Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Engineering Applications of Artificial Intelligence 16 (2003) 607–613
Engineering Applications of
ARTIFICIAL INTELLIGENCE
www.elsevier.com/locate/engappai

# Neural network model for rapid forecasting of freeway link travel time

Abhijit Dharia, Hojjat Adeli*

*Departments of Biomedical Informatics, Civil and Environmental Engineering and Geodetic Science, and Neuroscience, The Ohio State University, 470 Hitchcock Hall, 2070 Neil Avenue, Columbus, OH 43210, USA*

Accepted 11 September 2003

**Abstract**

Estimation of freeway travel time with reasonable accuracy is essential for successful implementation of an advanced traveler information system (ATIS) for use in an intelligent transportation system (ITS). An ATIS consists of a route guiding system that recommends the most suitable route based on the traveler's requirements using the information gathered from various sources such as loop detectors and probe vehicles. This information can be disseminated through mass media or on on-board satellite-based navigational system. Based on the estimated travel times for various routes, the traveler can make a route choice. In this article, a neural network model is presented for forecasting the freeway link travel time using the counter propagation neural (CPN) network. The performance of the model is compared with a recently reported freeway link travel forecasting model using the backpropagation (BP) neural network algorithm. It is shown that the new model based on the CPN network, and the learning coefficients proposed by Adeli and Park, is nearly two orders of magnitude faster than the BP network. As such, the proposed freeway link travel-forecasting model is particularly suitable for real-time advanced travel information and management systems.


*Keywords:* Backpropagation; Counter propagation; Intelligent transportation systems; Neural networks; Traffic engineering

## 1. Introduction

Recent economic and environmental concerns have placed the focus on efficient and intelligent utilization of the existing transportation infrastructure, rather than adding to it. Traditional models of traffic congestion and management lack the adaptability and sophistication needed to effectively and reliably deal with increasing traffic volume on the freeway. Intelligent transportation systems are intended to provide a high level of automation in the freeway system through the use of advanced and adaptive technologies and implementation of advanced traffic management systems (ATMS) and advanced traveler information systems (ATIS).

Estimation of freeway travel time with reasonable accuracy is essential for successful implementation of an ATIS for use in an intelligent transportation system (ITS). An ATIS consists of a route guiding system (RGS) that recommends the most suitable route based on the traveler's requirements, using the information gathered from various sources such as loop detectors and probe vehicles. This information can be disseminated through mass media, such as radio and the Internet, or on board satellite-based navigational system. Based on the estimated travel times for various routes, the traveler can make route choices.

The freeway network can be considered as consisting of a number of links. A link can be a portion of the freeway between upstream and downstream loop detectors. Logically, the link travel times used in the RGS should be estimated from the time the driver actually arrives at the initial point of the link. Hence, freeway link travel times forecasting must be done over multiple time steps or periods, particularly when the travel time on the link under consideration is relatively long. For such links, it is unlikely that the travel time will remain constant over a period of time. The success of an RGS, however, depends on its ability to predict the anticipatory link travel time in addition to the historical and real-time link travel time.

*Corresponding author.

0952-1976/$ - see front matter © 2003 Published by Elsevier Ltd.
doi:10.1016/j.engappai.2003.09.011

Traditionally, short-term freeway link travel time has been forecast by time series models (Nihan and Holmesland, 1980; Dailey, 1993; Van Arem et al., 1997), Kalman filtering model (Okutani and Stephanedes, 1984) and historical and real-time profiles (Boyce et al., 1993). These models are effective for predicting the travel time one time step ahead but deteriorate when the forecasting has to be done over multiple time steps (Park and Rilett, 1999).

Neural network computing appears to be a promising approach to overcome this shortcoming. A neural network provides a mapping between a set of inputs and corresponding outputs (Adeli and Hung, 1995). The network is trained to learn this mapping using a number of training examples. The training is achieved by determining the network's weights. A review of civil engineering applications of neural networks up to 2000 was presented by Adeli (2001). Recent research on transportation engineering applications of neural networks is presented by Adeli and Samant (2000), Adeli and Karim (2001), Samant and Adeli (2001), Karim and Adeli (2002, 2003a ,b), Adeli and Jiang (2003), Jiang and Adeli (2003), and Ghosh-Dastidar and Adeli (2003) among other.

