Advances in Computer Science Research, volume 87
3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019)

Urban Traffic Flow Fore-casting based on Deep Learning Model

Shengdong Mou [a], Zhengxian Xiong [b]
Green development and collaborative innovation center in Wuling Mountain area of Yangtze Normal University Fuling, Chongqing 408100, China.
[a]mushengd@163.com, [b]121806967@qq.com

Abstract. Short-term traffic flow forecasting is very important for realizing urban intelligent traffic system. In the past, many neural network models have been proposed to predict traffic flow, but the effect was not very significant. The reason is that most of them are based on shallow model learning and they are prone to fall into local extreme values and cannot simulate more complex mathematical operations. Therefore, they are not suitable for simulating realistic traffic conditions. As a new subject of machine learning, deep learning has achieved remarkable results in speech and image processing. It can unsupervised abstract effective features from the data for prediction, so in this study, deep learning modeling is used for urban trunk road traffic flow forecasting. The experimental results of this study show that the model is effective in traffic flow forecasting.

Keywords: Deep learning; Traffic flow forecasting; Neural network; Machine learning.

1. Introduction

With the rapid development of society and the rapid increase of the number of motor vehicles, urban traffic is increasingly prone to congestion, and traffic accidents and air pollution are further aggravated. In China, the average vehicle speed of big cities like Beijing and Shanghai in the downtown area is less than 20 km/h during the peak period. Traffic congestion further leads to the aggravation of energy consumption and environmental pollution. The research results show that [1] when the vehicle speed is reduced from 40 km/h to 10 km/h, the energy loss will be doubled and the environmental pollution will be increased by more than three times. The nitrogen oxides and carbon monoxide emitted by cars in Beijing account for 46% and 63% respectively. Therefore, Intelligent Transportation Systems (ITS) has become a key way to solve the urban congestion problem. ITS makes effective use of the current advanced hardware and software to manage the traffic effectively. As the core technology of intelligent transportation system, short-term traffic state fore-casting makes use of existing data to predict traffic flow conditions, and then helps travelers with path planning and traffic flow guidance, thus alleviating traffic congestion and reducing environmental pollution. Different forecasting models have been proposed for traffic flow forecasting. However, the accuracy of forecast results is challenged by the complexity of urban traffic flow in time and space.

As a new subject of machine learning, deep learning has attracted extensive attention since it was proposed. Google, Microsoft and Baidu are all studying and utilizing deep learning. It has been successfully applied to classification task, natural language processing, dimension reduction, image recognition and so on. Deep learning effectively and unsupervised extracts the underlying typical characteristics of the underlying data by using a multi-tier architecture, and then provides it to the high-level for classification and regression. Traffic flow itself is a complex process, and deep learning architecture can help us effectively learn and grasp the inherent complex characteristics without prior knowledge, so as to realize effective forecasting of traffic flow.

This paper presents a traffic flow forecasting model based on deep learning. The experimental results show that this method is accurate in traffic flow forecasting.

2. Background

2.1 Traffic Flow Forecasting

Traffic flow fore-casting has always been regarded as the key technology of urban intelligent transportation system (ITS), which uses historical data to forecast future traffic flows in a certain period of time, with the forecasting period being generally set to 5-30 minutes.

Suppose X_i^t represents the traffic flow of the i-th traffic road at time t, then, according to given an observed sequence of traffic flow i = 1, 2,…, m ; t = 1, 2, …, T, the traffic flow forecasting is to forecast the traffic flow of a certain road within the period of {T + Δt} based on the previous sequence. Including Δt can be adjusted.

