A Comparison of Traffic Flow Prediction Methods Based on DBN

Huachun Tan[1,*]; Xuan Xuan[2]; Yuankai Wu[3]; Zhiyu Zhong[4]; and Bin Ran[5]

[1]Dept. of Transportation Engineering, Beijing Institute of Technology, Beijing 100081, China (corresponding author). E-mail: tanhc@bit.edu.cn
[2]Dept. of Transportation Engineering, Beijing Institute of Technology, Beijing 100081, China. E-mail: 2120140500@bit.edu.cn
[3]Dept. of Transportation Engineering, Beijing Institute of Technology, Beijing 100081, China. E-mail: 5433809@bit.edu.cn
[4]Dept. of Transportation Engineering, Beijing Institute of Technology, Beijing 100081, China. E-mail: 451828104@qq.com
[5]Dept. of Civil and Environmental Engineering, Univ. of Wisconsin-Madison, 1415 Engineering Dr., Madison, WI 53706. E-mail: bran@wisc.edu

Abstract

Accurate and real-time traffic flow prediction nowadays shows more and more dependence on big transportation data. Deep learning, a powerful method for feature learning, has turned out to be an effective tool to cope with these explosive data. Recently, deep models, especially unsupervised models like Deep Belief Networks (DBN) and Stacked Autoencoder (SAE), are being employed into the field of traffic research and have shown great prospect. However, there is still a vacancy in the exploration on comparing the performances of different kinds of deep architectures to find an optimal solution. In this paper, we set up two deep-learning-based traffic flow prediction models for feature extraction and performances comparison: One is a Deep Belief Networks (DBN) based on Restricted Boltzmann machines (RBMs) that have Gaussian visible units and binary hidden units, and the other is a DBN based on RBMs with all units being binary. A conclusion is drawn where the former one performs better in traffic flow prediction after a series of experiments.

Keywords: Deep learning; Deep Belief Networks (DBN); Restricted Boltzmann machines (RBM); Traffic flow prediction.

INTRODUCTION

The explosion of traffic information, along with development of data collection techniques, has made research in this area increasingly significant for many applications. For example, data describing call detail records (CDRs) help to provide rich spatiotemporal information about human mobility patterns and offers digital footprints at a scale and resolution which make sense (Jiang and Murga et al. 2015). Also, data from whole urban networks enable us to accurately forecast traffic flow based on proper model which can fully utilize the underlying spatial-temporal information of such a big data (Li and Wang et al. 2015). Obviously, all these applications based on abundant data rely on a model that can fully extract a meaningful pattern for the specific target.

Deep Learning (Huang et al. 2014) is an efficient and valuable tool that features extraction from big data, especially the Deep Belief Networks (DBN) model, and has recently intrigued the transportation researches. For instance, Wenhao Huang et al. (2014) made the first step to employ DBN-based models into transportation research in. They built a model with DBN and found that deep learning is a promising method for transportation research. Except for the application of DBN methods, traffic state prediction had also been done with the Stacked Autoencoder (SAE) model (Bengio et al. 2007), which is another structure of deep neural network which shares a similar mechanism with DBN. Yisheng Lv et al. (2015) applied a SAE model to predict expressway traffic flow and found that the SAE model performed better in discovering the latent traffic flow feature compared with the Back Propagation Neural Network (BP-NN), the Random Walk (RW), the Support Vector Machine (SVM), and the Radial Basis Function Neural Network (RBF-NN) models. Recently, a research on continuous travel time prediction using SAE-based Deep Network was conducted by Gang Xiong et al. (2015), and found that the deep model can meet the requirement of practical application.

Considering the strong feature learning capacity of a deep model, the studies mentioned above all took deep architectures as a means of feature extraction providing assistance to the final targets of the built model. However, the deep architectures are diverse and even the same structure can obtain different results responding to different formation mechanisms (Bengio et al. 2007; Bengio 2009; Hinton 2010). Choosing an appropriate model is also important for a research since it takes an important role in feature representation.

In this paper, we pay more attention to the deep architecture itself. Firstly, we build two DBN models which have different kinds of modules and compare these models to find the best one to deal with traffic flow prediction. Then conclusions are drawn according to the performance of the two architectures, DBN based on the Gaussian visible units, and binary hidden units-RBMs, which one performs better and if it deserves further research.