Recently, Park and Rilett (1999) presented a multiplayer feedforward neural network for freeway link travel time forecasting using the backpropagation learning rule (Rumelhart et al., 1986) for training the network. They forecast up to five periods (time steps) into the future with a time step of 5 min. They have investigated various input-output combinations and concluded that *"when predicting three through five time periods into the future, the ANN models that employed spatial data in the form of link travel times on links immediately upstream and downstream from the target link gave the lowest error"* based on a 5-min time step. They compared this approach with a Kalman Filtering model, the ALI_SCOUT method (Hoffman and Janko, 1990), the historical travel time profile, the real-time travel time profile, and an exponential smoothing model (Chassakios and Stephanedes, 1994) and reported that, overall, the neural networks were 20% more accurate in the forecasting of freeway link travel time as compared to other models that they employed for the same purpose.

Backpropagation (BP) is the most widely used neural network model in civil engineering applications, primarily due to its simplicity (Arditi et al., 1998; Cattan and Mohammadi, 1997; Deo and Chaudhari, 1998; Owusu-Ababia, 1998; Thirumalaiah and Deo, 1998). However, backpropagation has shortcomings, including a very slow rate of convergence and arbitrary and problem-dependent selection of the learning and momentum ratios, as pointed out by Adeli and Hung (1994) and others.

In this article we present a neural network model for forecasting the freeway link travel time using the counter propagation neural (CPN) network and demonstrate its superiority over the backpropagation learning model.

## 2. Counterpropagation network

CPN employs a combination of supervised and unsupervised learning (Hecht-Nielsen, 1998; Adeli and Park, 1995, 1998). The topology of a CPN network consists of three layers: input, competition, and interpolation (Fig. 1). The training of a CPN network is carried out in two stages. In the first stage, the weights of the links connecting the input layer to the competition layer are computed using the Kohonen learning rule. Let $\boldsymbol{X}$ denote the input vector and $\boldsymbol{W}_j$ denote the weight vector for the links connecting the nodes in the input layer to the $j$th node in the competition layer. The Euclidean distance between the input vector and the weight vector corresponding to the $j$th node in the competition layer is given by

$$d_j = |\mathbf{W}_j - \mathbf{X}|. \tag{1}$$

For any given training instance, a competition is created among the nodes in the competition layer. The node with the smallest Euclidean distance wins. Inhibitory interlayer connections between the nodes in the

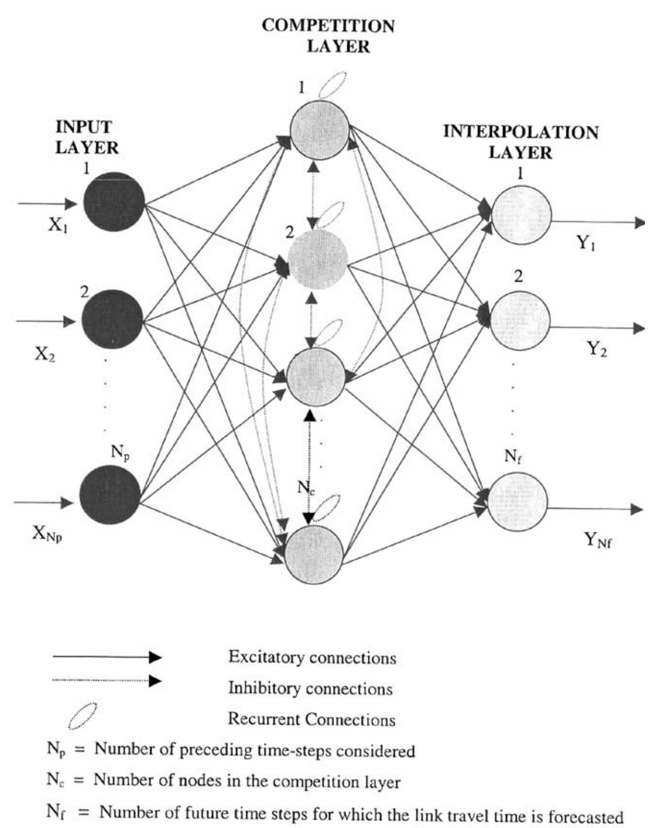


Fig. 1. Topology of counterpropagation neural network for freeway link travel time forecasting.

competition layer are used to conduct this competition and set the output of the winning node to 1 and all other nodes to 0. If the $k$th node is the winning node in the competition layer, then the outputs of the nodes in the competition layer are assigned as follows:

$$O_i = \begin{cases} 0 & \text{for all nodes except the winning nodes } (i \neq k), \\ 1 & \text{for the winning node } (i = k). \end{cases} \tag{2}$$

