Evaluation of Artificial Neural Network Applications in Transportation Engineering

ARDESHIR FAGHRI AND JIUYI HUA

The increased interest in artificial neural networks (ANNs) seen in government and private research as well as business and industry has included relatively little activity in transportation engineering. The position that ANNs, as a branch of artificial intelligence, hold in the transportation engineering field is discussed, including the differences between ANNs and biological neural networks and expert systems, respectively. The characteristics of ANNs in different fields are discussed and summarized, and their potential applications in transportation engineering are explored. A case study of trip generation forecasting using one traditional method and two ANN models is presented to show the application potential of ANNs in transportation engineering. The results of each method are compared and analyzed, and it is concluded that the potential for using ANNs to enhance both software and hardware in transportation engineering applications is high, even in comparison with expert systems and other types of artificial intelligence technique.

Artificial neural networks (ANNs) have proven to be an important development in a variety of problem solving areas. Increasing research activity in ANN applications has been accompanied by equally rapid growth in the commercial mainstream use of ANNs. However, there is relatively little research or practical application of ANNs taking place in the field of transportation engineering. This paper summarizes the characteristics of ANNs, evaluates the applicability of ANNs to transportation engineering, and explores the interface of ANN techniques and different transportation engineering problems.

DEFINITION OF ANNs

The human brain consists of 10 billion to 500 billion neurons. A cell body, an axon, and dendrites make up a biological neuron such as the one shown in Figure 1 (*top*). The connections between the neurons are called synapses, and each neuron is connected to 100 to 10,000 other neurons. A neuron executes a very simple task: when presented with a stimulus, it emits an output into other neurons connected to it via the synapses (*1*,*2*).

Artificial neurons (also called processing units or processing elements) mimic the functions of biological neurons by adding the inputs presented to them and computing the total value as an output with a transfer function. Figure 1 (*bottom*) shows a simple example of an artificial neuron. The artificial neuron also connects to other artificial neurons as the biological neuron does. The strength of the connections is called weight.

An ANN is a system composed of artificial neurons and artificial synapses that simulates the activities of the biological neural network. The ANNs can be single layer or multilayer, depending on their structure. In a single-layer ANN, all the processing units of the ANN take inputs from the outside of the network and their outputs go to the outside of the network; otherwise, it is a multilayer ANN. The principle that can approximately compute any reasonable function in an ANN is called architecture. The weights are adjustable, the programming for adjusting the weights is called training, and the training effect is called learning. The learning can be done either by being given weights computed from a set of training data or by automatically adjusting the weights according to some criterion.

A general definition for an ANN can be given as "a computing system made up of a number of simple, highly interconnected processing elements that process information by dynamic state response to external inputs" (*3*).

Differences Between ANNs and Biological Neural Networks

Although ANNs attempt to simulate real neural networks, they operate differently in many ways. The primary differences between ANNs and biological neural networks follow.

1. The processing speed is different. Cycle time is the time taken to process a single piece of information from input to output. The effective cycle time of a biological neuron is about 10 to 100 msec; the effective cycle time of an advanced computer's CPU is in the nanosecond range (*1*,*2*).
2. There are more than 100 kinds of biological neuron. ANNs contain only a few kinds of processing unit.
3. The "computations" (chemical reactions) in biological neual networks occur not only in neurons, but also in dendrites and synapses (*2*).
4. ANNs seldom have more than a few hundred processing units; a brain has 10 billion to 500 billion neurons.
5. The human brain has stronger error removal capability than an ANN. An upside-down letter may cause a lot of error in an ANN recognition system, but the human brain can recognize it easily.
6. Knowledge in ANNs is replaceable, but knowledge in biological neural networks is adaptable (*2*).

Department of Civil Engineering, University of Delaware, Newark, Del. 19716.

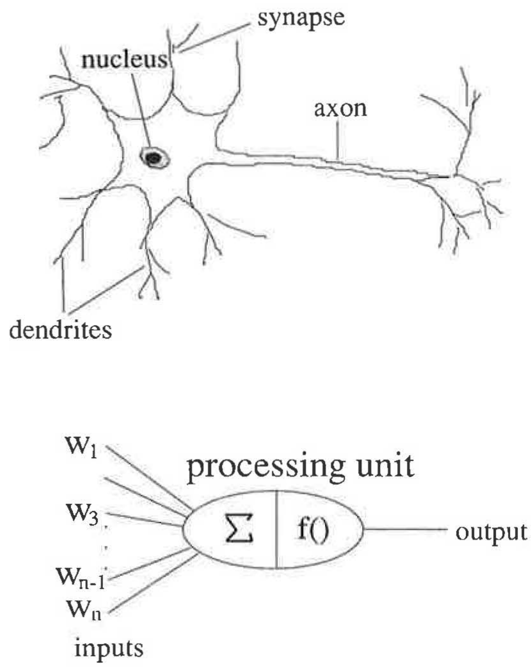

FIGURE 1 Schematic drawing of typical biological neuron and artificial neuron.

