Predicting the Metro Passengers Flow
by Long-Short Term Memory

Zhen Hu[✉], Yi Zuo, Zhuyin Xue, Wenting Ma, and Guilin Zhang

Science and Technology on Information Systems Engineering Laboratory,
Nanjing Research Institute of Electronic and Engineering, Nanjing 210007, China
`49859211@qq.com`

Abstract. In this paper, LSTM is proposed to predict metro passengers flow to avoid traffic jams for the city governors. The model is validated by manual counted data and the results show that LSTM can report an instructive prediction.

Keywords: Traffic prediction · LSTM · Neural network

1 Introduction

Long-Short Term Memory, also known as LSTM, has been widely used in the field of Natural Language Processing (NLP) and has exposed great potential. It's generally known that words in sentences were determined not only by the near-by words but also words far away in the sentence. For example the third "he" in the sentence "He says that he sow a cute cat with the color of white that he loves most" was determined by the near-by word "love" and the far-away words the first and the second "he". By introducing the forget gate, LSTM can merge memories by marking near-by and far-away words simultaneously, and thus has shown a good performance in NLP.

As the growing number of citizens in large metropolises such as Beijing, Shanghai and Shenzhen, public traffic jam has become a big problem. Due to the large transport capacity, metro, or subway, is a critical cure to the over-population in big cities. However, the routes of metro were set in advance and the rail was exclusive. The ceiling of metro was surpassed easily with the development of the cities. Metropolises in China have now introduced the tide strategy to relief the burden of traffic, which is to increase the number of metro in rush hours. Yet, the number of passengers varies widely between different days, and traffic jams are heavily time to time. It will be helpful if the governors know the amount of passengers in advance while the governors can make wise decisions.

Apparently, the number of metro passengers was determined not only by the passenger flow at previous moment but also the passenger flow at the same time in the

days before. It's reasonable that LSTM handles the passenger flow by combining the influence from near-by and far-away flows.

In this paper, we introduce LSTM to estimate the metro passenger flow. We trained the LSTM model by the manual-counted data for 30 days and tested the model by the manual-counted data for the following 30 days.

2   Related Work

LSTM was firstly introduced in 1997 By Hochreiter and Schmidhuber [1]. LSTM was proposed to overcome the problem of gradient vanishing in Recurrent Neural Network (RNN). Since RNN was trained by unfolding it to a truncated deep neural network whose weight matrix was the same between layers, the gradient vanished quickly as the depth of the truncated network growing. By introducing a self-connected with fixed weight one in RNN, the gradient can transmitted without vanishing or exploding.

The name, Long-Short Term Memory, came from the intuition that with gradient vanishing, RNN can remember the long-term memories as the slowly changing normal weights and the fixed weight one can remember the short-term memories as gradient transmitted constantly. Units in LSTM, which were called cells customarily, were composed by different types of nodes which were listed below [9].

- **Memory cell input node:** takes a normal activation function to the input and set the output to the input gate;
- **Input gate:** the unique construction of LSTM [8] compared with traditional RNN which takes the memory cell input as well as the output of memory cell input node at the previous time step and then takes a normal activation function to the combined input.
- **Forget gate:** introduced by Gers et al. which provides a mechanism that makes the memory cell forget the value at hand [2].
- **Memory cell output gate:** similar to the input gate, the output gate takes the product of the value of the memory cell and weight kept by the output gate as the output of the memory cell.

Due to the limitation of computational hardware, LSTM was far more than practice, let alone NLP. LSTM was introduced to NLP by Müller et al. in 2013 [3]. They extracted LDA features as well as the traditional bag of words features as the input of the LSTM model and output the conditional probability of the coming word given the current word. The experimental results showed the outperformance of traditional n-gram method. The network used by Müller et al. is shown in Fig. 1(a).
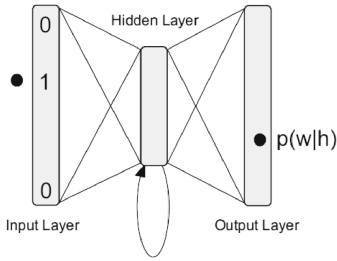
(a) Model used by Müller et al.[3]
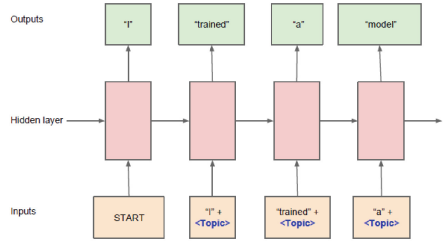

(b) Model used by Ghosh et al.[5]

**Fig. 1.** Model used by Müller et al. and Ghosh et al. respectively.

Similar to Müller et al., Le et al. proposed a mixture-of-expert system which used LSTM as expert selection model. The whole system was designed to generate a dialogue sentence. In 2016, Ghosh et al. proposed the Contexture-LSTM model, [5] which takes the topic of words as the input of the LSTM model. The architecture of their model is shown in Fig. 1(b).

3   Model

There are numerous metro stations in the investigated city and for every station we introduced one LSTM cell to simulate the memory. Our model was implemented by Theano and we used the original LSTM cell provided by Theano [6, 7]. The architecture of LSTM cell is shown in Fig. 2(a). The model was trained and tested on Tesla K80 GPUs.
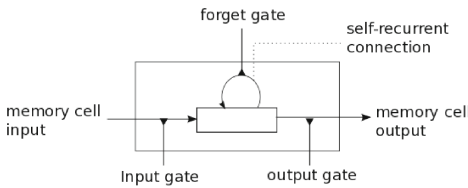
(a) Architecture of LSTM cell used.
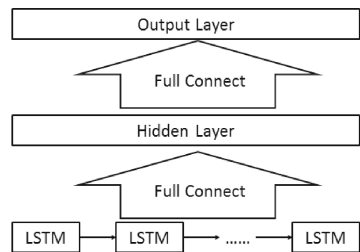
(b) Architecture of LSTM model used.