The weight $W_{ij}$ of the link connecting the input node to the node in the competition layer is computed from the Kohonen learning rule (Kohonen, 1988) as follows:

$$W_{ij}(n+1) = W_{ij}(n) + \alpha\left[X_i - W_{ij}(n)\right]O_i \tag{3}$$

where $n$ is the iteration number, $X_i$ is the input corresponding to the $i$th node in the input layer and $\alpha$ is the learning csoefficient. Hecht-Nielsen (1988) suggested a value in the range of 0 and 0.8 for the learning ratio. We use the following expression proposed by Adeli and Park (1995):

$$\alpha = \frac{1}{(n+1)^2}. \tag{4}$$

This expression provides several advantages. First, it circumvents the arbitrary and problem-dependent selection of the learning parameter. Second, it provides a variable value as a function of the iteration number. Third, it ensures that the weight changes are reduced after each iteration, thus stabilizing the weight computations and improving the convergence performance. This process is repeated for all the training instances. In other words, the values of weights are updated after addition of each successive training example.

It should be noted that in a CPN network only the weights of the links fanning out of the winning nodes are updated in every iteration (one winning node for each training example). This is in contrast to the back-propagation neural network, where all the weights are updated in every iteration. In a CPN network, a winning node for a particular training example is temporarily deactivated for all other input patterns and is not allowed to participate in the competition for other training instances in the current iteration. This idea is based on the so-called *conscience mechanism* originally proposed by DeSieno (1988) where each node can win only once in a single iteration of a particular training instance.

In the second stage of training a CPN network, the weights of the links connecting the competition layer to the output or interpolation layer are found using the Grossberg learning rule (Grossberg, 1982):

$$V_{jk}(n+1) = V_{jk}(n) + \beta\left[Y_k - V_{jk}(n)\right], \tag{5}$$

where $V_{jk}$ is the weight of link connecting the $j$th node in the competition layer to the $k$th node in the output layer, $Y_k$ is the output corresponding to the $k$th node in the output layer, and $\beta$ is the learning coefficient. We use the same Eq. (4), proposed by Adeli and Park (1995), to evaluate $\beta$.

To test the learning accuracy of the CPN network we define an error term in the following form:

$$E = \frac{1}{2}\sum_k [O_k - Y_k]^2, \tag{6}$$

where $Y_k$ and $O_k$ are the actual and the computed outputs, respectively.

## 3. Freeway link travel time forecasting using CPN

The topology of a CPN network for forecasting the freeway link travel time is shown in Fig. 1. The number of nodes in the input layer is equal to the number of preceding time intervals, $N_\mathrm{p}$. The number of nodes in the output layer is equal to the number of future time intervals, $N_\mathrm{f}$. We select the number of nodes in the competition layer, $N_\mathrm{c}$ to be equal to the number of training instances. We found that to be the minimum number of nodes needed in the hidden (competition) layer in order to obtain satisfactory results. Choosing a number greater than the number of training instances for the number of nodes in the competition layer increases the computational costs without any improvement in accuracy. As such, we have a logical way of selecting the number of nodes in the competition layer. This is in contrast to the BP algorithm where one cannot find a similar logical rule for selection of the number of nodes in the hidden layer.

## 4. Backpropagation network

For the sake of comparison, we have also implemented the BP training algorithm. The topology of the BP network is shown in Fig. 2. The numbers of nodes in the input and output layers are $N_\mathrm{p}$ and $N_\mathrm{f}$, respectively; the same as those of the CPN network shown in Fig. 1. Using a larger number of hidden nodes can potentially improve the accuracy and convergence of the BP algorithm at the cost of increasing the computational processing time. Park and Rillet (1999) use four nodes in the hidden layer while considering seven input time steps and five output time steps and report no significant improvement when a larger number of nodes is used. In this work, we also used four nodes in the hidden layer for a similar configuration for the sake of comparison.

The BP training rule is a steepest descent algorithm in the parlor of optimization (Adeli, 1994) in the following form:

$$W_{jk}(n+1) = W_{jk}(n) + \eta\left(\frac{\partial E}{\partial W_{jk}}\right) + \lambda\Delta W_{jk}, \tag{7}$$

