Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Feature Fusion Models for Deep Autoencoders: Application to Traffic Flow Prediction

Arezu Moussavi-Khalkhali & Mo Jamshidi

To cite this article: Arezu Moussavi-Khalkhali & Mo Jamshidi (2019) Feature Fusion Models for Deep Autoencoders: Application to Traffic Flow Prediction, Applied Artificial Intelligence, 33:13, 1179-1198, DOI: 10.1080/08839514.2019.1677312

To link to this article: https://doi.org/10.1080/08839514.2019.1677312

APPLIED ARTIFICIAL INTELLIGENCE
2019, VOL. 33, NO. 13, 1179–1198
https://doi.org/10.1080/08839514.2019.1677312

Taylor & Francis
Taylor & Francis Group

Feature Fusion Models for Deep Autoencoders: Application to Traffic Flow Prediction

Arezu Moussavi-Khalkhali and Mo Jamshidi

Department of Electrical and Computer Engineering, University of Texas-San Antonio, San Antonio, TX, USA

ABSTRACT

Due to reduction in dimensionality and extraction of the definitive features of input data, deep architectures have achieved significant success in various machine learning applications. Considering their successful applications in speech recognition and image classification, the main goal of this research is to investigate the performance of the sparse autoencoders utilized in regression analysis. To this end, deep sparse autoencoders with the standard method of training, *cascaded, and partially cascaded architectures*, fed with the fusion of low- and high-level features, are proposed and implemented. The regression task is to forecast the vehicular flow rate of a location on an arterial highway using different traffic variables of several locations ahead in the Twin Cities Metro area of Minneapolis. The results demonstrate that the partially cascaded model exhibits advancements in yielding more accurate results than the other two architectures fed with the features that correlate the most to the traffic flow rate.

Introduction

Many studies continue to develop methods that are capable of forecasting short-term and mid-term vehicular traffic flow as precisely as possible. The network of terrestrial roads and highways is highly correlated, and a subtle change in weather conditions or in the congestion of a single link may affect the neighboring roads drastically. Because of this stochastic property inherent in traffic data, it is very hard to predict the future traffic variables, as they do not follow a special trend. Figure 1 shows some phenomena affecting the traffic variables. Traffic flow or travel time predictions are of interest in many applications. Some motivations for precision advancements in forecasting traffic flow are:

- Improving the functionality of advanced traffic management systems (ATMS) and ATMS subsystems, such as advanced traveler information

CONTACT Arezu Moussavi-Khalkhali arezou.moussavi@gmail.com Department of Electrical and Computer Engineering, University of Texas-San Antonio, San Antonio, TX, USA

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

Figure 1. Factors affecting traffic conditions.

systems (AITS), in supplying on time and accurate decisions and responses (Ezell 2010; Noonan and Shearer 1998)

- Improving traffic conditions by using all available resources (roads, paths); for instance, en-route information provided by AITS helps the drivers take alternative routes in case of congestions (Noonan and Shearer 1998)
- Providing more accurate trip advisory systems for travelers, such as mobile phone apps

In addition to the above-mentioned cases, some other applications and systems that also benefit from travel time estimation are introduced in (Ezell 2010). Furthermore, intelligent roads and intelligent vehicles will take advantage of accurate traffic flow predictions by informing other vehicles through vehicle-to-vehicle or infrastructure-to-vehicle communications.

To this end, there is abundant research on forecasting traffic flow using model-based and unmodelled approaches, also known as parametric and non-parametric models.

Linear regression (Davis and Nihan 1991; Sun et al. 2003), variations of ARIMA models (Williams 2001; Williams and Hoel 1999, 2003; Yu and Zhang 2004), and Kalman filtering (Guo, Huang, and Williams 2014;

Ojeda, Kibangou, and De Wit 2013) are among parametric approaches that are widely used for traffic predictions. Whereas, *k*-nearest neighbor (Li, Shen, and Xiong 2012; Oswald, Scherer, and Smith 2000; Zhang et al. 2013), Fuzzy logic (Dimitriou, Tsekeris, and Stathopoulos 2008; Li, Lin, and Liu 2006; Zhang and Ye 2008), artificial neural networks (Abdulhai, Porwal, and Recker 2002; Chan et al. 2012; Dia 2001; Dougherty and Cobbett 1997; Vlahogianni, Karlaftis, and Golias 2005, 2007), support vector machine (SVM) regression, also known as support vector regression (SVR) (Su, Zhang, and Yu 2007; Wu, Ho, and Lee 2004), and hybrids (Chang, Ge, and Li 2012; Van Der Voort, Dougherty, and Watson 1996) comprise some of the well-researched methods of non-parametric models. Several methods, such as Bayesian networks and wavelets, are surveyed in (Bolshinsky and Freidman 2012; Van Hinsbergen and Sanders 2007), which declare that no technique surpasses the other approaches in prediction precision. However, due to their ability to capture and model the stochastic nature of traffic data, neural networks, being unmodelled methods, have gained more attention among researchers since the 1990s. The reason is that non-parametric mechanisms are independent of the underlying mathematical assumptions and environmental uncertainties of the traffic model. Consequently, using neural networks, which are well-known to approximate a non-linear function without being exposed to the bounding function of input-output variables, is a promising approach to tackle such complex problems (Lee 2000; Zhang and Liu 2009). On the other hand, the state-of-the-art versions of neural networks, deep architectures, have gained a lot of attention in different applications, especially those related to classification tasks. The unsupervised pre-training phase used to initialize the weights is believed to be the driving force behind the success of deep structures (Bengio 2009). Therefore, the focus of this research is on non-parametric techniques, specifically deep structures applied to traffic flow predictions.