Differences Between ANNs and Expert Systems

As a part of artificial intelligence, ANNs possess some similarities with expert systems, such as storing knowledge and having learning processes in their operation. However, each has its own characteristics. The relationship between ANNs and expert systems is one not of replacement but of partnership. The differences between ANNs and expert systems follow.

1. ANNs and expert systems differ in the method of developing an intelligent system. The expert system approach uses a domain expert to identify the explicit heuristics used to solve problems, whereas the ANN approach assumes the problem-solving steps are to be derived without direct attention as to how a human actually performs the task. Thus, expert systems try to figure out how the human mind is working, and ANNs mimic the most primitive mechanisms of the brain and allow the external input and output to designate the proper functioning (*4*).
2. ANNs are flexible with knowledge. The knowledge stored in expert systems is restricted to the human knowledge domain. In contrast, ANNs can be trained with some data acquired in the past for the particular problems to be solved; the data may not even to be a type of knowledge. For some problems, it is not necessary for humans to understand the knowledge about the data; the ANNs give answers according to their own internal criteria.
3. ANNs have different ways of learning. The learning of expert systems is a procedure designed to store some knowledge in the system. The learning of ANNs is to adjust the strengths of the connections between the processing units. The knowledge in an ANN is more like a function than a content. Some ANNs could have a self-learning capability.
4. ANNs have different ways of computing. The results in expert systems depend on how the knowledge has been represented. The computation method used by an ANN is determined by its architecture, and the results depend on how the network is structured. In a multilayer ANN, if the number of hidden units is changed, the accuracy of the results will be different.
5. Once an ANN is developed, no more programming is required; the only requirement is to feed data to the ANN and train it. However, in expert systems, programming may be required if additional knowledge is to be introduced to the system.

ANNs and expert systems can cooperate with each other, and for some problems they overlap in use. Once integrated, it is expected that they will enable artificial intelligence (also called AI) to cover a much wider variety of applications.

Applications of ANNs

The special characteristics make ANNs especially useful in a variety of applications. An ANN can provide an approach that is closer to human perception and recognition than traditional methods. In situations in which input is noisy or incomplete, ANNs can still produce reasonable results.

Although ANNs have not been widely explored in transportation engineering, their unique properties indicate great potential. We summarize two applications in the following.

Self-Organizing Traffic Control System

Nakatsuji developed an optimizing splits traffic control system using a four-layer ANN (*5*). The development is based on two assumptions: (a) cycle length is common over the road network and does not vary with time, and (b) there are no offsets between adjacent intersections. The purpose of this project was to estimate optimal splits of signal phases using ANN technology. The inputs to the ANN are control variables, that is, split lengths of signal phases and the traffic volumes on inflow links; the outputs from the ANN are the measures of effectiveness such as queue lengths and the performance index. Through a case study, Nakatsuji reported that his system was effective in adjusting the synaptic weights in the training process and was able to improve the convergence into global minimum, and the solutions achieved were in accord with analytical ones.

Intelligent System for Automated Pavement Evaluation

The intelligent system for automated pavement evaluation was developed by Ritchie et al. (*6*). The focus in this research was the development of an advanced sensor-processing capability using ANN technology to determine the type, severity, and extent of distresses from digitized video image representations of the pavement surface acquired in real time. A three-layer ANN was used in this system. The results of the initial case study presented in this paper clearly show the potential for application of ANNs for distress classification of pavement images as part of the proposed innovative noncontact intelligent system.

CLASSIFICATION OF ANNs

In the operation of ANNs, two issues should be addressed: (a) How are the processing units and the interconnection configured? that is, What is the structure of the ANN? and (b) How will weight values be assigned to the interconnections? that is, What are the ANN's learning rules? Here, the general classification of architectures and the learning of ANNs are presented.

The types of ANN architecture can be divided into four categories.

1. Mapping ANNs—Using a transfer function, mapping ANNs compute the sum of all the products of corresponding inputs and weights to make the outputs, that is,

$$Y = f(X,W)$$

where

Y = outputs,
X = inputs,
W = weights, and
f = transfer function.

2. Recurrent ANNs—In recurrent ANNs, some (or all) of the outputs are connected to the inputs. The outputs of the ANNs are the function of the inputs, the weights, and some (or all) outputs, that is,

$$Y = f(Y,X,W)$$

or

$$y(t + 1) = f[Y(t), X(t), W(t)]$$

where t denotes the time.

3. Temporal ANNs—Temporal ANNs compute the rates of the changes of their outputs as a function of the outputs, the inputs and the weights. In mathematical notation,

$$dY/dt = f(Y,X,W)$$

4. Hybrid ANNs—Hybrid ANNs integrate different kinds of learning into one network.

The learning of ANNs is generally classified as

- Supervised learning—ANNs are trained on a set of input-output pairs. The weights are adjusted to minimize error of the outputs. Another set of input-output pairs, called testing data, is provided to test the effects of training.
- Self-organizing learning—The network is trained on a set of inputs. No guidance is presented to the network about what it is supposed to learn. The ANN adjusts the weights to meet its own built-in criterion.
- Reinforcement learning—ANNs are trained on a set of inputs. The target values are not provided for learning; instead, error signals of the output are given to the ANNs. This process is analogous to reward or punishment.

