Condition rating of rigid pavements by neural networks

Neil N. Eldin and Ahmed B. Senouci

Abstract: This paper discusses the development and the implementation of a neural network for the condition rating of rigid concrete pavements. The condition rating scheme employed by Oregon State Department of Transportation was used as the basis for the development of the network presented. A training set of 298 cases was used to train the network. The network adequately learned the training examples with an average training error of 0.011. A testing set of 3902 cases was used to check the generalization ability of the system. The network was able to determine the correct condition ratings with an average testing error of 0.022. The network ability of dealing with noisy data was also tested. Up to 40% noise was added to the data and introduced to the network. The results showed that the network presented could accurately identify condition rating relationships at high level of noise. Finally, a statistical hypothesis test was conducted to demonstrate the system's fault-tolerance and generalization properties.

Key words: neural networks, condition rating, condition index, rigid pavements, pavement distresses, pavement maintenance, fault-tolerance, generalization, noisy data.

Résumé : Cet article traite du développement et de la mise en oeuvre d'un réseau neuronal pour l'évaluation de l'état des chaussées rigides. Le système de notation de l'Oregon State Department of Transportation a été utilisé comme base d'élaboration du réseau présenté. Un ensemble de 298 cas a été utilisé pour former le réseau. Le réseau a assimilé les exemples de formation avec une erreur moyenne de 0,011. Un ensemble d'essai composé de 3902 cas a été utilisé pour vérifier la capacité de généralisation du système. Le réseau a été en mesure d'établir les bonnes notations avec une erreur d'essai moyenne de 0,022. La capacité du réseau à tenir compte de données bruitées a également fait l'objet d'essais. Un maximum de 40% de bruit a été ajouté aux données et intégré au réseau. Les résultats indiquent que le réseau présenté pouvait identifier avec précision des relations de notation de l'état à des niveaux élevés de bruit. Enfin, un essai théorique statistique a été réalisé afin de démontrer la tolérance du système aux pannes ainsi que ses caractéristiques de généralisation.

Mots clés : réseaux neuronaux, évaluation de l'état, indice de l'état, chaussées rigides, stress des chaussées, entretien des chaussées, faille de tolérance, généralisation, données bruitées.
[Traduit par la rédaction]

Introduction

Pavement maintenance has become increasingly a challenging task for many highway agencies as maintenance costs have steadily risen. One of the important activities of highway engineers is the determination of pavement condition ratings. It is on the basis of these condition ratings that maintenance priorities of road sections are assigned.

Pavement condition rating means assignment of relative weights to (or deducting points from) various levels of pavement distresses in order to obtain a combined score that indicates the current condition of a roadway section. Accurate determination of relative rating scores of highway sections is crucial to the management of highway maintenance programs.

Generally, condition ratings are subjectively determined by highway engineers and maintenance personnel. However, highway agencies have recently employed mathematical models to objectively determine pavement condition indices. Examples of condition indices include defect rating index (Snaith and Burrow 1984), pavement condition index (Uzarski 1984), maintenance control index (Kikukawa and Anzaki 1987), and maintenance needs index (Therberge 1987). These models, however, are based primarily on numerical formulas that attempt to assess condition indices.

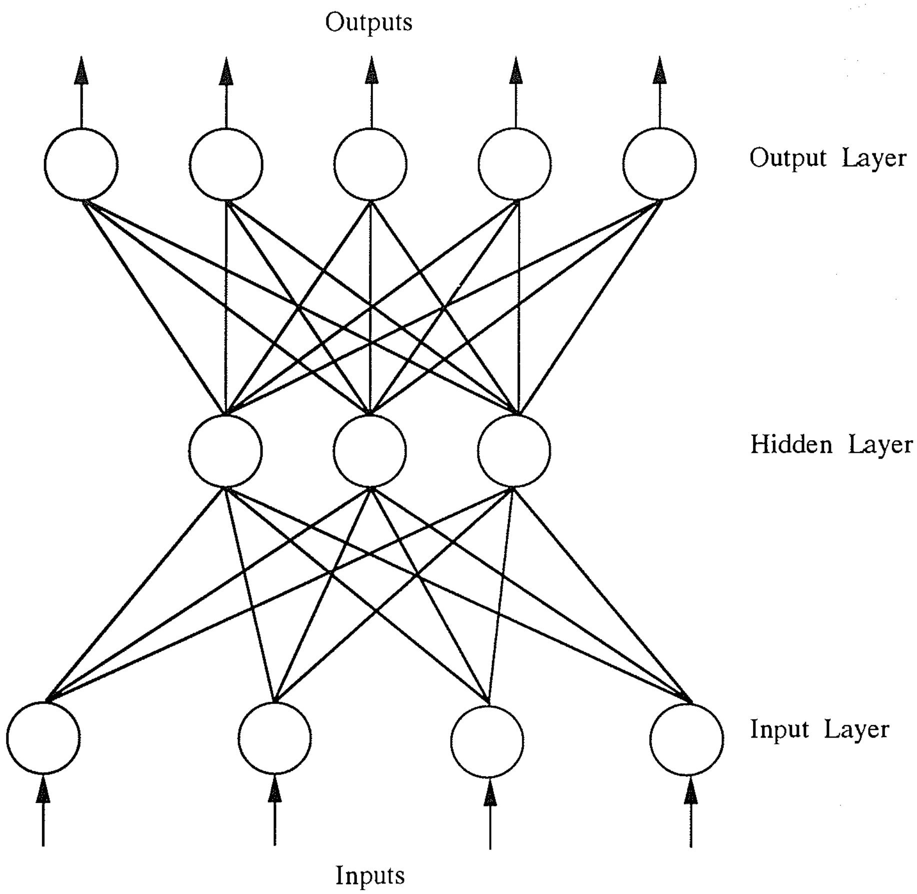
With the recent development in the field of neural networks, an effective tool can be added to assist highway maintenance personnel in managing pavement programs. Neural networks provide more advantageous technique for determining condition rating for road sections than many of the existing ones. Neural networks are computational models capable of mimicking the human decision-making process. They possess interesting properties such as self-organization, fault-tolerance, and massively parallel processing. Fwa and

Received August 2, 1994.
Revised manuscript accepted January 24, 1995.

