SHORT-TERM TRAFFIC VOLUME PREDICTION TECHNIQUES: REVIEW

1Archna R. Shinde, 2Kajal P. Visrani
1Assistant Professor, 2 Assistant Professor
1Department of Computer Engineering,
2SSBT’s College of Engineering and Technology, Jalgaon, India

Abstract: The monitoring and controlling of road traffic is becoming a major problem in many countries. With the ever-increasing number of vehicles on the road, the Traffic Monitoring Authority has to find new methods of overcoming such a problem. Now a day’s many intelligent transport systems use modern technologies to predict traffic flow, to minimize accidents on road, to predict speed of a vehicle and etc. There have been various Neural Network based approaches proposed for short-term traffic state prediction that are surveyed in this paper.

Index Terms - Intelligent Transport Systems; Neural Network; Deep Learning.

I. INTRODUCTION

There has been a continuous increase in road traffic in recent years which results in congestion, accidents and pollution. To solve traffic related problems scientifically and reasonably has become a society-wide important. Building transportation infrastructures can solve the traffic congestion up to a certain level and for a limited period of time only. Initially, the important way to increase transport efficiency, reduce traffic congestion and improve traffic safety situation, is to implement traffic guidance and control, effectively use the road resource. In developing countries where resources are limited, so traffic congestion problem is becoming a major challenge for administrators of transportation system. Recently an intelligent strategies are used for increasing the transportation systems efficiency and sustainability, which is known as Intelligent Transportation Systems (ITS). In an ITS, it is important to predict traffic flow on a short-term basis, using existing information of present traffic conditions and previous traffic observations. Short-term traffic flow prediction involves forecasting traffic volumes in each next time interval, which is generally in the range of five minutes to one hour, decided by traffic authorities according to the corresponding requirements and situations. Applications for Intelligent Transportation Systems are those which improve the efficiency of surface transportation systems and solve transportation problems using modern information and communication technologies. In order to meet increasing demand for traffic, it is required to implement ITS for the efficient use of the transport infrastructure. One of the most significant requirements of these systems is to predict accurately the nature of the traffic stream.

This paper discuss the various Intelligent Transportation Systems (ITS) techniques like an Artificial Neural Network, Deep learning. The current paper study a novel approach for traffic congestion using intelligent agents. These intelligent agents will collect, store and process the road traffic data in order to predict the forthcoming traffic flow enabling road operators to be able to proactively take appropriate measures, such as changing the traffic light strategy, and individuals to be able to use alternative routes to their destination. Deep learning approach use multilayer to find hidden information in transaction.

II. RELATED WORK

There many studies in the literature that deal with the road traffic prediction problem.

Kumar, Parida, Katiyar (2013) implemented the Artificial Neural Network (ANN) for short term prediction of traffic flow using past traffic data. The model includes traffic volume, speed, density, time and day of week as input variables. Instead of average speed of combined traffic flow, speed of each category of vehicles was considered separately.

Manoranjitham, Raj, Lal (2018) reviewed the deep learning techniques: The recurrent neural network (RNN) and Long short-term memory (LSTM) in order to predict the short term traffic flow. In recurrent neural network (RNN) and long short-term memory (LSTM), the available data on traffic flow can be used to train a deep neural network to recognize patterns and give a short-term forecast for traffic flow for a particular area.

Shafqat, Malik, Byun, Kim (2019) presented short-term traffic flow prediction using Recurrent Neural Networks (RNN) for Road Transportation Control in ITS. They did some pre-processing on the input data to get it into the desired format. Data was aggregated based on some time intervals e.g., time interval of 15 minutes etc. The proposed model will take traffic flow data as an input and forecast the traffic flow rate in future i.e., number of vehicles at a given time.

In Karthika, UmaMaheswari, Venkatesh (2019), Deep learning techniques can be used with technological progress to prevent information from real time. Deep algorithms are discussed to predict real-world traffic data. Deep learning is employed because it automatically extracts features from large set of data. Deep learning approach use multilayer to find hidden information in transaction.