Fig. 2. Architecture of LSTM cell and model in this paper.

The overall structure is shown in Fig. 2(b). The outputs of LSTMs were taken as the input for a 3-layer neural network whose output indicates the passenger flow at the next time step.

4   Experiment

In our experiment, the data was collected by more than one collectors and the average value was taken as the input of LSTM both in-direction and out-direction for every metro station. The data was collected on rush hours that maybe slightly different between holidays and working days. So totally, we have about 6,000 training samples and 6,000 testing samples in total. Since our model was designed to avoid traffic jam, we only considered the time steps when no less than 100 passengers get into/out of the station.
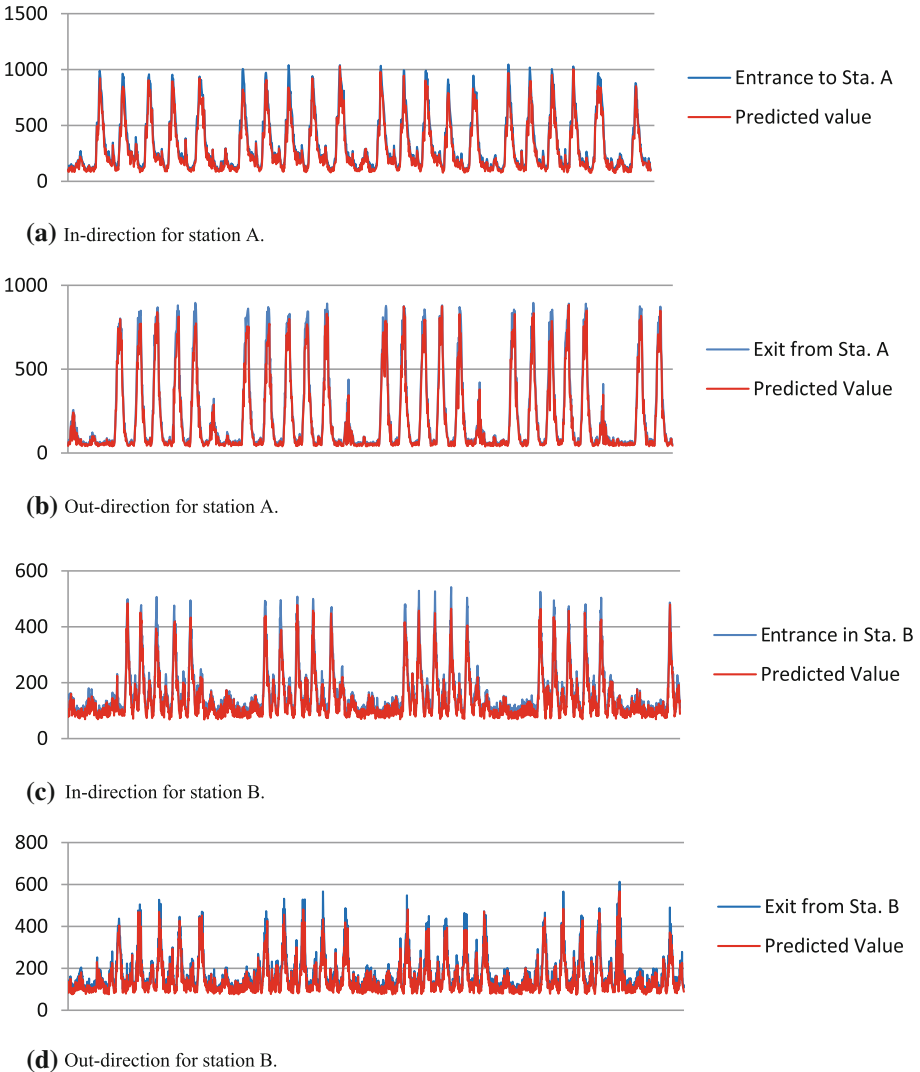

(a) In-direction for station A.

(b) Out-direction for station A.

(c) In-direction for station B.

(d) Out-direction for station B.

Fig. 3. Some typical results.

And the mean error for all stations and both directions are 24.6% and 27.2% for the in-direction and 22.1% for the out-direction. We present some typical results in Fig. 3.

5   Conclusion

In this paper, we explored the potential of LSTM on metro passenger flow prediction. The experiment is promising while the computation cost remains too high without GPUs. In our future work, we shall propose variant LSTM model to improve our model both in accuracy and in computation efficiency.

References

1. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735 (1997)
2. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. Neural Comput. **12**(10), 2451–2471 (1997)
3. Soutner, D., Müller, L.: Application of LSTM neural networks in language modelling. In: International Conference on Text, Speech and Dialogue, pp. 105–112. Springer, Heidelberg (2013)
4. Le, P., Dymetman, M., Renders, J.M.: LSTM-based mixture-of-experts for knowledge-aware dialogues. arXiv preprint arXiv:1605.01652 (2016)
5. Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., Heck, L.: Contextual LSTM (CLSTM) models for large scale NLP tasks. arXiv preprint arXiv:1602.06291 (2016)
6. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., Bengio, Y.: Theano: new features and speed improvements. arXiv preprint arXiv:1211.5590 (2012)
7. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy), Austin, TX, June 30–July 3, 2010 (2010)
8. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Interspeech, vol. 31, pp. 601–608 (2012)
9. Gers, F.A., Schraudolph, N.N.: Learning precise timing with LSTM recurrent networks. JMLR.org (2003)