The overarching goal of this paper is to exploit the stacked sparse autoencoders to extract the high- and low-level features from the stochastic vehicular traffic data, so as to integrate the extracted representations to achieve high accuracy predictions. In addition to dimensionality reduction, autoencoders are capable of performing non-linear Principal Components Analysis (PCA) for feature extraction. Therefore, applying autoencoders to data with high nonlinearity is superior to linear PCA. Consequently, this paper discusses the experimental results of implementing deep architectures using stochastic gradient descent and non-linear SVR, and compares the performance to that of the feedforward neural networks (FFNN) augmented by linear PCA, and FFNN without PCA. Unlike what was expected, the stacked sparse autoencoder trained with the standard method (Hinton and Salakhutdinov 2006) does not does not outperform the FFNN amplified with PCA. Therefore, two novel training architectures, the *cascaded and partially cascaded architectures* are

implemented and evaluated to improve the functionality of stacked sparse autoencoders utilized in regression analysis. It was previously shown that the cascaded model based on SGD surpass the standard method (Moussavi-Khalkhali and Jamshidi 2016); therefore, cascaded models based on SVR and further novel models, partially cascaded methods, are proposed in this study. Finally, an FFNN amplified with PCA and a non-linear SVR trained by the fusion of raw inputs and high-level features, derived from the deep regression models, are implemented and analyzed. The aforementioned non-parametric predictive methods are implemented to predict the flow rate of a location down a Trunk highway (target point), using all the available traffic variables throughout the highway ahead of the target point. Ten-minute interval traffic flows are extracted from 60 loop detectors, including the ones located throughout the highway, off-ramps, and on-ramps, to detect the traffic flow rate of a station ahead of them. The data comprised three months of traffic flow starting from 6 AM to 10 PM during weekdays from August 2013 to November 2013. The dataset is split into 60% training, 20% validation, and 20% testing set. Accordingly, the prediction horizon comprises 13 days of 10-min interval flow rate from 6 AM to 10 PM each day.

The main contributions of this study can be recapitulated as follows: (1) To the best of our knowledge this is the first work that considers feature fusion in the stacked sparse autoencoders using in traffic flow analysis. (2) The inclusion of the following properties of traffic data to predict the flow rate makes this study different from the previous works: abnormalities of traffic data; and the correlation between the traffic flow rate and other traffic variables. Unlike many studies abnormalities and special days or hours are not excluded from the dataset. Moreover, in order to predict the flow rate of a location, variables other than flow are extracted from the neighboring locations. (3) The prediction accuracy is significantly improved by enhancing the standard training of deep architecture using cascaded and partially cascaded models.

The organization of this paper is as follows. Section 2 reviews the relevant literature. Section 3 continues with a brief introduction to autoencoders and the different methods proposed to train deep regression models. Section 4 explains the data extracted for the purpose of this study. Section 5 provides the implementation details. Section 6 discusses the experimental results, and Section 7 concludes the paper.

Literature Review

Traffic flow forecasting has a key role in deploying intelligent roads, and reverberates throughout daily life. As a result, it draws attention from various disciplines of science, including statistics, civil engineering, computer science, and electrical engineering. For several decades, both parametric and non-parametric models

have been studied to a large extent. There are some surveys and research about parametric models, such as Kalman filters (Guo, Huang, and Williams 2014; Ojeda, Kibangou, and De Wit 2013), and non-parametric models, including but not limited to artificial neural networks (ANN) (Chan et al. 2012; Dia 2001; Dougherty and Cobbett 1997; Vlahogianni, Karlaftis, and Golias 2005, 2007). Among non-parametric machine learning techniques, neural networks are identified for their powerful ability to learn and model the nonlinearity of complex functions. However, due to their structure and the training method, deep architectures have presented new advancements in feature extraction and precise predictions when subject to complex data. Since the main focus of this paper is on deep regression methods, the following explains the recent studies conducted on deep architectures and traditional neural networks utilized in traffic prediction.

A very recent study carried out in (Lv et al. 2015) is used stacked sparse autoencoders to predict the traffic flow rate of freeways using the week days flow rate of all the detectors across the freeways in California. The data extracted from detectors of a single freeway were averaged to predict the flow rate of that freeway. 15-, 30-, 45-, and 60-minute traffic flow rates are predicted. Greed search was performed to select the number of hidden units. However, it was not mentioned how other parameters of the network were selected. Also, the authors mentioned that a logistic regression was used as the top layer of SAEs, whereas logistic regression is appropriate in applications that the task is to predict a nominal value. In other words, the details of the network implementation is not clear. The results shows the SAEs are superior to ANNs, random walk models, and SVMs.

Counterparts of stacked autoencoders, stacked Restricted Boltzmann Machines (RBM) – Deep Belief Networks (DBN)-, are used to predict the flow rate of a freeway in California and a highway in China (Huang et al. 2013). The authors used the one-year flow rate of several observation points to predict the flow of a target point. The feature vectors consisted of 15-min interval flow rates of the observation points. However, the number of neurons in each layer was set based on the results of performing different experiments instead of using a cross validation technique. The results show a better performance using RBMs compared to some other approaches such as SVR and neural networks. However, their deep architecture (DBN) is only effective in peak times with a large traffic flow, and cannot perform well in off-peak hours.

Traditional neural networks and their variants have been of interest to traffic flow prediction. In the study performed in (Dia 2001), time-lag recurrent neural networks are trained with historical data collected in 5 hours over 2 days. The future speed measurements of a location on a highway in Australia were predicted using the historical speed information of the same location. The prediction accuracy of their method surpasses the performance of multilayer perceptron (MLP).

A model based on time delayed neural networks (TDNN) optimum of which is selected using the genetic algorithm is trained based on the synthetic data and tested on the real dataset to predict the flow and occupancy of a freeway section in California (Abdulhai, Porwal, and Recker 2002). The training and test data spans two peak hours. The performance of the model is not compared with other methods.

In research conducted in (Vlahogianni, Karlaftis, and Golias 2005, 2007), a genetically optimized MLP and TDNN are utilized to predict the flow of a location down an urban signalized arterial road in Greece using several points ahead. The input data consisted of eighteen days of 3-min interval traffic volumes of 4 locations ahead of the point of interest. The results show the superiority of their method to ARIMA models.