N.N. Eldin and A.B. Senouci. Department of Civil Engineering, Oregon State University, Corvallis, OR 97331, U.S.A.

Written discussion of this paper is welcomed and will be received by the Editor until February 29, 1996 (address inside front cover).

Fig. 1. Typical backpropagation network with one hidden layer.



Chan (1992) explored the feasibility of using neural network models for priority assessment of pavement maintenance needs, and concluded that neural networks are extremely promising tools for pavement maintenance.

This paper provides an overview of neural network basics and presents the development of a neural network system for determining the condition rating of continuously reinforced concrete pavement. The system was validated by data and criteria provided by the Oregon State Department of Transportation (ODOT). The fault-tolerance and generalization properties of the proposed system was verified by statistical hypothesis testing.

Overview of neural networks

Neural computing is one of the fastest growing areas of artificial intelligence. The reason for this growth is that neural networks hold great promise for solving problems that have proven to be extremely difficult for standard digital computers. There are two key differences between neural and digital computation. First, neural networks are inherently parallel machines and as a result they can solve problems much faster than serial digital computers. Second, and perhaps more important, many neural networks have the ability to "learn." Rather than programming (hard coding) these networks, one presents them with a series of examples from which the network learns the governing relationships contained in the training examples. Thus, neural networks can offer an effective approach for handling nonlinear relationships such as that governing pavement condition ratings.

Figure 1 presents an example of a typical neural network. The network consists of processing neurons (shown as circles) and information flow channels (shown as solid lines) between the neurons, called connections or links. Each processing neuron typically has a small amount of local memory and it carries out a local computation that translates received inputs to the neuron outputs. This computation is called the transfer function of the neuron. Transfer functions can be linear or nonlinear and may consist of algebraic or differential equations.

At present, backpropagation networks are the most widely used. This paper employs a backpropagation neural network for determining pavement condition rating. A detailed discussion of neural networks can be found elsewhere (Werbos 1984; Rumelhart and McClelland 1986; Pao 1989; Simpson 1990). However, a brief description of some basic background is included here to benefit the unfamiliar reader.

Backpropagation networks

Backpropagation has been applied to a wide variety of practical problems and has proven very successful in its ability to model nonlinear relationships such as encountered in pavement condition rating. Figure 1 shows a typical backpropagation neural network with one hidden layer. As shown in the figure, the processing elements of the network are structured in three layers: input, hidden, and output. Each of these processing elements is called a node. Each node in the hidden layer receives input from all nodes in both the input and output layers through connections. Each connection (relationship between two connected nodes) is associated with a certain weight (W_{ji}) that can take a positive or negative value. In addition, each element is associated with a bias term, called threshold (θ_j). This bias term works as a horizontal shift for the origin of the transfer function to suit the magnitude of signals incoming to the processing elements. Values for both W_{ji} and θ_j are determined for a neural network during the training phase.

The output O_j of each neuron (j) in either the hidden layer or the output layer is given by the following sigmoid transfer function (Pao 1989):

$$[1] \qquad O_j = \frac{1}{1 + \exp(-h_j)}$$

where h_j is the total input of neuron j, and is determined by the following equations:

$$[2] \qquad h_j = \sum_{i=1}^{\mathrm{NIN}} x_i W_{ij} - \theta_j \qquad (\forall j \in \text{hidden layer};\ i \in \text{input layer})$$

$$[3] \qquad h_j = \sum_{i=1}^{\mathrm{NHN}} x_i W_{ij} - \theta_j \qquad (\forall j \in \text{output layer};\ i \in \text{hidden layer})$$

where NIN and NHN represent the number of neurons in the input and hidden layers, respectively.

Thresholds (θ_j) are usually omitted from [1] because they can always be treated as connections to an input neuron that is permanently clamped at -1. The sigmoid transfer function acts as a nonlinear gain for the output of a neuron. The nonlinear gain characteristics of the sigmoid transfer function helps the network to handle the small as well as the large inputs. If h_j of a neuron is too large in magnitude, the output of the neuron receives a very small gain. Conversely, if h_j is too small, the output of the neuron receives a high gain. The sigmoid function compresses the range of h_j such that O_j lies between 0 and 1, which is the desired output range in the present problem. The backpropagation training algorithm requires that the neuron transfer function be differentiable everywhere. The sigmoid function satisfies this condition.

At the beginning of the network training, connecting weights (W_{ji}) are initialized to random values. At the end of a prescribed training process, the knowledge implied by the training examples is captured in the form of weight values. Previous research suggests that initializing weights with small numbers ensures that network is not saturated by the large values of the weights before learning is complete (Rumelhart and Zisper 1986). It was also found that initialization with the same value does not allow network learning to take place.