The whole article is structured as follows. We introduce several prior researches conducted on applying the deep learning method into the field of transportation. Then, we give illustrations to the core theories of RBMs, whom the different kinds of DBN models are built based on, with details followed by an introduction of the comparison methodology. We then conduct a series of experiments with different models and give a description of the experimental results with an emphasis on the comparison experiment. And lastly, discussions and conclusions are presented where we discuss the problems that remain to be taken into consideration and possible directions for future research.

BACKGROUND

It has been confirmed that the deep learning method is efficient and promising in learning the patterns of a transportation system by the existing deep model-based traffic researches (Lv et al. 2015;Gang et al. 2015; Huang et al. 2014). Here, we give a review on the successful cases for a better understanding of the deep learning-based prediction methodologies.

Deep learning approaches are first introduced into transportation research by Wenhao Huang et al. in 2013. They used DBN models based on real-valued units which have Gaussian noise to model the traffic flow data in their experiments.

In their study they conducted a series of experiments on single road traffic flow prediction by employing a DBN for feature extraction and then a sigmoid regression layer above the DBN for prediction. A database was formed in this way: they aggregated the data obtained from the Caltrans Performance Measurement System (PeMS), a system where data is continuously collected every 30 seconds by over 15000 individual detectors distributed in more than 8100 freeway locations throughout the State of California.

They also applied DBN into Multi Task Learning (MTL), which learns several tasks at the same time with the aim of a mutual benefit using a shared representation. They took the prediction of one road or station as a task and jointed different tasks to form a MTL. The objective was to learn a shared feature representation for all tasks. A database was formed by data that was collected from the highway system of China. This article was the first work employing deep learning into a transportation area which had verified the availableness and effectiveness of the DBN method in traffic researches with successful experiments.

Another deep structure named Stacked Autoencoder (SAE) (Bengio 2009) has also been used into traffic study (Lv et al. 2015). In 2013, Yisheng Lv et al. applied a stacked autoencoder (SAE) model to predict expressway traffic flow.

Sharing a similar mechanism with DBN, a SAE model is created by stacking autoencoders hierarchically to form a deep network which takes the output of the autoencoder found on the lower layer as the input of the current layer (Bengio et al. 2007).

According to Yisheng Lv et al. (2015), they employed a logistic regression layer as a predictor on the top of the SAE to make the SAE network a tool for traffic flow prediction. To train this tool, they used a greedy layer wise unsupervised learning algorithm for pre-training and a Back Propagation algorithm for fine-tuning in the same way as the DBN architecture introduced previously. The model was then tested on a dataset of data collected by PeMS. They observed the changing performance of the model in dealing with predication tasks with the parameters of the model being adjusted regularly, and then made comparisons between the SAE model and existing methods.

Thanks to the help of this article, deep method, compared with the BP NN, the RW, the SVM, and the RBF NN models, turned out to be instrumental in discovering the latent traffic flow feature representation and was superior to the competing methods.

Besides traffic flow prediction, other traffic state predictions based on deep learning were taken into consideration as well. Recently, a new research on continuous travel time prediction using SAE-based Deep Network was conducted by Gang Xiong et al.(2015). Actually, the model they built in this article was all the same in both structure and training method with the one built by Yisheng Lv et al. in the study in 2015.

Differences came from the conditions of the dataset and the selected location for prediction. In this paper, the study site was designed to be a typical four-approach signalized intersection and travel time dataset were organized referring to the data information at two points of which one is the start point and the other is the end point of data collection. They collected 150694 samples in total by the API (Application Programming Interface) functions in a simulative way. The SAE model was proposed to predict the continuous time it took to travel some distance.

Conclusions were drawn where the proposed model, as a deep learning-based method, can well meet the requirement of a practical application directed by indexes of mean absolute error (MAE) and mean absolute percentage error (MAPE). They conducted this experiment with simulate data and the results were supposed to be more powerful when being validated against data set collected in real world.

