IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Neural Networks Used in the Prediction of Traffic Flow

Nishanth Vaidya, Nikhil Bharadwaj

Student, Student

Department of CSE,

Sambhram Institute of Technology, Bangalore, India

Abstract- Neural network approaches for traffic flow forecasting are usually even currently based on STL models, which do not take the data from the relative tasks. But instead of using STL and MTL, using Neural Networks can improve on generalization by transferring information in training signals of extra tasks. In our Deep Learning and Neural Networks are used in the prediction of traffic flow. Using DL and NN deep architecture to predict the short-term traffic flow in an urban traffic network have been used and are effective. Comprehensive experiments will be done to show that this method is effective in forecasting traffic.

Keywords – Deep Learning, Neural Networks.

1. INTRODUCTION

In the current generation, every individual old enough to drive will mostly own an automobile and every family may own multiple. But due to this, it is becoming more and more common for people to experience congestion in traffic, therefore it can be averted, if we can predict the traffic flows anytime, especially at peak times, people can be informed about it, allowing them to take a different route and thus saving time. Smooth and free-flowing traffic will be a boon to society. The establishment of traffic flow forecasting models based on predicting traffic flows to avoid the congestion situation is an answer to remove traffic congestion. For example, it could provide valuable traffic information to Intelligent Transportation Systems (ITS)to anticipate congestion events that may occur, as early as possible. Currently, there are varying kinds of models that are being used that are the ARIMA model [1], the time series model [2], the Kalman filtering model, and the exponential smoothing model. As it is difficult to construct and solve the mathematical model used in these methods, we use other models such as Neural Networks, SVM, etc. These methods have no need to build a complex model as they make use of real-world datasets to make predictions. In [7], a Neural Network architecture was utilized in the model, and the traffic congestion was predicted based on the Global Positioning System (GPS) data from a cab. In [8], a hybrid model for traffic flow prediction has been developed, the data was taken in the First In First Out order and classified data into clusters and made a coarse prediction, and the multilayer feed-forward neural network architecture was optimized to provide an accurate prediction. In [9] a neuro-fuzzy model was employed for traffic prediction. It used a gate network to categorize the input data based on a fuzzy approach and a Neural Network approach was used to maintain the input-output relationship.

Traffic flow datasets are found to have significant characteristics; for example, it has both time and space dimensions and is variable due to multiple conditions In [10] the various characteristics of traffic flow were considered, and the spatial and temporal correlation was introduced to improve prediction accuracy.

Based on the preliminary investigation, a certain characteristic the caused variation in the congestion of traffic in both upstream and downstream traffic conditions were found out to be traffic accidents. By making use of this characteristic and making full use of huge real-world data, this paper proposes a deep architecture for traffic flow prediction. The performance of the proposed model was compared with conventional neural network models.

2. MACHINE LEARNING METHODS AND MODEL ARCHITECTURE

Machine learning is one of the hottest topics in research and industry, with new methodologies being developed all the time. The speed and complexity of the field makes keeping up with new techniques difficult for experts and beginners alike.

A) Supervised Learning - Supervised learning is a type of machine learning task which learns a function that maps an input to an output based on example or training input-output pairs. It includes algorithms such as Support Vector Machines (SVM), k-Nearest Neighbours (k-NN) and Artificial Neural Networks (ANN), Genetic algorithms and Decision Trees (DT).

- **Naïve Bayes (NB)** - These algorithms entail probabilistic classifiers that make the prior assumption that the features of the input data are independent of each other. They are scalable and only require small to medium training datasets to produce appreciable results.
- **Support Vector Machines (SVM) - Support Vector Machines** or SVM for short, are models with associated learning algorithms which analyze the given datasets for both, classification as well as regression. Consider examples consisting of training data, with each example being marked as belonging to either of two categories. For these examples, the SVM algorithm builds a model that separates new examples to either category. Doing so makes it a non-probabilistic classifier. An SVM is a represents its examples as varying points in space, mapped so that the examples of the separate categories are clearly distinct.

B) Unsupervised Learning – Unsupervised Learning is a type of machine learning algorithm that makes decisions from datasets consisting of input data without any labelled outputs. Unsupervised Learning consists of two tasks being Association and Clustering.