Traffic flow forecast model generally includes two steps, namely, feature learning and model learning. Feature learning is unsupervised learning. Through training, a feature representative model h, which represents the historical traffic time series in the past, can be obtained. After feature training, the previous traffic flow sequence X can be transformed into another feature space Y through h, that is, h(x) → Y. Model learning is supervised learning. Given a pair of feature Y and target task Z {(Y_1, Z_1), (Y_2, Z_2), …, (Yn, Zn)}, and learning prediction model Zn + 1 = g (Y), by minimizing the objective loss function L, the appropriate parameters of the prediction model can be obtained:

$$L(Z;W) = \|Z - g(Y)\|^2 \tag{1}$$

Although the forecasting models differ from each other, the objective loss function is mostly the same. Previous forecasting models can be divided into three categories [2]:

Time series method based on historical data. Autoregressive integrated sliding average model [3] (ARIMA) achieves forecasting by finding the pattern of traffic flow changing with time. Similarly, there are subset ARIMA [4], expression variable ARIMA (ARIMAX) [5], vector autoregressive moving average (ARMA) and ARIMA [6] based on time and space, as well as seasonal ARIMA (SARIMA) [7] and Kalman filtering method [8].

Method based on probability graph model. The method of probability graph is used to model and forecast the traffic flow. Common methods include: bayesian network [9], markov chain, markov random field (MRFs), fuzzy logic [10], etc.

Nonparametric statistical methods, such as neural network (NNs), support vector regression (SVR), local weighted learning (LWL), k-NN [11] method, support vector machine (SVR) [12], stochastic differential equation [13], etc. This kind of methods is more effective than other methods because it can simulate the characteristics of traffic flow, such as uncertainty, complexity and nonlinearity.

In general, with the development of intelligent transportation, many forecasting models have been proposed. However, it is difficult to have one particular approach that is better than the another one in all areas of traffic flow. Because most of these methods are based on some specific data modeling, and the forecasting results also depend on the accuracy of the collected data. However, neural network (NN) can obtain more robust forecast results as it can effectively utilize the relationship between historical data and current data. Most of the existing neural networks are based on the shallow architecture. When building a multi-layer architecture, the method based on gradient descent cannot effectively adjust the parameters. As a new neural network, deep learning solves the problem of traditional NN training and achieves better experimental results than traditional neural network. Therefore, deep belief network (DBN) can be used to improve traffic flow model.

2.2 Deep Belief Network

As the most common deep learning model, deep belief network (DBN) is built by a series of RBMs. Each layer of RBM has only one hidden layer, and the output of each layer serves as the input of the next layer. Hinton et al. proposed a method that can quickly train DBN layer by layer each time [14], namely, one layer each time.

RBM, i.e. restricted Boltzmann machine, is a special case of markov random fields (MRFs). If a bipartite graph has no connection between nodes in each layer, and the first layer is visible and the other layer is hidden, and all nodes are assumed to be random and binary-distributed, the second layer is connected by a symmetric matrix, and the probability distribution satisfies the Boltzmann distribution, then this is RBM. The visual layers correspond to inputs because their states have been observed. The hidden layers correspond to the feature detection, and their joint configuration energy equation (v, h) is:

$$E(v,h) = -\sum_{i\in v}^{n} b_i v_i - \sum_{i,j}^{n} v_i h_i w_{ij} \tag{2}$$

Where v_i and h_i are input i and feature j, respectively; b_i and a_i are offsets corresponding to v_i and h_i ,respectively; w_{ij} is the weight matrix between them. Since the hidden layers are mutually independent, that is:

$$P(h|v) = \prod_j p(h_j|v) \tag{3}$$

$$P(h_j = 1|v) = \frac{1}{1 + \exp(-\sum_{i=1}^{n} w_{ij} v_i - a_j)}$$
$$P(v_i = 1|h) = \frac{1}{1 + \exp(-\sum_{j=1}^{n} w_{ij} h_i - b_i)} \tag{4}$$

Then, given a set of training sets $(V^c | c \in (1,2,\ldots.. c))$, the objective is to maximize the logarithmic likelihood function of this model:

$$\sum_{c=1}^{c} \log p(V^c) = \sum_{c=1}^{c} \log \frac{\sum_g e^{-E(V^C,g)}}{\sum_u \sum_g e^{-E(U,g)}} \tag{5}$$