Preprocessing techniques, such as feature selection, dimensionality reduction, and smoothing can highly affect the prediction results. The exponential smoothing technique is applied to the traffic flow to smooth the peaks and valleys of the data before feeding it into the ANN in (Chan et al. 2012). The data were collected in two peak hours of the day with 1-minute intervals for six weeks from a highway in Australia. This study concludes that smoothing can improve the results of applying ANN.

SVRs are also studied and utilized to some extent for traffic flow or travel time predictions. In the study performed in (Wu, Ho, and Lee 2004), the travel time of several locations in a highway in Taiwan is predicted using SVRs. The data were collected through a 5-week period without holidays. Compared with historical-mean and current-time predictor methods, the results of SVR show improvements in prediction accuracy.

Incremental support vector regression is utilized by (Su, Zhang, and Yu 2007) to predict the flow of a point on a freeway in California. In this method, the new samples were added to the training set to generate more recent predictions. To perform the study, the data were collected from the I-880 database (Skabardonis et al.) during almost 30 days and the traffic flow of the day 31 is predicted. The results show a better precision compared to the neural networks.

However, in current studies one or both of the following factors are over looked:

- Abnormalities, like accidents or holidays are excluded from the dataset.
- The correlation between the variable to be predicted and other traffic features is overlooked; i.e., the input and output traffic variables are the same (such as predicting future flow rate from historical flow rate measurements)

To address these issues, the dataset used in this study spans typical and atypical traffic situations such as different weather conditions, crashes, traffic

jams, peak and off-peak hours, special days like holidays and sporting events, and almost all of the uncertainties and stochastic nature associated with traffic data. Furthermore, the correlated features to flow rate are derived through the exhaustive search in the pool of traffic variables. Instead of the yearly data, three-month historical data are utilized to perform this research, which is an advantage in the presence of resource constraints.

Boosting the prediction accuracy of vehicular traffic flow prediction with respect to spatio-temporal properties of traffic data in highly correlated terrestrial roads and highways using deep regression models is the main research objective. Since the focus of this research is on non-parametric techniques, specifically deep structures, the performance of the proposed architectures are compared to that of the simple ANNs, ANNs augmented by PCA, and ANNs using PCA fed with the high-level features extracted from deep models.

Autoencoders: The Standard, Cascaded, and Partially Cascaded Methods for Training Deep Architectures

The first part of this section reviews sparse autoencoders briefly. The next part explores the different architectures proposed to train a stacked sparse autoencoder.

Sparse Autoencoders

From a layer-wise perspective, a single layer of an autoencoder consisted of the input layer, one hidden layer, and the output layer with the same units as the input layer (See Figure 2).

Output is identical to the input, since each autoencoder in a deep architecture aims to reconstruct its input. The reconstruction cost is the l_2^2 error function; i.e., computation of $\left(X-\tilde{X}\right)^2$ over all the training set, where X is the original input and $\tilde{X}$ is the reconstructed input. The optimal weight is found by minimizing the l2 regularized cost function w.r.t. W:

$$W_{opt} = \underset{w}{^{argmin}} \left(\| y - XW \|_2^2 + \lambda \| w \|_2^2 \right) \tag{1}$$

The goal of training an autoencoder is to minimize the above-mentioned cost function through performing the backpropagation algorithm. This phase of training autoencoders is called pre-training phase.

When the number of hidden layer units are less than or equal to the number of input units, and the activation function of hidden units is linear, the single-layer autoencoder acts as principal components analysis (PCA). Autoencoders implemented in this fashion are called sparse autoencoders (SAE). SAEs benefit from the sparse distribution of the hidden units; i.e., SAEs exploit deactivating

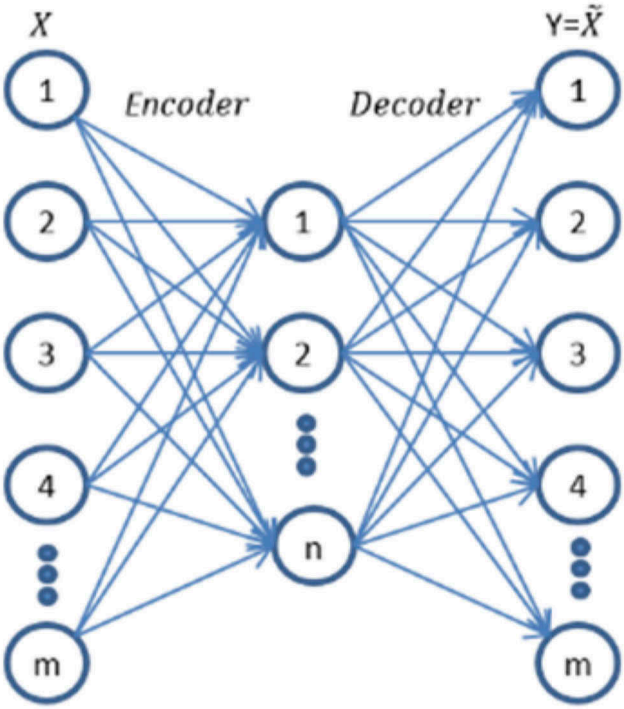

Figure 2. A typical autoencoder with n hidden units, input x, and output y (reconstructed input).

some of the latent neurons to cause sparsity in the hidden layer. Hence, the sparsity parameter is defined as the average activations of each neuron over the training set. To penalize the activations with values far from a certain value of the sparsity parameter, sparsity regularization is applied to the model, i.e., a penalty term is added to the objective function. Hence the cost function of an SAE including l2 regularization term is as follows:

$$W_{opt} = \underset{w}{^{argmin}}(\| y - XW \|_2^2 + \lambda \| w \|_2^2 + \beta KL(\tilde{\rho}, \rho)) \tag{2}$$

Where β is the sparsity regularization term and *KL* is the Kullback-Leibler divergence or the relative entropy between the desired average activation of neurons (ρ) and their actual activations ($\tilde{\rho}$). Training several autoencoders and stacking them up, then adding a classification technique or a regression method, will construct a deep classifier or a deep regression model. When the entire structure is built, the last phase of training, aka the fine-tuning phase, may be applied to adjust the parameters. Stochastic gradient descent and SVR are applied to optimize the last layers cost function (regression cost function) in this study.

