Spectral Basis Neural Networks for Real-Time Travel Time Forecasting

By Dongjoo Park,[1] Laurence R. Rilett,[2] and Gunhee Han[3]

(Reviewed by the Urban Transportation Division)

Abstract: This paper examines how real-time information gathered as part of intelligent transportation systems can be used to predict link travel times for one through five time periods ahead (of 5-min duration). The study employed a spectral basis artificial neural network (SNN) that utilizes a sinusoidal transformation technique to increase the linear separability of the input features. Link travel times from Houston that had been collected as part of the automatic vehicle identification system of the TranStar system were used as a test bed. It was found that the SNN outperformed a conventional artificial neural network and gave similar results to that of modular neural networks. However, the SNN requires significantly less effort on the part of the modeler than modular neural networks. The results of the best SNN were compared with conventional link travel time prediction techniques including a Kalman filtering model, exponential smoothing model, historical profile, and real-time profile. It was found that the SNN gave the best overall results.

INTRODUCTION

With the advent of advanced traveler information systems and in particular route guidance systems (RGS), the prediction of short-term link travel times has become increasingly important. Intuitively, the RGS's route selection algorithms should use link travel times that are based on the time at which the driver is expected to arrive at a given link rather than use link travel times that are based on current conditions. This would be particularly important for trips where the expected arrival time at a link is relatively far into the future, and it is unlikely that the current link travel time will remain stable. Because drivers implicitly base their routes on the anticipated link travel time when they arrive at a particular link, the RGS should have the same capabilities (Hoffman and Janko 1990; Chen and Underwood 1991; Rilett and Van Aerde 1991; Rilett 1992; Boyce et al. 1993; Tarko and Rouphail 1993). This paper examines how real-time information gathered as part of an intelligent transportation system can be used to predict link travel times for the near future.

Artificial neural networks (ANN) and in particular multilayer perceptron neural networks that utilize a back-propagation algorithm have been applied successfully for forecasting link travel time and other traffic parameters. Hereafter, the term conventional ANN will refer to multilayer perceptron neural networks that utilize a back-propagation algorithm unless mentioned otherwise. The successful applications of conventional ANN may be attributed to the nonlinear and multidimensional nature of many transportation problems. However, there are numerous practical shortcomings associated with conventional ANN and a number of methodologies including modular neural networks (MNN) have been proposed to alleviate these shortcomings.

The focus of this paper is on using spectral basis neural networks (SNN) that combine a pretransformation of the input features and conventional ANN for forecasting multiple-periods link travel times into the future. The pretransformation is based on sinusoidal functions and is performed on the input features (in this paper, recent average link travel times) to obtain linearly separable input features. The goal is to convert the complex, nonmonotonic function that relates future travel times to the input features into a monotonic function. The data used as input were link travel times from Houston that were collected as part of the Automatic Vehicle Identification (AVI) system of the Houston Transtar system.

This paper is organized as follows. A state-of-the-art literature review of the short-term travel time forecasting models, with an emphasis of ANN applications, is presented in Section 2. The shortcomings of conventional ANN and MNN with respect to the short-term link travel time forecasting problem are discussed and an SNN is proposed in Section 3. The test freeway corridor in Houston and the data collected for calibration and validation are described in Section 4. The proposed SNN is then implemented, and the results are compared with the results from a conventional ANN and an MNN in Section 5. The best SNN is identified and compared with existing link travel time prediction techniques including a Kalman filtering model, an exponential smoothing model, a historical profile, and a real-time profile in Section 6. Finally a concluding discussion follows in Section 7 and includes a summary of the proposed approach and future extensions.

LITERATURE REVIEW

Historically, most forecasting models in the transportation area have tended to focus on short-term traffic flow prediction for use in the operation and control of traffic signals and ramp meters (Okutani and Stephanedes 1984; Davis and Nihan 1991; Dougherty and Kirby 1993; Smith and Demetsky 1994). Since the mid-1980s, the estimation and prediction of short-term link travel times has become increasingly important. This is in recognition of the fact that for distributed RGS to be successful, the calculated routes should be based not only on historical and real-time link travel time but also on anticipatory link travel time information. Most travel time studies have focused on link travel time estimation, rather than link travel time forecasting, and the input data typically consists of traffic volume and occupancy from loop detectors, probe vehicles, and simulation programs (Dailey 1993; Pushkar et al. 1995; Sisiopiku and Rouphail 1995).

The existing short-term link travel time forecasting models

[1]Postdoctoral Res., Texas Transp. Inst., Texas A&M Univ. Sys., College Station, TX 77843-3135. E-mail: dpark@tamu.edu

[2]Asst. Prof., Dept. of Civ. Engrg. and Asst. Res. Engr., Texas Transp. Inst., Texas A&M Univ. Sys., College Station, TX. E-mail: rilett@tamu.edu

[3]Asst. Prof., Dept. of Electr. Engrg., Yonsei Univ., Seoul, South Korea. E-mail: gunhee@cvlsi13.yonsei.ac.kr

Note. Discussion open until May 1, 2000. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on October 5, 1998. This paper is part of the ***Journal of Transportation Engineering***, Vol. 125, No. 6, November/December, 1999. ©ASCE, ISSN 0733-947X/99/0006-0515–0523/$8.00 + $.50 per page. Paper No. 19380.

include time series models (Anderson et al. 1994; Van Arem et al. 1997), Kalman filtering models (Park and Rilett 1997, 1998), neural network models (Florio and Mussone 1994; Park and Rilett 1997, 1998; Park et al. 1998), and historical and real-time profiles of the several RGS projects including ADVANCE in Chicago and ALI_SCOUT in Berlin (Hoffman and Janko 1990; Boyce et al. 1993; Tarko and Rouphail 1993).

Although there have been a number of studies using ANN for predicting traffic flow-related parameters such as volume and density, relatively few studies have used ANN for forecasting short-term link speeds or travel times. Florio and Mussone (1994) applied a neural network to simultaneously predict density, flow, and speed on sections of a freeway. Density, flow, meteorology, and percent of heavy vehicles at instant t were used as input variables and the outputs were density, flow, and speed at time $t + \Delta t$, where Δt was 10 min. In this paper, predicting variables at time $t + \Delta t$ will be referred to as ''one time period ahead prediction,'' predicting variables at time $t + 2\Delta t$ will be referred to as ''two time periods ahead prediction,'' etc. Park and Rilett (1997) proposed several ANN link travel time prediction models for US 290 in Houston using 1994 AVI travel time data. The ANN used link travel time patterns not only from the target link but also from upstream and downstream links as input and predicted link travel times for one through five time periods (Δt = 5 min) ahead. It was found that, when predicting one and two time periods ahead, the ANN that used only the preceding travel times from the target link gave the best results. However, when predicting three through five time periods ahead, the ANN that employed link travel times not only from the target link but also from upstream and downstream links gave superior results. In this latter situation, the five most recent link travel times from the three links gave the best results. These ANN applications all used conventional ANN for forecasting the link travel time.