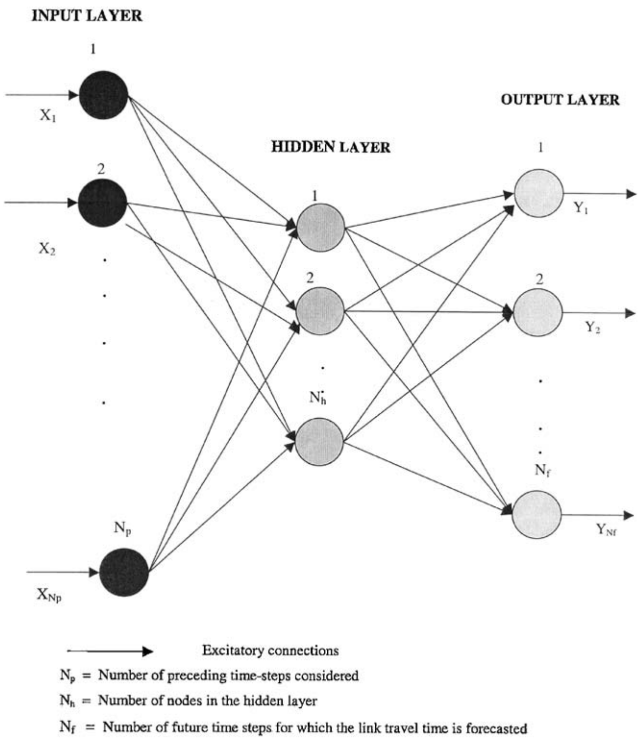


Fig. 2. Topology of backpropagation neural network for freeway link travel time forecasting.

where $W_{jk}(n)$ is the weight of a link connecting the $j$th node in the hidden layer to the $k$th node in the output layer at the end of the $n$th iteration, $E$ is the error term, $\eta$ is the learning ratio, $\lambda$ is the momentum ratio, and $\Delta W_{jk}$ is the change of weight during the last iteration. The error term $E$ is defined the same as Eq. (6), however, the computed output in the BP algorithm is expressed as follows:

$$Y_k = \sum g\left(\sum_j W_{jk} g\left(\sum_i W_{ij} X_i\right)\right), \tag{8}$$

where $X_i$ is the value of the $i$th input node and $g$ is the activation function defined by the following sigmoid function:

$$g(x) = \frac{1}{1+e^{-x}}. \tag{9}$$

Researchers have reported that using a large value for the learning ratio may result in convergence oscillation, while a small value may result in unsatisfactory learning (Adeli and Hung, 1995). The recommended value for the momentum ratio is in the range of zero to 1.0 (Hertz et al, 1991). After trying several different numbers, values of 0.8 and 0.5 were selected for the learning and momentum ratios, respectively, in this work.

## 5. Training the network

The data for this project was obtained from the simulation package TSIS (Traffic Software Integrated Systems, Version 4.21, developed by the Federal Highway Administration (FHWA) (http://www.fhwa-tsis.com/). TSIS is a versatile package for microscopic traffic simulation. At the heart of the TSIS package is the FHWA's microscopic traffic Corridor Simulation (CORSIM). CORSIM provides FRESIM (Freeway Simulation Model) for simulating freeway traffic. The TSIS environment has several attractive features, including an intuitive, user-friendly graphical interface, scrollable screen output, and on-line context-sensitive help that encompasses the CORSIM User's Guides. The package also comes with a user-friendly object-oriented graphics post-processor, TRAFVU (TRAF Visualization Utility).

Fig. 3 shows the geometry of a four-lane freeway (in each direction) simulated using the TSIS software. The numbers in the figure refer to the freeway nodes (locations of the loop detectors). TSIS software requires that all the entry and exit nodes be numbered in the range of 8000 and 8999. A sample of simulated travel time data for link 1–2 in Fig. 3, which is 5000 feet (0.95 miles or 1.5 km) long, for a 120-min period using 5-min time steps is shown in Fig. 4. To train the neural network several such simulations were performed.

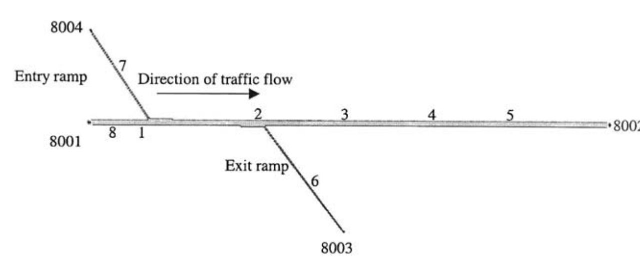


Fig. 3. Geometry of a simulated four-lane freeway.