The next section discusses two training models, cascaded and partially cascaded architectures, proposed to train the deep structures. The performance of these two methods is compared to that of the standard method of training

deep architectures. Later in the paper, it is shown that the following architectures results in remarkably more accurate predictions compared with the standard method of training deep regression models.

Cascaded and Partially Cascaded Architectures

Figure 3 shows the standard model of training deep architectures (Hinton and Salakhutdinov 2006). In this Fig., each layer of sparse autoencoder (SAE) is trained with the activations of its previous SAE. The inputs for the first SAE consist of the input data.

Figure 4 illustrates the cascaded and partially cascaded architecture used to train stacked autoencoders. In the cascaded model, each layer of SAE is fed with the combination of features obtained from its immediate previous layer and features from preceding layers. Algorithm 1 specifies the training procedure of a stacked cascaded sparse autoencoder. The *partially cascaded* architecture benefits from less complex design where merely the last layer receives inputs from all preceding layers.

As mentioned earlier, adding a regression method to the top layer of a deep architecture tailors the entire structure to a deep regression model. Moreover, using linear regression as the last layer does not contribute to good results, even while the features were extracted by using several layers of autoencoders. Alternatively, applying the stochastic gradient descent (SGD) or RBF kernel based SVR to optimize the cost function of the last layer works well, the reason being non-linear relationships between the dependent and independent variables of the dataset.

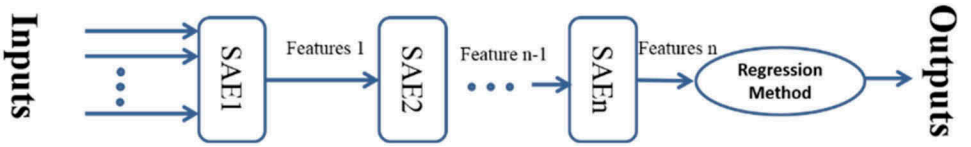

Figure 3. The standard method of training deep architectures.

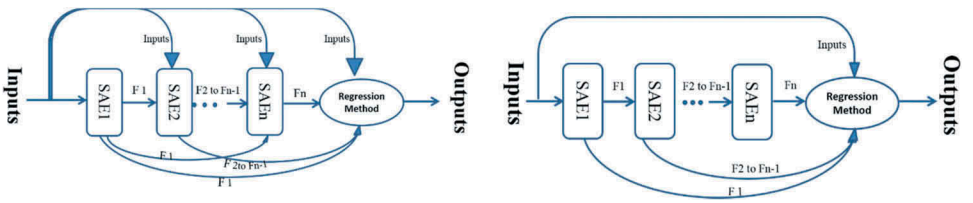

Figure 4. The cascaded (left) and partially cascaded (right) methods of training deep architectures.

Algorithm 1: Training a Cascaded Stacked Sparse Autoencoder

AE stands for autoencoder; parameters are weights and biases

output ← 0;

For *i* = 1 to *number of SAEs*

Initialize weights ~ U (-a, a), biases = 0, y ← concatenate (input data; output)

// a is a real number where 0 < a < 1; y is the output for the SAE

// Initialize the sparsity parameter and the weight of the sparsity penalty term

While stopping criterion not met

train the SAE using backpropagation algorithm

// the cost function has an additional sparsity term

End while

Compute the activation of hidden layer (*Fi*)

Fi ← A_f (y**wi*); //A_f is the activation function

output ← concatenate (*Fi*, output)

End for

Implementation Area and the Data Extraction/Source

The data used for this study is extracted from the data repository of the Minnesota Department of Transportation (MnDOT) (Minnesota Department of Transportation 2014). The area being studied is Trunk highway 10 (TH10), which is a major arterial roadway connecting the Minneapolis/St. Paul and St. Cloud metropolitan areas in Minnesota. There are several loop detectors across this highway and also on the on-ramps and off-ramps of TH10, which measure occupancy and volume. Other variables like headway, flow, density, and speed are calculated based on occupancy and volume, and all are available in the data repository.

Figure 5 shows an excerpt of the map of the region of interest. Ten-minute interval traffic flows are extracted from 73 loop detectors. The detectors are placed in on-ramps and off-ramps of TH10 and throughout TH10 about 0.8 km apart. Based on the importance and the accuracy of the data, the information of 60 loop detectors are utilized to predict the flow of station S972, including three detectors (L1 4078, L2 4079, L3 4080 shown in Figure 5).

Different time intervals were taken into consideration: 2-, 5-, 10-, and 15-minute interval traffic flows. However, the 10-minute intervals are solely utilized in this study for two reasons: the first one is heuristically getting better prediction accuracy using 10-minute intervals because this duration is long enough to include the drastic changes embedded in traffic data. The second reason why 10-minute intervals are preferred instead of 2- and

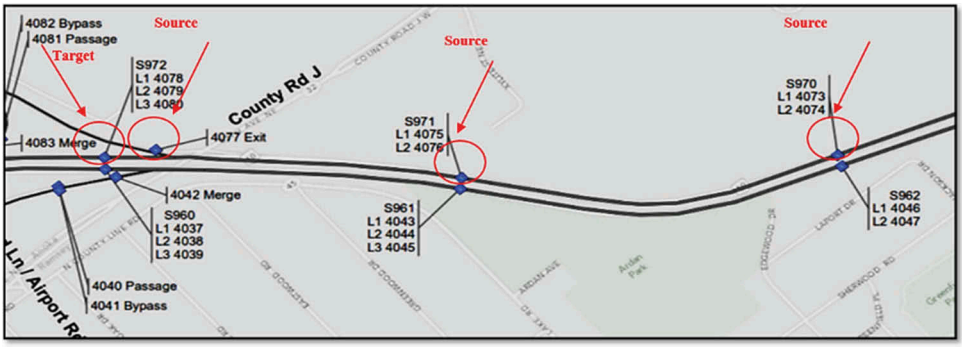

Figure 5. The map excerpt of the region under study (Minnesota Department of Transportation 2014).