Generally, the parameters wij, bi and aj are obtained by gradient descent method, but it can be approximated by gibbs sampling method, i.e., the hidden layer H can be sampled according to the specified rules for the visible layer V, and then the visible layer V can be sampled in turn. This process can be repeated many times. After several iterations, the model forgets its initial starting point so that it can be sampled from their balanced distribution. Finally, the function is expected to obtain an approximate value within a finite number of times by using the contrast bifurcation (CD) method. The N + 1 sampling algorithm is labeled as CD-N. In practice, you can usually get the appropriate value with CD-1. Then the update rule of weight wij can be obtained:

$$\Delta W_{ij} = \varepsilon_{\varpi} (E_{data} [v_i h_j] - E_{\text{mod}} [v_i h_j]) \tag{6}$$

Where ε_{ϖ} represents learning rate; Edata is the desired output of the layer when the visual layer inputs according to the initial model distribution; Emod is the expected output estimated by the CD algorithm. Similarly, the update rules of bi and aj are similar to that of wij.

2.3 GBRBM Gauss-bernoulli GBRBM

In ordinary limiting boltzmann machine (RBM), the input limit of the visual layer is 0 or 1, which is very inconvenient for simulating continuous values like traffic flow in reality. Therefore, real data can be simulated by gauss-bernoulli GBRBM[15]. It simulates real data by adding the continuous value of gaussian noise, thus replacing the binary visual input of ordinary RBM, and its energy function is changed to:

$$E(v,h) = \sum_{i=1}^{v} \frac{(v_i - b_i^v)^2}{2\varphi_i^2} - \sum_{j=1}^{H} a_j^h h_j - \sum_{i=1}^{v} \sum_{j=1}^{H} \frac{v_i}{\varphi_i} h_j w_{ij} \tag{7}$$

Where vi represents the i-th true value of the visible layer; φ is the standard deviation of the Gaussian. The priority of a specific continuous value can be obtained through the expression of the visible layer. According to the energy equation [16], the conditional probability distribution of them can be obtained as:

$$p = (v_i | h) = N(b_i + \varphi_i \sum_{j=1}^{n} h_j w_{ij}, \varphi_i^2) \tag{8}$$

$$p(h_j | v) = sigm(\sum_{i=1}^{n} v_i w_{ij} + a_j) \tag{9}$$

The training and parameter adjustment process of gauss-bernoulli GBRBM is no different from that of ordinary RBM, and the parameters of both RMB and gauss-bernoulli GBRBM can be adjusted by using the CD process.

3. System Structure

In this wor, a deep architecture is established. The underlying architecture is the DBN architecture composed of GBRBM and RBM, which is used for unsupervised feature learning. A regression layer is added to the top layer for forecasting, which can also be replaced by a support vector machine (SVM).

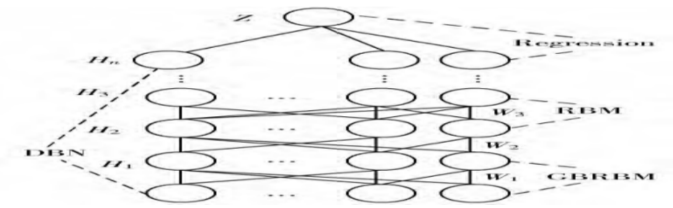

Fig.1 Depth system architecture of single road prediction

After the pre-training of DBN, the top layer carries out BP algorithm through the labeled data to adjust the parameters. This method is better than the traditional neural network which directly uses BP algorithm for gradient descent adjustment. This is because the parameters after the pre-training of DBN are already close to those after the training, so the BP algorithm only needs to conduct a local search within the known parameters, and the training speed and convergence speed are much faster. The training steps of this model are as follows:

(1)Through normalizing the traffic flow data to [0,1], then the input vector X can be expressed as:

$$X = (x_i^t | t \in T, i \in N) \tag{10}$$

(2)Using the vector X as input, the first layer of GBRBM is trained through the CD process

(3)Train the RBM by taking the output of GBRBM as the input of the upper RBM.

(4)Train RBM by uing the output of RBM as the input of upper RBM .

(5)Repeat step (4) until the given number of layers is completed

(6)Take the output of the last RBM as the input of the top-level regression layer and randomly initialize its parameters.

