Deep Neural Networks for Traffic Flow Prediction

Hongsuk Yi, HeeJin Jung
Supercomputing Center
Korea Institute of Science and Technology Information
Daejeon, Republic of Korea
hsyi@kisti.re.kr, bigbear@kisti.re.kr

Sanghoon Bae
Department of Spatial Information Engineering
Pukyung National University
Busan, Republic of Korea
sbae@pknu.ac.kr

***Abstract*—Traffic flow prediction is an essential function of t raffic information systems. Conventional approaches, using artif icial neural networks with narrow network architecture and poo r training samples for supervised learning, have been only parti ally successful. In this paper, a deep-learning neural-network ba sed on TensorFlow™ is suggested for the prediction traffic flow conditions, using real-time traffic data. Until now, no research h as applied the TensorFlow™ deep learning neural network mod el to the estimation of traffic conditions. The suggested supervise d model is trained by a deep learning algorithm, which uses real traffic data aggregated every five minutes. Results demonstrate that the model's accuracy rate is around 99%.**

Keywords—traffic prediction; deep learning neural networks; transportation big data

I. Introduction

Deep-learning is increasingly being recognized as an essential tool for artificial intelligence research, with applications in several areas. These include speech recognition and image recognition [1], however, researchers have applied deep-learning algorithms to find solutions to problems in various other fields [2]. Deep-learning algorithms can be roughly classified into four types: Deep Neural Network (DNN), Convolution Neural Network (CNN), Recurrent Neural Network (RNN) and Q-learning. These deep-learning types are rapidly evolving, with several software packages including Theano, CuDNN, Caffee, and Keras now open to the public [3].

Of particular interest however, is the TensorFlow™ dataflow-based deep-learning software package unveiled by Google Inc. in 2015 [4]. TensorFlow™ implements a version that deploys RNN, DNN and CNN not only to multi-core CPUs, but also to acceleration devices like GPUs [5]. It also supports the AdaGrad, Dropout, and ReLu functions, which are all very important elements and options for successful deep-learning. Despite this, most applied research has deployed the TensorFlow™ software to limited fields such as image recognition and speech recognition only.

In this paper, the TensorFlow™ DNN architecture is developed in order to analyze and estimate traffic conditions, using real-time transportation big data. The suggested DNN model aims to distinguish congested conditions from non-congested conditions through logistics regression analysis. Traffic Performance Index (TPI) is used for logistic regression analyses, and a hyper tangent function is used as an activation function for each layer. The suggested model is effective in identifying real traffic flow conditions, with an estimated 99% accuracy.

II. Transportation Big Data Usage

The traffic data were obtained from about 0.5 million probe vehicles with an on-board device (OBD). The OBD is a car navigation system which includes a real-time communication system; each OBD recognizes the location of the probe vehicle using real-time Global Positioning System (GPS) technology, and computes an average link travel speed when the probe vehicle completes a specified link between intersections. Using the real-time communication system, the individual OBD device reports the average link travel speed with identification and a turn code to a central server. The central server then aggregates the average link travel speeds, representing traffic flow conditions of each link at a specific time period, and infers the traffic flow conditions of 3,290,392 links every five minutes. The traffic data used in this paper are those aggregated by the central server from June 14th 2016 to July 10th 2016 (23 days excluding holidays).

The aggregated traffic data is defined as the link-based average speed in kilometers per hour (KPH), and divided into three turn movements including left-turn, right-turn and straight on for each link. The central server produces 6,260,603 traffic condition datasets every five minutes, 288 times per day. That is, the traffic flow condition data consists of 180,305,364 values (6,260,603-by-288 matrix) for each day.

The traffic data represent similar traffic conditions at the same day of the week and at the same time, and were already verified by conventional traffic theories and practical experience. Traffic information systems usually categorize historical traffic data corresponding to the day of the week, and then use them for traffic condition forecasting. The aggregated traffic data by day of week are defined as traffic condition pattern data; in this paper, the data for a Friday are built using 4 days' link-based average speed data in KPH. Note that the traffic condition pattern data for a Friday is a 6,260,603-by-288 matrix, with only integer values.

III. TEST DATA FOR TRAINING A ARTIFICIAL NEURAL NETWORK MODEL

The traffic performance index as an indicator of traffic flow conditions, is as follows

$$TPI=(V_{max} - V_i)/V_{max},$$

where V_{max} is the maximum speed of traffic data, and V_i is the average link travel speed at an i-th time period.

The TPI is a measure of magnitude of congestion, and gives a value between 0 and 1 inclusive, where 1 is a traffic jam state and 0 is a free flow state. A traffic jam state means that no vehicles can move, whilst a free flow state means that all vehicles travel at maximum speed with no influence from other vehicles.

