Using Convolutional Neural Network with Asymmetrical Kernels to Predict Speed of Elevated Highway

Di Zang[1,2](✉), Jiawei Ling[1,2], Jiujun Cheng[1,2], Keshuang Tang[3], and Xin Li[4]

[1] Department of Computer Science and Technology, Tongji University, Shanghai, China
zangdi@tongji.edu.cn

[2] The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, China

[3] Department of Transportation Information and Control Engineering, Tongji University, Shanghai, China

[4] Shanghai Lujie Electronic Technology Co., Ltd., Pudong, Shanghai, China

Abstract. In this paper, we present a deep learning based approach to performing the whole-day prediction of the traffic speed for the elevated highway. In order to learn the temporal features of traffic speed data in a hierarchical way, an improved convolutional neural network (CNN) with asymmetric kernels is proposed. Speed data are collected from loop detectors of Yan'an elevated highway of Shanghai. To test the performance of the presented method, we compare it with some conventional approaches of traffic speed estimation. Experimental results demonstrate that our method outperforms all of them.

Keywords: Convolutional neural network · Kernels · Intelligent transportation system · Speed prediction

1 Introduction

It is an attractive topic for human beings to have the ability to foresee the future, and it is the same in transportation management. It is of great importance for traffic management department to learn the traffic evolution to provide a guide of tomorrow's traffic for people to select an unobstructed route. It is also of value for traffic management department to adjust the traffic strategy in advance [1, 2].

However, it is challenging to define a high-performance prediction model, because the utilization of spatiotemporal relationship was not high, we did not have the ability to form a more efficient prediction model to deal with the spatiotemporal correlation of traffic flow in roads expanding on a two-dimensional field, we were not able to forecast long-term future. Conventional traffic data prediction models usually treat the traffic data as sequential data, so these models usually cannot have a good performance, because of the limitations in implementation, assumptions and hypotheses, noisy or missing data, ineptness to deal with outliers and incapability to determine dimensions [3].

Z. Shi et al. (Eds.): ICIS 2017, IFIP AICT 510, pp. 212–221, 2017.
DOI: 10.1007/978-3-319-68121-4_22

In the existing models, there are two main research methods which dominate the study in traffic forecasting: methods based on statistic and methods based on neural networks [3].

In traffic prediction, statistical methods are widely used. The classic method is autoregressive integrated moving average (ARIMA) model. It is a time-series prediction model which considers the correlations in successive time sequences of traffic variables. Seasonal ARIMA model [4], KARIMA model [5] and ARIMAX model [6] which are the extensions of basic ARIMA model are widely researched and applied. In [7], k-nearest neighbors (KNN) has been used to forecast traffic flow. In [8], support vector machines (SVM) were employed in traffic prediction. Online-SVM and Seasonal SVM were used in [9] and [10] to improve the prediction accuracy. Methods based on statistics have been widely applied in traffic prediction because of their easy implementation and promising results. However, these models did not consider the significant spatiotemporal feature of traffic data, so these models cannot achieve a higher accuracy than models based on neural networks. Besides, some statistical methods are powerless because the model takes a very long time and consumes copious computer memory when it deals with big data.

Neural network based methods, such as artificial neural network (ANN) are usually applied to solve traffic prediction problems. ANN is able to deal with multi-dimensional traffic data. Because of its easy and flexible implementation, strong generalization ability and high performance in traffic prediction, ANN model is favored in recent research in traffic prediction. In [11], ANN was used to predict traffic speed with consideration of weather conditions. In [12], a real-time traffic speed prediction algorithm based on ANN was proposed by Park et al. A model based on ANN combined with conventional Bayes theorem to predict short-term freeway traffic flow was proposed in [13]. Moretti et al. [14] used statistical and ANN bagging ensemble model to predict city traffic flow.