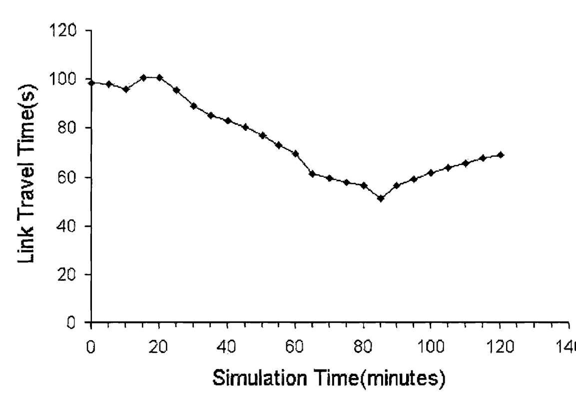


Fig. 4. Simulated link travel times for link 1–2 shown in Fig. 3 during a 120-min simulation period using 5-min time steps.

We will investigate the relation between the length of the forecasting time step and the average forecasting error. Four different cases are considered with the same number of time steps and total duration of 30 min for both input and travel forecasting times. In Case A, ten 3-min time steps were used as input and the link travel times are forecast for ten 3-min time steps into the future. In Case B, six 5-min time steps are used as input and the link travel times are forecast for six 5-min time steps into the future. In Case C, three 10-min time steps are used as input and the link travel times are forecast for three 10-min time steps. In Case D, two 15-min time steps are used as input and the link travel times are forecast for two 15-min time steps. The number of nodes in the input, competition, and interpolation layers for the neural networks used for these combinations are presented in Table 1.

Table 1
Number of nodes in the input, hidden (competition) and output (interpolation) layers for the CPN networks

| Model | Duration of time step (min) | No. of input nodes | No. of hidden nodes | No. of output nodes |
|---|---|---|---|---|
| A | 3 | 10 | 210 | 10 |
| B | 5 | 6 | 210 | 6 |
| C | 10 | 3 | 210 | 3 |
| D | 15 | 2 | 210 | 2 |

## 6. Training results

The CPN and BP models for freeway travel forecasting have been implemented in C++ on a Pentium 300 MHz computer. Using the TSIS simulation package, two hundred and ten training examples were generated for the freeway segment shown in Fig. 3 and used to train the CPN and BP networks.

The convergence results for training the BP network for the four cases A–D are presented in terms of normalized error (error divided by the largest error during the iterations) versus the number of iterations in Fig. 5. Similar results for the CPN network are presented in Fig. 6 (competition layer) and Fig. 7 (interpolation or output layer). In all cases, the same tolerance limit of 0.005 was used. Table 2 shows the training times for the two approaches on the Pentium 300 MHz machine. The superiority of the CPN network over the BP network are clearly demonstrated in Figs. 5–7 and in Table 2. The BP algorithm requires thousands of iterations versus only 5–17 iterations for the CPN algorithm. In terms of processing time, the CPN network takes 2.9–4.1 s versus 213–398 s for the BP