Mapping ANNs

Of the four kinds of ANN, mapping ANNs have the most models. Basically, for mapping ANNs, if Y_i denotes the ith output, X_j denotes the jth input ($j = 0,1, \ldots, n$) and W_{ij} denotes the corresponding weight, then we have

$$Y_i = g\left(\sum_{j=0 \text{ to } n} W_{ij} X_j\right)$$

where g is the transfer function. Generally, g will be one of the following types:

- Linear function:

$$g(x) = x$$

- Threshold function:

$$g(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1(\text{or } 0) & \text{else} \end{cases}$$

- Sigmoid function:

$$g(x) = \tan h(x) \text{ or } g(x) = \frac{1}{1 + e^{-x}}$$

X_1 to X_n are the external inputs. X_0 is the artificial or the bias input, and it is added to simplify the network implementation. Without it the formula for the network would be

$$Y_i = g\left(\sum_{j=1 \text{ to } n} W_{ij}X_j - \text{threshold}_j\right)$$

The major models of mapping ANNs are given in the following.

Linear Associator

The linear associator is one of the earliest basic mapping ANNs; it was invented by several people during the period 1968 through 1972. It can learn at most L input-output vector pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_L, y_L)$. When one of these input vectors, say x_k, is entered in the network, the output vector y should be y_k. When a vector $x_k + \varepsilon$ (close to x_k) is entered into the network, the output vector should be $y_k + \delta$ (close to y_k)(*1*).

Learning Matrix

The learning matrix is a crossbar, heteroassociative, nearest-neighbor classifier. It was applied to problems such as highly distorted handwritten chracters and diagnoses of mechanical failures to reduce downtime of machines.

ADALINE and MADALINE

ADALINE (Adaptive Linear Element) is described as a combinatorial logical circuit that accepts several inputs and produces one output. MADALINE (Multiple ADALINE) is a more comprehensive network consisting of many ADALINEs. Both ADALINE and MADALINE are commonly used models of mapping ANNs, and generally they operate with a least mean square error-correcting learning rule. The applications of ADALINE and MADALINE have been developed in control, pattern recognition, image processing, noise cancellation, and antenna systems.

Back Propagation

Back propagation is one of the best-known ANNs. The typical back-propagation ANN always has at least one hidden layer. There is no theoretical limit on the number of hidden layers, but typically there are no more than two. During the learning process, the error information is propagated back from the output layer through the network to the first hidden layer. Back propagation is a very powerful technique for constructing nonlinear transfer functions between a number of continuously valued inputs and one or more continuously valued outputs. This property leads to many interesting applications. Back-propagation ANNs have been used in solving many real-world problems such as image processing, speech processing, optimization, prediction, diagnostics, control, signal processing, noise filtering, and forecasting (*1,2,7*).

Self-Organizing Mapping

The self-organizing mapping ANNs can be used to sort items into appropriate categories. One of the most famous of these ANNs is the Kohonen layer, which was developed in Finland by Kohonen of Helsinki University of Technology between 1979 and 1982. The Kohonen layer basically implements a clustering algorithm to the network. In the operation, only one unit fires and takes a value of 1; the others will be 0. This process is accomplished by a winner-takes-all strategy. Self-organizing mapping can be easily adapted to handle categorization problems (*7*).

Adaptive Resonance Theory

The adaptive resonance theory (ART) was developed by Carpenter and Grossberg of Boston University in 1987. It includes three implementations: ART1 for binary inputs, ART2 for continuous-valued inputs, and ART3, which is the refinement of ART2. An ART network can classify and recognize input patterns without the presence of an omniscient teacher; that is, no instructor tells the network to which category each particular stimulus input belongs (*7*).

Recurrent ANNs

In recurrent ANNs the outputs are fed back to the network as a part of their inputs. This process is also described as a recurrent connection. The major models of recurrent ANNs are summarized in the following.

Hopfield

The Hopfield network was proposed in 1982 by Hopfield of the Biophysics Division of Bell Laboratories. It is a primary example of a recurrent network. The Hopfield network acts as an associative memory, that is, it passes through a sequence of several patterns and chooses one that most closely resembles the input pattern as the output. The stable patterns into which the network settles are called attractors, and the possible states of the network are called the configuration space. The sets of the states that eventually transform into the same attractor are called basins of attraction.

The Hopfield networks have been used not only in the common areas of ANNs such as image processing, signal processing, pattern matching, and prediction, but also to solve some classical combinatorial problems such as the "traveling salesperson problem." Its ability in optimization should also be highlighted.