Liu, Li, Wu, and Li (2018) visualize the possible and board usage of deep learning in predictions of various traffic indicators, for example, traffic speed, traffic flow, and accident risk. Here instead of directly passing mobility data to classic machine learning models, the raw data are first fed into deep learning models to learn abstractions by many hidden layers. In general, firstly low-level abstractions are extracted from the input data and then fed to following layers to form higher-level abstractions. At last, such a tree of abstractions automatically selects some high-level features that are simultaneously sensitive to subtle details, e.g., different times of day, and insensitive to irrelevant variations, e.g., the types of passed vehicles on roads. Building on such features, the derived traffic models will be more informative, stable, and robust, and thus they can achieve much better prediction performance. In essence, deep learning can be viewed as an excellent feature extractor, which avoids burdensome feature engineering while automatically learning good features using a general-purpose learning procedure.
Xiao and Yin (2019), hybrid Long Short-Term Memory (LSTM) neural network is proposed, based on the LSTM model which able to provide precise prediction results and suitable for different traffic conditions in the actual traffic network. It consists of one input combination layer, zero or more intermediate combination layers and one output combination layer. The input combination layer includes a LSTM layer, an Activation layer and a Dropout layer, the intermediate combination layer includes a dense layer, an Activation layer and a Dropout layer, and the output combination layer includes a dense layer and an Activation layer. Then, the structure and parameters of the hybrid LSTM neural network are optimized for the large traffic flow set as well as the small traffic flow set. The hybrid LSTM is used to model the vehicle flow prediction of each road section and intersection in the actual traffic network.

Zhang, Yao, Hu, Zhao, Li and Hu (2019), proposed an accessible and general workflow to acquire large-scale traffic congestion data and to create traffic congestion datasets based on image analysis. With this workflow a dataset was created named Seattle Area Traffic Congestion Status (SATCS) based on traffic congestion map snapshots from a publicly available online traffic service provider Washington State Department of Transportation. They also proposed a deep autoencoder-based neural network model with symmetrical layers for the encoder and the decoder to learn temporal correlations of a transportation network and predicting traffic congestion.

Do, Taherifar, Vu (2018), in-depth discussion is provided to demonstrate how different types of Neural Networks have been used for different aspects of short-term traffic state prediction. Here further research directions are also suggested for additional applications of NN models, especially using deep architectures, to address the dynamic nature in complex transportation networks.

III. METHODOLOGY

Deep learning models have various form one of the common architecture is shown in figure 1.1. The architecture has an input layer, an output layer, and in between number of hidden layers. For input layer will uses the values of raw data and it will give the desired output in the form of output layer. All hidden layers are responsible for converting input layer states into predicted output layer inferences by capturing the high-level abstractions. Each layer in the network contains several units, and the sizes could vary from one layer to another links exist between units of any two adjacent layers, and each connection is associated with a weight. Each unit has an activation function that determines how to calculate its own state based on units from the immediately preceding layer and then exposes its condition to the next layer (Liu, Li, Wu, and Li, 2018).

Figure 1.1: General Architecture of Neural Network

Neural networks are data-driven, self-adaptive models capable of capturing the underlying relationships without requiring a priori hypothesis regarding the problem examined. They can learn from data, even if the underlying relationships are not obvious. Their non-linear nature and their ability to generalize means that they can generate predictions from a part of the data that was not used for training after being trained on sample data makes them a useful tool for working with noise databases.

There is no limit on the number of variables in ANN modeling, i.e. one can select the desired number of input or output variables depending on the problem. A number of input variables, network structure and number of hidden layers, activation mechanism and range of learning or training algorithms are important features that are important for the design of the neural network. (Kumar, Parida, Katiyar, 2013).

3.1 Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) model is basically designed to process two-dimensional data, like an images. As shown in Figure 1.2, a CNN model is consist of an input layer and an output layer, as well as multiple hidden layers, which could be the convolutional, pooling, or fully connected layers. The convolutional layers uses convolutional filters, which apply certain transformations on the input data to capture their properties. Next is pooling layers that combine the output of unit clusters of a previous layer into a single unit in the next layer by using the max or min filter. A pooling layer learns more abstract representations of the data, and acts as a form of dimensionality reduction to simplify the whole model. A fully connected layer is used to complete the inference (Liu et al, 2018).
3.2 Recurrent Neural Network (RNN)

The Recurrent Neural Network (RNN) model is mainly used for tasks that are involved with sequential inputs, for example, speech and language, due to its “memory” design in the form of a loop as shown in Figure 1.3. A loop allows information to be passed from one step to the next. RNN process an input sequence one element at a time, maintaining output results in the hidden units that implicitly persist information about the history of all past elements. When unfolding the loop, an RNN can be viewed as a stack of separate neural networks with some parameters of each network fed from the previous one. Such parameters act as the memory of RNN models. Inside the repeating neural networks of an RNN, the input element \( P_t \) at time step \( t \) is concatenated with the output \( Q_{t-1} \) of previous time step and then are together fed into an activation function (e.g., tanh) to derive output \( Q_t \) of the current time step. Such an architecture allows RNNs to capture temporal dynamics, but RNNs cannot support long-term dependency. Thus, an improved RNN called a Long Short Term Memory network (LSTM) is proposed, which uses special hidden units (i.e., memory cells) to remember inputs for a long time. LSTM models are able to learn long sequences and automatically determine the optimal time lags for prediction (Liu et al, 2018).