ANN can make use of large amounts of data, but it cannot take advantages of spatiotemporal correlations from large amounts of traffic data. ANN are not able to achieve a better performance than methods based on deep learning. Recently, more and more deep learning models are applied to predict traffic flow because deep learning models are able to learn the deeper level features from the given data. Nowadays, Deep Belief Networks (DBNs) are widely used in traffic volume prediction. The model [15] used the method of heterogeneous multitask learning and K-means clustering to improve the prediction accuracy. Ridha et al. [16] combined DBN with weather condition to predict traffic flow using streams of data. Ma et al. [17] proposed a new model combined Restricted Boltzmann Machine (RBM) and Recurrent Neural Network (RNN) forming a RBM-RNN model, it achieves the advantages of both RNN and RBM. In [18], a Stacked Autoencoder based model was proposed to forecast traffic flow. Based on [18], Duan et al. [19] further improved the SAE model by choosing different appropriate hyperparameters at different times. Tian and Pan [20] first introduce Long Short-term Memory (LSTM) into traffic prediction. The LSTM model outperforms other neural networks in both stability and accuracy. In [21], a deep spatio-temporal residual network was applied in crowd flows prediction. Ma et al. [22] proposed a Convolutional Neural Network (CNN) based model which learns traffic data as image, the model achieves a good result in Beijing road network speed prediction. However, the CNN model in [22] treats the traffic dynamic in time series and space equally, and cannot work with the whole-day traffic data.

To solve the problems in [22], this paper introduces asymmetrical kernels to CNN, which can treat the spatial features and temporal features of traffic data differently. Because of different treatment between spatial features and temporal features, our model gets lower mean squared error (MSE) and mean related error (MRE) than common CNN. In addition, the improved model is applied to predict the whole-day traffic speed of the next day with the help of whole-day traffic speed data of the previous day.

2 Proposed Approach

In this section, we will introduce the method of transforming the loop detectors' data to matrix and the basic theory of our CNN model.

2.1 Loop Detector Data Transformation

The traffic speed of elevated highway can be provided by the loop detectors deployed on the highway. In the time dimension, the loop detectors' data range from 0:00 am to 12:00 pm. The time intervals are usually 5 min. On the elevated highway, each two loop detectors are deployed between 400 m. The loop detector data of elevated highway also can be converted to matrices by a similar method. We let x-axes represents time, and y-axes represents space. We arrange the loop detector data of elevated highway in the order of loop detectors' position and time series to form a 2D matrix. Each row in the matrix denotes speed data in different time periods recorded from a same loop detector in the elevated highway. Each column in the matrix denotes speed data from different loop detectors at a same time period. The time-space traffic speed matrix can be represented as follow:

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & & s_{2n} \\ \vdots & & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{bmatrix} \tag{1}$$

In the matrix, m denotes the amount of loop detectors, n denotes the length of time intervals and s_{ij} denotes the average speed on the loop detector i at time period j. We also can represent the matrix as a heat map. Figure 1 is the illustration of a heat map transformed from the matrix.

2.2 The Architecture of the Improved CNN

CNN has been widely used in the research of image understanding, because of its strong ability in extracting critical features from images. In the field of image classification, CNN performs better than other deep learning models, even surpasses human beings. As shown in Fig. 2, the Input of our model are matrices of spatiotemporal traffic data. In our model, there are 3 convolution layers and 2 pooling layers. The input matrix goes through two convolution layers, one pooling layer, one convolution layers, one pooling layer and one fully-connected layer in turn. The output of the model is a vector which can be reshaped to a matrix with the same size of the input matrix.