Backpropagation learning algorithm is based on the error reduction between the actual output of a processing element and its desired output by modifying incoming connection weights. This is done through an iterative process during which the network modifies the initial random values of W_{ji} to converge the results to the desired (known) output. The algorithm can be thought of as a minimization problem in which the network error, E, is minimized. The network error, E, is defined as the average of output square errors over all training examples (input–output pairs), and is computed using the following formula:

$$[4] \qquad E = \frac{1}{2P} \sum_{j=1}^{P} \sum_{i=1}^{\mathrm{NON}} (D_{ij} - O_{ij})^2$$

where NON is the number of neurons in the output layer; P is the number of training examples; D_{ij} is the desired output value of the ith processing element in the output layer, pertaining to the jth training example; and O_{ij} is the computed output value of the ith processing element in the output layer, pertaining to the jth training example.

The method for finding the minimum E is summarized in the following step-by-step procedure:

1. Initialize the weights to small random values.
2. Increment the training iteration counter.
3. Initialize the network square error, E, to zero.
4. Select a training example from the training set. If no training examples are left, go to step 9.
5. Compute the output vector, $\boldsymbol{O}$. The computation is performed on a layer-by-layer basis using [3]. The outputs of the processing elements in a layer are used as inputs to the succeeding layer.
6. Compute the network square error, E, which is defined as

$$[5] \qquad E = E + \frac{1}{2P} \sum_{i=1}^{\mathrm{NON}} (D_j - O_j)^2 \qquad (j \in \text{output layer})$$

   where D_j is the desired output of the jth neuron in the output layer; O_j is the computed output for the jth neuron in the output layer, and NON is the number of neurons in the output layer.

7. Adaptively modify connection weights. In order to minimize the mean network square error E, it is necessary to adaptively modify each weight W_{ji} and threshold θ_j of the network. Following are the equations used to modify these weights.

$$[6] \qquad W_{ji} = W_{ji} + \eta O_j(1 - O_j)(D_j - O_j)O_i \qquad (\forall j \in \text{output layer};\ \forall i \in \text{hidden layer})$$

$$[7] \qquad W_{ji} = W_{ji} + \eta O_j(1 - O_j) \times \left[\sum_{k=1}^{\mathrm{NON}} O_k(1 - O_k)(D_k - O_k)W_{kj}\right] O_i$$

$$(\forall j \in \text{hidden layer};\ \forall i \in \text{input layer};\ k \in \text{input layer})$$

   where η is the learning rate which is chosen as large as possible without leading to oscillation of the convergence of the network when encountered with sharp curvatures in the error surface in the weight space. In our application, the learning rate was taken equal to 0.90.

8. Go back to step 4.
9. End the training process. If E is less than the target mean square error E_t, the training process is over; otherwise go back to step 2 for another iteration of the training procedure. The target mean square error E_t is the acceptable network error at the termination of the training process. This nonzero E_t leads to a certain amount of error in the actual output of each output neuron.

Table 1. Condition index.

Lowest condition index	Condition rating	Category
99–100	1	Very good
76–98	2	Good
46–75	3	Fair
11–45	4	Poor
0–10	5	Very poor

Neural network for pavement condition rating

The neural network presented here was developed using the condition rating scheme established by Oregon Department of Transportation (ODOT). In this computational scheme, the pavement condition rating is computed based on the cracking and rutting indices. These two indices represent the pavement condition based on distresses observed in the field. The lowest of these two indices is then transformed into a condition rating, as shown in Table 1. The condition index values are a function of distress type, distress severity, and distress quantity (extent) present in the pavement surface. The index values have been established to range from 0 to 100. Larger index values indicate better pavement conditions. The distance of 0.16 km (0.1 mile) has been selected as the length of a standard section for which distresses are categorized and quantified for calculation of condition index values.

For calculation of an index value of a given 0.16 km (0.1 mile) section, the distress(es) found in the pavement surface is categorized by type (rutting, transverse cracking, longitudinal cracking, etc.) and severity (low, moderate, or high) and then quantified in appropriate units (e.g., LF and SF). For each distress severity and for each distress type, an index value is computed using [8] as follows:

$$[8] \quad \text{Index(type X)}_{(\text{sev. X})} = 1.0 - A(\text{measured distress/maximum distress})^{B}$$

The coefficient A and exponent B represent the relative importance of the type and severity of each distress. The values assigned to these coefficients control the sensitivity of the index to measured distress. Dividing the measured distress by the maximum distress generates a dimensionless value (distress density, or extent) which ranges from 0 (no distress) to 1 (maximum distress).

The coefficient A can range in value from 0 to 1.0 and establishes the importance of a particular severity level and distress type relative to all the other severity levels and distress types. The exponent B also ranges in value from 0 to 1.0 and sets the curvature of the equation which controls the relative effect of the extent of a particular distress type. When $B = 1.0$, the equation generates a straight line with a slope equal to A, and the index calculated is directly proportional to measured distress. As B approaches 0, the equation becomes highly nonlinear and a small distress density would drastically decrease the index.

After computing index values based on distress severity and distress type using [8], a composite index value is calculated for each distress type by using [9] as follows.

Table 2. Rigid pavement deduct coefficients.