5-minute measurements is to reduce the amount of training information, thus accelerating the training process. Since shorter prediction horizons are preferred, 10-minute intervals are chosen instead of 15-minute intervals.

The data comprised three months of traffic flow starting from 6 AM to 10 PM during weekdays from August 2013 to November 2013. Where there are multiple lanes, the average flow of all lanes are calculated and used. The flow rate of station S972 consisting of the average of three loop detectors (L1 4078, L2 4079, and L3 4080) is the target data to be estimated in this study. The prediction horizon comprises 13 days of 10-min interval flow rate from 6 AM to 10 PM each day.

A noteworthy feature of the extracted data is that they bear different weather conditions, crashes, traffic jams, peak and off-peak hours, special days like holidays and sporting events, and almost all of the uncertainty and stochastic attributes associated with traffic data.

Implementation Details

In this section, deep regression models with different architectures (Figure 3–4) are implemented to fit the traffic data. Models constructed of two to four layers of SAE based on the typical, cascaded, and partially cascaded models are trained.

The traffic variables highly correlated to the traffic flow rate are determined through exhaustive search in features pool and based on the results produced by deep architectures. In other words, during the implementation, different combinations of traffic variables are tried, and the best answer is derived without using "density." The reason could be the high correlation between the "density" and other variables. Also, density is measured over a length that is not reasonable to use it for point measurements (Hall 1996).

Preprocessing phase. The following demonstrates the steps that are done on preprocessing the data before fitting to the model:

- In multilane roads, the mean value of traffic variables is taken into account.
- Mean imputation is used to replace the missing data.
- Feature scaling is done on input data consisting of four traffic variables of 60 locations ahead of the target. Two temporal variables are added to measure the time during each day of each week. Specifically, the training set is composed of flow, occupancy, headway, and speed, plus time of the day, and weekdays, which is transformed into [0, 1].
- The observation points include the measurements from detectors located on TH10, plus measurements related to all on-ramps and off-ramps of TH10.
- The dataset is divided into three parts consisting of the training set, validation set, and test set. In particular, from 6,305 data values, 5,000 values are used for training and validation, and the rest are used to test the model (60% training, 20% validation, and 20% test data).

Model Selection phase. Cross validation and data splitting techniques are commonly used to find the optimal value of free parameters and assess the empirical and generalization errors. Two motivations contribute to choosing data splitting over cross validation in this research. The first one is having abundant traffic data, so putting a portion of data aside just for the validation purpose does not confine the test or the training set. The second one is this method uses less processing time compared to k-fold cross validation where training and validation must be performed k times before making a decision over an optimal parameter by averaging over k different answers. Repeating the process to set the four hyper-parameters (the number of hidden units, the sparsity parameter, the weight of the sparsity penalty term, and the weight decay (regularization) parameter) in SAEs will add to the complexity of the process.

The above-mentioned reasons weigh in favor of using three-way data splitting for this case study (60% training set, 20% validation set, and 20% test set). To further ease the process of parameter selection, each combination of parameters has given a probability of 0.6 to be engaged in the model selection technique during the training process.

Training phase. The unnormalized input data ranges from 0 to 2619 with few values of "-1" indicating the failure of the detectors to log variables. A new approach is taken to decrease the error of the reconstruction step in the SAE by scaling the logistic function. We hypothesize that, as a consequence of scaling the input data to [0, 1], the deviation of the normalized features is very small. Small deviations of input data affect the output of the sigmoidal function, so that the transferred data has very small values with very small deviation, hence smaller gradients. The small gradients affect the accuracy of

learning; therefore, the scaled version of logistic function is used to produce the larger gradients, increasing the accuracy of reconstruction by powers of 10.

Experimental Results

Two to four layers of stacked SAEs are trained according to the standard, cascaded, and partially cascaded architectures (See Figure 3–4). Table 1 through Table 3 show the results of different implementations for two-, three-, and four-layer architectures. Each table is assigned to a certain depth. It is apparent that for a two-layer model the cascaded and partially cascaded architectures in Table 1 are the same. The effect of increasing layers on the prediction precision is discussed later. Evaluations are based on the MSE (Mean Squared Error) and MAPE (Mean Absolute Percentage Error). MSE measures the deviation between the estimated values and real measurements, whereas MAPE evaluates the size of the error in percentage terms.

Table 1. Two-layer deep regression models.

Architectures/Specs	2SAEs-Standard	2SAEs-Cascaded/Partially cascaded
No.of hidden units/layer	114, 114	114, 114
L2 regularization parameter/layer	$1*10^{-3}$, $1*10^{-4}$	$1*10^{-3}$, $1*10^{-4}$
Sparsity parameter/layer	0.1, 0.1	0.1, 0.1
Sparsity regularization parameter/layer	0.2, 0.1	0.2, 0.1
MSE	973	300
MAPE (%)	2.772	1.605

Table 2. Three-layer deep regression models.

Architectures/Specs	3SAEs-Standard	3SAEs-Cascaded	3SAEs-Partially cascaded
No.of hidden units/layer	114, 228, 114	114, 114, 114	114, 570, 114
L2 regularization parameter/layer	$1*10^{-3}$, $1*10^{-4}$, $1*10^{-5}$	$3*10^{-4}$, $1*10^{-3}$, $1*10^{-3}$	$1*10^{-3}$, $1*10^{-4}$, $1*10^{-5}$
Sparsity parameter/layer	0.1for all layers	0.1for all layers	0.1for all layers
Sparsity regularization parameter/layer	0.1, 0.1, 2	0.1for all layers	0.1for all layers
MSE	536	298	286
MAPE (%)	2.431	1.586	1.588

Table 3. Four-layer deep regression models.

