Traffic Flow Prediction Based on Gated Recurrent Graph Convolutional Networks: Postprint

Authors: Wang Ming, Peng Jian, Huang Feihu

Date: 2022-04-07T00:00:00+00:00

Abstract

Traffic flow prediction plays a critical role in the development of intelligent transportation systems; however, existing prediction methods fail to accurately capture the underlying spatio-temporal correlations, and most employ fully connected networks for single-step prediction. To further exploit the spatio-temporal characteristics of data and enhance the accuracy of both long-term and short-term predictions, we propose a gated recurrent graph convolutional network (GR-GCN) model. First, spectral graph convolution is integrated with Gated Recurrent Units (GRU) to construct a spatial-temporal component (STC) that simultaneously captures the spatio-temporal correlations of nodes, thereby comprehensively extracting spatio-temporal features from the data. Subsequently, this spatial-temporal component is utilized to form an encoder unit, into which time series data and road network structure data are input. Finally, GRU is employed as the decoder unit, and the two are assembled into an Encoder-Decoder structure in chronological order to sequentially decode the prediction results for each time step. Experiments were conducted on the highway datasets PeMSD4 and PeMSD8 from the California Department of Transportation (Caltrans) Performance Measurement System. The results demonstrate that the proposed GR-GCN model outperforms most existing baseline models in predicting traffic flow for the next 15, 30, 45, and 60 minutes, particularly in long-term prediction.

Full Text

Traffic Flow Prediction Based on Gated Recurrent Graph Convolutional Network

Wang Ming[1], Peng Jian[1]†, Huang Feihu[2],[1] 1. College of Computer Science, Sichuan University, Chengdu 610065, China 2. Aostar Information Technologies Co., Ltd, Chengdu 610041, China

Abstract: Traffic flow prediction plays a key role in the construction of intelligent transportation systems. However, existing prediction methods cannot accurately mine the underlying spatiotemporal correlations in data, and most adopt fully connected networks for single-step prediction. To further exploit the spatiotemporal characteristics of data and improve the accuracy of both long-term and short-term predictions, this paper proposes a Gated Recurrent Graph Convolutional Network (GR-GCN) model. First, spectral graph convolution is combined with Gated Recurrent Units (GRU) to construct a Spatial-Temporal Component (STC) that simultaneously captures the spatiotemporal correlations of nodes, thereby fully extracting spatiotemporal features from the data. Then, this spatial-temporal component is used to form encoder units, into which time series data and road network structure data are fed. Finally, GRU is employed as the decoder unit, and the two are arranged in chronological order to form an Encoder-Decoder structure that sequentially decodes the prediction results for each time step. Experiments were conducted on the highway datasets PeMSD4 and PeMSD8 from the California Department of Transportation (Caltrans) Performance Measurement System. The results demonstrate that the proposed GR-GCN model outperforms most existing benchmark models in predicting traffic flow for the next 15, 30, 45, and 60 minutes, particularly in long-term prediction.

Keywords: traffic flow prediction; graph convolutional network; gated recurrent unit; encoder-decoder architecture

0 Introduction

With the acceleration of urbanization in China, people' s demand for travel is increasing, leading to increasingly severe traffic congestion problems that constrain urban development. There is an urgent need to establish an intelligent transportation system [1] to help people plan routes rationally and alleviate traffic congestion. The core of intelligent transportation systems lies in accurately predicting future traffic flow conditions to assist system tasks.

Current traffic flow prediction models can be mainly divided into three categories. The first category comprises statistical theory-based methods, including historical average analysis, Autoregressive Integrated Moving Average [2] (ARIMA), and Kalman filter analysis [3]. The second category is machine learning-based methods, which can model the nonlinear state of traffic flow and continuously adjust parameters to optimal values through adaptive learning, thereby obtaining more accurate results and gradually replacing statistical methods. Mainstream machine learning algorithms for prediction include KNN [4] (K-Nearest Neighbor) and Support Vector Machines [5,6]. The third category is deep learning-based methods, which represent the current research focus. Deep learning can not only learn data features and correlations but also 挖掘 more complex features from simple ones without manual feature engineering. By stacking neural network structures and extracting features from shallow information, it offers significant advantages for solving complex real-world problems.