Park and Rilett (1998) used an MNN that combined artificial intelligence clustering techniques (Kohonen Self-Organizing Feature Maps and the fuzzy c-means clustering algorithm) with a conventional ANN for predicting link travel times. In the study the complex link travel time function that was approximated was modeled as a combination of simpler functions where a separate ANN model was calibrated for each separate classification. Because an individual ANN model is calibrated for a simpler function of each class, the MNN are less likely to experience some of the deficiencies associated with conventional ANN such as identifying a local minimum and being sensitive to outliers. As expected, the MNN reduced the forecasting error with an overall 17% relative improvement as compared with the conventional singular ANN. A number of other neurofuzzy or hybrid ANN approaches also have been applied successfully for nontransportation applications (Poddar and Rao 1993; Chang 1996), although the specific methodologies for partitioning and approximating the function are different. In a transportation application, Van Der Voort et al. (1996) combined a Kohonen Self-Organizing Feature Maps with an ARIMA time series model to forecast traffic flow.

Although the MNN results from the previous study were encouraging, it was found that the MNN approach required a more labor intensive effort on the part of the modeler as compared with that required by a conventional ANN. For example, the MNN requires an extensive process for partitioning the input features that includes selecting a clustering algorithm, identifying the classification parameters, extracting the features, and deciding the appropriate weight vectors (Park and Rilett 1998).

SNN

Shortcomings of Conventional Multilayer Perceptron Neural Networks

Hornik et al. (1989) theoretically showed that conventional ANN utilizing a single hidden layer and using arbitrary squashing functions can approximate any measurable function from a finite-dimensional space to another finite-dimensional space to any desired degree of accuracy, provided that a sufficient number of hidden units (neurons) are available. In this sense, the ANN are considered as a class of universal approximators. Note that the majority of the neural network application in the transportation field have been conventional ANN (Dougherty 1993; Dougherty and Kirby 1993; Ritchie and Cheu 1993; Vythoulkas 1993; Florio and Mussone 1994; Smith and Demetsky 1994; Chang and Su 1995; Gilmore and Abe 1995; Park and Rilett 1997).

In practice, however, it has been found that a conventional ANN works efficiently when the function that is approximated is relatively monotonic with only a few dimensions of the input features, but may not be as efficient when this is not the case (Haykin 1994; Musavi et al. 1994). The ANN encounters the greatest difficulty in approximating functions when the input features are not linearly separable. Because the linear separability in a classification problem is equivalent to the monotonicity in the function approximation problem, an input feature that is not linearly separable implies that the approximated function has a higher complexity. The difficulty becomes more severe as the total number of classes or patterns of the function that is approximated increases (i.e., higher nonmonotonicity), and accordingly the distribution of classes becomes more complicated. That is, most errors occur when the input features to be classified are linearly nonseparable or close to class boundaries. These input features are referred to as ''hard vectors.''

In general, as the total number of classes that need to be identified increases, the ANN needs to become an expanded network. That is, additional hidden layers and/or neurons are required to have enough capacity to allow all the classes to be modeled with one network. This expanded network may result in long computation time and implementation difficulties (Schmidt 1993; Musavi et al. 1994; Chang 1996). Often, numerous experiments are needed to identify the best network structures because the practitioner does not know beforehand the form of the higher dimensional function that is being approximated. In addition, the convergence of the ANN depends upon several factors including the complexity of the function that is being approximated, the order of the presentation of the input features, and the choice of parameters in the training algorithm. In general, however, there is no clear methodology for deciding the appropriate values of these factors (Hornik et al. 1989; Boulard and Wellekens 1990; Haykin 1994; Hagan et al. 1995; Chang 1996; Park and Rilett 1998). As a rule of thumb, it is best to try several different initial conditions with respect to the neuron attributes (i.e., the weights and biases) to ensure that an optimum solution has been obtained. Momentum, variable learning rate, and conjugate gradient optimization techniques have all been proposed as heuristic modifications of the conventional ANN for use when modeling complex functions (Haykin 1994; Hagan et al. 1995). However, they are mainly designed to speed up the convergence of the ANN rather than to improve the solution quality.

MNN: Input Partitioning (Preclassification)

There are two possible techniques for reducing the problems associated with the conventional ANN when approximating a highly nonlinear function. The first technique employs ''input-partitioning (i.e., preclassification),'' whereas the second technique is based on ''input-transformation (i.e., premapping).'' The basic strategy of the input-partitioning, or MNN, is to replace a complex function that is to be approximated with a combination of relatively simpler functions. An ANN is said to be modular if the computation performed by the network can be decomposed into two or more subsystems that operate on distinct inputs without communicating with each other (Haykin 1994). Because an individual ANN model or subnetwork is calibrated based on a simpler function of each class, the MNN are less likely to experience the deficiencies possible with a conventional ANN (Jacobs et al. 1991; Horikawa et al. 1992; Anand et al. 1995; Park and Rilett 1998) [see also Haykin (1994) for details].

Proposed Approach: Input Feature Transformation (SNN)

This paper proposes an SNN that incorporates an input transformation or premapping. The SNN is also known as an expanded input classifier, functional expansion, and higher order or augmented neural network. The motivation for this architecture is that classification errors can be reduced and the ANN performance improved, if the hard vectors are transformed into another set of vectors (that is, into a different dimension) so that they are easier to classify. The hard vectors are converted using a nonlinear transformation such that the input features become simpler to classify without actually introducing any new information (Han 1997). In this paper the nonlinear transformation is based on a sinusoidal transformation. Pao (1989), Ersoy and Hong (1990), and Eck and Shih (1994) introduced similar approaches using variant transformation functions.

The proposed SNN is explained with a simple example. Fig. 1 represents an example of a nonmonotonic function with 1D input x. The output y has a bipolar value (i.e., 1 or -1). Assume that A_1 and A_2 belong to class ''A'' that has a target value of 1 for y and that B_1 and B_2 belong to class ''B'' with a target value of -1 for y. The requirement is to construct a pattern classifier that produces output 1 in response to input patterns A_1 and A_2 and output -1 in response to input patterns B_1 and B_2. However, note that the y value cannot be calculated from the input value x using any linear function. It is relatively easy to show that this problem is not linearly separable with a one-layer structure and would require a three-layer structure to classify the A and B regions (Hagan et al. 1995).

In contrast to the above linear perceptron approach, the problem can be classified using only one layer if an appropriate nonlinear transformation is employed. Eq. (1) shows a nonlinear transformation based on sinusoidal functions

$$x_1 = \sin(2\pi x); \quad x_2 = \sin(4\pi x) \qquad (1a,b)$$

If (1) is applied to the nonmonotonic function shown in Fig. 1, a one-to-one relationship between the input and output patterns may be obtained. Accordingly, the partitions, labeled as A_1, A_2, B_1, and B_2 in Fig. 1, form a linearly separable configuration, and the classified problem can be solved readily by any linear classifier, such as the linear perceptron. In other words, the use of nonlinear transformation functions is sufficient to transform a linearly nonseparable problem into a linearly separable one. Note that for the example in Fig. 1, the MNN approach would first classify the given input patterns into two groups—one that has a target value of 1 for y and the other that has a target value of -1 for y—and would develop a separate ANN for each input pattern.