DBN architecture and implementation

Essentially, DBN is a deep Artificial Neural Network (ANN) containing more hidden layers compared with the ordinary Artificial Neural Network (Bengio et al. 2007; Bengio 2009; Hinton 2010). Because each layer of the model shares parameters with a specific Restricted

Boltzmann Machine, we always say that DBN model is a stack of RBMs and RBMs are always treated as building blocks of DBN (Bengio et al. 2007; Hinton 2010). Different kinds of RBMs lead to different adjusted parameters followed by the different results of feature extraction. We denote DBNs with different kinds of 'building blocks' as different models in spite of their similar structures. Here the models used in this paper are introduced.

RBM

An RBM is an undirected graphical model where visible units are stochasticly connected to hidden units using undirected weighted connections. Meanwhile, there are no connections within hidden variables or visible variables (Hinton 2010). Figure 1 gives an illustration of an RBM, which has two layers in total: one visible layer and one hidden layer.

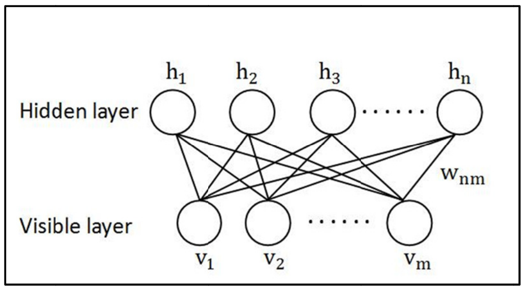


Figure 1. Structure of an RBM

Unlike normal Neural Network (NN) models, the RBM block works based on the theories of entropy and energy. A connected pair of the visible and hidden units, denoted as (v, h), can define an energy E(v, h) computed by a certain formula to describe the distribution information in this connected pair, and this energy then defines a probability distribution $P(v, h)$ through which we can compute the marginal probability P(v) or P(h). Anyone who is interested in the very formulas here can see research performed by Hinton (2010) for details.

Next, the log probability of P(v) is used as a cost function. Our entire training target is to minimize this cost function whose derivative with respect to the parameters in the model responds to the updating of the parameters during the whole training process. Taking the learning process of weight w_{ij} as example, the updating is done with Δw_{ij} defined as follows:

$$\Delta w_{ij} = \epsilon\left(\frac{\partial \log P(v)}{\partial w_{ij}}\right) \ ,$$

where, $\log P(v)$ is the cost function and ϵ is a learning rate.

Contrastive Divergence is in charge of updating the state or value of a layer according to the marginal probability $P(v)$ or $P(h)$ during which process iterations of alternating Gibbs sampling are used. The procedures can be shown briefly as follows (Bengio et al. 2007; Hinton et al. 2006; Hinton 2010):

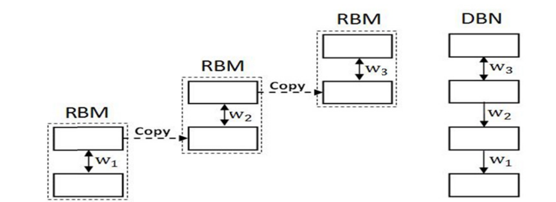
$$P(h|v_0) \to \mathbf{h_0},\ P(v|h_0) \to \mathbf{v_1},$$
$$P(h|v_1) \to \mathbf{h_1},\ P(v|h_1) \to \mathbf{v_2},$$
$$\ldots\ldots$$

While different kinds of units here lead to different calculation results of energy function and marginal probability (Hinton 2010), this then leads to different parameter adjustments. Sharing parameters with DBN, RBMs with different kinds of units cause differences between similar DBN models (Hinton et al. 2006).

In this paper, two different kinds of RBMs are used to find DBN models. They are RBM with binary visible and hidden units, and RBM with Gaussian visible units and binary hidden units. We call them B-B RBM and G-B RBM, for short. The relevant energy formulas of these models are introduced in research by Hinton in 2010.

DBN

A DBN model is built by stacking RBMs. The word stack here refers to the process of taking the output of a hidden layer as the input of the higher layer next to it. This process we call copy as shown in Figure 2, and in this way multiple RBMs can be stacked hierarchically (Bengio et al. 2007; Bengio 2009; Hinton 2006).



a) Hierarchical training b) RBMs stack into DBN

Figure 2. DBN Stacked by RBMs

Two different models are used in this article of which one is composed by B-B RBM-based hidden layers, and the other contains the G-B RBM-based hidden layers. As well as the different RBMs, the two different kinds of DBNs are named briefly to be B-B DBN and G-B DBN.