Distress				
Type	Severity	A	B	Maximum distress
Rutting	Low	0.05	1.00	
Rutting	Moderate	0.45	1.00	—
Rutting	High	0.85	1.00	
Lane Joint	Low	0.02	1.00	
Lane Joint	Moderate	0.04	1.00	—
Lane Joint	High	0.06	1.00	
Shoulder joint	Low	0.01	1.00	
Shoulder joint	Moderate	0.02	1.00	—
Shoulder joint	High	0.03	1.00	
Transverse	Low	0.33	0.30	
Transverse	Moderate	0.67	0.30	6000 SF
Transverse	High	1.00	0.30	
Longitudinal	Low	0.33	0.50	
Longitudinal	Moderate	0.67	0.50	44 each
Longitudinal	High	1.00	0.50	
Patching	Low	0.50	0.50	
Patching	Moderate	0.75	0.50	1500 LF
Patching	High	1.00	0.50	
Punchout	Low	0.33	0.50	
Punchout	Moderate	0.67	0.50	32 each
Punchout	High	1.00	0.50	

$$[9] \quad \text{Index(type X)} = \text{Index(type X)}_{(\text{sev.1})} \times \text{Index(type X)}_{(\text{sev.2})} \times \text{Index(Type X)}_{(\text{sev.3})}$$

Once an index value is calculated for each distress type, a pavement condition index is determined for a standard section using [10]–[12]. This condition index accounts for all appropriate distresses found in the given section:

$$[10] \quad \text{Rutt index} = 100 \times \text{Index(Rutting)}$$

$$[11] \quad \text{Crack index} = 100 \times \text{Index(Lane Joint)} \times \text{Index(Shoulder Joint)} \times \ldots \times \text{Index(Patching)}$$

$$[12] \quad \text{Condition index} = \text{Min(Crack index, Rutt index)}$$

The coefficients, exponents, and maximum values for the various distress types are presented in Table 2. Most of the distress types have three levels of severity: low, moderate, and high. The total measured distress for all three severity levels of a particular distress type cannot exceed the maximum value listed in Table 2 (e.g., patching (low) + patching (moderate) + patching (high) $\leq$ 6000 SF).

Network architecture

In developing a neural network for pavement condition rating, a backpropagation neural network with one hidden layer was used.The neural network, which has been chosen for the present research, has 15 input nodes and one output node. An input vector contains information about the type, severity, and quantity of pavement distresses while an output vector

Table 3. Network input and output components.

Component type	Component number	Attribute	Range of discrete values	Range of continuous values
Input	1	Rutting	0–3	—
	2	Lane joint	0–3	—
	3	Shoulder joint	0–3	—
	4	Transverse (low severity)	—	0–1
	5	Transverse (moderate severity)	—	0–1
	6	Transverse (high severity)	—	0–1
	7	Longitudinal (low severity)	—	0–1
	8	Longitudinal (moderate severity)	—	0–1
	9	Longitudinal (high severity)	—	0–1
	10	Patching (low severity)	—	0–1
	11	Patching (moderate severity)	—	0–1
	12	Patching (high severity)	—	0–1
	13	Punchout (low severity)	—	0–1
	14	Punchout (moderate severity)	—	0–1
	15	Punchout (high severity)	—	0–1
Output	1	Condition index	—	0.1–0.5

contains the condition rating of the pavement measured on a 0–0.5 scale. Table 3 summarizes the input and output components of the network. The number of hidden neurons was found to be 6 on a trial-and-error basis. Attempting a fewer number of hidden nodes resulted in the network's failure to converge to a desired level. Upon the selection of the network paradigm and initial network architecture, training and testing examples were prepared using ODOT rating scheme.

The first three components in the input vector take discrete values between 0 and 3. These values, respectively, correspond to the state of no distresses, low severity distresses, medium severity distresses, and high severity distresses. The remaining twelve components of the input vector take continuous values between 0 and 1. Each of these components represents the density (extent) of a specific severity level of a certain distress. It is important to note that the sum of components describing the same type of distress should be less than or equal to 1. The output component, which represents the pavement condition rating, takes a continuous value between 0.1 and 0.5. For example, if the output component takes a value of 0.3, the condition rating of the pavement section under consideration is equal to 3 (pavement in fair condition) (Table 1).

Network training

A set of 298 training examples was used to train the neural network. The training process was continued until the average absolute training error reached the value of 0.011. Table 4 summarizes the mean, the standard deviation, and the maximum training error. Figure 2 shows the training error distribution for the network. The results show that the network was capable of adequately learning the training examples.

Network testing

A set of 3902 examples was used to validate the developed neural network. The testing was performed periodically during the training phase in order to check the convergence of the network. The training phase was terminated when the average absolute testing error reached the value of 0.022. Table 4 summarizes the mean, the standard deviation, and the maximum testing error. Figure 3 summarizes the testing error distribution for the network. These results show that the network has a good generalization feature; the condition rating prediction of the network was very acceptable.

Neural network response to noisy data

In order to determine the performance of the presented network when subjected to noisy data, several levels of noise were tested. Noise was introduced to the network output component as uniformly distributed variables over the interval $[-k, k]$, where k is a fraction representing the level of noise expected and takes values between 0 and 1. Since the condition indices vary from 0.1 to a maximum value of 0.5, a k value of, say, 0.20 refers to a noise level of up to 40% of the maximum value in the condition rating.

The neural network was trained independently for each of the following three levels of noise: $k = 0.05$, 0.1, and 0.2. Figure 4 presents the average absolute testing errors corresponding to the four levels of noise, $k = 0$, 0.05, 0.1, and 0.2, respectively. The results show that the network was able to accurately predict the pavement condition indices even when trained with data containing a high level of noise.

Fault-tolerance and generalization properties

A trained network should be capable of generalizing the governing rules to accurately determine an output from new (not previously introduced) or incomplete inputs. In order to verify the fault-tolerance and generalization properties of the developed system, a statistical hypothesis test was conducted.

Table 5 summarizes the condition ratings obtained by both the ODOT model and the network system for 30 test cases. Each test case had two experiments, which are referred to as

Fig. 2. Training error distribution.