Many researchers have applied deep learning to traffic flow prediction with remarkable results. Representative models include ConvLSTM proposed by Liu et al. [7], which combines CNN and LSTM to capture spatiotemporal correlations; DCRNN proposed by Li et al. [8], which uses diffusion graph convolution for spatial feature modeling and a GRU-based encoder-decoder structure for temporal dimension modeling; T-GCN proposed by Zhao et al. [9] and STGCN proposed by Yu et al. [10], both of which use spectral graph convolution to capture spatial dependencies between nodes and employ one-dimensional dilated convolution for temporal feature modeling. These models predict traffic flow data by modeling spatiotemporal correlations. However, the above methods either fail to effectively model the spatiotemporal correlations and accurately capture spatiotemporal features, or use single-step prediction to forecast data for several future time steps, which cannot accurately reflect the temporal relationships in future traffic flow data [11].

Traffic flow, as a typical spatiotemporal data, requires consideration of both spatial dependencies in the road network and temporal dependencies in historical and future data during prediction. Considering only temporal or spatial dependencies separately, or generally modeling relationships among future prediction data, will lose important information that contributes to prediction results. To address this issue, this paper proposes the Gated Recurrent Graph Convolutional Network (GR-GCN) model, which utilizes spectral domain graph convolution to capture spatial dependencies between nodes, feeds nodes with spatial relations into gated recurrent units to simultaneously extract spatiotemporal features of traffic flow data, and then performs multi-step prediction through an encoder-decoder structure, accurately modeling dependencies among historical data and among future prediction data. Experimental results on real traffic flow prediction datasets demonstrate that our model outperforms baseline models.

1.1 Traffic Flow Prediction Task

Traffic flow prediction is a typical time series forecasting task. This study only involves traffic flow data, so only one attribute needs to be retained from the traffic state data. The traffic flow prediction problem is defined as: given historical observations where is the window size of historical sequence input, predict the traffic flow data for all nodes in future time periods , where represents the traffic flow state of the entire network during future time periods . In essence, the traffic prediction problem learns a function that predicts future traffic flow data for all nodes within time period based on historical observation data and the corresponding road network structure . The specific formula is as follows:

where and represent the historical observation data and road network structure at corresponding time steps.

1.2 Graph Convolutional Networks

Convolutional Neural Networks are feedforward neural networks based on convolution operations that can efficiently process Euclidean data such as images and speech. However, much real-world data is non-Euclidean, which cannot be directly processed using convolutional networks for feature extraction. Therefore, Graph Convolutional Networks [12,13] have been proposed as a neural network that extends convolution operations to graph structures, with the core idea being that central nodes aggregate information from their neighboring nodes. Table 1 shows the symbols involved in graph convolution operations.

Table 1 Graph-related definitions

Symbol	Definition
	Set of edges in the graph
	Set of nodes in the graph
	Adjacency matrix of graph nodes
	Degree matrix of graph nodes
	Laplacian matrix of the graph
	Features of layer in the network

The adjacency matrix reflects the adjacency relationships between nodes, while the degree matrix reflects the number of edges connected to each node. The Laplacian matrix best reflects the structural properties of a graph and is a core research object in graph theory. It describes the difference between signals of central nodes and neighbor nodes and can reflect the smoothness of node neighborhoods, specifically defined as . The GCN model principle utilizes the fact that convolution is essentially filtering in the frequency domain [14]. Node features are treated as signals, transformed to the frequency domain space through Fourier transform, and finally obtained back to the graph domain through inverse Fourier transform. Each convolution layer only processes first-order neighborhood information, and multi-order neighborhood information transfer can be achieved by stacking multiple convolution layers [15]. The propagation rule for each convolution layer is:

where represents the nonlinear activation function. In short, each layer of GCN aggregates neighbor features for each vertex by multiplying the adjacency matrix with the feature matrix, then multiplies by a parameter matrix, and finally uses an activation function to obtain a matrix aggregating neighbor node features. A two-layer graph convolutional network model can be represented as:

where represents the feature matrix, represents the adjacency matrix of the topology network, represents the adjacency matrix after Laplacian regularization, and are the weight parameters of the first and second layers respectively, and represents the activation function used by the network.