A predictor then should be attached on the top layer to fulfill the task of traffic flow prediction. In this paper, we put a linear regression layer using ordinary least squares on top of the network. The linear regression layer can be also replaced with other regression models such as a sigmoid regression layer or an SVR (support vector regression) layer (Lv et al. 2015). For fair comparison, different DBN models must share a same regression algorithm.

The DBN attaching a predictor on its top layer comprise the whole deep architecture model for traffic flow prediction. This is illustrated in Figure 3.

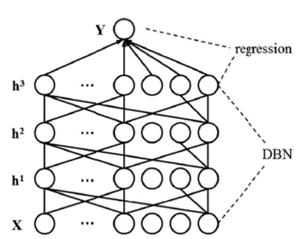


Figure 3. Structure of a Finished DBN-based Model

Comparisons

With the built models, prediction can be done followed by the comparisons of the two models.

Comparisons will be done at the following aspects:

1) Graphs demonstrating the performance of these predicating models should be given firstly for a visual comparison.
2) The models are built mainly for achieving the practical goal of prediction, so the predicted results are significant. In this paper, we analyze the predicted traffic flow with three evaluation metrics, namely Mean Absolute Error (MAE), Mean Absolute Percent Error (MAPE), and Root Mean Square Error (RMSE). They are calculated as follows：

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - \hat{f}_i|$$

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

$$\mathrm{RMAE} = \left[\frac{1}{n}\sum_{i=1}^{n}|f_i - \hat{f}_i|^2\right]^{\frac{1}{2}}$$

3) Discussions to explain the different results should be given aiming at a final conclusion of the study.
4) Comparisons should also be conducted between deep architecture and the shallow one.

EXPERIMENTS

Datasets Description

In this paper, we test two DBN models based on a same database of which the data are obtained from the Caltrans Performance Measurement System (PeMS). As explained earlier, PeMS is a system where the data are continuously collected every 30 seconds by over 15000 individual detectors distributed in more than 8100 freeway locations throughout the State of California. The collected data are then aggregated into time intervals of 5-min before being uploaded to the Internet (Caltrans Performance Measurement System 2014; Lv et al. 2015).

As shown below in Figure 4, among all the data displayed in this system we firstly choose a target location， O_i, whose records we are to predicate with our models. Taking the randomness of the traffic state into consideration, this location choice is all made randomly. With the location O_i being chosen， the two points, O_{i+1} and O_{i+2}, lying in its upstream as well as the two in downstream O_{i-2} O_{i-1}, are selected at the same time. Let q_i^t denote the quantities of the observed cars at O_i during the t-th time interval, then the predication problem can be expressed as a process to learn the value of q_i^t based on the observed traffic flow sequence $\{q_{i+\Delta}^{t-\tau}\}$, where, Δ = -2, -1,0,1,2 as shown in Figure 4. Here t denotes an arbitrary time interval in which the traffic flow remains to be predicted. τ=1,2,…,m, where m means a previous m time interval. To decide the value of m we test our models with m ranging from 1 to 12 (5 min–1 h), and finally set m to be 5. That is to say we employ a block of observed data in size of 25 to predicate q_i^t, i. e. the traffic flow of O_i at the t-th time interval.

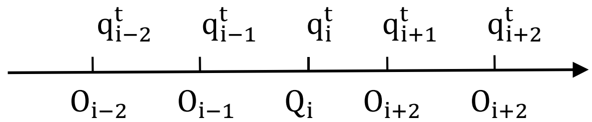


Figure 4. Description of the Selected Collecting Points

In the experiment, all of the data collected by these selected detectors in the periods from 2014.04.01 to 2015.06.01 are employed, and we divide the data into three parts. The training set comprises the data of the first 42 weeks. The validating set contains data collected from the 43rd week to the 52nd week, and the testing one is formed with data of the remaining 9 weeks. Then, each database is made into a matrix in the size of (T*25), where a vector in a row represents a learning sample, and the length of the column T means the number of examples.

Deep Structure Setting

In this paper, we build two DBNs: the B-B DBN, a deep network stacked by RBMs with binary-binary units, and the G-B DBN, one stacked by RBMs with Gaussian-binary RBMs and binary-binary RBMs (Hinton 2010).