Brain-State-in-a-Box

Compared with the Hopfield network, whose processing units are allowed to take only binary values, the brain-state-in-a-box (BSB) network has processing units that are allowed to take on any value from -1 to $+1$. BSB has been used in pattern classification, diagnostics, knowledge processing, image processing, and psychological experiments (*2*). It is reported that one of the BSB applications, the instant physician system developed by Anderson of Brown University in 1985, is performing surprisingly well (*1*).

Bidirectional Associative Memory

The bidirectional associative memory (BAM) network was developed in 1987 by Koskonow of the University of Southern California. It allows associations between two arbitrary patterns. A BAM network consists of two layers, with every unit in one layer connected to every unit in the other layer. BAM applications in image processing, control, resource allocation, and optical/electrooptical have been reported.

Boltzmann Machine

The Boltzmann machine was developed in 1984 by Hinton of the University of Toronto and colleagues. This network is a trainable, stochastic version of the Hopfield network. Hidden units constitute an important part of the architecture. Boltzmann machines have many applications, such as image processing, speech processing, temporal processing, prediction, optimization, diagnostics, character recognition, knowledge processing, and signal processing (2).

Recurrent Back Propagation

Recurrent back-propagation ANNs can recognize dynamic patterns whose input vectors change with time; that is, a sequence of input vectors X(0), X(1), . . . , X(t_{stop} − 1) are represented to the network. A typical recurrent back-propagation ANN consists of two layers, a functional layer and a register layer. It handles a set of time-dependent input-output data instead of static input-output of back-propagation ANNs.

Temporal ANNs

The operation of temporal ANNs can be represented by a differential equation, and the output and input vectors of the network are changed with time. Fewer models of temporal ANNs have been developed. Generally speaking, temporal ANNs are suitable for dealing with the dynamic types of problems.

Hybrid ANNs

Hybrid ANNs combine supervised and unsupervised learning in one network. We summarize two models.

Counter Propagation

The counter-propagation network is a typical hybrid ANN initially proposed by Hecht-Nielsen. Counter-propagation networks consist of a linear mapping layer on top of a Kohonen layer. The Kohonen layer is trained in the usual way, and the linear mapping layer is trained by a simplified version of delta rule that is called ouster learning. Counter-propagation networks often run 10 to 100 times as fast as back-propagation networks when they are applicable to a situation.

GMDH

The GMDH ANN combines simple nonlinear processing units into an effective multilayer network. The most important feature of this kind of ANN is that it includes a procedure for building a near-optimal network.

In short, compared with traditional computation methods, ANNs behave as if they are depending on some kind of "intuitional reaction." They are concerned not with the principles of their operation but with the effects of their behaviors. This makes ANNs special and superior in solving certain problems, especially for recognition, control, image processing, optimization, and diagnostics. The structures of ANNs seem to be problem-dependent—for different types of problems, different structures should be used. In Table 1 we present application problems divided into 11 categories and rate the performances of 13 major ANN models in these categories.

APPLICATIONS OF ANNs IN TRANSPORTATION ENGINEERING

Transportation problems can be divided into five categories: planning, operation and control, administration and finance, construction and maintenance, and design. The applications are identified and the suitabilities of several major ANN models in different transportation engineering domains are discussed.

Planning

Transportation planning identifies a set of actions that will lead to the achievement of given objectives. The prediction of traffic conditions is a basic problem of transportation planning. Two of the most common traffic planning problems, trip generation and the origin-destination (O-D) distribution, are identified as suitable for the ADALINE and back-propagation ANN models, respectively.

Trip Generation

Trip Generation forecasting analysis predicts the zonal amount of traffic. The primary approach used for trip generation forecasting now is the regression method, which uses the statistical data of the past to establish a mathematical model used to compute the number of traffic trips required in the future. The ADALINE is identified as being suitable for this problem. The weights between the inputs and the outputs could be considered the regression coefficients in regression analysis (linear case). The training data sets are taken from past survey data. The output of the network can be a variable or a vector. If we take only one output, the model predicts the value of trip generation at a specific time in the future. If we take a sequence of outputs, it predicts the values of the trip generation in a time series in the future. This idea is also expected to be used in traffic attraction forecasting.

O-D Distribution

The trip generation problem described traffic volume production for a specific area. The O-D distribution describes the traffic volume between specific areas (called zones). The O-D forecasting is implemented according to the data sets of traffic generation and traffic attraction of the zones. A number of O-D forecasting methods have been developed; basically, they are a traffic O-D distribution pattern estimation model that predicts the distribution pattern on the basis of three factors: the future traffic generation, the future attraction (or one of them), and the present distribution pattern. The back-propagation ANN is identified as suitable for O-D forecasting. The future zonal trip generations and attractions can be identified as the external inputs of the network, and the outputs of the network represent future O-D distribution. Through proper training, which is a process of minimizing the total