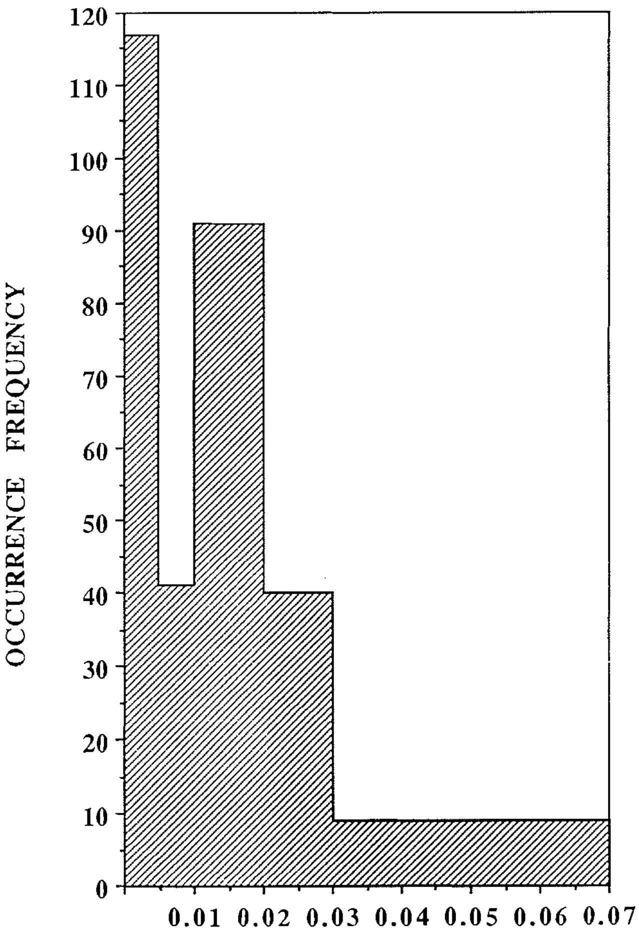


Fig. 3. Testing error distribution.

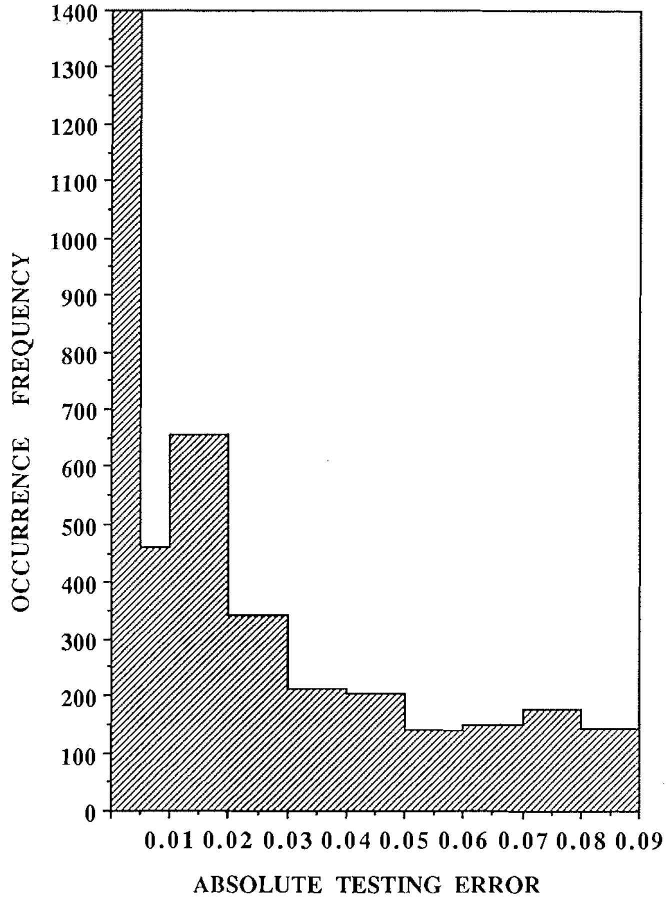


Table 4. Training and testing errors.

Phase	Mean error	Standard deviation	Maximum error
Training	0.011	0.010	0.064
Testing	0.022	0.023	0.089

experiments 1 and 2. The data for each test case are given in the table in two consecutive rows; the first row corresponds to experiment 1 and the second to experiment 2. In experiment 1, a complete set of inputs was used while in experiment 2 only a partial set was provided. A missing input is shown by a dash (—) in the input field. Selection of the missing input in each test case was done randomly. The fault-tolerance of both the ODOT model and the neural network was tested by conducting two paired t-test experiments. The first t-test experiment was to determine if the results provided by the ODOT model in experiments 1 and 2 were not significantly different. The second t-test experiment was to determine if the results provided by the network in experiments 1 and 2 were not significantly different.

With one condition rating per test case, one can make statistical inference on 30 paired differences of results (experiments 1 and 2). The paired differences were computed as

$$[13a] \quad D_j = \mathrm{Cond}_{1j} - \mathrm{Cond}_{2j} \qquad (j = 1, 2, \ldots, 30)$$

$$[13b] \quad D_i = \mathrm{Cond}_{1i} - \mathrm{Cond}_{2i} \qquad (i = 1, 2, \ldots, 30)$$

where Cond_{1j} and Cond_{2j} are the condition ratings obtained by the ODOT model in experiments 1 and 2 and Cond_{1i} and Cond_{2i} are the condition ratings obtained by the neural network system in experiments 1 and 2.