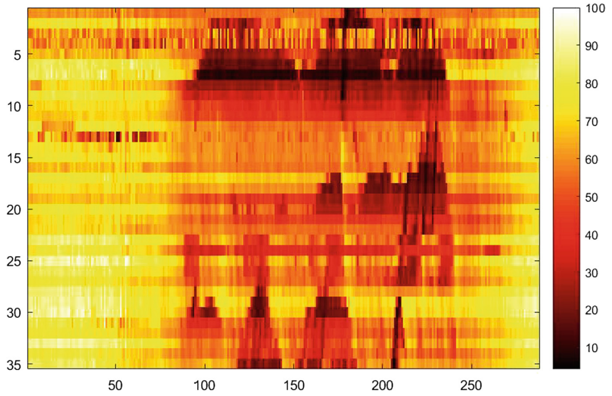

Fig. 1. The visualization of whole-day traffic speed of Shanghai Yan'an elevated highway.

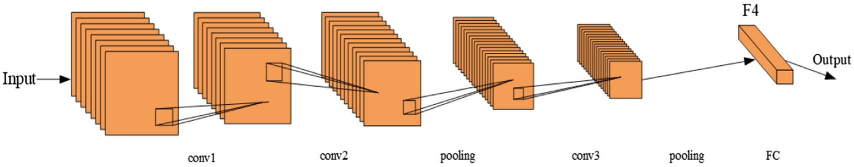

Fig. 2. The architecture of our improved CNN model.

2.2.1 The Model's Input and Output

Like common CNNs, our improved CNN accepts matrices (images) as input. However, in this paper, instead of being used to solve classification problems, we use CNN to finish regressive task. Thus, the output of our model is a vector that can be reshaped to a matrix just the same size as the input matrix which is the prediction of the next day.

2.2.2 Convolution Layers

The previous layer's feature maps or the input matrices are convolved with trainable kernels and then the feature maps are put through the activation function to make up the output feature maps. Each output feature map combines convolutions with multiple input feature maps. In common, the relationship between input and output maps of convolution layers is as follow:

$$x_j^l = f\left(\sum x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{2}$$

where x^{l-1} denotes the input of the convolution layer, k^l means the convolution layer's kernels, b denotes an additive bias, f is the activation function. We usually use sigmoid (3) function or ReLu (4) function as the activation function.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

$$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \tag{4}$$

2.2.3 Asymmetrical Kernels

In common CNN models, the convolution layers' kernels are square, however, in our model, kernels in convolution layers are asymmetrical rectangle matrices, because we consider spatial dynamics and temporal dynamics differently. A traffic congestion event can impact for a long time, sometimes for several hours, while the time intervals in matrices are too small. To solve the problem demonstrated before, we use asymmetrical rectangle kernels, which can capture more temporal dynamics of traffic data. In the models, we use asymmetrical rectangle kernels with 3 × 13 size, 3 × 5 size and 3 × 5 size in different convolution layers.

2.2.4 Pooling Layers

The output of a pooling layer are down-sampled versions of input maps. If there are N input maps, then the number of output maps will be exactly N, but the size of output maps will be smaller. More formally,

$$x_j^l = f\left(\sum \beta_j^l down\left(x_i^{l-1}\right) + b_j^l\right) \tag{5}$$

where the function *down()* denotes a sub-sampling (pooling) function, usually we use max pooling function or average pooling function. Generally, the pooling function will transform each distinct n-by-n block into one pixel of the output map. Then the output maps will be multiplied by a multiplicative bias β and add a bias b.

Although the pooling layer can reduce the amount of model's trainable parameters, it also brings some information loss. In order to reduce the information loss, in our model, we cancel the pooling layer after the first convolution layer. This operation can achieve a bit performance improvement.

2.2.5 Fully Connected Layer

The fully connected layer is similar to the artificial neural network, if we use x to denote the input of fully connected layer, y to represent the output of the layer, the corresponding relation between x and y is as follow:

$$y_j^l = f\left(\sum w_j^l x_i^{l-1}\right) + b_j^l\right) \tag{6}$$

In the formula (6), w denotes the trainable weights between the input and output, f represents the activation function described before.