Architectures/Specs	4SAEs-Standard	4SAEs-Cascaded	4SAEs-Partially cascaded
No. of hidden units/layer	114, 570, 57, 114	114, 228, 114, 114	114, 570, 456, 57
L2 regularization parameter/layer	$1*10^{-3}$, $1*10^{-5}$, $1*10^{-3}$, $1*10^{-5}$	$3*10^{-4}$, $1*10^{-3}$, $1*10^{-3}$, $1*10^{-3}$	$1*10^{-3}$, $1*10^{-4}$, $1*10^{-5}$, $3*10^{-4}$
Sparsity parameter/layer	0.1, 0.1, 0.1, 0.2	0.1 for all layers	0.1 for all layers
Sparsity regularization parameter/layer	0.1, 2, 2, 1	0.1 for all layers	0.1, 1, 2, 1
MSE	646	329	295
MAPE (%)	6.957	1.636	1.595

The actual measurement and predicted values for i^{th} observation are y_i and $\hat{y}_i$. MSE and MAPE are measured using the following equations:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) * 100 \tag{4}$$

Where n is the number of observations. Since the R-value measures the linearity strength between the parameters, it is not a valid goodness-of-fit for non-linear models, so it is not included in the results.

The value of parameters related to each layer is shown in the tables. However, retraining the networks can lead to different final sets of parameter values. Several experiments for this application reveal that the factors and multiples of the number of input features render more precise results if chosen as the number of hidden neurons. Each column shows the parameters and the results obtained by training standard, cascaded, and partially cascaded architectures. Although the advantage of cascaded models over the typical structures was previously shown (Moussavi-Khalkhali and Jamshidi 2016), the entire models are implemented for comparison reasons, this time by exploiting SVRs in addition to SGD as the top layers.

To compare the performance of deep regression models, several non-parametric models are implemented with two sets of training data: the input data and the fusion of the input data with the high-level features extracted from the deep architectures.

Models using a nu-SVR (Chang and Lin 2011) with RBF[1] as its kernel function is performed, and the results are demonstrated in Table 4. Since the partially cascaded models surpass the other architectures using SGD, SAEs with SVR fed by a training set, consisting of the fusion of input and high-level features, are implemented. To train a nu-SVR two parameters need tuning: the regularization parameter (C) and the variable to control the number of support vectors (υ). The validation set is used to find the optimal parameters.

Table 5 demonstrates the results of training several traditional neural networks. Particularly, 10 neural networks are trained, and the one with the least generalization error is shown in Table 5. The average performance of these

Table 4. Deep regression models based on SVR and partially cascaded training.

Architectures/Specs	1SAE- Partially cascaded	2SAEs-Partially cascaded	3SAEs-Partially cascaded	4SAEs- Partially cascaded
Regularization parameter C; υ	C = 35000 υ = 0.68	C = 100000 υ = 0.66	C = 110000 υ = 0.66	C = 120000 υ = 0.66
MSE	301	300	299	299
MAPE (%)	1.575	1.573	1.573	1.574

Table 5. Traditional neural networks and PCA-amplified neural networks.

Architectures/Specs	An ANN	An ANN Augmented by PCA	An Amplified ANN (An ANN Augmented by PCA -Fed with input and high-level features)
No.of hidden units	20	20	20
No. of PCs	NA	20 out of 114	20 out of 114
MSE (The best perf.)	656	348	346
MAPE (%)(The best perf.)	2.107	1.734	1.724
Average MSE	2100	398	385
Average MAPE (%)	2.815	1.779	1.770

networks is also mentioned in the table. For further comparisons, models consisted of linear PCA and ANN are also implemented and the results are displayed in Table 5. The average performance and the results of the model with the least MSE and MAPE are shown in Table 5. In another experiment, the PCA is applied to the input data in conjunction with the high-level features extracted from deep models. This model, called amplified ANN, results in slightly more accurate predictions than the preceding models trained with ANN.

Studying the results from Tables 4 and 5 implies that the features extracted from the deep models boost the performance of traditional neural networks to some extent.

The following conclusions are derived from studying the results presented in Table 1 through 5.

- The Prediction precision of deep structures that are trained based on the proposed cascaded and partially cascaded models surpasses the precision of the structures trained with the standard method and even outperforms the amplified ANNs.
- Between the partially cascaded and cascaded models the former delivers the least generalization error in terms of MSE and MAPE. Specifically, the partially cascaded method using three-layer SAE renders the most accurate predictions.
- Comparing the MAPE values SAEs based on SVRs lead to better results. Although MSE values implies that SAEs based on SGD render more accurate results, fewer adjustments to set the hyper-parameters makes SVR appropriate in the presence of time and resource constraints. In other words, SVRs benefit from less complex training procedure, hence less process and time. The two parameters for SVR, the regularization parameter (C) and the variable which controls the number of support vectors (υ) are adjusted using the validation set. C takes a broad range of numbers, but υ is between 0 and 1.
- The prediction accuracy of the ANN augmented by linear PCA appears to be superior to that of the ANN without using PCA (See Table 5). This

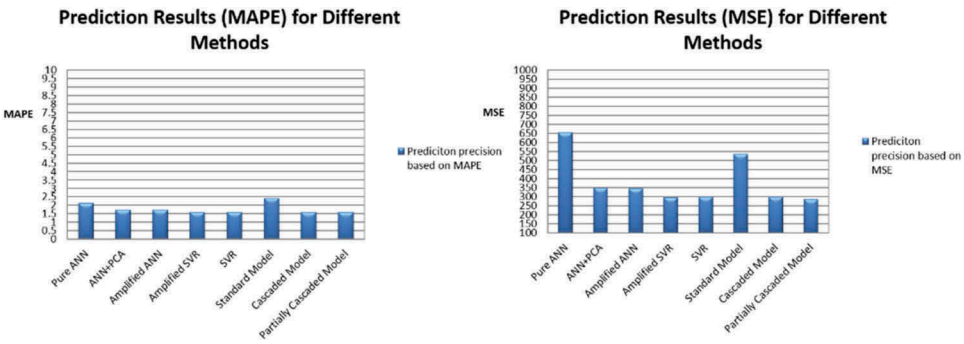

Figure 6. Performance comparison of different methods.

is not far from expectation as PCA selects the most relevant features that best describes the variance in the input data. Subsequently, the ANN combined with PCA delivers the better precision if it is fed by the fusion of high-level features and the input data.