1.3 Gated Recurrent Unit

The Gated Recurrent Unit network (GRU) [16] is an improved model based on Recurrent Neural Networks (RNN). Compared with RNN, it can solve problems such as long-term memory and gradient vanishing in backpropagation. Compared with Long Short-Term Memory (LSTM) networks, GRU can achieve comparable performance while being easier to train, significantly improving training efficiency.

The hidden unit in GRU is a special cell structure rather than nodes in traditional neural networks. It contains two gates internally: a reset gate and an update gate. Intuitively, the reset gate determines how to combine new input information with previous memory, while the update gate determines how much of the previous memory is preserved to the current time step. The cell structure of GRU is shown in Figure 1 [Figure 1: see original paper].

The calculation formulas for each gate are as follows:

where the symbol represents element-wise multiplication between two vectors; represents the sigmoid activation function; represents the update gate; represents the reset gate; represents the current memory content; represents the final memory at the current time step. Due to its cyclic structure, the final memory of the previous cell and the input at the current time step together serve as the input to the current cell. By concatenating these unit structures and feeding data into them in chronological order, the model can capture potential temporal correlations in the data.

1.4 Encoder-Decoder Structure

The encoder-decoder model (Seq2Seq) was originally a successful structure used in machine translation [17]. This model accepts a sequence as input, encodes the information in the sequence into an intermediate representation, and finally uses a decoder to decode the intermediate representation into a target sequence sequentially. Since the prediction results are generally unknown, the prediction result from the previous time step often needs to be fed together with the encoded hidden representation as input to the next time step. As shown in Figure 2 [Figure 2: see original paper].

This structure is highly flexible, and its encoder and decoder can be different types of structures. In existing research, the encoder can be any combination of RNN (recurrent neural network), GRU (gated recurrent units), LSTM (long short-term memory), GCN (graph convolutional network), or spatial-temporal attention blocks, while the decoder can be any combination of RNN, GRU, LSTM, or multilayer perceptrons [18,19]. It only needs to satisfy the following mapping relationship:

where and are the lengths of the encoder and decoder respectively. These two can be unequal, so it can be applied to traffic prediction tasks for multi-step prediction. Specifically, the encoder encodes the historical sequence into a hidden intermediate representation , which is then fed into the decoder layer to decode it into future traffic states.

1.5 Gated Recurrent Graph Convolutional Network Model

To simultaneously capture the spatial and temporal correlations in traffic flow data, this paper proposes a Gated Recurrent Graph Neural Network model based on graph convolutional networks, gated fusion units, and encoder-decoder structures, as shown in Figure 3 [Figure 3: see original paper].

The left half of Figure 3 belongs to the encoder, which feeds historical sequence data into each unit of the encoder in chronological order. This paper refers to such units as Temporal Graph Convolution Units. The unit operation process is as follows: first, the graph convolutional network in the unit is used to capture the traffic topology network structure and obtain its spatial features. Then, the time series data with spatial information is fed into the gated recurrent unit, where information dynamically changes to obtain temporal features of the sequence data. When the entire historical sequence has been input and processed, the output of the last unit is the intermediate representation C. The encoder process is as follows:

where is the hidden representation associated with input , and is a randomly initialized non-zero matrix. The right half of Figure 3 belongs to the decoder, which uses GRU units to decode the intermediate representation C and obtain hidden representations. Through a feedforward neural network, predictions for future time steps are made. The specific calculation methods for the hidden states and predicted values at each time step are as follows:

where is the hidden representation associated with output , and is the start of the output sequence, belonging to an initial signal of the sequence that needs initialization. In the model, the traffic features from the previous time step are used to initialize it to facilitate decoder iteration.

For the Spatial-Temporal GCN unit (TGCN), its internal structure is shown in Figure 4 [Figure 4: see original paper]. In the figure, represents the output at time , represents the graph convolution processing, and represent the reset gate and update gate at time respectively, represents the output at time , and represents the output of after transformation through a fully connected layer. The specific calculation process is as follows:

and represent the weight coefficients and biases during training. represents the graph convolution processing, with the calculation method as in Equation (5). The decoding process is similar to the encoding process. Since the input data has already undergone spatial dependency capture through graph convolution in the encoder, it does not need graph convolution operation again. Therefore, in the decoder part, this paper only uses GRU to decode the intermediate variables, with the decoding method as in Equation (6).