In both t-test experiments, the expected values of D_j and D_i should be equal to 0 if the results obtained in experiments 1 and 2 were to have the same mean value, μ. Therefore the hypothesis test was described as

$$[14a] \quad H_0\colon \quad \mu_D = 0$$

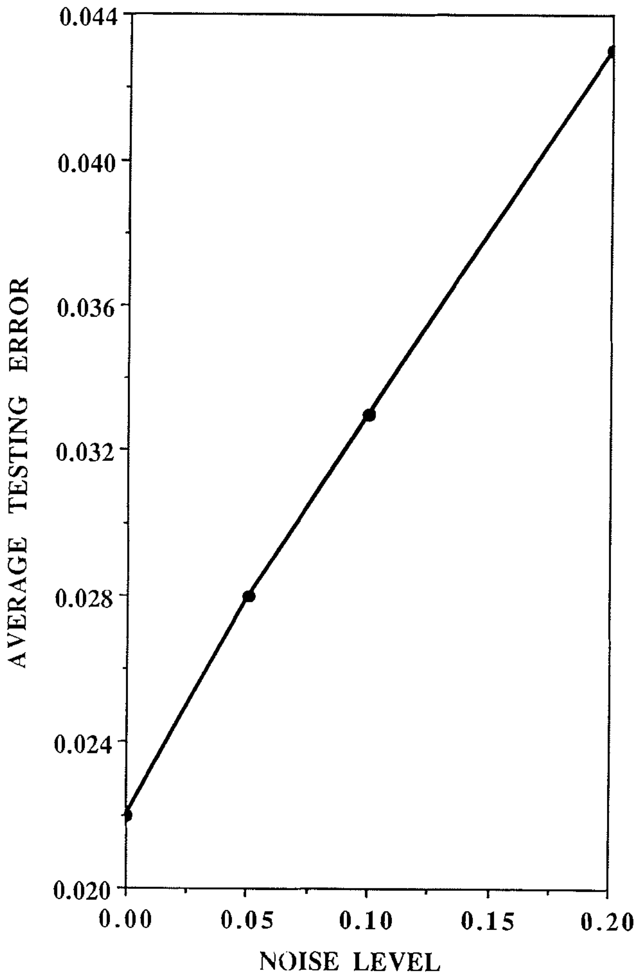
$$[14b] \quad H_1\colon \quad \mu_D \neq 0$$

The test statistic for this hypothesis is

$$[15] \quad t_0 = \frac{\bar{D}}{S_D/\sqrt{n}}$$

where $\bar{D}$ and S_D are the mean and standard deviation of the paired differences in each t-test experiment; and n is the number of paired differences (equal to 30). In the first t-test

Fig. 4. Average testing error versus noise level.



experiment (ODOT model), $\bar{D}$ and S_D equal to 0.0533 and 0.0629, respectively. In the second t-test experiment (neural network), $\bar{D}$ and S_D equal to 0.0100 and 0.0305, respectively.

For a type-I error, α, the null hypothesis H_0 would be rejected if

$$[16] \quad |t_0| > t_{\alpha/2,n-1}$$

where $n - 1$ is the number of degree of freedom, α is the significance level or the probability that the null hypothesis is rejected when it is true (the probability of type-I error), and $t_{\alpha/2,n-1}$ is the percentage points of the t-distribution. Choosing $\alpha = 0.05$, the value of $t_{\alpha/2,n-1}$ is equal to 2.045. In the first t-test experiment, the absolute value of t_0 is equal to 9.574. Since the absolute value of t_0 is larger than $t_{\alpha/2,n-1}$, the null hypothesis H_0 is rejected and the results obtained in experiment 2 are statistically different from those obtained in experiment 1. In the second t-test experiment, the absolute value of t_0 is equal to 1.795. Since the absolute value of t_0 is smaller than $t_{\alpha/2,n-1}$, the null hypothesis H_0 is accepted and the results obtained in experiment 2 are not statistically different from those obtained in experiment 1. Therefore, unlike the ODOT model, the developed network can provide the same condition ratings, with 95% confidence level, whether provided with a set of complete inputs or a set of partial inputs.

Advantages of the presented system

Existing condition rating models attempt to mathematically express the expert's decision-making. This approach has a number of limitations: (*i*) since mathematical models are often based on empirical equations, they need to be frequently updated to reflect improvements in maintenance technologies and maintenance strategies; (*ii*) as models used by different highway agencies vary considerably in their mathematical forms and types of defects addressed, transferability of knowledge among agencies may not be feasible; and (*iii*) partially inaccurate or incomplete input can result in wrong decisions.

In contrast, neural networks do not require that users specify mathematical relationships between input and output variables. Neural networks have the ability to learn from actual historical data and accurately establish the relationship between the input and output variables. After the training phase, a neural network can provide accurate condition ratings for road sections based on simple condition descriptions. Unlike mathematical model, neural networks have high fault tolerance. Neural networks are capable of accurately determining condition ratings even when presented with incomplete or inaccurate data. The presented system can be easily incorporated as a module into the pavement management system of the responsible agencies. The interaction between the proposed neural network and an agency's pavement management system can be achieved through an interface program.

The writers plan on expanding the network presented into a more comprehensive system by integrating advanced sensor processing capabilities for data collection. Data collection can be vastly improved by using digitized video image technologies.

Conclusions

In this study, a neural network system for the determination of condition rating for rigid pavement was developed. The study suggests that the use of neural networks has several advantages over other condition rating models. The following summarizes the findings of the study.

1. The network was able to adequately learn from the training examples with an average absolute error reached the value of 0.011.

2. The network showed a good generalization capability, and was able to predict condition ratings within an average absolute error of 0.022.

3. When noise was introduced in the input data, the network was able to accurately identify condition ratings with noise as high as 40% of the maximum value of the condition rating scale. At this noise level, the average absolute error was about 0.043.

4. Unlike the ODOT condition rating model, the network showed a good fault-tolerance capability. The network was able to provide the same condition ratings, with 95% confidence level, when provided with a set of incomplete input.