It should be noted that any orthonormal (orthogonal) basis functions can be used to define the nonlinear transformation. This study employs a spectral (i.e., Fourier) expansion of the input features. Assume that an input vector is n dimensional and that the number of expansion coefficients is k. Then, the new expanded input vector is $n \times k$ dimensional. The orthonormalized nonlinear transformation of the input vector is defined in (2), which includes both cosine and sine functions (Han 1997). Fig. 2 shows the structure of the proposed SNN

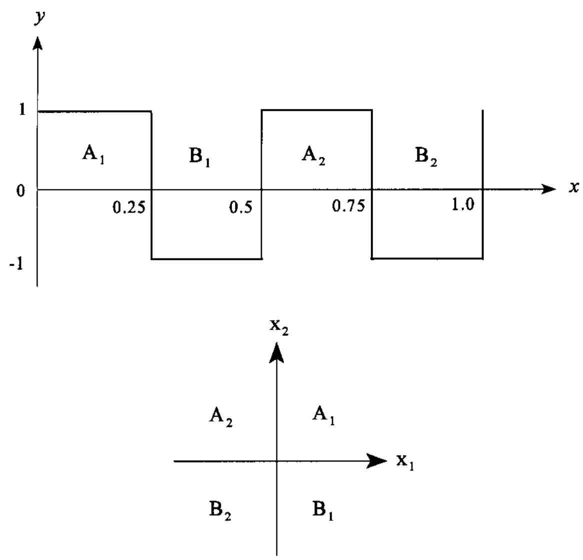

FIG. 1. Nonmonotonic Function and New Partition Configuration

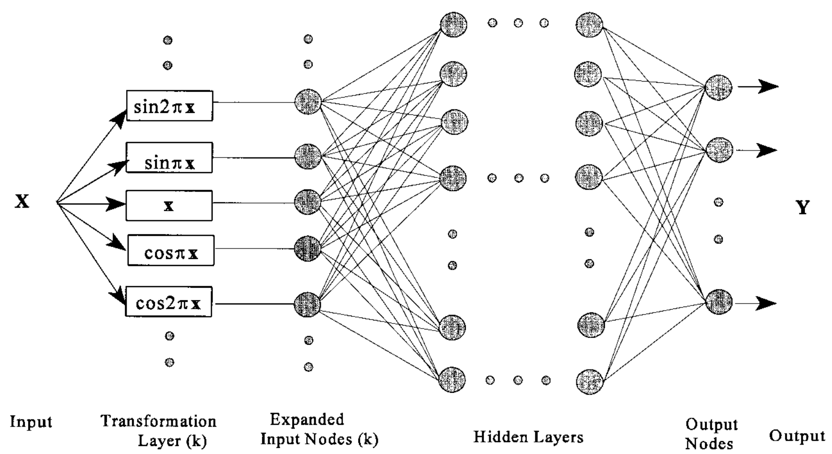

FIG. 2. Proposed SNN with Expanded Input Nodes

$$\mathbf{x}(r) = \begin{cases} \mathbf{x} & \text{if } r = 1 \\ \sin\left(\frac{r}{2} \cdot \pi \cdot \mathbf{x}\right) & \text{if } r \text{ is even} \\ \cos\left(\frac{r-1}{2} \cdot \pi \cdot \mathbf{x}\right) & \text{if } r \text{ is odd and } r > 1 \end{cases}, \quad r = 1, \ldots, k \tag{2}$$

The number of required expansion nodes of the SNN, which corresponds to the frequency if a sinusoidal function is utilized, should be decided based on the complexity of the function that is being approximated and the network structure. In general, as the degree of nonlinearity of the function that is being approximated increases, the number of expansion nodes increases as well. It is important to note that spectral analysis has been used to predict a time series that exhibits periodic behavior (Nicholson and Swann 1974) or to characterize speed in the frequency domain (Lu 1992), whereas the objective of the spectral analysis used in the proposed approach is to increase linear separability of input features. Because of the similarity between traditional spectral analysis based on Nyquist theorem and SNN, it is hypothesized that the Nyquist theorem may provide insight into the number of expansion nodes needed to accomplish this. However, it is unlikely that the Nyquist theorem or spectral analysis would provide an optimal value because the proposed SNN utilizes a different model structure and optimization technique than that employed in spectral analysis. Consequently, the best number of expanded nodes still must be obtained from experimentation.

STUDY FREEWAY AND DATA COLLECTION

The SNN models described in this paper were trained and tested on travel time data collected through an AVI system in Houston. A preliminary analysis of the data was undertaken for three reasons. First, it allows for a direct examination of the dynamic evolution of the freeway travel time pattern on a given link and provides insight into what input features are important. Second, it provides both the real-time and historical travel time base, which can then be compared with the results of the proposed SNN. Third, it shows the highly nonlinear nature of travel time prediction function, which is the motivation of the proposed approach.

Study Freeway Corridor: US 290 in Houston

The test bed for this study was US 290, which is a radial six-lane urban freeway in Houston as shown in Fig. 3. It has a barrier-separated HOV lane that runs along the centerline of the freeway for approximately 19 km, and the data utilized were from the non-HOV section of the freeway.

Data Collection and Aggregation

Travel time data were collected over a 27.6-km stretch of US 290 from seven AVI reader stations (yielding six links) as