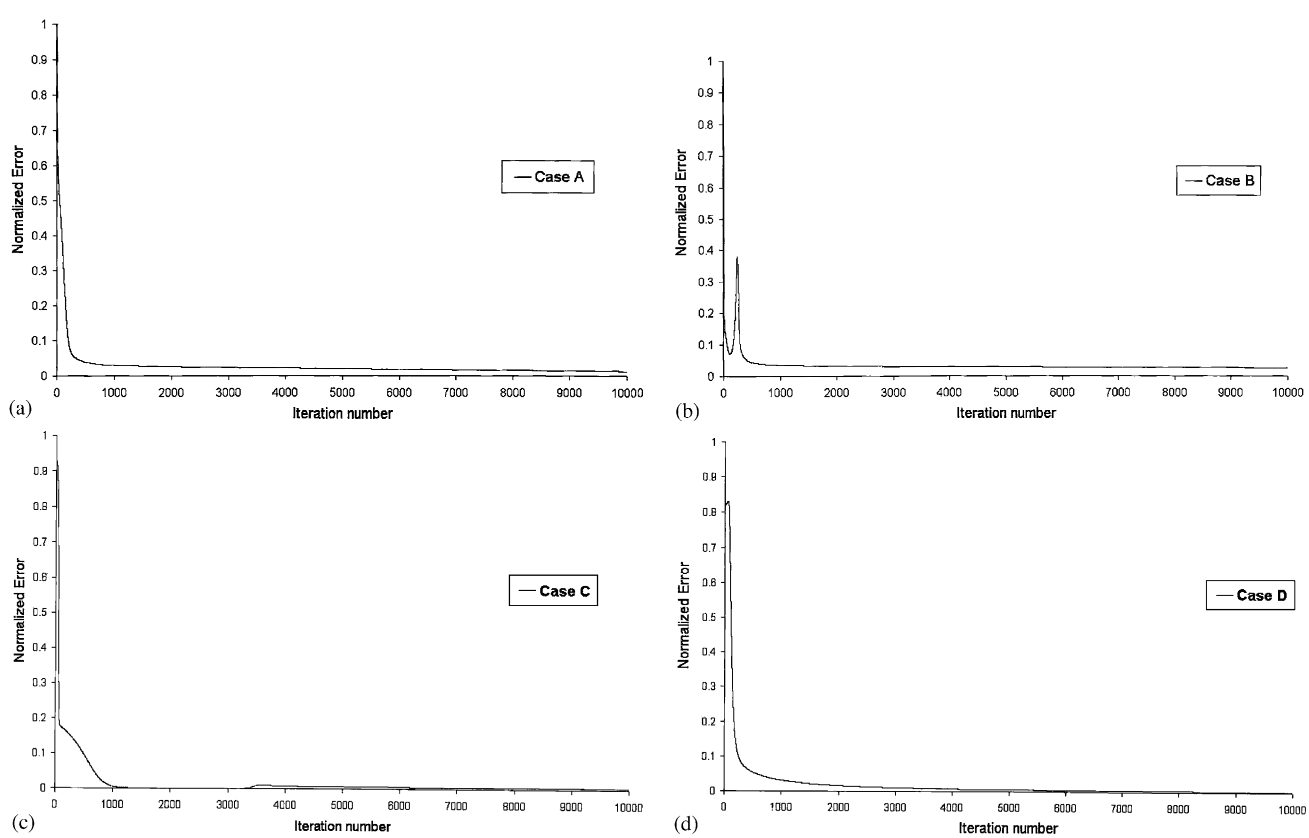


Fig. 5. Convergence curves for the BPN network.

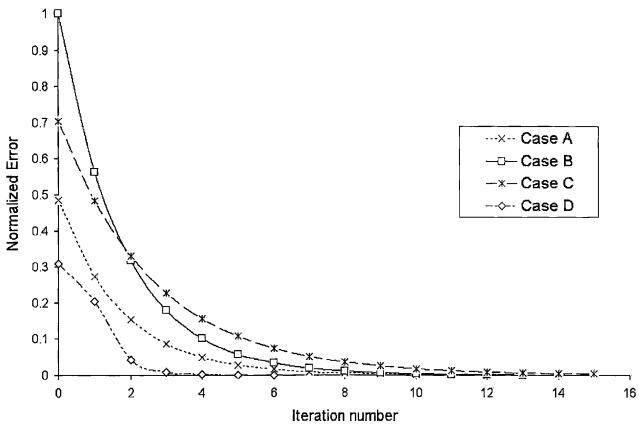


Fig. 6. Convergence curves for the competition layer of CPN network.

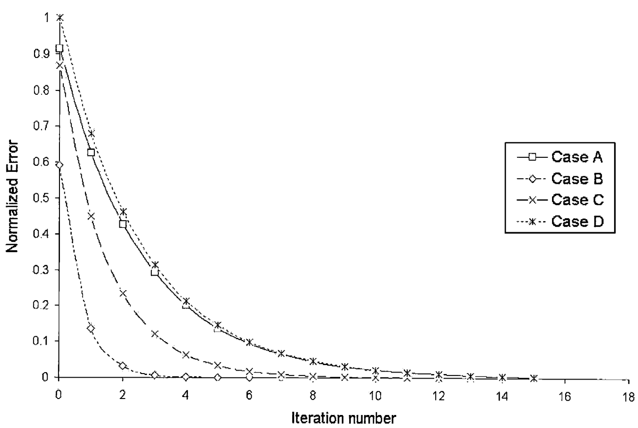


Fig. 7. Convergence curves for the interpolation layer of CPN network.

Table 2
Training time (s)

| Model | Backpropagation | Counterpropagation |
|---|---|---|
| A | 398.3 | 4.1 |
| B | 312.7 | 3.8 |
| C | 276.4 | 3.4 |
| D | 212.8 | 2.9 |

network to achieve the same level of accuracy. The CPN network is nearly two orders of magnitude more efficient than the BP network.

## 7. Forecasting results

In order to test the travel time forecasting capability of the CPN we generated fifty new sets of freeway traffic link travel times for each one of the Cases A–D using the TSIS simulation package. The link travel-time obtained from the simulation package is then compared with the computed travel times using the CPN and BP networks. The averages of the error percentages for Cases A–D are summarized in Table 3.