2.2.6 Model Optimization

To get an optimized model, we use stochastic gradient decent method to minimize the model's mean squared error (presented in formula 7) with the batch size is one.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{7}$$

In formula (7), y denotes the model output, and $\hat{y}$ represents the model's expected value.

3 Experimental Results

The model is evaluated using loop detectors data of Yan'an elevated highway for the year of 2011. In the experiments, we use a whole day's speed data, to predict the next day's whole-day speed data. The first 320 days of data are selected for training and the left days of data are used for testing. We use the matrix of previous day as the input of the model, the reshaped output vector is applied as the predicted value.

3.1 Handle the Data

We handle the data in the way which is similar to the method used in [22]. There are 35 loop detectors deployed on the Shanghai Yan'an elevated highway. In addition, the observed data is recorded every 5 min. Because of some limitation of loop detectors, we did some work in data cleaning. First, we reset the abnormal data, for example, some elements in the speed matrix are larger than 200 km/h, we set these data to be 100, because there are few cars can run at this speed which is also against Chinese law. According to Chinese law, the max speed on the elevated highway is 80 km/s, we take the slight overspeed into account, thus, we choose 100 km/s as the max speed in the matrix. Second, sometimes, the loop detectors employed on the Yan'an elevated highway do not work during 0:00 am to 4:00 am, in addition, we think that most people do not travel from 0:00 am to 6:00 am so we take no account of the data from 0:00 am to 6:00 am. Third, there are 3 loop detectors often cannot work, thus, 3 rows in the matrix are deleted. Last, in order to reduce the impact of abnormal elements in the matrix, we aggregate the data in time-dimension to obtain a 20-min interval. Eventually, the size of matrix is 32 × 54.

3.2 Experimental Settings

There are three convolution layers and two pooling layers. There 16 kernels with the size of 3 × 13 in the first convolution layer, 512 kernels with 3 × 11 size in the second convolution layer and 1024 kernels with 3 × 5 size in the last convolution layer. In the pooling layers, we do max pooling on the input. The experiments are conducted on a sever with i7-5820 K CPU, 48 GB memory and NVIDIA GeForce GTX1080 GPU. We implement these models on TensorFlow framework of deep learning. The configurations of our CNN model are listed as follow (Table 1):

Table 1. The configurations of our CNN model

Layer	Name	Description
Input	–	A matrix with 32 × 54 size
Layer1	Convolution	16 kernels with 3 × 13 size
Layer2	Convolution	512 kernels with 3 × 11 size
Layer3	Pooling	2 × 2 max pooling
Layer4	Convolution	1024 kernels with 3 × 5 size
Layer5	Pooling	2 × 2 max pooling
Layer6	Fully-connected	
Output	–	A vector with 1728 elements

3.3 Evaluation Metrics

The accuracy of traffic speed prediction is mainly assessed by two performance metrics which are Mean Relative error (MRE) and Mean Squared Error (MSE). MSE evaluate the model's absolute error while MRE shows the relative error of the model. MSE is demonstrated before and MRE is presented as the following:

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \tag{8}$$

where y denotes model's predicted value using the data of the previous day as the model's input, $\hat{y}$ denotes the observed traffic speed value of the next day and n denotes the number of samples.

3.4 Experiment Result

As shown in Figs. 3, 4 and 5, we visualize some kernels in different convolution layers, different feature maps and the output matrix during the experiment. In Fig. 3, The three images in the first row are kernels we choose from the first convolution layers, then the images in the next row are kernels selected from the second convolution layers, the last images are kernels chosen from the third convolution layers. In Fig. 4, The images in

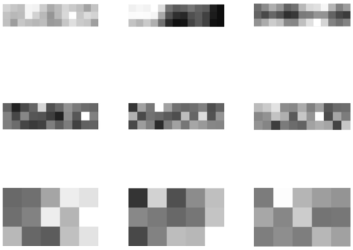

Fig. 3. Kernels' visualization from different convolution layers. The images in the first line are asymmetrical kernels of the first convolution layer and so forth.