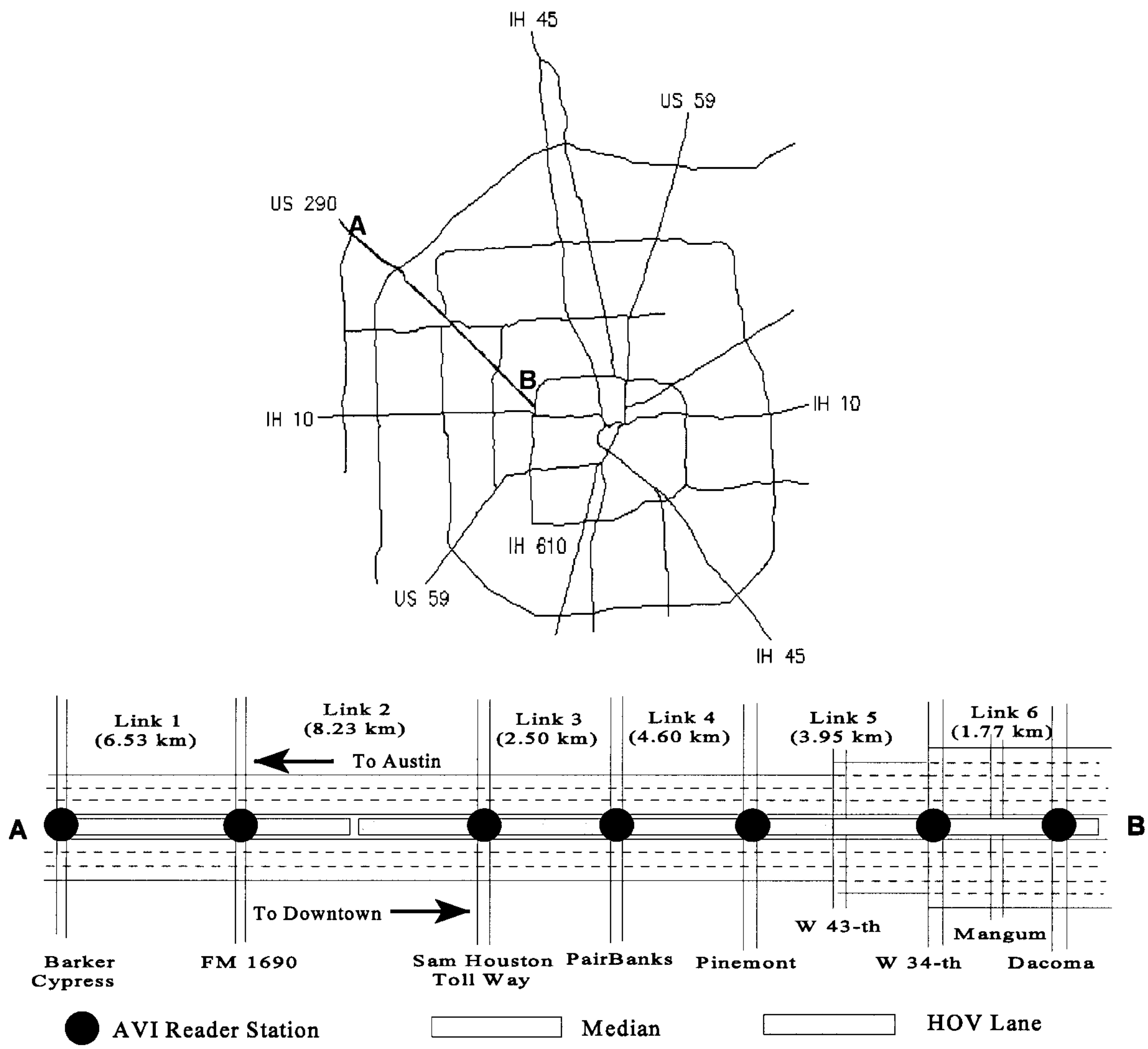

FIG. 3. Study Urban Freeway Corridor US 290 in Houston

shown in Fig. 3. The data were collected over a 24-h period each weekday in both directions of travel for 12 months in 1996 yielding 231 weekdays. The travel time data from the AVI vehicles were subsequently aggregated at 5-min periods for each link. Only the eastbound a.m. peak travel time data were employed in this study because these links experienced more severe congestion than the westbound links. Based on a visual inspection of the travel time patterns, the peak period was defined as lasting from 6:00 to 10:00 a.m., and the data collected over this time period were used for this study. During this peak period, an average of approximately 15–30 vehicles per 5-min period passed links 3, 4, 5, and 6, whereas an average of approximately 6–10 vehicles passed links 1 and 2 per 5-min period.

Space Mean Speed

For visual comparison purposes, the link travel speeds are better indicators of conditions on the highway than link travel times because the links have different lengths. Fig. 4 shows the peak period space mean speeds averaged over a 5-min period for 231 days on the six links. Because of the nature of the AVI data, the ''space'' mean speed rather than the ''time'' mean speed is obtained for any given time period and will be referred to as the mean speed in this paper. The mean speed on links 3, 4, and 5 ranges between approximately 30 and 40 km/h from 7:10 to 7:40 a.m. The mean speed on link 6 is approximately 25 km/h between 7:30 and 8:10 a.m., indicating that link 6 experiences a more severe and later period of congestion than the other links. Link 2 shows a relatively small speed reduction with the lowest mean speed of 70 km/h. Conversely, link 1 experiences an average travel speed pattern that is almost equal to the free-flow travel speed of approximately 110 km/h. The lowest speeds on links 2, 3, 4, and 5 occur at approximately 7:30 a.m. whereas that of link 6 occurs at approximately 7:50 a.m.

Travel Time Variance Analysis Using Nested ANOVA

An analysis of the link travel time variability using a nested ANOVA was performed to identify the variance associated with each of the error components and the relative importance of each of the components to the overall variance. The nested ANOVA had four levels—month, day, interval, and within interval (between vehicles). Travel time data for 12 intervals from 7:00 to 8:00 a.m. was incorporated in this analysis. Links 1 and 2 were excluded from this analysis because there was an insufficient number of vehicles for statistical analysis. The total variability was identified, and estimates of these four components were derived.

The results of the nested ANOVA are shown in Table 1. It was found for all links that the largest source of the variation

TABLE 1. Travel Time Variance Sources Using Nested ANOVA

Link (1)	Statistics (s) (2)	Variance Source: Within intervals (3)	Between intervals (4)	Between days (5)	Between months (6)
3	Variance (σ^2)	725.3	6,260.6	3,236.1	810.9
	SD (σ)	26.9	79.1	56.9	28.5
	Variance (%)	6.6	56.7	29.3	7.4
4	Variance (σ^2)	1,093.7	9,254.9	5,985.6	1,327.7
	SD (σ)	33.0	96.2	77.4	36.4
	Variance (%)	6.2	52.4	33.9	7.5
5	Variance (σ^2)	1,191.9	13,307.8	8,152.7	2,502.9
	SD (σ)	34.5	115.4	90.3	50.0
	Variance (%)	4.7	52.9	32.4	9.9
6	Variance (σ^2)	1,163.5	6,559.1	3,333.0	277.9
	SD (σ)	34.1	80.9	57.7	16.7
	Variance (%)	10.3	57.9	29.4	2.5

may be explained by the variance between intervals (52.9–57.9%), and the second largest was the between-days variance of 29.3–33.9%. The contribution by between-vehicles and between-months variances ranged from 4.7 to 10.3% and 2.5 to 9.9%, respectively. The standard deviations of the within-intervals variance of four links ranged from 26.9 to 34.5 s. These results imply that (1) the major sources of the travel time variation during the peak hour come from the fluctuations between intervals and between days; and (2) using a real-time or historical profile may not be a reliable methodology to accurately predict average link travel times.

IMPLEMENTATION AND RESULTS ANALYSIS

Study Design of SNN

The proposed SNN was used to forecast travel times on link 4 for one through five time periods into the future. The sigmoid function was used as a neuron transfer function for the hidden layer(s) and a linear function was used for the output layer (Haykin 1994). The number of nuerons in the hidden layer depends on the pattern and complexity of the approximated function and the transfer function of the layers. In this paper the appropriate number of neurons was chosen through a preliminary analysis. The best SNN structure was identified after a sensitivity analysis involving the number of hidden layers and the number of neurons. To minimize the squared error during training, a steepest descent algorithm was incorporated. The SNN was entirely coded in FORTRAN and run on a 133-MHz Pentium desktop computer.