The main parameters needed to be setup in these two models are nearly the same: the size of input and output layers, the number of hidden layers, as well as the number of units in each hidden layer.

The input layer is just the original data input, so the size of it is 25, as illustrated above. The size of the output layer is decided by the destination of the model. For example, in our experiment, the goal is to obtain the temporal flow of location O_i at a fixed delta-t，so the size of the output layer in our models are all 1. If the experiment is designed to estimate joint locations as a multi-task, the number of the units in the output layer should matches with that of the target locations (Huang et al. 2014).

We conduct a series of experiments separately on the two models to find out their respective perfect parameters leading to the best performance. Sharing the same parameter setting, models are confirmed as follows: number of hidden layers=3, units in layers = 100, batch size=10, pre-training learning rate = 0.06. Predictions are done by these two models followed by comprehensive comparisons.

Neural Network Architecture

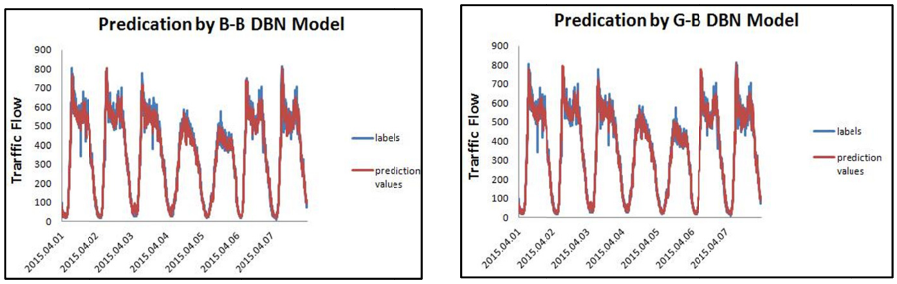
We also design comparisons between the winner of the former experiments and a normal Neural Network (NN) architecture so that we can make a further verification on the effectiveness of the Deep Belief Networks model based on these comparisons.

The NN model is formed with only one hidden layer who has 100 nodes and trained by the Back-Propagation algorithm (Smith and Demetsky 1994).

Additionally, we build these models on a Python console, and Theano is the Python library that we employ to define, optimize, and evaluate mathematical expressions (Bastien et al. 2012).

Results

We use the best performance of the two models respectively for comparison. The results of the prediction tasks are reported separately in Figure 5.



a) prediction by B-B DBN model b) prediction by G-B DBN model

Figure 5. Performances of the Two Models

From the pictures in Figure 5 we can see that the performances are nearly the same and match well with the observed one from a general view. This phenomenon validates DBN models to be applicable tools in transportation research once again.

However, although the traffic flow predicted by those two well trained models shared similar traffic patterns with the observed flow sets, the fitting results show differences to a certain degree according to Figure 6.

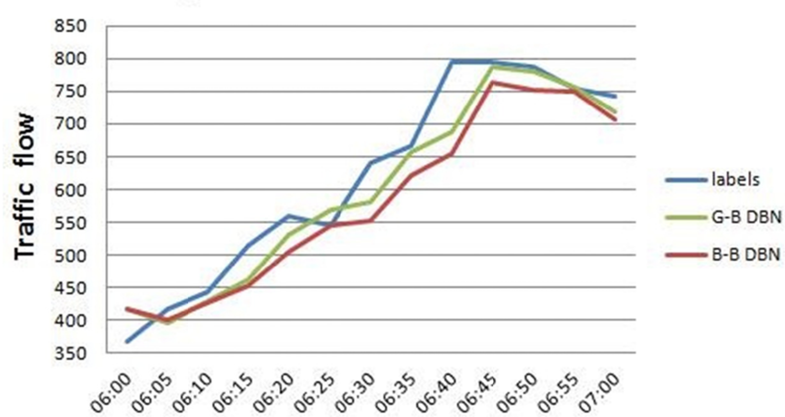


Figure 6. Comparison Result Between Different DBNs

In Figure 6, we give a result of the target comparison visually by showing the output of the two proposed models for the traffic flow predicted in a typical time interval.

Intuitively speaking, the distance, which can be treated as an absolute error of the prediction result, between the green and the blue curves is smaller than that between the red and the blue curves for the most part, i.e., the G-B DBN architecture performs better. This conclusion can further be made with more reliability by analyzing the statistic data presented in Table 1.