(7)Fine-tune the parameters of this architecture using an supervised BP approach.

Finally, the model obtained through training can be used as the forecasting model. When a set of input vectors are given, the forecasting output of the corresponding road can be obtained.

4. Experiment and Result Analysis

4.1 Experimental Data Description

The traffic flow data used in the experiment were obtained from the UK's official traffic flow data center. The data set provides the average journey time, speed and traffic flow of traffic every 15 minutes, and covers the highways and class A roads (that is, urban main roads) in England. In the experiment, five major urban roads (AL1065, AL1596, AL566, AL543 and LM69) between Newcastle and Sunderland in England were selected. The data of September 2018 (totally 30 days) were selected. Data of the first 29 days were used to train the model, and data of the last day were used to test.

4.2 Performance Index

The two most common performance indexes are mean absolute error (MAE) and mean relative error (MRE), which are defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Z_i - \hat{Z}_i\right| \tag{11}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|Z_i - \hat{Z}_i\right|}{Z_i} \tag{12}$$

Where Zi is the value of actual traffic flow; $\hat{Z}_i$ is the forecasted value. MAE and MRE were chosen as the measurement criteria.

4.3 Architecture Implementation

In a deep architecture, it is necessary to determine the size of input layer, the number of hidden layers, and the number of nodes in each hidden layer. Here, the traffic flow of AL1065, AL1596, AL566, AL543 and LM69 during the first two time periods are selected as inputs, that is, there are 10 inputs in total, and road LM69 is taken as its forecast output. Among them, the hidden layer contains the shallow structure {10,10,4} and the deep structure {10,12,10,8,6,4}. After the training, the forecasting results on the 30th day of road LM69 are shown in figure 2.

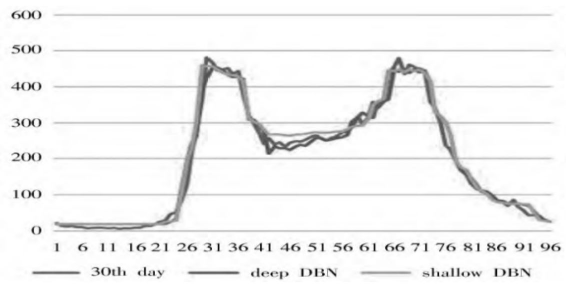

Fig.2 The results of different depth forecast for a single road

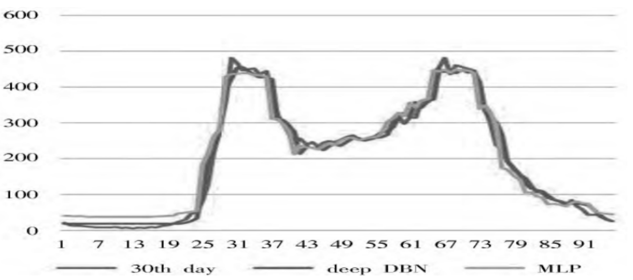

Fig.3 Results of DBN and MLP

In Fig. 2, the abscissa represents the i-th 15 min, and the ordinate represents the traffic flow. It can be seen from the figure that the result of deep structure forecast is closer to the real value.

4.4 Results Comparison and Analysis

Experimental program was developed on Windows 7, with Intel®CoreTM i7-710MQ as hardware, 4 GB of memory, graphics card NVIDIA Ge-force 840 MHz. Each operation time was generally more than 30 min, and most of them can be completed within 1 hour.

In the experiment, the adjustment test of BP process tuning times epochs of DBN was conducted, and it was found that when the number was larger than a certain number, the prediction result of DBN had no great influence on the number adjustment. This further confirms that the parameter adjustment of DBN in the pre-training stage has been almost better judgment.

In order to measure the performance of the deep architecture, the traditional MLP neural network was also adopted in BP algorithm training for forecasting, and the comparison was carried out. The results showed that the accuracy of BP algorithm decreased when the number of training layers increased, which indicates that the traditional BP algorithm is not suitable for deep architecture.In the experiment, MLP adopts {4,4,5} shallow architecture.