Intuitively, the average travel times of the target link in the preceding time periods are important parameters for identifying future travel time patterns. In this paper, travel time information from the previous five time periods was used, as

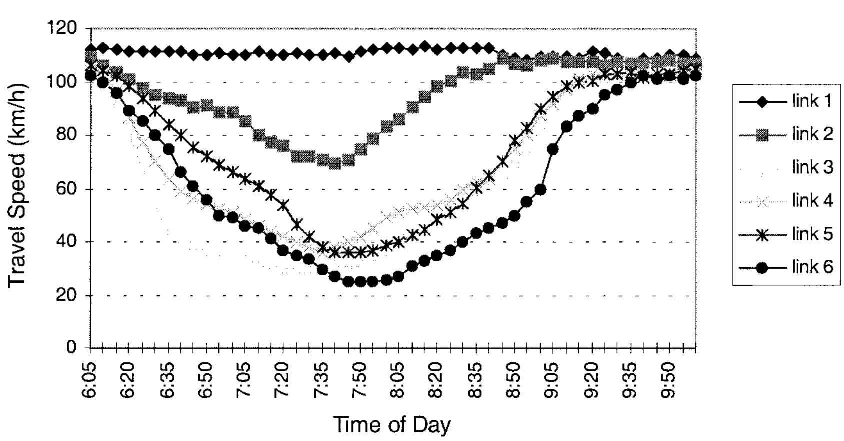

FIG. 4. Mean Travel Speed versus Time of Day

this range had given the best results in a previous study (Park and Rilett 1997). Another important parameter is the link travel times experienced on the upstream and downstream links during the preceding time periods. A shockwave that formed upstream or downstream from the target link has the potential to affect the target link some time in the future (May 1990). In this study, the link travel times of the links immediately upstream (link 3) and downstream (link 5) were chosen for inclusion based on previous results (Park and Rilett 1997). Given the nature of the AVI system, other important parameters such as volume and density were not available and therefore could not be used in the analysis. The output space consists of the link travel forecasts for the next five consecutive time periods. In summary, the SNN is designed to map the 15D input space to the 5D output space, written as $\mathbf{s}:\mathbb{R}^{15} \rightarrow \mathbb{R}^{5}$.

Of 231 weekdays in 1996, 131 days and 100 days were selected randomly for training and testing, respectively. The training ended when the error of the testing data set increases or 10,000 iterations were reached. The value of 0.01 was chosen as a learning rate. The mean absolute percent error (MAPE) described in (3) was used to measure model performance. It may be seen that the performance of the proposed SNN models for each time period (i.e., $1 \leq \delta \leq 5$) is based on the average error taken over n testing samples. The best model was chosen based on the average prediction error for all five time periods

$$\mathrm{MAPE}_{\delta} = \frac{\displaystyle\sum_{k=1}^{n} \frac{|tt(k+\delta)_p - tt(k+\delta)_o|}{tt(k+\delta)_o} \times 100}{n}, \quad \delta = 1, 2, \ldots, 5 \tag{3}$$

where MAPE_{δ} = mean absolute percent error for δ time period(s) ahead forecasting (%); k = time interval of the kth sample; n = number of testing samples (4,800 = 100 days × 48 intervals a day); $tt(k+\delta)_p$ = predicted link travel time for δ time period(s) ahead at the kth time interval (s); and $tt(k+\delta)_o$ = observed link travel time for δ time period(s) ahead at the kth time interval (s).

SNN Implementation with One Hidden Layer

The proposed SNN was first run with a single hidden layer, and these models are denoted by SN1L. To examine the performance of the SN1L, the number of expanded input nodes was varied from 1 (i.e., a conventional ANN or SN1L_E1 in this paper) to 15 (i.e., SN1L_E15) in increments of 2. Additionally the number of neurons in the hidden layer for each model was varied from 3 to 30. The best number of expanded nodes and hidden neurons were decided based on the performances of the proposed models.

The prediction results from the eight SN1Ls are shown in Fig. 5 where the MAPE is shown on the Y-axis for each of the eight SNNs. Note that the 1 on the X-axis denotes the SN1L_E1 model, the 3 denotes the SN1L_E3 model, etc. Although the MAPE from the conventional ANN (SN1L_E1) with seven neurons for the hidden layer were 8.7, 12.1, 14.5, 16.2, and 18.3% (with overall average of 14.0%) for one through five time periods ahead predictions, respectively, the SNN with seven expanded input nodes (SN1L_E7) and 15 neurons for the hidden layer gave the best results. The MAPE for one through five time periods were 7.2, 9.7, 12.2, 14.1, and 15.7%, respectively, and the overall average MAPE was 11.8%. This result corresponds to an approximately 17, 20, 16, 13, and 14% improvement as compared with the conventional ANN (SN1L_E1) for one through five time periods predictions, respectively.

The fact that the MAPE forms a convex function as the number of expanded input nodes increases indicates that a threshold point exists after which increased expanded input nodes decrease the solution quality. This behavior is attributed to the fact that as the number of expanded input nodes increases, the capacity of the neural network increases as well. The enhanced network capacity makes it possible for the network to capture more easily the highly nonlinear pattern of the training data. However, once the threshold value of the number of expanded input nodes is passed, the capacity is such that the network is susceptible to being overtrained.

The absolute error (AE) for the δ time period(s) ahead at the kth time interval is defined in (4) and the mean absolute percent (MAE) is defined in (5). Following standard practice, the output or target values of the training and testing data set are normalized to values between 0.1 and 0.9

$$\mathrm{AE}_{\delta}^{k} = |ntt(k+\delta)_p - ntt(k+\delta)_o| \tag{4}$$

$$\mathrm{MAE} = \frac{\displaystyle\sum_{k=1}^{n}\sum_{\delta=1}^{5} \mathrm{AE}_{\delta}^{k}}{5n} \tag{5}$$

where $ntt(k+\delta)_p$ = normalized predicted travel time for δ time period(s) ahead at the kth time interval; $ntt(k+\delta)_o$ = normalized observed travel time for δ time period(s) ahead at the kth time interval; k = time interval of the kth sample; n = number of samples (4,800 = 100 days × 48 intervals for testing, and 6,288 = 131 days × 48 intervals for training); AE_{δ}^{k} = absolute error for the δ time period(s) ahead at the kth time interval; and MAE = mean absolute percent. It was found that as the number of expanded input nodes increases, the MAE of the training and testing data set decreased to a minimum at seven and then increased again. As expected, these patterns were similar to the relationship between the MAPE and the

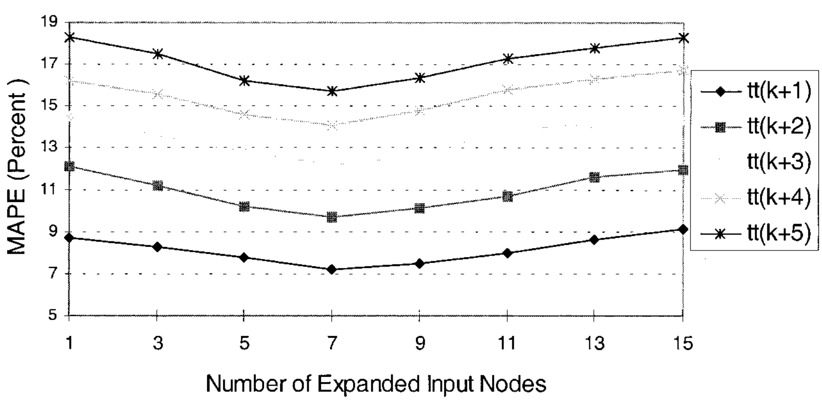

FIG. 5. Travel Time Forecasting Errors with Proposed SNN (SN1L)

number of expanded input nodes shown in Fig. 5. In other words, as the number of expanded input nodes increases, the generalization ability of the SNN increases and then decreases after a threshold.