Overall, the Spatial-Temporal Graph Convolutional Network model can handle the complex spatial dependencies and dynamic temporal dependencies in traffic data. On one hand, graph convolutional networks are used to capture spatial dependencies between nodes in the traffic urban network topology. On the other hand, gated recurrent units are used to capture the dynamic changes of information on traffic roads to obtain the temporal dependencies of nodes themselves. Finally, the encoder-decoder (Seq2Seq) structure is used to complete multi-step prediction of traffic flow. Through the effective combination of the above modules, the model well captures the spatial and temporal dependencies of traffic flow and completes the prediction task.

2.1 Dataset

This paper conducts experiments using two real highway datasets, PeMSD4 and PeMSD8. Traffic volume data is used as the experimental dataset, with raw data collected at 30-second intervals aggregated into 5-minute intervals, meaning each point contains 12 traffic data samples per hour.

This paper uses traffic flow data from the past hour (12 monitoring samples) to predict traffic flow for the next 15 minutes (3 samples), 30 minutes (6 samples), 45 minutes (9 samples), and 60 minutes (12 samples). The dataset is divided into training, validation, and test sets at a ratio of 7:1:2, and normalized using the mean and variance of the training set. Detailed data information can be found in Table 2 .

Table 2 Dataset description

Dataset	Sensors	Training Samples	Validation Samples	Test Samples
PeMSD4				
PeMSD8				

2.3 Experimental Hyperparameter Settings and Analysis

The performance of the model is evaluated through two commonly used metrics for regression tasks: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), with specific calculation formulas as:

This paper selects Huber loss, which combines MSE and MAE, as the loss function. By adjusting hyperparameters to determine the scope of MAE and MSE, the model can converge to the global optimal solution more quickly. The specific calculation formula is:

The main hyperparameters involved in the GR-GCN model include learning rate, number of training epochs, batch size, and number of hidden layers. This paper sets the learning rate to 0.001, batch size to 64, and training epochs to 200.

The number of hidden units in the neural network is an important parameter that significantly affects the prediction accuracy of the model. To select the most suitable value, relevant experiments were designed. On the PeMSD4 dataset, the number of hidden units was set to 8, 16, 32, 64, and 128 respectively, and the changes in prediction error were analyzed. As shown in Figure 5 [Figure 5: see original paper], the horizontal axis represents the number of hidden units, and the vertical axis represents prediction error. It can be observed that as the number of hidden units increases, the error first decreases and then increases. When the number of hidden units is 64, the model’ s prediction error is minimized. This is mainly because when hidden units gradually increase beyond a certain value, the model complexity and computational complexity increase exponentially. Therefore, throughout the experiments, the number of hidden layers was set to 64.

2.4 Model Ablation Experiments

Additionally, this paper conducted ablation experiments on the GR-GCN model. The operation method is similar to the control variable method, aiming to prove the contribution of each submodule in the model to prediction performance improvement. The graph convolution module (Our-GC) and the decoder module (Our-De) were removed for analysis respectively.

As shown in the figure, when the graph convolutional network is removed from the model, prediction errors increase significantly for both short-term (15min) and long-term (60min) prediction tasks. This again demonstrates that considering both temporal and spatial features of traffic data is more beneficial for prediction than considering only temporal features. For the case where the decoder module is removed and replaced with fully connected single-step prediction, the error gap with the proposed model is not large for short-term prediction tasks, but the gap is obvious for long-term prediction tasks. This again proves that using a decoder capable of multi-step prediction is beneficial for long-term prediction.

In summary, each module of the proposed model not only improves performance for short-term prediction tasks but also helps reduce errors in long-term prediction tasks.