Table 5. Results of experiments 1 and 2.

Test case	Rutting	Lane joint	Shoulder joint	Transverse cracking Severity Low	Transverse cracking Severity Moderate	Transverse cracking Severity High	Longitudinal cracking Severity Low	Longitudinal cracking Severity Moderate	Longitudinal cracking Severity High	Patching Severity Low	Patching Severity Moderate	Patching Severity High	Punchout Severity Low	Punchout Severity Moderate	Punchout Severity High	Condition rating ODOT model	Condition rating Neural network
1	0	2	2	0.056	0.082	0	0.031	0.059	0.133	0.126	0.012	0	0.091	0.046	0	0.4	0.4
	0	2	2	0.056	0.082	0	0.031	0.059	0.133	0.126	0.012	0	—	0.046	0	0.4	0.4
2	2	0	2	0.325	0.043	0	0.275	0.093	0	0.119	0.148	0.115	0.02	0.292	0.111	0.5	0.4
	2	0	2	0.325	0.043	0	—	0.093	0	0.119	0.148	0.115	0.02	0.292	0.111	0.5	0.4
3	0	2	0	0.193	0.025	0	0.134	0.084	0	0.206	0.012	0	0.033	0.084	0.302	0.5	0.4
	0	2	0	—	0.025	0	0.134	0.084	0	0.206	0.012	0	0.033	0.084	0.302	0.4	0.4
4	2	0	0	0.002	0.001	0	0	0.002	0.001	0.001	0.002	0	0.001	0.002	0	0.3	0.4
	2	0	0	—	0.001	0	0	0.002	0.001	0.001	—	0	0.001	0.002	0	0.3	0.4
5	2	2	0	0.639	0.05	0.01	0.533	0.166	0	0.478	0.221	0	0.408	0.218	0.073	0.5	0.5
	2	—	0	0.639	0.05	0.01	—	0.166	0	0.478	0.221	0	0.408	0.218	0.073	0.5	0.5
6	2	2	2	0.059	0.099	0.064	0.096	0.074	0	0.159	0.011	0	0.13	0.04	0	0.4	0.4
	2	2	2	0.059	0.099	—	0.096	0.074	0	—	0.011	0	0.13	0.04	0	0.4	0.4
7	0	0	2	0.227	0.016	0	0.021	0.113	0.074	0.188	0.055	0	0.033	0.02	0.163	0.5	0.4
	0	0	2	0.227	0.016	0	0.021	0.113	0.074	0.188	0.055	0	—	0.02	—	0.4	0.4
8	0	2	2	0.047	0.012	0.016	0.061	0.006	0	0.053	0.015	0	0.059	0.008	0	0.4	0.4
	0	2	2	0.047	0.012	0.016	0.061	0.006	0	—	0.015	0	0.059	—	0	0.4	0.4
9	2	0	2	0.232	0.179	0.151	0.33	0.231	0	0.363	0.199	0	0.346	0.216	0	0.5	0.5
	2	0	2	0.232	0.179	0.151	0.33	—	0	—	0.199	0	0.346	0.216	0	0.5	0.5
10	2	2	0	0.042	0.006	0	0.044	0.004	0	0.018	0.029	0.005	0.018	0.011	0.06	0.4	0.4
	2	2	0	—	0.006	0	—	0.004	0	0.018	—	0.005	0.018	0.011	0.06	0.4	0.4
11	0	0	0	0.045	0.197	0	0.063	0.171	0.025	0.24	0.002	0	0.188	0.054	0	0.4	0.4
	0	0	0	0.045	0.197	0	0.063	0.171	0.025	—	0.002	0	—	—	0	0.4	0.4
12	0	0	0	0.228	0.129	0.103	0.264	0.089	0.161	0.246	0.187	0	0.169	0.124	0.281	0.5	0.5
	0	0	0	0.228	0.129	—	0.264	0.089	—	0.246	0.187	0	—	0.124	0.281	0.5	0.5
13	2	0	0	0.02	0.113	0.034	0.051	0.016	0.079	0.112	0.056	0	0.16	0.008	0	0.4	0.4
	—	0	0	—	—	0.034	0.051	0.016	0.079	0.112	0.056	0	0.16	0.008	0	0.4	0.4
14	2	0	2	0.032	0.011	0.012	0.022	0.024	0	0.037	0.009	0	0.042	0.003	0	0.4	0.4
	2	0	—	0.032	0.011	0.012	0.022	—	0	0.037	0.009	0	—	0.003	0	0.4	0.4
15	0	0	2	0.197	0.098	0	0.228	0.021	0.092	0.209	0.087	0	0.286	0.009	0	0.4	0.3
	0	0	—	—	0.098	0	0.228	—	0.092	0.209	0.087	0	—	0.009	0	0.4	0.4
16	0	2	2	0.479	0.033	0	0.107	0.358	0.047	0.321	0.191	0	0.143	0.369	0	0.5	0.4
	0	2	—	0.479	—	0	0.107	0.358	0.047	0.321	0.191	0	—	—	0	0.4	0.4
17	2	0	0	0.102	0.079	0	0.027	0.108	0.233	0.172	0.003	0.032	0.152	0.029	0	0.5	0.4
	—	0	0	—	—	0	0.027	0.108	0.233	0.172	0.003	—	0.152	0.029	0	0.4	0.4
18	2	2	0	0.129	0.085	0	0.188	0.027	0	0.131	0.081	0.007	0.136	0.073	0.023	0.4	0.4
	2	2	0	0.129	—	0	—	—	0	0.131	—	0.007	0.136	0.073	0.023	0.4	0.4
19	0	0	0	0.191	0.072	0	0.242	0.021	0	0.112	0.004	0.02	0.166	0.098	0	0.4	0.4
	0	0	0	—	0.072	0	0.242	0.021	0	—	0.004	0.02	0.166	—	0	0.4	0.4
20	0	2	2	0.02	0.023	0	0.041	0.002	0	0.008	0.032	0.015	0.004	0.031	0.039	0.4	0.4
	0	2	2	0.02	—	0	—	—	0	0.008	—	0.015	—	0.031	0.039	0.3	0.4

Table 5 (*concluded*).