To summarize, in contrast to the theory of Hornik et al. (1989), the conventional ANN with a single hidden layer may not be a satisfactory universal function approximator in practice. This is particularly true when the function approximated is highly nonlinear and multidimensional, which are typical of the problems experienced in many transportation applications.

In addition to the generalization ability (i.e., model accuracy or capacity) of the model, there are secondary factors, such as time efficiency, memory size, and complexity, that need to be considered when selecting ANN models for a specific problem (Musavi et al. 1994). The average number of iterations and the CPU time of the five implementations of the best conventional ANN (SN1L_1E) were 3,257 and 140 min, respectively), whereas those of the best SNN (SN1L_7E) were 715 and 159 min, respectively. As expected, the SNN requires marginally more CPU time and significantly fewer iterations. These results may be attributed to the fact that the SNN starts the function approximation (i.e., mapping) with more linearly separable input features (i.e., travel time patterns), and therefore the highly nonlinear patterns of the link travel time dynamics can be classified with fewer iterations. However, because of the expansion of the input features, the SNN requires more computation time to handle the increased number of inputs (15 versus 15 times 7 expansion) and increased number of neurons in the hidden layer (7 versus 15). It was found that the SNN reaches the near optimal point (e.g., 0.025 MAE for the testing data set) in <50 iterations, whereas the conventional ANN required >1,500 iterations.

SNN Implementation with Two Hidden Layers

It has been found in practice that ANN capacity increases as the number of hidden layers increases (Haykin 1994; Hagan et al. 1995). Therefore, the proposed SNN architecture was run using two hidden layers. Similar to the ANN analysis with one hidden layer (NN1L or SN1L_E1), the number of expanded input nodes was varied from 1 (SN2L_E1) to 15 (SN2L_E9) and the number of neurons for the two hidden layers was varied from 3 to 15.

The best conventional ANN with two hidden layers (NN2L or SN2L_E1) had five neurons in both the first and second hidden layers, and the MAPE were 8.7, 10.8, 13.4, 15.3, and 16.8% for the one through five time periods ahead predictions, respectively. This corresponds to an average MAPE of 13% over all time periods. As expected, the two hidden layer ANN gave better results when compared with the single layer ANN but inferior performance with respect to the best SNN with either one or two hidden layers. The average number of iterations and the CPU time were 2,635 and 157 min, respectively. As expected, the conventional ANN with two hidden layers requires less iteration and more CPU time compared with the conventional ANN with one hidden layer. Note that the CPU time of the conventional ANN with two hidden layers is similar to that of the proposed SNN. In addition, the best network structure of the SNN with two hidden layers had three expanded input nodes (i.e., SN2L_E3), and nine and five neurons for the first and second layers, respectively. The average MAPE of this model was 12.0%, and the MAPE for the one through five time periods predictions were 7.4, 9.9, 12.3, 14.4, and 15.9%, respectively. These results are approximately equivalent to that obtained from the SN1L_7 model.

The important point to note regarding the above analyses is that the best number of expanded input nodes of SN2L was three whereas that of the SN1L was seven. This result implies that, as discussed earlier, the nonlinear transformation of the input space corresponds to the addition of hidden layers. If more than two hidden layers are incorporated into the conventional ANN, it may be possible to obtain the same level of network generalization ability as the SNN; however, training such a conventional ANN may require significantly increased effort and time for selecting the best ANN architecture.

Comparison with MNN

As stated in Section 3, the motivation behind MNN is similar to that of the SNN, although the methodology employed to enhance the conventional ANN is different. Therefore, it is worthwhile (1) to compare the result from the SNN with that of MNN; and (2) to examine the result from combining these two methods for the same problem. It was found that the overall average MAPE of the MNN that employed a fuzzy c-means clustering technique with a conventional single hidden layer ANN was 12.0% (Park and Rilett 1998). This level of accuracy is approximately the same as that identified by the best SNN.

To identify the effect of combining these two networks, an SN1L was implemented with the MNN. The best network structure of the combined network was found to have three expanded input nodes, eight clusters, and 11 neurons for the hidden layer. The combined network gave the average MAPE of 11.9%, which is a similar forecasting error compared with the SNN and/or MNN. All three neural network parameters (i.e., the numbers of classes, expansions, and neurons) were found to be less than those for the SNN and MNN. This result is not surprising because the combined network utilizes input transformation and input partition at the same time. It should be noted, however, that although the SNN and MNN showed almost the same prediction accuracy for the given problem, the MNN needs significantly more effort as discussed in Section 2.

COMPARISON WITH OTHER TRAVEL TIME FORECASTING MODELS

Although it is important to identify the best SNN configuration and compare it with other neural network approaches, it is equally worthwhile to compare the best SNN design with other standard prediction models. In this paper the best SNN (SN1L_E7) was compared with a Kalman filtering model, a historical profile, a real-time profile, and an exponential smoothing model. The historical profile technique simply forecasts the future travel time based on the historical average. Conversely, the real-time profile assumed that the current link travel time would continue in the future. The Kalman filtering and exponential smoothing models predicted link travel times using a recursive formula. For the exponential smoothing model, the smoothing coefficient was identified through a calibration process. The results are shown in Table 2.

The average MAPE of SN1L_E7 for all forecasting periods was 11.8%, whereas those of the Kalman filtering, exponential

TABLE 2. Travel Time Prediction Comparisons of SNN1 with Other Models

Model (1)	MAPE (%) Average (2)	$tt(k+1)$ (3)	$tt(k+2)$ (4)	$tt(k+3)$ (5)	$tt(k+4)$ (6)	$tt(k+5)$ (7)
Conventional ANN	14.0	8.7	12.1	14.5	16.2	18.3
SNN: SNN1[7]	**11.8**	7.2	**9.7**	**12.2**	**14.1**	**15.7**
Historical profile	24.4	24.4	24.4	24.4	24.4	24.4
Real-time profile	18.4	9.0	14.1	18.8	23.2	27.0
Exponential smoothing	16.7	8.3	12.9	17.1	20.9	24.3
Kalman filtering	16.1	**6.2**	12.1	17.0	20.8	24.5

Note: Boldface values represent model that gave best results.

smoothing, historical profile, and real-time profile were 16.1, 16.7, 18.4, and 24.4%, respectively. As would be expected the MAPE of the historical profile are the same for all prediction periods, whereas the MAPE from other models increases as the prediction time periods increase. It is interesting to note that the historical profile gave the worst results for all prediction levels. Similarly, simply using the real-time data never resulted in the best solution for any situation. That is, simply assuming that the characteristics of the current time period will continue into the near future, or using historical averages, is not as applicable for RGS as it may be for other prediction techniques. Given the variability in the data discussed earlier, these results would be expected.