Table 3
Average error in forecasting time

| Duration of time step (min) | BPN Percentage error | CPN percentage error |
|---|---|---|
| 3 | 9.6 | 8.9 |
| 5 | 11.5 | 10.9 |
| 10 | 14.3 | 16.1 |
| 15 | 21.0 | 20.6 |

The verification results indicate that the CPN and BP models predict the freeway link travel times with similar accuracy. Table 3 shows that the smaller the forecasting time step, the smaller the error. In other words, the higher the resolution of the data provided to the neural network, the more accurate its prediction will be. It is interesting to note that the magnitude of the error increases with the magnitude of the forecasting time step roughly linearly. The average error for Case A with the smallest time step of 3 min is around 9%. Park and Rillet (1999) have reported prediction errors in the range of 12.5–23.4% using other methods, while predicting freeway link travel times 25 min into the future.

## 8. Conclusion

The BP training algorithm has been popular primarily because of its simplicity. In this paper we presented a CPN model and network with learning coefficients as proposed by Adeli and Park for forecasting the freeway link travel time and showed that it is nearly two orders of magnitudes faster than the BP training algorithm for the same level of accuracy.

The BP algorithm uses the steepest descent rule for minimization of the mean square error. Hence, the inherent entrapment pitfall of the steepest descent algorithm is also inherited by the BP algorithm. The BP algorithm is very sensitive to the choice of initial weights. It will converge to a local minimum in the vicinity of the initial solution (Fig. 8). Different initial weights will result in different local minima if more than one local minimum are present. Consequently, the convergence behavior of the BP algorithm is often non-smooth and jagged, as noted in Fig. 5. The convergence of the CPN algorithm, on the other hand, is very smooth, as noted in Figs. 6 and 7.

In the BP algorithm, the addition of any new pattern affects the weights of all the links; the same weight may be pulled in different directions by different training patterns. This results in excessive computational time for training the network. In the CPN algorithm, the effect of a particular training pattern is localized to the weight of its winning node only. Thus, computational time required for training is drastically reduced.

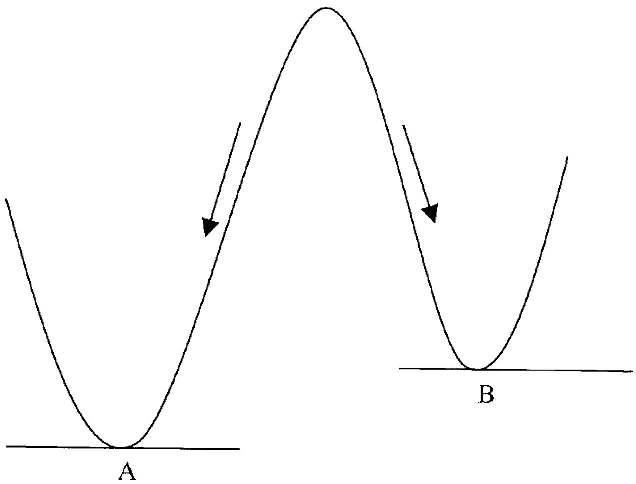


Fig. 8. Error minimization using BPN.

An appropriate topology for the BP algorithm, including the number of hidden layers and the number of nodes in the hidden layer, is selected by a trial-and-error process. Conversely, a CPN network always has three layers: input, competition, and interpolation. The number of nodes in the competition, or hidden layer, is governed by the number of training patterns presented to the network.

## References


Adeli, H. (Ed.), 1994. Advances in Design Optimization, Chapman & Hall, London.

Adeli, H., 2001. Neural networks in civil engineering: 1989–2000. Computer-Aided Civil and Infrastructure Engineering 16 (2), 126–142.

Adeli, H., Hung, S.L., 1994. An adaptive conjugate gradient learning algorithm for efficient training of neural networks. Applied Mathematics and Computation 62 (1), 81–100.

Adeli, H., Hung, S.L., 1995. Machine Learning-Neural Networks, Genetic Algorithms, and Fuzzy Systems. Wiley, New York.

Adeli, H., Jiang, X., 2003. Neuro-fuzzy logic model for freeway work zone capacity estimation. Journal of Transportation Engineering, ASCE 129 (5), 484–493.

Adeli, H., Karim, A., 2001. Fuzzy-wavelet RBFNN model for freeway incident detection. Journal of Transportation Engineering, ASCE 126 (6), 464–471.

Adeli, H., Park, H.S., 1995. Counterpropagation Neural Networks in Structural Engineering. Journal of Structural Engineering, ASCE 121 (8), 1205–1212.

Adeli, H., Park, H.S., 1998. Neurocomputing in Design Automation. CRC Press, Boca Raton, FL.