As can be seen from figure 3,compared with the traditional MLP neural network, the DBN architecture obtained more accurate results in both the peak period and the minimum traffic flow. The comparison of performance indexes between MLP and DBN are shown in table 1.

Table 1. Comparison of Performance Indexes

	MAE	MRE
DBN shallow	14.83	0.281
DBN deep	14.80	0.279
MLP	20.93	0.752

5. Conclusion

This paper for the first time applies the deep learning architecture to the traffic flow forecasting of urban main road. Compared with urban expressway, urban main road traffic is of greater uncertainty and variability, which brings greater challenges to forecasting. At the same time, the model also successfully discovered the potential characteristics between traffic roads through deep learning, such as the non-linear relationship in time and space. In this study, a deep architecture was firstly established, and the potential characteristics were mined by layer-by-layer unsupervised pre-training, and the global fine-tuning of parameters were realized by using the regression layer to further optimize the forecasting results. Then, the comparison between DBN and MLP showed that the accuracy of DBN was much better than MLP model.

The model proposed in this paper can still be further improved. For example, the top layer is changed to SVM and other forecast models, and the application scenarios of the model can be extended to the traffic flow forecasting of the whole city.

References

[1]. Balakrishnan A, Sundaresan S, Zhang B. Browse-and-Switch: Retail-Online Competition under Value Uncertainty [J]. Production and Operations Management, 2014, 23(7): 1129-1145.

[2]. HUANG Wenhao, SONG Guojie, XIE Kunqing. Deep architecture for traffic flow prediction: deep belief networks with multi-task learning [J]. IEEE Transactions on Intelligent Transportation Systems,2014,15(5):2191-2201.

[3]. AHMED S A, COOK A R. Analysis of freeway traffic time-series data by using Box-Jenkins techniques [J]. Transportation Research Record,1979,722:214-221.

[4]. LEE S, FAMBRO D B. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting [J]. Transportation Research Record, 1999, 1678: 179-188.

[5]. WILLIAMS B M. Multivariate vehicular traffic flow prediction-evaluation of ARIMAX modeling [J]. Transportation Research Record,2001,1776:194-200.

[6]. KAMARIANAKIS Y, PRASTACOS P. Forecasting traffic flowconditions in an urban network-comparison of multivariate and univariate approaches [J]. Transportation Research Record, 2003,1857:74-84.

[7]. WILLIAMS B M, HOEL L A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results [J]. Journal of Transportation Engineering, 2003,129(6):664672.

[8]. YANG F, YIN Z Z, LIU H, et al. Online recursive algorithm for short-term traffic prediction [J]. Transportation Research Record,2004,1879:1-5.

[9]. SUN Shiliang, ZHANG Changshui, YU Guoqiang. A Bayesiannetwork approach to traffic flow forecasting [J]. IEEE Transactions on Intelligent Transportation Systems,2006,7(1):124-132.

[10]. ZARGARI S A, SIABIL S Z, ALAVI A H, et al. A computa-tional intelligence-based approach for short-term traffic flowprediction [J]. Expert Systems,2012,29(2):124-132.

[11]. DAVIS G A, NIHAN N L. Nonparametric regression and term freeway traffic forecasting [J]. Journal of Transportation Engineering,1991,117(2):178-188.

[12]. JEONG Y S, BYON Y J, MENDONCA C N, EASA S M. Supervised weighting-online learning algorithm for short -termtraffic flow prediction [J]. IEEE Transactions on Intelligent Transportation Systems,2013,14(4):1700-1707.

[13]. TAHMASBI R, HASHEMI S M. Modeling and forecasting the urban volume using stochastic differential equations [J]. IEEE Transactions on Intelligent Transportation Systems, 2014,15 (1):250-259.

[14]. HINTON G E, OSINDERO S, THE Y W. A fast learning algorithm for deep belief nets [J]. Neural Computation,2006,18(7):1527-1554.

[15]. HINTON G E, SALAKHUTDINOV R R. Reducing the dimensionality of data with neural networks [J]. Science,2006,313(5786):50.