Table 1. Comparison results between different DBNs

Model	MAE	MAPE (%)	RMSE
B-B DBN	23.55	10.06	32.65
G-B DBN	22.26	9.21	30.82

In Table 1, the analysis of the prediction results are displayed. The average accuracy (1-MAPE) obtained by the G-B DBN is improved slightly (about 0.8%) than that of which comes from a B-B DBN model. It verifies the prior conclusion that the G-B DBN method, instead of a B-B DBN-based one, performs more effectively for traffic flow prediction showing the lower MAE and higher average accuracy (1-MAPE).

The reason for this phenomenon is supposed to be that a B-B RBM is a two-layer network where the hidden units and reconstructed visible units are selected to be alternative states of 0 or 1, and this approach can cause unnecessary sampling noise (Erhan et al. 2009; Hinton et al. 2006). Relatively, a G-B RBM is a variant of B-B RBM whose visible units are replaced by linear units with independent Gaussian noise. The real-value visible units in G-B RBM can reduce sampling noise thus allowing faster learning.

Figure 7 gives a clear presentation about how differences came into being by way of visualizing the features extracted by each internal layer (Zeiler and Fergus 2014). We choose four examples from the testing data set and show them out with gray images in Figure 7.

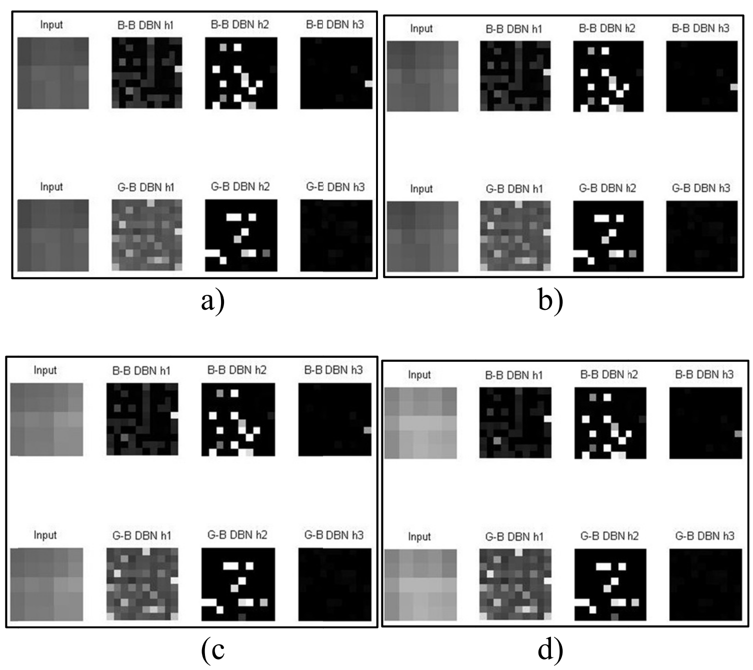


Figure 7. Visualization of the Feature Extraction Processes

From the four examples we can find that the differences between these two models are obvious in the first hidden layer. Based on RBMs who have Gaussian visible units in it, the G-B DBN model can map the input data into the first layer better than that of the B-B DBN.

An additive experiment of the comparison between G-B DBN and original Neural Network (Smith and Demetsky 1994) is then carried out for a further validation on the effectiveness of the DBN method. Comparison results are represented in the same way by a Figure 8 and a Table 2.

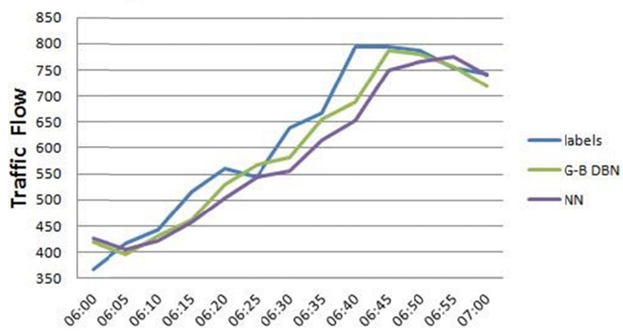


Figure 8. Comparison Results Between G-B DBN and NN

Table 2. Comparison results between G-B DBN and NN

Model	MAE	MAPE (%)	RMSE
G-B DBN	22.26	9.21	30.82
NN	22.68	9.63	31.48

It can be summarized that the DBN architecture can reach better prediction results almost at any time, with an improvement on the accuracy by over 0.43%. Thus, conclusions are organized where the DBN-based method is effective in traffic flow prediction and it is promising to apply DBN into transportation research.