The SN1L_E7 gave the best results for predicting two through five time periods ahead, whereas the Kalman filtering model gave superior results for predicting one time period into the future. The Kalman filtering model and exponential smoothing showed similar MAPE for forecasting two through five time periods ahead. The difference in prediction accuracy between the best SNN (SN1L_E7) and the other models was found to increase as the prediction period increases. It should be pointed out that the MAPE for forecasting five time periods ahead from the Kalman filtering, exponential smoothing, and real-time profile were either very similar or worse when compared with that from the historical profile. This result means that for long-term predictions (e.g., >25 min), the historical average may be better than some of the standard forecasting techniques. Note that whereas the Kalman filtering minimizes the error for one time period ahead forecasting, the SNN minimizes the overall errors (i.e., one through five time periods). If the SNN was developed for forecasting one time period ahead only, the absolute percent error was found to be 6.7%, which is similar to that of the Kalman filtering approach.

It is hypothesized that the superior performance of the SNN for forecasting travel times is related to the neural network's structure that (1) considers the real-time travel time dynamics and the historical travel time dynamics simultaneously by recognizing travel time patterns present in the historical data and therefore is well suited to multiple periods forecasting; (2) recognizes the travel time dynamics between neighboring links; and (3) is nonlinear. It is also important to note that although the other models use less input data, the data collection requirements of the SNN would not increase beyond those that are currently being performed.

CONCLUDING REMARKS

This paper proposed SNN for forecasting multiple periods freeway link travel times. The basic idea is to convert a nonmonotonic function approximation problem into a linearly separable classification problem. Actual link travel times from Houston that were collected as part of the AVI system of the Houston Transtar system were used as a test bed. It was found that the SNN gave superior results when compared with the conventional ANN and similar results to the modular ANN when forecasting one through five time periods ahead. The MNN, however, requires significant additional effort on the part of the modeler for clustering design. The best SNN was subsequently compared with existing travel time prediction models, and it was found that the proposed SNN gave the best results. It appears that the travel time patterns on US 290 exhibit a high degree of nonlinearity and that the nonlinear input feature transformation scheme of the proposed SNN approach is the appropriate technique in these situations.

The proposed SNN has significant advantages over the conventional ANN because it increases the network capacity (i.e., better function approximation accuracy), particularly when the function approximated is highly complex and nonlinear. However, the SNN requires that the number of expanded input nodes should be properly calibrated. It was also demonstrated that when an improper number of expanded input nodes is chosen, the generalization ability of the SNN may be compromised. It also should be noted that the technique requires larger memory size for the input feature transformation compared with a conventional one hidden layer ANN.

Although the results of the proposed SNN approach are promising, a number of issues still need to be resolved before the SNN can be implemented. This study was based on a 5-min aggregation level. Obviously, a sensitivity analysis is required to examine the optimal level of aggregation. For example, a 10- or 15-min aggregation level may provide more adequate results with a corresponding reduction in computer resources. It would also be interesting to look at a more fundamental issue pertaining to whether the data should be aggregated at all. Incorporating other information such as traffic flow, density, and occupancy, in addition to the AVI-based travel times, might improve the quality of the ANN. In addition, combining SNN with other ANN structures such as a finite-impulse response network (Wan 1994), nonlinear autoregressive moving average network (Narendra and Parthasarathy 1990), real-time recurrent network (Williams and Zipser 1989) may lead to better performance. The spatial and temporal transferability of the neural network models also should be studied before the neural networks can be applied successfully.

Finally, it should be noted that the forecasting of link travel times is not an end in itself but is rather an input to other transportation applications. For example, in RGS the goal is to forecast the route travel time, and therefore the link forecasting method chosen should reflect this purpose (Rilett and Park 1999). It is easy to hypothesize that the variance of the link travel time and the covariance of the link travel times also should be identified (Fu and Rilett 1998).

ACKNOWLEDGMENTS

Support for this research was provided by a Federal Highway Administration cooperative agreement to the Texas A&M Intelligent Transportation Systems Research Center of Excellence, College Station, Tex., and is gratefully acknowledged. The writers also would like to thank Shawn Turner and Darrell Borchardt of the Texas Transportation Institute, College Station, Tex., and the Texas Department of Transportation for help in collecting the AVI data used in this paper.

APPENDIX I. REFERENCES

Anand, R., Mehrotra, K., Mohan, C. K., and Ranka, S. (1995). ''Efficient classification for multiclass problems using modular neural networks.'' *IEEE Trans. on Neural Networks*, 6(1), 117–124.

Anderson, J. M., Bell, M. G. H., Sayers, T. M., Busch, F. M., and Heymann, G. (1994). ''The short-term prediction of link travel times in signal controlled road networks.'' *Proc., Transp. Sys.: Theory and Application of Advanced Technol., IFAC Symp.*, 621–626.

Boyce, D., Rouphail, N., and Kirson, A. (1993). ''Estimation and measurement of link travel times in the ADVANCE project.'' *Proc., IEEE-IEE Vehicle Navigation and Information Sys. Conf.*, 62–66.

Chang, G.-L., and Su, C.-C. (1995). ''Prediction intersection queue with neural network models.'' *Transp. Res.*, Part C, 3(3), 175–191.

Chang, J. (1996). ''A hierarchical structure of neural network for the complicated data set classification problems,'' PhD dissertation, Dept. of Electr. Engrg., Texas A&M University, College Station, Tex.

Chen, K., and Underwood, S. E. (1991). ''Research on anticipatory route guidance.'' *Proc., 2nd Vehicle Information and Navigation Sys. Conf.*, Vol. 1, Society of Automatic Engineering, Dearborn, Mich., 427–440.

Dailey, D. J. (1993). ''Travel-time estimation using cross-correlation techniques.'' *Transp. Res.*, Part B, 27(2), 97–107.

Davis, G. A., and Nihan, N. L. (1991). ''Nonparametric regression and short-term freeway traffic forecasting.'' *J. Transp. Engrg.*, ASCE, 117(2), 178–188.

Dougherty, M. S. (1993). ''A review of neural networks applied to transport.'' *Transp. Res.*, Part C, 2(4), 247–260.

Dougherty, M. S., and Kirby, H. R. (1993). ''The use of neural networks to recognize and predict traffic congestion.'' *Traffic Engrg. and Control*, 34(2), 311–314.