- **Clustering** – Clustering consists of grouping data points that present alike characteristics. Well known approaches typically include algorithms such as k-means and hierarchical clustering. Clustering methods are scalable but the scalability is limited in a sense, but they represent a feasible solution that is used as a phase before adopting a supervised algorithm or for other detection purposes.
- **Association**. The aim of association is to identify unknown patterns across data, making them suitable for the purpose of prediction. However, they tend to produce an large output of sometimes invalid rules, hence they require a human overseer to function.

MODEL ARCHITECTURE

Since the spatial difference between normal and incident conditions can be very large, only using one neural network will not be able to capture the dynamics effectively. Therefore, the traffic data is split into two clusters and we train a neural network for each cluster. The pretraining module makes use of two functions, one to extract data through classification module and the other to initialize weights and perform fine-tuning. After this, we use the pre-training model that serves to provide deep learning via DBN to achieve the automatic extraction of abstract representations.

DBN is a stack of Boltzmann Machines that are energy-based models. Therefore, an energy function for the model is first and foremost initialized [7]. And at the same time, the real-world values with Gaussian noise are used to represent traffic data. Then, for a given set of states (v, h), the energy functions and conditional probability distributions are calculated and recorded. Previous studies have come to show that differences of early parameters produce major influence on the final choice during the training process. Therefore, the by limiting the parameters by pretraining, we can achieve a more efficient optimization. A very good feature of a DBN is that it can infer the states of the layers of hidden units only in forward pass after which the inference can be used in deriving the variational bound [9]. So, after pretraining through a stack of RBMs, we can streamline the whole probabilistic framework and simply use our own generative weights in the reverse direction as a way of initializing the weights of all except the last layer. Then, the "recognition" weights of the DBN become the weights of a standard neural network for each traffic condition respectively. In conclusion, with pre-training step completed in the first section, we add a regression layer above the DBN to forming a classifier. And the classifier is trained using the named data. Thus, we can apply the complete module to predict the traffic flow accurately.

3. LITERATURE SURVEY

- M. van der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

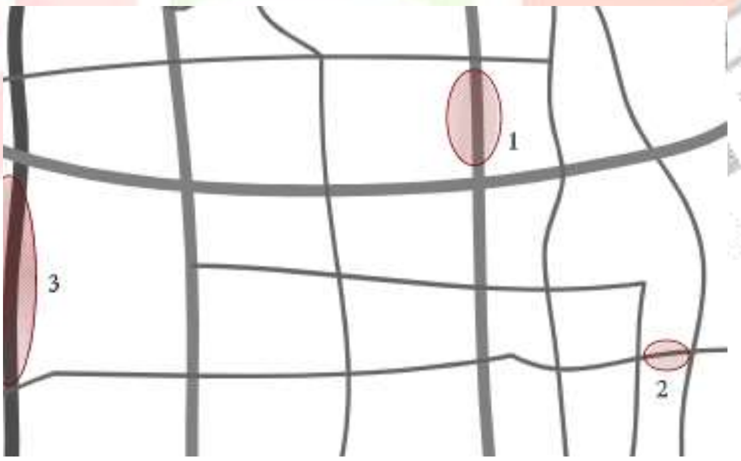
  ABSTRACT- A hybrid method of short-term traffic forecasting is introduced; the KARIMA method. The technique uses a Kohonen self-organizing map as an initial classifier; each class has an individually tuned ARIMA model associated with it. Using a Kohonen map which is hexagonal in layout eases the problem of defining the classes. The explicit separation of the tasks of classification and functional approximation greatly improves forecasting performance compared to either a single ARIMA model or a backpropagation neural network. The model is demonstrated by producing forecasts of traffic flow, at horizons of half an hour and an hour, for a French motorway. Performance is similar to that exhibited by other layered models, but the number of classes needed is much smaller (typically between two and four). Because the number of classes is small, it is concluded that the algorithm could be easily retrained in order to track long-term changes in traffic flow and should also prove to be readily transferrable.

- C. K. Moorthy and B. G. Ratcliffe, "Short term traffic forecasting using time series methods," *Transportation Planning and Technology*, vol. 12, no. 1, pp. 45–56, 1988.