				Transverse cracking			Longitudinal cracking			Patching			Punchout			Condition rating	
				Severity			Severity			Severity			Severity				
Test case	Rutting	Lane joint	Shoulder joint	Low	Moderate	High	Low	Moderate	High	Low	Moderate	High	Low	Moderate	High	ODOT model	Neural network
21	2	0	2	0.102	0.013	0	0.054	0.061	0	0.059	0.056	0	0.115	0	0	0.4	0.4
	2	0	—	0.102	0.013	0	—	—	0	0.059	—	0	—	0	0	0.3	0.4
22	2	2	2	0.084	0.016	0.042	0.117	0.048	0.015	0.071	0.099	0	0.16	0.011	0	0.4	0.4
	—	—	—	0.084	0.016	0.042	0.117	0.048	0.015	0.071	0.099	0	—	—	0	0.4	0.4
23	0	0	2	0.506	0.225	0	0.273	0.318	0.002	0.139	0.591	0	0.347	0.384	0	0.5	0.5
	0	0	2	—	0.225	0	—	0.318	0.002	0.139	—	0	—	—	0	0.4	0.4
24	2	2	0	0.05	0.042	0.041	0.056	0.05	0	0.071	0.035	0	0.027	0.079	0	0.4	0.4
	—	2	0	0.05	—	—	—	—	0	0.071	0.035	0	—	—	0	0.3	0.4
25	0	0	0	0.231	0.511	0	0.434	0.308	0	0.297	0.445	0	0.652	0.091	0	0.5	0.5
	0	0	0	—	—	0	—	—	0	—	—	0	0.652	0.091	0	0.3	0.4
26	0	2	2	0.059	0.022	0.158	0.036	0.011	0.273	0.074	0.09	0.026	0.062	0.108	0	0.5	0.4
	0	2	2	0.059	0.022	—	—	0.011	—	—	0.09	—	—	0.108	0	0.4	0.4
27	2	0	2	0.236	0.051	0	0.02	0.087	0	0.216	0.02	0.152	0.031	0.256	0	0.5	0.5
	2	0	2	—	—	0	0.02	0.087	0	—	—	0.152	—	—	0	0.4	0.4
28	2	2	2	0.023	0.039	0.003	0.043	0.02	0	0.056	0.007	0.001	0.046	0.018	0	0.4	0.4
	—	—	—	—	—	—	—	—	0	0.056	0.007	0.001	0.046	0.018	0	0.3	0.4
29	0	2	0	0.189	0.024	0	0.142	0.071	0	0.109	0.104	0	0.131	0.057	0.102	0.4	0.4
	0	2	0	—	—	0	0.142	0.071	0	—	—	0	—	—	—	0.3	0.4
30	2	2	0	0.435	0.001	0	0.154	0.282	0	0.127	0.069	0.388	0.398	0.038	0	0.5	0.4
	—	—	0	—	0.001	0	0.154	—	0	0.127	—	—	—	—	0	0.3	0.4

Acknowledgment

The writers would like to acknowledge the cooperation of Oregon State Department of Transportation staff. Special acknowledgment is due to Ms. Lucinda Moore and Mr. Danny Hori.

References

Fwa, T.F., and Chan, W.T. 1992. Priority rating of highway maintenance needs by neural networks. ASCE Journal of Transportation Engineering, **119**(3): 419–431.

Kikukawa, S., and Anzaki, Y. 1987. Present situation and prospect of pavement management system in Japan. Proceedings of the 2nd North American Conference on Managing Pavements, Vol. 1, Ontario Ministry of Transportation, Downsview, Ont., pp. 1.403–1.414.

Pao, Y.H. 1989. Adaptive pattern recognition and neural networks. Addison-Wesley Publishing Company, Reading, Mass.

Rumelhart, D.E., and McClelland, J. 1986. Parallel distributed processing: explorations in the microstructure of cognition. MIT Press, Cambridge, Mass., Vol. 1, Chap. 8.

Rumelhart, D.E., and Zisper, D. 1986. Feature discovery by competitive learning. *In* Parallel distributed processing: explorations in the micro-structure of cognition. Vol. 1. *Edited by* D.E. Rumelhart and J.L. McClelland. MIT Press, Cambridge, Mass.

Simpson, P.K. 1990. Artificial neural systems, foundations, paradigms, applications, and implementations. Pergamon Press, Elmsford, N.Y.

Snaith, M.S., and Burrow, J.C. 1984. Priority assessment. Transportation Research Record No. 951, Transportation Research Board, Washington, D.C., pp. 9–13.

Therberge, P.E. 1987. Development of mathematical models to assess highway maintenance needs and establish rehabilitation threshold levels. Transportation Research Record No. 1109, Transportation Research Board, Washington, D.C., pp. 27–35.

Uzarski, D.R. 1984. Managing better with PAVER. Transportation Research Record No. 951, Transportation Research Board, Washington, D.C., pp. 41–51.

Werbos, P. 1984. Beyond regression: new tools for prediction and analysis in behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, Mass.