Eck, J. T., and Shih, F. Y. (1994). ''An automatic text-free speaker recognition system based on an enhanced ART 2 neural architecture.'' *Information Sci.*, 76, 233–253.
Ersoy, O. K., and Hong, D. (1990). ''Parallel, self-organizing, hierarchical neural networks.'' *IEEE Trans. on Neural Networks*, 1(2), 167–178.
Florio, L., and Mussone, L. (1994). ''Neural network models for classification and forecasting of freeway traffic flow stability.'' *Proc., Transp. Sys.: Theory and Application of Advanced Technol., IFAC Symp.*, 773–784.
Fu, L., and Rilett, L. R. (1998). ''Expected shortest paths in dynamic and stochastic traffic networks.'' *Transp. Res.*, Part B, 32(7), 499–511.
Gilmore, J. G., and Abe, N. (1995). ''Neural network models for traffic control and congestion prediction.'' *IVHS J.*, 2(3), 231–252.
Hagan, M. T., Demuth, H. B., and Beale, M. (1995). *Neural network designs.* PWS Publishing Co., Boston.
Han, G. (1997). ''Function approximation by multi-layer perceptron network with expanded input nodes.'' *Working Paper*, Dept. of Electr. Engrg., Texas A&M University, College Station, Tex.
Haykin, S. (1994). *Neural networks: A comprehensive foundation*. Prentice-Hall, Upper Saddle River, N.J.
Hoffman, C., and Janko, J. (1990). ''Travel time as a basic of the LISB guidance strategy.'' *Proc., IEEE Road Traffic Control Conf.*, 6–10.
Horikawa, S., Furuhashi, T., and Uchikawa, Y. (1992). ''On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm.'' *IEEE Trans. on Neural Networks*, 3(5), 801–806.
Hornik, K. M., Stinchcombe, M., and White, H. (1989). ''Multilayer feedforward networks are universal approximators.'' *Neural Networks*, 2(5), 359–366.
Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). ''Adaptive mixtures of local experts.'' *Neural Computation*, 3, 79–87.
Lu, J. (1992). ''Spectral analysis of vehicle speed characteristics.'' *Transp. Res. Rec. 1375*, Transportation Research Board, 26–36.
May, A. D. (1990). *Traffic flow fundamentals*. Prentice-Hall, Englewood Cliffs, N.J.
Musavi, M. T., Chan, K. H., and Kalantri, K. (1994). ''On the generalization ability of neural network classifiers.'' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6), 659–663.
Narendra, K. S., and Parthasarathy, K. (1990). ''Identification and control of dynamical systems using neural networks.'' *IEEE Trans. on Neural Networks*, 1(1), 4–27.
Nicholson, H., and Swann, C. D. (1974). ''The prediction of traffic flow volumes based on spectral analysis.'' *Transp. Res.*, 8, 533–538.
Okutani, I., and Stephanedes, Y. J. (1984). ''Dynamic prediction of traffic volume through Kalman filtering theory.'' *Transp. Res.*, Part B, 18(1), 1–11.
Pao, Y.-H. (1989). *Adaptive pattern recognition and neural networks*. Addison-Wesley, New York.
Park, D., and Rilett, L. R. (1998). ''Forecasting multiple-period freeway link travel times using modular neural networks.'' *Transp. Res. Rec. 1617*, Transportation Research Board, 63–70.
Park, D., and Rilett, L. R. (1999). ''Forecasting freeway link travel times with a feedforward multilayer neural network.'' *Comp.-Aided Civ. and Infrastruct. Engrg.*, 14, 357–367.
Park, D., Rilett, L. R., and Han, G. (1998). ''Forecasting multiple-period freeway link travel times using neural networks with expanded input nodes.'' *Proc., 5th Int. Conf. of Advanced Technol. Application in Transp. Engrg.*, 325–332.
Poddar, P., and Rao, P. V. S. (1993). ''Hierarchical ensemble of neural networks.'' *Proc., IEEE Int. Conf. on Neural Networks*, 287–292.
Pushkar, A., Hall, F. L., and Acha-Daza, J. A. (1995). ''Estimation of speeds from single-loop freeway flow and occupancy data using cusp catastrophe theory model.'' *Transp. Res. Rec. 1457*, Transportation Research Board, 149–157.
Rilett, L. R. (1992). ''Modeling of TravTek's dynamic route guidance logic using the INTEGRATION model,'' PhD dissertation, Queen's University at Kingston, Kingston, Ont., Canada.
Rilett, L. R., and Park, D. (1999). ''Direct forecasting of freeway corridor travel times using spectral basis neural networks.'' *Proc., 78th Transp. Res. Board Annu. Meeting*, Transportation Research Board, Washington, D.C.
Rilett, L. R., and Van Aerde, M. (1991). ''Routing based on anticipated travel times.'' *Proc., ASCE 2nd Int. Conf. on Applications of Advanced Technologies in Transp. Engrg.*, ASCE, New York, 183–187.
Ritchie, S. G., and Cheu, R. L. (1993). ''Simulation of freeway incident detection using artificial neural networks.'' *Transp. Res.*, Part C, 1, 203–217.
Schmidt, W. F. (1993). ''Initialization, backpropagation and generalization of feed-forward classifier.'' *Proc., Int. Conf. on Neural Networks*, 598–604.
Sisiopiku, V. P., and Rouphail, N. M. (1995). ''Toward the use of detector output for arterial link travel time estimation: A literature review.'' *Transp. Res. Rec. 1457*, Transportation Research Board, 158–165.
Smith, B. L., and Demetsky, M. J. (1994). ''Short-term traffic flow prediction: Neural network approach.'' *Transp. Res. Rec. 1453*, Transportation Research Board, 98–104.
Tarko, A., and Rouphail, N. M. (1993). ''Travel time data fusion in ADVANCE.'' *Proc., ASCE 3rd Int. Conf. on Applications of Advanced Technologies in Transp. Engrg.*, ASCE, New York, 36–42.
Van Arem, B., Van Der Vlist, M. J. M., Muste, M. R., and Smulders, S. A. (1997). ''Travel time estimation in the GERDIEN project.'' *Int. J. Forecasting*, 13, 73–85.
Van Der Voort, M., Dougherty, M., and Watson, S. (1996). ''Combing Kohonen maps with ARIMA time series models to forecast traffic flow.'' *Transp. Res.*, Part C, 4(5), 307–318.
Vythoulkas, P. C. (1993). ''Alternative approaches to short term traffic forecasting for use in driver information systems.'' *Proc., 12th Int. Symp. on Transp. and Traffic Theory*, C. F. Daganzo, ed., Elsevier Science, New York, 495–506.
Wan, E. A. (1994). ''Time series prediction by using a connectionist network with internal delay lines.'' *Time series prediction: Forecasting the future and understanding the past*, A. S. Weigend and N. A. Gershenfeld, eds., Addison-Wesley, Reading, Mass., 195–217.
Williams, R. J., and Zipser, D. (1989). ''A learning algorithm for continually running fully recurrent neural networks.'' *Neural Computation*, 1, 270–280.

APPENDIX II. NOTATION

The following symbols are used in this paper:

AE_{δ}^{k} = absolute error for δ time period(s) ahead at kth time interval;
k = time interval of kth sample;
MAE = mean absolute percent;
$MAPE_{\delta}$ = mean absolute percent error for δ time period(s) ahead forecasting;
n = number of testing samples;
$ntt(k + \delta)_o$ = normalized observed travel time for δ time period(s) ahead at kth time interval;
$ntt(k + \delta)_p$ = normalized predicted travel time for δ time period(s) ahead at kth time interval;
r = number of node expansions;
t = time of day;
$tt(k + \delta)_o$ = observed link travel time for δ time period(s) ahead at kth time interval (s);
$tt(k + \delta)_p$ = predicted link travel time for δ time period(s) ahead at kth time interval (s);
x = input value (scalar);
$\mathbf{x}$ = input value (vector);
x_i = ith input value with nonlinear transformation;
$\mathbf{x}(r)$ = input vector with r node expansions;
y = output value; and
Δt = aggregation interval size.

Subscripts

i = input unit index;
o = observation index;
p = prediction index; and
δ = time period index.