TABLE 1 Application Evaluation of ANN Models

Category \ Model	Mada./Ada.	ART	BAM	Hopf.	Boltz. Mach.	Back Prop.	BSB	Count. Prop.	Lin. Asso.	Learn Matr.	Koho.	GMDH
Recognition	●			●	●	●		○				
Control	◍	◍	◍	◍		●					○	
Forecasting/ Prediction	●					●						
Classification		●				●	◍	◍			●	
Diagnosis		◍				◍	●					
Optimization				●	◍	●						●
Noise Filtering	◍					●						
Image Processing	◍	◍	◍	◍	◍	◍	◍					
Association			◍	●	○		○		○	◍		
Decision Making		◍						●				
Temporal Processing	◍	◍				●						

Key:
- Strong Applicability ●
- Moderate Applicability ◍
- Applicable ○
- Difficult to Evaluate □

error of outputs, the noise in past O-D patterns can be removed. Our prototype development of an O-D forecasting model using a back-propagation network showed reliable feasibility.

Operation and Control

Operation and control of transportation deal with problems such as traffic congestion diagnosis and control, hazardous material transportation, air traffic control, and ground traffic signal timing control. Two applications are identified in this area.

Traffic Pattern Recognition System

Because of the difficulty of dynamic traffic assignment modeling, the most commonly used coordinated area traffic control systems are based on time interval–dependent control on the assumption that in one time interval the network traffic state (called the traffic pattern) is static. Obviously, the current method has two major defects. The first is that there is no proper criterion to identify the traffic patterns—that is, if the traffic pattern is changed we need a criterion that will identify the new traffic pattern and measure how much the new pattern differs from the old one so that a set of optimized control parameters can be presented. The second defect is that the tolerance of the traffic pattern changes—that is, when the traffic pattern does not change too much until it matches another set of control parameters, we usually do not want to change the control parameters because the changes may bring some traffic disorder. Against these two defects, the traffic pattern recognition system is proposed. At present, the ANN model used in this system is ART1, but it will soon be replaced with ART2. The ART properties of parallel process and tolerance adjustability can provide more reasonable traffic pattern recognition results than the traditional methods can.

Automatic Vehicle Identification

The inefficiency of toll booths has attracted the attention of many researchers. Several systems that may replace the functions of toll booths have been developed in the past few years. An automatic vehicle identification (AVI) system can provide the convenience of electronic vehicle identification, through the use of subscribed purchase transponders, as vehicles move through a toll plaza or checkpoint without stopping or, in some cases, even slowing down. The system automatically charges or debits tolls from the drivers' accounts. In 1988, such a system was set up on the Dallas (Texas) North Tollway, and 15,000 tags were issued. Three AVI toll roads being built in Orange County, California, reportedly will feature the most extensive use of AVI in the United States (*8*).

The most important advantage of the license plate recognition system is that no special devices are required for the vehicles passing through. In this system, the ANN is used to complete the task of character recognition. Several conventional computing methods are used to complete the statistical data processing and calculate the amount of toll fee for each motor vehicle that passes the checkpoint. This system is under development, and the character recognition software has been successfully completed. Another important advantage of this system is that it will be very helpful for the O-D survey.

Administration and Finance

The area of administration and finance deals with the sources and distribution of the money used for financing transportation systems and their improvement. Topics include innovative transportation financing techniques; the effect of deregulation on different modes, pricing, user payments, and cost; and the prioritization, allocation, and distribution of funds. Here, potential applications of ANNs in the innovative transportation financing field are described.

Many state and local transportation agencies have used innovative transportation financing techniques to cope with financial problems. Their efforts to attract private funding, along with specific case studies, may be found in published sources. A back-propagation ANN can be trained to learn the experiences of many state and local transportation agencies that have succeeded in generating funds by using innovative transportation financing techniques. The input to the network consists of such parameters as the transportation agency's size, financial needs, institutional elements, and the mode (i.e., transit, highway, pedestrian) for which financing is requested. The output includes the most appropriate financing techniques for the transportation agency to adopt and the amount of money that will potentially be generated.

Construction and Maintenance

Transportation construction involves all aspects of the construction process, including preparing the surface, placing the pavement materials, and preparing the roadway for use by traffic. Transportation maintenance starts some time after the construction and involves all the required work and procedures to keep the facility in working order. Under this topic the pavement traffic mark recognition system using appropriate ANN techniques is proposed.

Pavement traffic marks can inform motorists, guide traffic flow, and ensure traffic safety. The damage sustained by traffic marks painted on the roadway surface varies depending on the traffic that travels over them as well as the climatic conditions. Investigating the existence and the degree of damage to traffic marks is usually expensive and disruptive of the normal flow of traffic. The system currently under development uses a hybrid ANN (consisting of a Hopfield ANN and two multilayer back-propagation ANNs) to recognize the damaged traffic mark and classify it into one of several categories that represent the different degrees of damage.

