained prediction model comes to unknown traffic data, it can fast and accurately calculate the outputs and achieve the future TM.

Simulation results: In our experiments, the real-world data from Abilene network [2] are used to simulate the TM prediction performance of DLTMP. Concretely, we select 600 time-slots in the data from 1st to 7th March 2004. The first 400 time-slots are used for training and others for testing (predicting). In our simulations, the number of deep LSTM layers m and the input horizon K are both set to 2 empirically. The optimiser *MSRProp* is used to optimise DLTMP. The entire data set is normalised between 0 and 1. We employ Keras [8] to implement DLTMP. We also compare our prediction approach with three typical methods, that is, TomoGravity, PCA, and BPNN. We use spatial relative error (SRE) and temporal relative error (TRE) in [3] to evaluate the performance of TM prediction approaches.

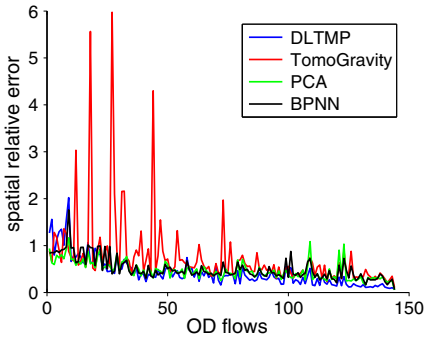

Fig. 2 *Comparison of SRE*

Fig. 2 shows the SREs of these four methods, where the x-axis denotes OD flows attained by sorting all OD flows in ascending order by the mean of their real values, and the y-axis indicates the SRE. Fig. 2 contributes to three facts. First, the trends of SRE with DLTMP, PCA, and BPNN are consistent and decrease with the increase of OD flow volume; however, the trend of SRE with TomoGravity fluctuates obviously. Second, these four methods cannot predict OD flows with small traffic volume well, but in contrast, the error with DLTMP is minimal. Finally, although PCA and BPNN have less error on the prediction results of partial OD flows, they are limited to a very small amount of OD flows.

Fig. 3 shows the TREs of these four methods. From Fig. 3, we find our DLTMP attains a less TRE than the other methods across the entire time-slots. In addition, over the whole time-slots, TREs with TomoGravity, PCA, and BPNN make up and down fluctuation near 1.0 and may yield the burst changes, e.g. at time-slot 172. However, TRE with DLTMP makes up and down fluctuation only near 0.2, and it does not yield the burst changes. Thus, as shown in Figs. 2 and 3, we can draw a conclusion that DLTMP can achieve accurate TM prediction performance with smaller prediction error than TomoGravity, PCA, and BPNN.

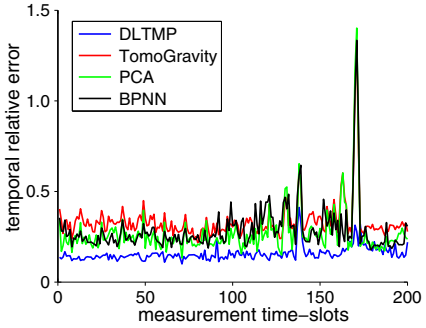

Fig. 3 *Comparison of TRE*

Conclusion: In this Letter, a novel deep architecture based on LSTM RNNs is proposed to model the spatio-temporal features of network traffic, and then a linear regression model is employed to accurately calculate the future predicted TM. The results of our simulations demonstrate that our TM prediction approach based on DLTMP architecture outperforms other typical TM prediction approaches and can achieve state-of-the-art TM prediction performance.

Acknowledgment: This work was supported by the National Natural Science Foundation of China (grant nos. 61531013 and 61371087).

Submitted: *2 February 2018* E-first: *27 March 2018*
doi: 10.1049/el.2018.0336
One or more of the Figures in this Letter are available in colour online.

Jianlong Zhao and Hua Qu (*School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, People's Republic of China*)

✉ E-mail: z.jl199235@stu.xjtu.edu.cn

Jihong Zhao (*School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, People's Republic of China*)

Dingchao Jiang (*School of Electronic and Information Engineering, Xi'an Jiantong University, Xi'an 710049, People's Republic of China*)

References

1 Tune, P., and Roughan, M.: 'Controlled synthesis of traffic matrices', *Trans. Netw.*, 2017, **25**, (3), pp. 1582–1592
2 Nie, L., Jiang, D., Guo, L., *et al.*: 'Traffic matrix prediction and estimation based on deep learning for data center networks'. Proc. IEEE GLOBECOM Workshops, 2016, pp. 1–6
3 Soule, A., Lakhina, A., Taft, N., *et al.*: 'Traffic matrices: balancing measurements, inference and modeling', *ACM SIGMETRICS Perform. Eval. Rev.*, 2005, **33**, (1), pp. 362–373
4 Zhang, Y., Roughan, M., Duffield, N., *et al.*: 'Fast accurate computation of large-scale IP traffic matrices from link loads', *ACM SIGMETRICS Perform. Eval. Rev.*, 2003, **31**, (1), pp. 206–217
5 Zhou, H., Tan, L., Zeng, Q., *et al.*: 'Traffic matrix estimation: a neural network approach with extended input and expectation maximization iteration', *J. Netw. Comput. Appl.*, 2016, **60**, (6), pp. 220–232
6 LeCun, Y., Bengio, Y., and Hinton, G.: 'Deep learning', *Nature*, 2015, **521**, (7553), pp. 436–444
7 Gers, F.A., Schmidhuber, J., and Cummins, F.: 'Learning to forget: continual prediction with LSTM', *Neural Comput.*, 2000, **12**, (10), pp. 2451–2471
8 Ghaderi, A., Sanandaji, B.M., and Ghaderi, F.: 'Deep forecast: deep learning-based spatio-temporal forecasting', arXiv preprint arXiv:1707.08110, 2017