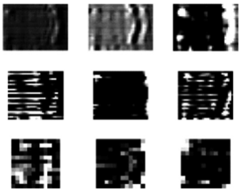

Fig. 4. Feature maps extracted from different convolution layers. The images in the first line are feature maps of the first convolution layer and so forth.

the first row are feature maps extracted from the first convolution layers, then second and third. The left image in Fig. 5 is the model's reshaped output which is used as prediction and the right is the visualized speed data in reality. In Fig. 6 the real data of the next day and the prediction of our model are represented as polylines.

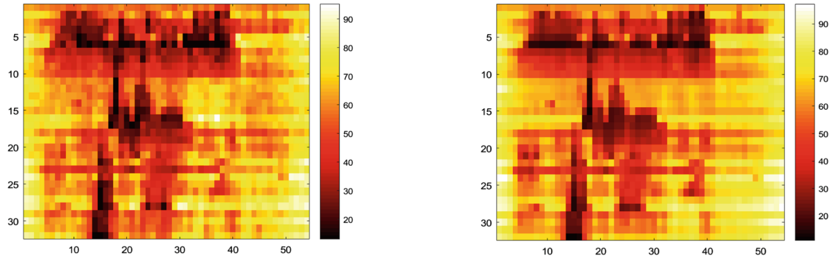

Fig. 5. The left image is the output of the model, which is transformed to heat map and the right image is the visualized real traffic speed of the next day. We can see that the output of our model is very similar to the real data.

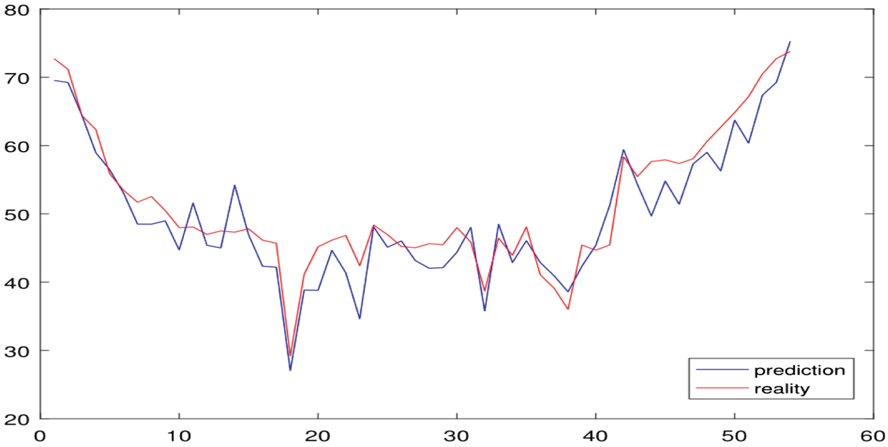

Fig. 6. The blue polyline is the model's prediction, and the red polyline represents the reality of the next day. The blue polyline can reflect the trend of the reality, and fit the reality well. (Color figure online)

We compare our CNN model with the most widely used methods of traffic flow prediction, such as ARIMA, KNN, ANN and common CNN. The performance of these models mentioned before is listed below (Table 2):

Table 2. The performance of different model mentioned before.

Model	MSE	MRE
Our Model	58.7607	0.0998
Common CNN	94.0019	0.1346
KNN	169.1917	0.2105
ANN	126.9821	0.1592
ARIMA	331.6745	0.2562

As is shown in the list, neural network based models get lower MSE and MRE than KNN and ARIMA who are not based on neural networks. In addition, our model achieves the lowest MSE and MRE. The MRE of our model is less than the rest models' over 3%. And for MSE, the MRE of our model is less than the second-best model about 30%.

4 Conclusion

In this paper, we proposed an CNN based deep learning model to predict whole-day traffic speed of elevated highway. In our model, we use asymmetrical kernels in the convolution layer. Our model focuses more on temporal dynamics which solve the problem that common methods cannot treat the special features and the temporal features differently. The experimental result proved that our model can achieve a good performance when comparing with other conventional method.