Design

In transportation, design includes the visible elements of the transportation facility. It deals with factors such as the grade line or profile, horizontal and vertical alignments, pavement of roadway, terminals, and stations. The input to the design process includes such parameters as the forecasted design demand rate, type of demand, and environmental conditions such as weather and temperature. The output consists of the specific numerical or the range of values of the design elements. ANNs can be used for a variety of design problems in transportation. The ANN knowledge-storing ability is especially useful in the development of design supporting systems. Through proper training of the networks, the models can learn to provide suitable output results when provided with input values. For human experience–based design supporting systems, the most suitable ANN models are ADALINE, MADALINE, and back propagation.

In brief, the transportation applicabilities of ANNs in image processing, control, optimization, and forecasting are especially impressive. For some problems, such as traffic pattern recognition, ANNs obviously appear to be stronger than traditional methods. Some ANNs' unique abilities, such as pattern classification, filled some blanks in transportation engineering such as the traffic mark classification system. Table 2 shows the ANN models appropriate to transportation engineering applications.

CASE STUDY

As a demonstration of the applicability of ANNs in transportation engineering, we aimed trip generation forecasting, which is one of the most common transportation planning problems, using the ADALINE and the back-propagation ANN models.

Definition of Problem

Past studies have identified the fact that human traveling activities are related to some socioeconomic indexes such as income, population, and employment. For a specific area (or zone), the trip production can be conjectured from these indexes. The purpose of trip generation analysis is to develop a mathematical model that can be used to forecast the traffic

TABLE 2 Related ANN Models to Transportation Applications

TRANS. DOM.	APPLICATIONS	BALZ. MACH	ADA./ MADA.	BACK. PRO.	HOP. NET.	ART 1	ART2	Cauchy MACH	NEO .COG.
		NEURAL NETWORK MODELS							
PLANNING	TRIP GEN.		X						
	OD DISTR.			X					
OPERA. and CONTR.	SIG. TIMING OPT.			X				X	
	TRAF. PATTR. CLASSIFIC.				X	X	X		
MANAG. and FINAN.	TSP OPT.	X			X				
	LICENSE NO. REC.			X					X
CONSTR. and MAINT.	PAV. DISTRESS DETECTION			X					
	TRAF. MARK CLASSIF.			X	X				
DESIGN	DESIGN SUPPORT SYSTEM		X	X					

trip production of the forecasting zone according to a number of socioeconomic indexes. For example, two indexes, zonal households and zonal population, are used in one major Canadian transport planning study to predict the zonal peak-hour trip production (*9*).

zonal peak-hour
trips produced = 0.3036 (zonal households)
+ 0.5638 (zonal population)

Current Solution

The previous approaches for trip generation forecasting include two main methods: regression analysis and category analysis. The majority of trip generation studies performed to date have used multiple regression analysis to develop the prediction equations for the trips generated by various types of land use. Generally, the least squares method is used to determine the constant *c* and the coefficients.

Description of ANN Models

An ADALINE ANN was used in this case study. This very simple ANN model has only four input units and one processing unit. One of the obvious differences between ADALINE and the regression method is the handling of the optimization of the weights and the coefficients. The regression method pursues the coefficients that will produce the minimum error on the surveyed data, which can be considered the training data sets for the ADALINE model. The training of ADALINE pursues the best value of the weights that will

TABLE 3 Observed Data of Trips and Index

No.of single family houses x1	Permanent single family residents x2	Number of multi-family dwelling units x3	Number of families with single auto ownership x4	trip rate y		
48	116	130	89	234	Training data	Regression data
71	126	24	48	132		
47	84	26	37	108		
26	46	10	18	49		
221	477	32	127	412		
114	275	41	78	247		
41	73	25	25	69		
167	360	31	99	320		
397	916	37	217	741		
264	748	23	144	549		
269	763	38	154	586		
171	485	43	107	388		
179	507	26	103	384		
233	661	8	121	471		
185	524	26	103	396		
252	714	48	147	554		
275	779	72	170	627		
175	470	58	114	400		
128	343	67	96	321		
83	179	15	49	159		
401	1073	148	269	934	Testing data	
52	93	20	36	99		
26	46	9	18	53		
83	179	15	49	158		
3	5	0	2	4		
4	8	2	3	8		
416	1146	468	433	1273		
8	14	0	4	12	Forecasting data	
30	65	11	22	64		
22	39	5	14	53		
41	99	173	107	271		
124	121	51	88	240		

allow the model to obtain good results on the testing data sets, but not on the training data sets. Even if a set of weights will allow the model to perform well on the training data sets, unless those values of the weights will allow the model to reach the approximate error minimum on the testing data sets, those weights are not considered good. In fact, this phenomenon is called over training. An ADALINE process such as this is expected to achieve better results than the regression method does. The belief in the training of ADALINE described here is also a general characteristic of ANNs.

A back-propagation model is also applied. This is a two-layer model with four input units to match the number of indexes and two processing units in the hidden layer. Unlike the usual way, which uses the sigmoid function, a linear transfer function is used in this model to match the ADALINE so that we can compare the performance of the two models. The other purpose is to examine its operation when a linear function is used. The learning of back propagation is done in the usual manner, that is, by using a delta learning rule.