  ABSTRACT - This paper explores the application of Time Series Analysis to produce short term forecasts using automatic traffic counts. Following a brief introduction to Time Series Analysis, model development and fitting is discussed in some detail. The performance of the forecasting models produced for ten sets of traffic data, collected by West Sussex Council, is investigated.

4. PROPOSED METHODS FOR STOCK MARKET PREDICTION

Data Description Part. Select three road segments randomly throughout "Named" District of any place as the research object. Below the diagram of the "Named" district is as shown, and the marked locations are the randomly selected road segments for the collection of data.



These raw data are then aggregated into minute-wise periods. Then three-time units of traffic data are taken as the input of the neural network, and the next time unit is taken as the future traffic data in order to make our predictions. Then we get 3 datasets with 1000s of groups of sample data for each road segments, the first few weeks of data as the training data, the last 10 days of data as the test data. And we repair the lost data and error data and make standardization of the data by normalizing the traffic speed into 0 or 1. *Evaluation Metrics.* We use the absolute percent error (APE), mean absolute percent error (MAPE), and root mean square error (RMSE) to check for the percentage of errors. They are defined as the representation of the predicted traffic flow and also represents the observed traffic. We then get the mean accuracy (MA) to evaluate the prediction performance of each algorithm. The MA is employed as follows:

$$APE\,(\%) = \left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100$$

$$MAPE\,(\%) = \frac{1}{n} \cdot \sum_{i=1}^{n} \left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100$$

$$RMSE = \left[\frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|)^2\right]^{1/2} \qquad MA = 1\text{-}\frac{1}{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|$$

Selection of Parameters. The network parameters of the deep architecture are chosen through numerous iterations of the data. The number of units in each layer is chosen from 40, to 2560. The layer size is set from 1 to 4 while the epoch is set from 10 to 100, with interval of 10. We chose the setting number of layers=3, number of units=160 and the epochs=40. In order to test the effect of each parameter on our deep architecture, we changed the value of a single parameter while keeping the other parameters fixed. The result of network size is reported. Taking the MA and the training time into account, 3 layers and 160 nodes in each layer were found out to be the best choice. The Neural Network algorithm was able to detect the areas with greater traffic with an accuracy of 81%.

5. CONCLUSIONS

In this paper, a deep machine learning architecture consisting of three modules as mentioned above, has been proposed to predict the short-term traffic flow. The pretraining module took a stack of RBMs at the bottom. The classification module and the fine-tuning module put regression layers at the top to classify the traffic data into two traffic states and predict the traffic flow. The performance of the proposed architecture clearly indicated the effectiveness in traffic flow prediction compared with the well-established NN models.

6. REFERENCES

[1] M. van der Voort, M. Dougherty, and S. Watson, “Combining Kohonen maps with ARIMA time series models to forecast traffic flow,” *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[2] C. K. Moorthy and B. G. Ratcliffe, “Short term traffic forecasting using time series methods,” *Transportation Planning and Technology*, vol. 12, no. 1, pp. 45–56, 1988.

[3] J. Ma, X.-D. Li, and Y. Meng, “Research of urban traffic flow forecasting based on neural network,” *Acta Electronica Sinica*, vol. 37, no. 5, pp. 1092–1094, 2009

[4] X. Ma, H. Yu, Y. Wang, and Y. Wang, “Large-scale transportation network congestion evolution prediction using deep learning theory,” *PLoS ONE*, vol. 10, no. 3, pp. 1–17, 2015.

[5] D. Srinivasan, C. Wai Chan, and P. G. Balaji, “Computational intelligence-based congestion prediction for a dynamic urban street network,” *Neurocomputing*, vol. 72, no. 10-12, pp. 2710–2716, 2009.

[6] H. B. Yin, S. C. Wong, J. Xu, and C. K. Wong, “Urban traffic flow prediction using a fuzzy-neural approach,” *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 2, pp. 85–98, 2002.

[7] A.-R. Mohamed, G. Dahl, and G. Hinton, “Deep belief networks for phone recognition,” in *Proceedings of the NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[8] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[9] G. E. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] B. Park, “Hybrid neuro-fuzzy application in short-term freeway traffic volume forecasting,” *Transportation Research Record*, no. 1802, pp. 190–196, 2002.