- ANNs enhanced by PCA outperform the deep regression models trained according to the standard method.

Figure 6 illustrates the performance comparison of different methods, which shows the superior performance of cascaded and partially cascaded models over the other methods. However, between the two former models, the partially cascaded method, where the last layer is fed with the activations of preceding layers, leads to better results. Another advantage of the partially cascaded model, compared to the cascaded model, is less complexity and, thus, more process and time proficiencies.

Generally, the partially cascaded architectures show a great potential alternative to the standard method of training sparse autoencoders with respect to regression analysis. In fact, this model shows the best performance compared to other methods as well. Furthermore, the results imply that, as the depth of the model increases, the accuracy in terms of MSE and MAPE rises. However, increasing the depth from three layers to four layers did not contribute to a drastic change in accuracy. In fact, after a specific number of layers, adding more depth to the model will deteriorate the accuracy of results while adding to the complexity of the model. However, the "right" number of hidden layers differs with the application and underlying distribution of the training data.

Discussion and Conclusion

The objective of this paper is to explore and utilize the capability of feature extraction of stacked sparse autoencoders in the context of regression analysis applied to traffic flow forecasting. Considering the aforementioned

objective, deep regression models employing sparse autoencoders are constructed based on the standard, cascaded, and partially cascaded training methods. This research represents that integrating features extracted from deep regression models is able to enhance the prediction accuracy of traffic flow rate, even in the presence of abnormalities and unusual phenomena such as accidents, diverse weather conditions, peak and off-peak hours. Since the training dataset required to train the deep structures is not a very large dataset (three months of historical traffic data), retraining the models with the newest available data is time efficient. Retraining the network in offline methods is required every so often to ensure the high prediction accuracy. Despite performance enhancement achieved by utilizing deep regression models, many hyper-parameters needed to be tuned and be set to their right values during the training procedure. The regularization term for sparse autoencoders, the parameters related to the sparsity criteria, the number of layers, the number of hidden units, and the regularization term for the fine-tuning phase are among factors that affect the precision accuracy of the regression model. Additionally, setting the number of iterations and the learning rate of the SGD algorithm in each of the models needed some heuristic searching. However, following the tricks of training the SGD algorithm mentioned by (Bottou 2012) and finding the right parameters for one model could result in achieving optimum values for other models more easily. Unfortunately, these hyper-parameters are not the only values needed to be determined. For this study, another variable requires adjustment, and that is the scale of narrowing the sigmoidal function so as to make the model robust to the outliers.

Considering the depth of the models reveals that adding more hidden layers to the structure does not necessarily contribute to better results. In fact, in some cases the best performance is achieved with the first layer and the accuracy worsens by adding more layers (Theis et al. 2011). Obviously, there is a tradeoff between the complexity of the model and the expected precision. Thus, the best decision can be made based on the application and the desired performance.

As a part of this research, the most correlated features to the flow variables are selected based on the exhaustive search through the pool of traffic variables; the best results are yielded using the flow, occupancy, headway, and speed as the attributes extracted from the other locations. All non-parametric methods delivered their best results using these variables.

Although utilizing the partially cascaded model in training deep architectures is considered as an encouraging approach, traditional neural networks amplified with PCA trained by the high-level features along with the input data demonstrate remarkably more accurate results than stacked SAEs trained based on the standard model. Partially cascaded models with SVR as top layer shows better performance when comparing MAPE values. However,

SAEs based on SGD render predictions with less MSE values. In general, training SAEs with SVR at top layer is a little challenging. Since the input of SVR is to be scaled such that all values are between 0 and 1, the high-level features, concatenated with the raw input, are rescaled using the statistics of the input data. As mentioned earlier, the two parameters for SVR, the regularization parameter and the parameter to control the support vectors, are tuned using the validation set. The granularity of the parameters, especially nu (υ), plays a major role in achieving the optimal solution.

To further explore the autoencoders in the regression task, we intend to implement deep denoising autoencoders, and apply them to the same traffic data, and eventually compare the results with the sparse autoencoders and some of the advanced parametric models in the next upcoming studies. Also, evaluating these methods on urban traffic data is a topic of interest, as the signalized urban road traffic data is affected by other factors such as signal timing not only in the target road, but in a large vicinity of neighboring roads.

Note

1. Radial Basis Function.

References

Abdulhai, B., H. Porwal, and W. Recker. 2002. Short-term traffic flow prediction using neuro-genetic algorithms. *ITS Journal-Intelligent Transportation Systems Journal* 7 (1):3–41. doi:10.1080/10248070212011.

Bengio, Y. 2009. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning* 2 (1):1–127. doi:10.1561/2200000006.

Bolshinsky, E., and R. Freidman. 2012. Traffic flow forecast survey. *Technion–Israel Institute of Technology.–2012.–Technical Report.–15 p.*

Chan, K. Y., T. S. Dillon, J. Singh, and E. Chang. 2012. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. *Intelligent Transportation Systems, IEEE Transactions On* 13 (2):644–54. doi:10.1109/TITS.2011.2174051.

Chang, C.-C., and C.-J. Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):27. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chang, G., T. Ge, and J. Li. 2012, April. Hybrid support vector machine and ARIMA model for short-term traffic flow forecasting. Proceedings of the 2012 Second International Conference on Electric Information and Control Engineering-Volume 02, (pp. 827–30), IEEE Computer Society, Washington, DC, USA.

Davis, G. A., and N. L. Nihan. 1991. Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering* 117 (2):178–88. doi:10.1061/(ASCE)0733-947X(1991)117:2(178).

Dia, H. 2001. An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research* 131 (2):253–61. doi:10.1016/S0377-2217(00)00125-9.

Dimitriou, L., T. Tsekeris, and A. Stathopoulos. 2008. Adaptive hybrid fuzzy rule-based system approach for modeling and predicting urban traffic flow. *Transportation Research Part C: Emerging Technologies* 16 (5):554–73. doi:10.1016/j.trc.2007.11.003.