Due to limitations to memory capacity and excessive CPU calculation times, it was not possible for the suggested model to be trained by TensorFlow™ using all traffic data. Instead, around 1 percent of traffic data were randomly selected for model optimization; in future however, the entire traffic data will be applied to the optimized model. The test data include link-based average speeds of 1,000 links per day; figure 1 shows variations in average TPI of the test data in time order.

In the field of transportation engineering, traffic conditions can be generally classified into two phases (congested and non-congested), three phases (congested, moderated and non-congested), or five phases (heavily congested, congested, lightly congested, non-congested, and free-flow). To precisely recognize traffic conditions according to their TPI value, in this paper the traffic data are labeled based on two phases only.

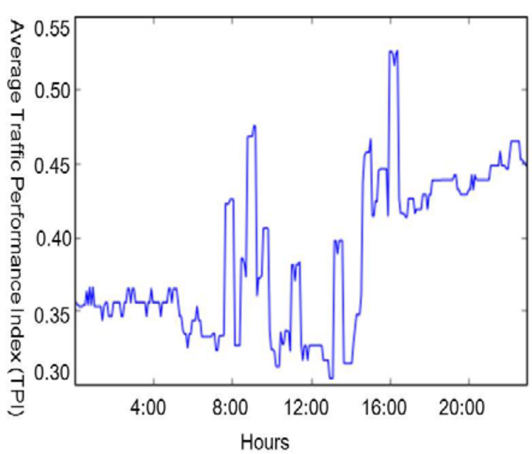

Fig. 1. Average TPI of 1,000 links randomly selected as the test data for model optimization, in time sequence. Note that the average TPI is aggregated using real-time traffic data every 5 minutes.

Figure 2 shows the average TPI of each link for a day. The dash line represents a critical value (0.5 TPI) to separate the traffic flow condition of each link into congested and non-congested. The target value for the supervised learning algorithm is a binary value; 0 is a non-congested condition where the average TPI is smaller than the criterion, and 1 is a congested condition where the average TPI is equal to or greater than the criterion. To allow an artificial neural network model to be successfully trained by a supervised learning algorithm, it is very important to appropriately label each traffic condition link.

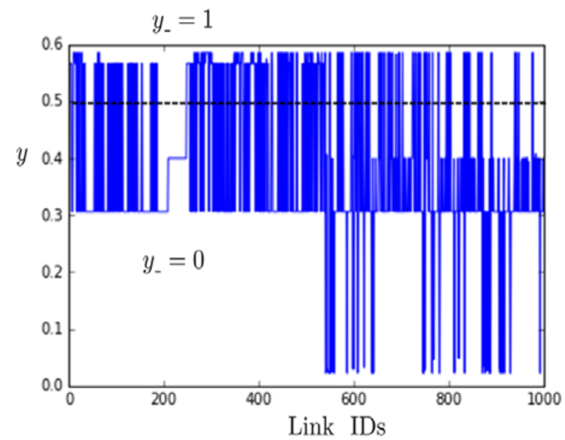

Fig. 2. Outputs of 1,000 links for supervised learning.

IV. DEEP NEURAL NETWORK ARCHITECTURE AND ACCURACY

The newest TensorFlow™ libraries were used to build the suggested model, and Pandas was used to analyze the traffic data from the transportation big data. Pandas is a very useful analysis tool for data in table format; the traffic data is imported into Pandas and then treated using Pandas DataFrames[6]. The suggested TensorFlow™ DNN architecture was constructed using the Scikit-learn API [7], and consists of 3 layers with 40, 50, and 40 neurons respectively.

Sigmoid functions are widely used as an Activation function in artificial neural network models; however, in this work, hyper tangent function (tanh) was used. It is widely known that the hyper tangent and ReLu functions are generally better than sigmoid for deep-learning models. ReLu is a linear function that an equivalent performance of DNN model can be expected to tanh . In this paper, the hyper tangent function is deployed to the suggested model, with the following code building the TensorFlow™ DNN architecture:

```
skflow.TensorFlowEstimator()

layers = skflow.ops.dnn(X,[40,50,40],tf.tanh)
```

The input data is a 288-by-1000 matrix of average link travel speeds, and the output data is a 288-by-1000 binary matrix of congestion labels. It is widely known that logistic regression models are more useful than linear regression models for categorical data; in this paper, a logistic regression model analyzes the binary value output data. The following code sets the logistic regression model to analyze the output data (layers.y):

```
skflow.models.logistic_regression(layers.y)
```

The optimization algorithm for the suggested model is the AdaGrad function; a Gradient Descent optimization algorithm which aims to minimize cross-entropy cost. The learning rate of the optimization algorithm is 0.05, and the maximum number of epochs is 2,000. In this paper, a min-batch technique which separates input data into 10 parts is deployed for model training. The following code shows the TensorFlow™ model optimization options:

```
Optimizer = Adagrad,

epoch = 2000,

Learning_rate = 0.05,

batch_size = 10.
```

To optimize weights and bias values of the model, 80 percent of the input data are used for training, with the remainder used to test the optimized model. In the model test, the optimized model estimates the TPI of each link, and the model accuracy measures the estimates against 20 percent of the input data. Results show that the accuracy of the suggested model is almost 99%.