Adeli, H., Samant, A., 2000. An adaptive conjugate gradient neural network—wavelet model for traffic incident detection. Computer-Aided Civil and Infrastructure Engineering 15 (4), 251–260.

Arditi, D., Oksay, F.E., Tokdemir, O., 1998. Predicting the outcome of construction litigation using neural networks. Computer-Aided Civil and Infrastructure Engineering 13 (2), 75–81.

Boyce, D., Rouphail, N., Kirson, A., 1993. Estimation and measurement of link travel times in ADVANCE project. Proceedings of Vehicle Navigation and Information Systems Conference, IEEE, Ottawa, Ont. Canada, Oct 12–15, pp. 62–66.

Cattan, J., Mohammadi, J., 1997. Analysis of bridge condition rating using neural networks. Computer-Aided Civil and Infrastructure Engineering 12 (6), 419–429.

Chassakios, A.P., Stephanedes, Y.J., 1994. Smoothing algorithms for incident detection. Transportation Research Record 1453, 75–82.

Dailey, D.J., 1993. Travel-time estimation using cross-correlation techniques. Transportation Research Part B 27B (2), 97–107.

Deo, M.C., Chaudhari, G., 1998. Tide prediction using neural networks. Computer-Aided Civil and Infrastructure Engineering 13 (2), 113–120.

DeSieno, D., 1988. Adding a conscience to competitive learning. IEEE International Conference on Neural Networks, vol. 1, San Diego, July 24–27, pp. 117–224.

Ghosh-Dastidar, S., Adeli, H., 2003. Wavelet-clustering-neural network model for freeway incident detection. Computer-Aided Civil and Infrastructure Engineering 18 (5), 325–338.

Grossberg, S., 1982. Studies of Mind and Brain. Reidel Press, Boston.

Hecht-Nielsen, R., 1988. Application of counterpropagation networks. Neural Networks 1 (2), 131–139.

Hertz, J., Krogh, A., palmer, R., 1991. Introduction to Theory of Neural Computation. Addison Wesley, Redwood City.

Hoffman, C., Janko, J., 1990. Travel time as a basic of the LISB guidance strategy. Proceedings, IEEE Road Traffic Control Conference, IEEE 1990, pp. 6–10.

Jiang, X., Adeli, H., 2003. Freeway work zone traffic delay and cost optimization model. Journal of Transportation Engineering, ASCE 129 (3), 230–241.

Karim, A., Adeli, H., 2002. Comparison of the Fuzzy—wavelet RBFNN freeway incident detection model with the California algorithm. Journal of Transportation Engineering, ASCE 128 (1), 21–30.

Karim, A., Adeli, H., 2003a. Fast automatic incident detection on urban and rural freeways using wavelet energy algorithm. Journal of Transportation Engineering, ASCE 129 (1), 57–68.

Karim, A., Adeli, H., 2003b. Radial basis function neural network for work zone capacity and queue estimation. Journal of Transportation Engineering Journal of Transportation Engineering, ASCE 129 (5), 494–503.

Kohonen, T., 1988. Self-organization and Associative Memory. Springer, New York.

Nihan, N.L., Holmesland, K.O., 1980. Use of box and the jenkins times series techniques in traffic forecasting. Transportation 9 (2), 125–143.

Okutani, I., Stephanedes, Y.J., 1984. Dynamic prediction of traffic volumes through Kalman filtering theory. Transportation Research-Part B 18B, 1–11.

Owusu-Ababia, S., 1998. Effect of neural network topology on flexible pavement cracking prediction. Computer-Aided Civil and Infrastructure Engineering 13 (5), 349–355.

Park, D., Rilett, L.R., 1999. Forecasting freeway link travel times with a multilayer feedforward neural network. Computer-Aided Civil and Infrastructure Engineering 14 (5), 357–367.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representation by error propagation. In: Rumelhart, D.E., James L. McClelland and the PDP Research Group (Eds.), Parallel Distributed Processes: Explorations in the Microstructure of Cognition, MIT Press, Cambridge, pp. 318–362.

Samant, A., Adeli, H., 2001. Enhancing neural network incident detection algorithms using wavelets. Computer-Aided Civil and Infrastructure Engineering 16 (4), 239–245.

Thirumalaiah, K., Deo, M.C., 1998. Real time flow forecasting using neural networks. Computer-Aided Civil and Infrastructure Engineering 13 (2), 101–111.

Van Arem, B., Van Der Vlish, M.J.M., Muste, M.R., Smulders, S.A., 1997. Travel time estimation in the GERDIEN project. International Journal of Forecasting 13, 73–85.