Computation

The data sets are shown in Table 3. The first 20 sets of the samples were used as the training data, and the next 7 sets of samples were used as the testing data for ANN models. The other five data sets of samples are used to verify the effects of the three methods, and hence we call them forecasting data. Both the training and testing data were put together for the regression analysis.

In the training of the ADALINE model, the mean squared error (*MSE*) is used to indicate how the training is going. When iterations were equal to 10,000, we reached the approximate minimum of which the $MSE = 0.000195$, and $MSE = 0.000053$. The training of the back-propagation model went much faster than the ADALINE. At iteration equal to 2,500, the minimum *MSE* on the testing data was detected at 0.00004242. The *MSE* on the training data at iteration equal to 2,500 was 0.000005035.

Figure 2 shows the forecasting results obtained using the three methods on the forecasting data, respectively. We can see that both of the ANN models forecasted values closer to the actually observed values than the regression does. The *MSE*s of these three methods are shown in Figure 3.

Discussion of Results

The results obtained indicate that ANN models work better than the linear regression method in this case. The results of ADALINE are slightly closer to the observed values than those of back propagation. This is considered to be the result of using floating data type in all the computations and then the stacked errors caused this small difference. However, the ADALINE took 10,000 iterations whereas the back propagation took only 2,500 iterations to reach the total error min-

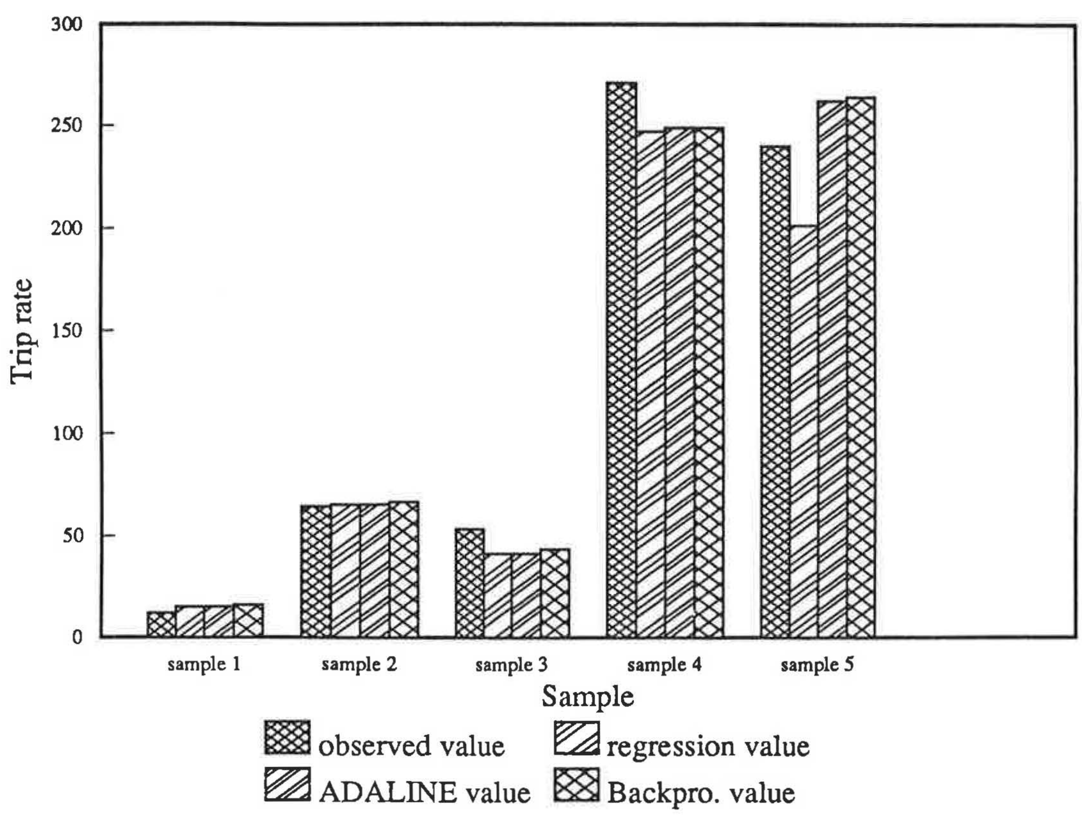

FIGURE 2 Results of regression, ADALINE, and back propagation.