Dougherty, M. S., and M. R. Cobbett. 1997. Short-term inter-urban traffic forecasts using neural networks. *International Journal of Forecasting* 13 (1):21–31. doi:10.1016/S0169-2070(96)00697-8.

Ezell, S. 2010. Explaining international it application leadership: Intelligent transportation systems.

Guo, J., W. Huang, and B. M. Williams. 2014. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies* 43:50–64. doi:10.1016/j.trc.2014.02.006.

Hall, F. L. 1996. *Traffic stream characteristics. Traffic flow theory*. US Federal Highway Administration, 36. McMaster University, Department of Civil Engineering and Department of Geography, Ontario, Canada.

Hinton, G. E., and R. R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313 (5786):504–07. doi:10.1126/science.1127647.

Huang, W., H. Hong, M. Li, W. Hu, G. Song, and K. Xie. 2013, December. Deep architecture for traffic flow prediction. International Conference on Advanced Data Mining and Applications, (pp. 165–76), Hangzhou, China, Springer Berlin Heidelberg. doi:10.1177/1753193412441124.

Lee, H. K. 2000. A framework for nonparametric regression using neural networks. Pacific Rim International Conference on Artificial Intelligence Melbourne, Australia.

Li, L., W. H. Lin, and H. Liu. 2006, March. Type-2 fuzzy logic approach for short-term traffic forecasting. IEE Proceedings-Intelligent Transport Systems(Vol. 153, No. 1, pp. 33–40), IET Digital Library. doi:10.1049/ip-its:20055009.

Li, S., Z. Shen, and G. Xiong. 2012, September. A k-nearest neighbor locally weighted regression method for short-term traffic flow forecasting. Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on(pp. 1596–1601), IEEE. Anchorage, USA.

Lv, Y., Y. Duan, W. Kang, Z. Li, and F. Y. Wang. 2015. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16 (2):865–73.

Minnesota Department of Transportation. 2014. Accessed 2014. http://data.dot.state.mn.us/datatools/.

Moussavi-Khalkhali, A. and Jamshidi, M., 2016, December. Constructing a Deep Regression Model Utilizing Cascaded Sparse Autoencoders and Stochastic Gradient Descent. In 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), (pp. 559-564). IEEE. Anaheim, California, USA.

Noonan, J., and O. Shearer. 1998. Intelligent transportation systems field operational test cross-cutting study: Advance traveler information systems (No. FHWA-JPO-99-038).

Ojeda, L. L., A. Y. Kibangou, and C. C. De Wit. 2013, June. Adaptive Kalman filtering for multi-step ahead traffic flow prediction. American Control Conference (ACC), 2013, (pp. 4724–29), IEEE. Washington, DC, USA.

Oswald, R. K., W. T. Scherer, and B. L. Smith. 2000. Traffic flow forecasting using approximate nearest neighbor nonparametric regression. *Final project of ITS Center project: Traffic forecasting: non-parametric regressions*.

Su, H., L. Zhang, and S. Yu. 2007, August. Short-term traffic flow prediction based on incremental support vector regression. Natural Computation, 2007. ICNC 2007. Third International Conference on (Vol. 1, pp. 640–645), IEEE. Haikou, China.

Sun, H., H. X. Liu, H. Xiao, R. R. He, and B. Ran. 2003, January. Short term traffic forecasting using the local linear regression model. 82nd Annual Meeting of the Transportation Research Board, Washington, DC.

Theis, L., S. Gerwinn, F. Sinz, and M. Bethge. 2011. In all likelihood, deep belief is not enough. *The Journal of Machine Learning Research* 12:3071–96.

Van Der Voort, M., M. Dougherty, and S. Watson. 1996. Combining Kohonen maps with ARIMA time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies* 4 (5):307–18. doi:10.1016/S0968-090X(97)82903-8.

Van Hinsbergen, J. W. C., and F. M. Sanders. 2007. Short Term traffic prediction models.

Vlahogianni, E. I., M. G. Karlaftis, and J. C. Golias. 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. *Transportation Research Part C: Emerging Technologies* 13 (3):211–34. doi:10.1016/j.trc.2005.04.007.

Vlahogianni, E. I., M. G. Karlaftis, and J. C. Golias. 2007. Spatio temporal short term urban traffic volume forecasting using genetically optimized modular networks. *Computer Aided Civil and Infrastructure Engineering* 22 (5):317–25. doi:10.1111/mice.2007.22.issue-5.

Williams, B. M. 2001. Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling. *Transportation Research Record: Journal of the Transportation Research Board* 1776 (1):194–200. doi:10.3141/1776-25.

Williams, B. M., and L. A. Hoel. 1999. Modeling and forecasting vehicular traffic flow as a seasonal stochastic time series process (No. LTVA/29242/CE99/103).

Williams, B. M., and L. A. Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering* 129 (6):664–72. doi:10.1061/(ASCE)0733-947X(2003)129:6(664).

Wu, C. H., J. M. Ho, and D. T. Lee. 2004. Travel-time prediction with support vector regression. *Intelligent Transportation Systems, IEEE Transactions On* 5 (4):276–81. doi:10.1109/TITS.2004.837813.

Yu, G., and C. Zhang. 2004, May. Switching ARIMA model based forecasting for traffic flow. Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04). IEEE International Conference on (Vol. 2, pp. ii-429), IEEE. Montreal, Que., Canada.

Zhang, L., Q. Liu, W. Yang, N. Wei, and D. Dong. 2013. An improved K-nearest neighbor model for short-term traffic flow prediction. *Procedia-Social and Behavioral Sciences* 96:653–62. doi:10.1016/j.sbspro.2013.08.076.

Zhang, Y., and Y. Liu. 2009. Comparison of parametric and nonparametric techniques for non-Peak traffic forecasting. *World Academic of Science and Engineering Technology* 51:50.

Zhang, Y., and Z. Ye. 2008. Short-term traffic flow forecasting using fuzzy logic system methods. *Journal of Intelligent Transportation Systems* 12 (3):102–12. doi:10.1080/15472450802262281.