The suggested model is evaluated on a laptop computer running Windows, with Virtualbox running Ubuntu (Version 14.04) installed on the same computer. TensorFlow™ (Version 0.9.0 CPU), Python (Version 2.7) and TFlearn (Version 0.2.1) were all installed on the virtual machine. Because TensorFlow™ in a virtualized environment is limited by memory capacity, to maintain optimum performance it was necessary to limit the number of links to 1,000.

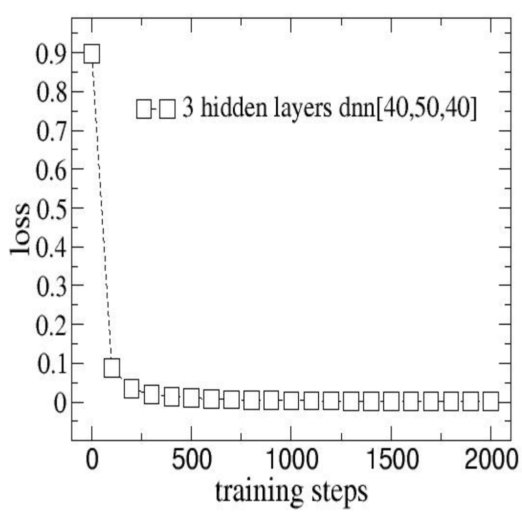

Fig. 3. Process of minimizing cost function of logistic regression analysis.

The TensorFlow™ software supplies Tensorboard visualizing data flows, used by setting a folder within the software to restore log files and then opening port 6006 on the local host. Figure 4 is a captured image of a Tensorboard data flow diagram. Tensorboard can visualize the input/output flows of data (tensor) in a hidden layer, and can also visualize variations in costs, weights and bias values during the optimization process.

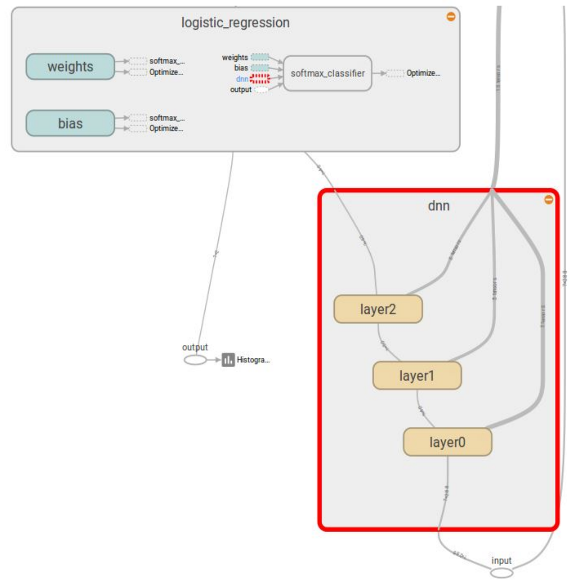

Fig. 4. DNN data flow and Tensorflow.

V. Conclusions and Future Works

In this paper, transportation big data collected from 0.5 million probe vehicles with OBD (car-navigation), was analyzed by Pandas. A DNN model with supervised learning was used to estimate link-based traffic flow conditions using real traffic data. The DNN model was built using TensorFlow™ from Google Inc., and coded using TFLearn. It should be noted that it is very important to label the input data when using supervised deep learning DNN. In this paper, TPI was used to distinguish congested traffic conditions from non-congested traffic conditions, and results show that the suggested 3 layer model could estimate congestion with 99% accuracy.

This research shows the potential for TensorFlow™ deep learning models for the accurate analysis of real-time traffic data, and precise estimation of traffic flow conditions. There are, however, several limitations in this research; due to memory capacity limitations for example, only 1 percent of traffic data for a given day could be used. In order to increase accuracy of estimation and improve the DNN architecture, it is now necessary to redefine the TPI according to transportation engineering knowledge. This research is the first work to analyze transportation big data using TensorFlow™, and it proves the considerable potential for future development of the technology. The research is a significant and promising approach to the application of TensorFlow™ to big data in the field of transportation engineering. If a high performance platform with multi GPUs could be built in future TensorFlow™ research, some of the limitations described above may be overcome.

REFERENCES

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning (Adaptive Computation and Machine Learning series), The MIT Press 2016.

[2] Deep Learning Tutorials, http://deeplearning.net/tutorial.

[3] Deep Learning Software, http://deeplearning.net/software_links.

[4] Tensorflow, https://www.tensorflow.org.

[5] Martın Abadi, et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, arXiv:1603.04467v2, 2015.

[6] Pandas, Python Data Analysis Library, http://pandas.pydata.org.

[7] TFLearn http://tflearn.org.

[8] Scikit-learn, http://scikit-learn.org/stable.