Acknowledgments. This work has been supported by the Fundamental Research Funds for the Central Universities of China and by National Natural Science Foundation of China under grant 61472284.

References

1. Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X., Chen, C.: Data-driven intelligent transportation systems: a survey. IEEE Trans. Intell. Transp. Syst. **12**(4), 1624–1639 (2011)
2. Park, J., Li, D., Murphey, Y.L., Kristinsson, J., McGee, R., Kuang, M., Phillips, T.: Real time vehicle speed prediction using a neural network traffic model, pp. 2991–2996. IEEE (2011)
3. Karlaftis, M.G., Vlahogianni, E.I.: Statistical methods versus neural networks in transportation research: differences, similarities and some insights. Transp. Res. Part C Emerg. Technol. **19**(3), 387–399 (2011)
4. Tran, Q.T., Ma, Z., Li, H., Hao, L., Trinh, Q.K.: A multiplicative seasonal ARIMA/GARCH model in EVN traffic prediction. Int. J. Commun. Netw. Syst. Sci. **08**(4), 43–49 (2015)
5. Voort, M.V.D., Dougherty, M., Watson, S.: Combining kohonen maps with ARIMA time series models to forecast traffic flow. Transp. Res. Part C Emerg. Technol. **4**(5), 307–318 (1996)
6. Williams, B.: Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transp. Res. Rec. J. Transp. Res. Board **1776**(1), 194–200 (2001)

7. Davis, G.A., Nihan, N.L.: Nonparametric regression and short-term freeway traffic forecasting. J. Transp. Eng. **117**(2), 178–188 (1991)
8. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. **5**(4), 276–281 (2004)
9. Castro-Neto, M., Jeong, Y.S., Jeong, M.K., Han, L.D.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst. Appl. **36**(3), 6164–6173 (2009)
10. Hong, W.C.: Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. Neurocomputing **74**(12–13), 2096–2107 (2011)
11. Asif, M.T., Dauwels, J., Goh, C.Y., Oran, A., Fathi, E., Xu, M., Dhanya, M.M., Mitrovic, N., Jaillet, P.: Spatiotemporal patterns in large-scale traffic speed prediction. IEEE Trans. Intell. Transp. Syst. **15**(2), 794–804 (2014)
12. Huang, S.H., Ran, B.: An application of neural network on traffic speed prediction under adverse weather condition. In: Transportation Research Board Annual Meeting (2003)
13. Zheng, W., Lee, D.H., Zheng, W., Lee, D.H.: Short-term freeway traffic flow prediction: Bayesian combined neural network approach. J. Transp. Eng. **132**(2), 114–121 (2006)
14. Moretti, F., Pizzuti, S., Panzieri, S., Annunziato, M.: Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling. Neurocomputing **167**(C), 3–7 (2015)
15. Huang, W., Song, G., Hong, H., et al.: Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. **15**(5), 2191–2201 (2014)
16. Soua, R., Koesdwiady, A., Karray, F.: Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 3195–3202. IEEE (2016)
17. Ma, X., Yu, H., Wang, Y., Wang, Y.: Large-scale transportation network congestion evolution prediction using deep learning theory. PLoS ONE **10**(3), e0119044 (2015)
18. Lv, Y., Duan, Y., Kang, W., et al.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2015)
19. Duan, Y., Lv, Y., Kang, W., Zhao, Y.: A deep learning based approach for traffic data imputation, pp. 912–917 (2014)
20. Tian Y., Pan L.: Predicting short-term traffic flow by long short-term memory recurrent neural network. In: IEEE International Conference on Smart City, pp. 153–158. IEEE (2015)
21. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: AAAI, pp. 1655–1661
22. Ma, X., Dai, Z., He, Z., et al.: Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors **17**(4), 818 (2017)