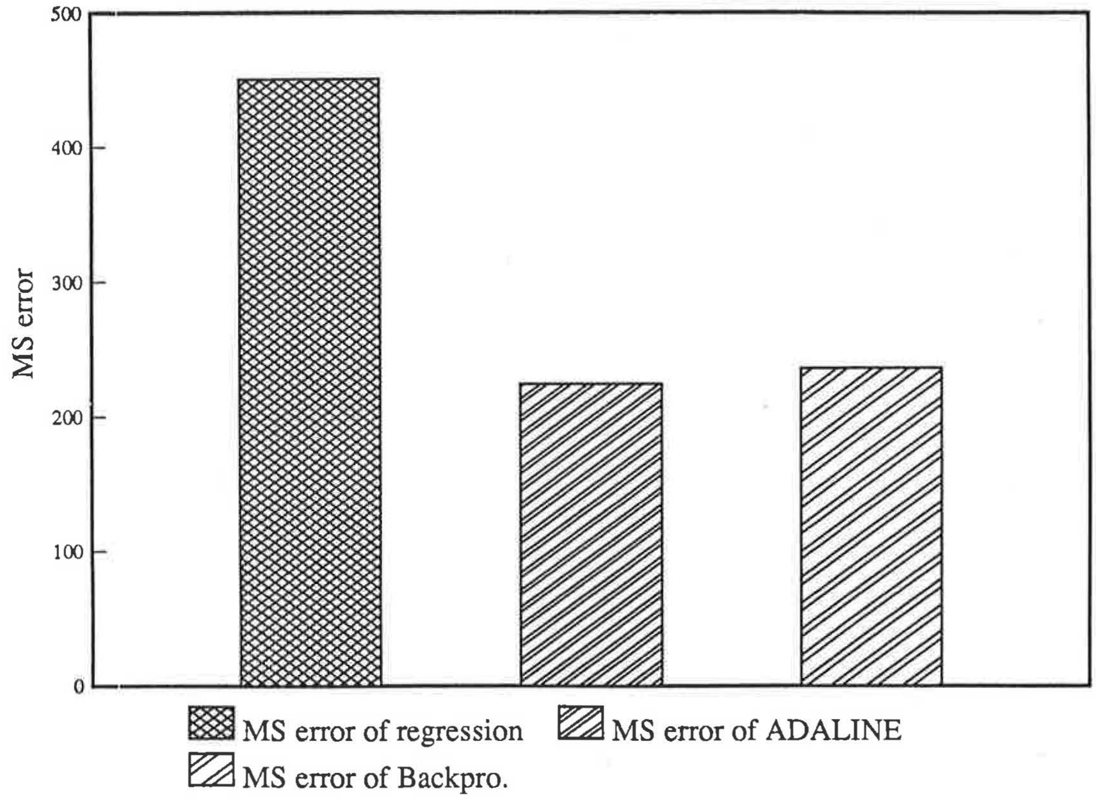

FIGURE 3 Comparison of *MSE* of regression, ADALINE, and back propagation.

imum on the testing data sets in the training. The training of the back-propagation model is much more efficient.

SUMMARY AND CONCLUSIONS

The fundamental concepts of ANNs were introduced, and the basic differences between ANNs and biological neural networks and expect systems were presented. On the basis of their unique architecture and learning techniques, ANNs were classified into four categories, and under each category, several ANN paradigms were explained. A matrix for relating each ANN paradigm to 11 attributes, including classification, pattern recognition, image processing and diagnosis, was then presented. On the basis of this matrix, the applicability of different ANN techniques for solving transportation problems was evaluated. Finally, a case study demonstrating the applications of two powerful ANN techniques—back propagation and ADALINE—in solving the trip generation problem was demonstrated. The results outperformed those obtained by conventional regression methods.

The strong applicability and suitability of different ANN techniques were demonstrated in this paper. Many of the example problems presented in the transportation application section have already been developed or are currently under development by the authors. ANNs are envisioned to become powerful tools not only for transportation, but also for enhancing the current state of such contemporary issues as artificial intelligence applications and intelligent vehicle-highway system technologies.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support for this project provided by a University of Delaware Research Foundation grant.

REFERENCES

1. R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley Publishing Company, Inc., Reading, Mass., 1990.
2. P. K. Simpson. *Artificial Neural System*. Pergamon Press, Inc., New York, N.Y., 1990.
3. M. Caudill. Neural Networks Primer Part 1. *AI Expert*, Dec. 1987, pp. 46–51.
4. B. L. Eliot. Neural Networks: A Comparison with Expert Systems. *Expert Systems Trends*, Winter 1990, pp. 72–75.
5. T. Nakatsuji and T. Kaku. Development of a Self-Organizing Traffic Control System Using Neural Network Models. In *Transportation Research Record 1324*, TRB, National Research Council, Washington, D.C., 1991.
6. S. G. Ritchie, M. Kaseko, and B. Bavarian. Development of an Intelligent System for Automated Pavement Evaluation. In *Transportation Research Record 1324*, TRB, National Research Council, Washington, D.C., 1991.
7. *Neural Computing Software Manual*. NeuralWare, Inc., Pittsburgh, Pa., 1991.
8. R. L. Hartje. Tomorrow's Toll Road. *Civil Engineering*, Feb. 1991, pp. 60–61.
9. B. G. Hutchinson. *Principles of Urban Transportation Planning*. Scripta Book Company, 1974.

Publication of this paper sponsored by Committee on Expert Systems.