3.3 Long Short-Term Memory Neural Network

Vanishing Gradient problem arises while training an Artificial Neural Network. This mainly occurs when the network parameters and hyper parameters are not properly set. LSTMs: Long Short Term Memory Networks are generally used to tackle Vanishing Gradient problem when you are working on RNN (Hochreiter & Schmidhuber, 1997). LSTMs are explicitly designed to avoid the long-term dependency problem. In this model gating mechanism was used to decide when and how to update its memory. Cell, an input gate, a forget gate and an output gate are mostly presented in the LSTM unit or block. The cell represents the memory, the input gate decides what new information should be stored in the memory, the forget gate specifics what information should be removed from the memory and the output gate determines how the memory is used to calculate the output of the LSTM unit (Figure 1.4a).

In their research, Ma, Tao, Wang, Yu, and Wang (2015) used a tri-layer LSTM network to model traffic dynamics. The hidden (secret) layer contains an LSTM block in this model, and the time delay was not fixed but is adaptive automatically through the training phase. This design allowed the network to capture temporal dependency in traffic data in a more flexible and effective way. A similar model was presented to the Ma, Tao et al. (2015a) model which incorporated information on rainfall as inputs (Jia et al., 2017). The model provided greater accuracy than all the baselines used, with MAPE being 5.89%. In addition to temporal correlations, Zhao et al. (2017) proposed an LSTM network (Figure 1.4 b) with several hidden layers, taking into account the spatial correlations. Origin – destination correlation matrices were proposed to derive the temporal – spatial correlations from observation position traffic states; the matrices were then used as network inputs. The proposed model outperformed conventional RNNs with the prediction horizons over 15 min in all prediction tasks, which reinforced the LSTM model’s robustness in capturing longer time-dependence. Inspired by the detrending system (Qi, Su, Zhang, Lin, & Li, 2015), a further model based on LSTM was recently proposed (Dai et al., 2017). Original flow series are decomposed into temporal (or trend) patterns and residual series in conventional (traditional) detrending-based models; the residual series are then used as inputs for the prediction models.

The given model is composed of an extraction layer and a prediction layer. The extraction layer was a fully connected layer which was used to remove temporal patterns from the original flows. The prediction layer was a layer with LSTM units which accepted both the extraction layer and the residual series features as predictive inputs. In addition, each layer was first per-trained, and then the entire network was fine tuned. The experiments showed better accuracy of the model compared with the basic LSTM model and the LSTM detrending-based model (Do, Taherifar, Vu, 2018).
3.4 Stacked Autoencoder (SAE)

An autoencoder is a three-layer neural network with an input layer, an output layer, and a hidden layer, as shown in the figure 1.5. In Stacked autoencoder (Liu, Li, Wu, and Mo Li, 2018), the target output is intentionally set as the input of the model, and thus the hidden layer aims to learn the representations of the input data, which can be viewed as a dimensionality reduction or encoding of input data and due to this function, the hidden layer of an autoencoder is also called the feature layer. The Stacked autoencoder model connects these layers of features in a stacked fashion to create abstractions of input data at a higher level, creating a deep architecture. One of the most popular variants of autoencoders is a denoising autoencoder that intentionally takes noisy or corrupted samples as the inputs while forcing the original uncorrupted data to be recovered. When stacking several denoising autoencoders, a SAE variant called a stacked denoising autoencoder (SdAE) is used. SdAE is able to discover relatively stable features compared to the SAE model, which makes it resilient against noisy inputs and thus far better performance.

IV. CONCLUSION

In this paper we studied different techniques for traffic prediction. The paper focused on some deep learning techniques which is based on neural networks such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory Networks (LSTM) and Stacked Autoencoder. Deep learning have advance traffic predictions through powerful representation learning and has shown initial success. A traditional Recurrent Neural Network (RNN) was having Vanishing Gradient problem, which was overcome with the help of Long Short Term Memory Networks (LSTM).
REFERENCES