3 Conclusion

This paper proposes a neural network model based on encoder-decoder structure to address the problem that existing traffic flow prediction methods fail to effectively combine spatiotemporal features and mostly use fully connected layers for single-step prediction. The model uses spatial-temporal GCN in the encoder part to extract spatial information of adjacent nodes’ traffic flow, then feeds nodes with spatial information through GRU to extract temporal information and obtain a hidden representation containing spatiotemporal features. Finally, the decoder sequentially decodes the prediction information for each time step to produce the final results. Experimental results on two real road network datasets show that the model basically outperforms other baseline algorithms in terms of MAE and RMSE metrics, and can effectively forecast traffic flow.

References

[1] Xu Hongbing, Deng Huijun. Application of artificial intelligence technology in urban intelligent transportation system [J]. Computer Products and Circulation, 2020 (01): 167.

[2] Williams B M, Hoel L A. Modeling and forecasting vehicular traffic flow as a seasonal stochastic time series process [J]. Dissertation Abstracts International, Volume: 60-01, Section: B, page: 0292; Adviser: Lester A. Hoel, 1999.

[3] Okutani I, Stephanedes Y J. Dynamic prediction of traffic volume through Kalman filtering theory [J]. Transportation Research Part B: Methodological, 1984, 18 (1): 1-11.

[4] Lint H V, Hinsbergen C V. Short-term traffic and travel time prediction models [J]. Transportation Research E-Circular, 2012.

[5] Drucker H, Burges C J C, Kaufman L, et al. Support vector regression machines [J]. Advances in Neural Information Processing Systems, 1997, 28 (7): 779-784.

[6] Liu Zhao, Du Wei, Yan Dongmei, et al. Short-term traffic flow prediction based on combination of K-nearest neighbor algorithm and support vector regression [J]. Journal of Highway and Transportation Research and Development, 2017, 34 (05): 122-128+158.

[7] Liu Yipeng, Zheng Haifeng, Feng Xinxin, et al. Short-term traffic flow prediction with Conv-LSTM [C]. 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), 2017: 1-6.

[8] Li Yaguang, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting [J]. arXiv preprint arXiv: 1707.01926, 2017.

[9] Bai Jiandong, Zhu Jiawei, Song Yujiao, et al. A3T-GCN: Attention temporal graph convolutional network for traffic forecasting [J]. International Journal of Geo-Information, 2021, 10 (7): 485.

[10] Yu Bing, Yin Haiteng, Zhu Zhanxing. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting [J]. arXiv preprint arXiv: 1709.04875, 2017.

[11] Ye Jiexia, Zhao Juanjuan, Ye Kejiang, et al. How to build a graph-based deep learning architecture in traffic domain: A survey [J], 2020.

[12] Niepert M, Ahmed M, Kutzkov K. Learning convolutional neural networks for graphs [C]. International conference on machine learning, 2016: 2014-2023.

[13] Wu Zonghan, Pan Shirui, Chen Fengwen, et al. A comprehensive survey on graph neural networks [J]. IEEE Trans on neural networks and learning systems, 2020, 32 (1): 4-24.

[14] Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs [J]. arXiv preprint arXiv: 1312.6203, 2013.

[15] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering [J]. Advances in neural information processing systems, 2016, 29: 3844-3852.

[16] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling [J]. Eprint Arxiv, 2014.

[17] Luong M T, Le Q V, Sutskever I, et al. Multi-task sequence to sequence learning [J]. Computer Science, 2015.

[18] Lu Mingming, Zhang Kunfang, Liu Haiying, et al. Graph hierarchical convolutional recurrent neural network (GHCRNN) for vehicle condition prediction [J], 2019.

[19] Chen Weiqi, Chen Ling, Xie Yu, et al. Multi-range attentive bicomponent graph convolutional network for traffic forecasting [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34 (4): 3529-3536.

[20] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [J]. arXiv preprint arXiv: 1609.02907, 2016.

[21] Zhao Ling, Song Yujiao, Zhang Chao, et al. T-GCN: A temporal graph convolutional network for traffic prediction [J]. IEEE Trans on Intelligent Transportation Systems, 2019, PP (99): 1-11.

[22] Bai Lei, Yao Lina, Li Can, et al. Adaptive graph convolutional recurrent network for traffic forecasting [J], Advances in Neural Information Processing Systems 2020, 33: 17804-17815.

Note: Figure translations are in progress. See original paper for figures.