CONCLUSIONS

In this paper, we investigate the effectiveness of DBN-based methods in traffic research with a special view on comparing different DBN models. The existing researches all set emphasis on validating the suitableness of deep architectures where as our method tells which one of the many DBN models is better.

We propose two DBN models for prediction and the comparison between them has shown that the difference of the capability in dealing with traffic data exists, although in an unobvious way. The DBN stacked by the G-B RBMs turned out to be a better model to extract features from traffic data compared with B-B RBMs.

Four possible perspectives are put forward here for further studies:

1) Mathematically explain why the differences exist between these two models, and then mine the deep information reflecting the operating rule of the traffic.
2) Analyze the effect of each parameter of the model to clarify how the prediction error develops, and then explore more possible improvements of deep learning-based method in traffic research.
3) Take other conditions such as weather conditions, driver behaviors and so on into consideration since they have a great relationship within the range of traffic state.
4) Carry out comparisons among more models based on different deep architectures such as Stacked Autoencoder (SAE), Convolutional Neural Network (CNN), and their respective variations to extend the study of finding an optimal deep learning-based traffic flow prediction model.

ACKNOWLEDGEMENT

The research was supported by NSFC(61271376，51308115), the National Basic Research Program of China (973 Program: No. 2012CB725405), and Beijing Natural Science Foundation (4122067).

REFERENCES

Alexander, L., Jiang, S., Murga, M., et al. (2015). "Origin–destination trips by purpose and time of day inferred from mobile phone data[J]." *Transportation Research Part C: Emerging Technologies.*

Bastien, F., Lamblin, P., Pascanu, R. et al. (2012). "Theano: new features and speed improvements[J]." *arXiv preprint arXiv*:1211.5590.

Bengio, Y., Lamblin, P., Popovici, D. et al. (2007). "Greedy layer-wise training of deep networks[J]." *Advances in neural information processing systems*, 2007, 19: 153.

Bengio, Y. (2009). "Learning deep architectures for AI[J]." *Foundations and trends® in Machine Learning,* 2(1): 1-127

Caltrans Performance Measurement System (PeMS). (2014). [Online].Available: http://pems.dot.ca.gov

Erhan, D., Bengio, Y., Courville, A., Vincent, P. (2009). "Visualizing higher-layer features of a deep network." *Technical report*, University of Montreal.

Gang, X., Kang, W., Wang, F. et al. (2015). "Continuous Travel Time Prediction for Transit Signal Priority Based on a Deep Network[C]." *Intelligent Transportation Systems (ITSC)*, 2015 IEEE 18th International Conference on. IEEE, 2015: 523-528

Hinton, G. E., and Salakhutdinov, R. R. (2006). "Reducing the dimensionality of data with neural networks." *Science*, vol. 313, no. 5786, pp. 504–507, Jul.

Hinton, G. E., Osindero, S., the, Y. W. (2006). "A fast learning algorithm for deep belief nets[J]." *Neural computation*, 18(7): 1527-1554.

Hinton, G. (2010). "A practical guide to training restricted Boltzmann machines[J]." *Momentum*, 9(1): 926.

Huang, W., Song, G., Hong, H. et al. (2014). "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning[J]." *Intelligent Transportation Systems, IEEE,* 15(5): 2191-2201.

Li, L., Su, X., Wang, Y., et al. (2015). "Robust causal dependence mining in big data network and its application to traffic flow predictions[J]." *Transportation Research Part C: Emerging Technologies.*

Lv, Y., Duan, Y., Kang, W., et al. (2015). "Traffic flow prediction with big data: a deep learning approach[J]." *Intelligent Transportation Systems, IEEE Transactions* ,16(2): 865-873.

Smith, B. L., and Demetsky, M. J. (1994). "Short-term traffic flow prediction: Neural network approach." *Transp. Res. Rec*., vol. 1453, pp. 98–104.

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks[M]." *Computer Vision–ECCV 2014*. Springer International Publishing, 2014: 818-833.