Structural Engineering and Mechanics, Vol. 13, No. 2 (2002) 187-209
DOI: http://dx.doi.org/10.12989/sem.2002.13.2.187

Evaluation of existing bridges using neural networks

Augusto V. Molina†

Parson Transportation Group, New York, U.S.A.

Karen C. Chou‡

Department of Mechanical & Civil Engineering, Minnesota State University, Mankato, MN 56001, U.S.A.

Abstract. The infrastructure system in the United States has been aging faster than the resource available to restore them. Therefore decision for allocating the resources is based in part on the condition of the structural system. This paper proposes to use neural network to predict the overall rating of the structural system because of the successful applications of neural network to other fields which require a "symptom-diagnostic" type relationship. The goal of this paper is to illustrate the potential of using neural network in civil engineering applications and, particularly, in bridge evaluations. Data collected by the Tennessee Department of Transportation were used as "test bed" for the study. Multi-layer feed forward networks were developed using the Levenberg-Marquardt training algorithm. All the neural networks consisted of at least one hidden layer of neurons. Hyperbolic tangent transfer functions were used in the first hidden layer and log-sigmoid transfer functions were used in the subsequent hidden and output layers. The best performing neural network consisted of three hidden layers. This network contained three neurons in the first hidden layer, two neurons in the second hidden layer and one neuron in the third hidden layer. The neural network performed well based on a target error of 10%. The results of this study indicate that the potential for using neural networks for the evaluation of infrastructure systems is very good.

Key words: infrastructure; systems; evaluation; bridges; ratings; neural networks.

---

1. Introduction

Evaluation of existing infrastructure system is an ongoing process for engineers. Most recently, the American Society of Civil Engineers (ASCE) gave a grade of D+ to the infrastructure systems in the United States (ASCE 2001). After much attention was placed on the inspection and rehabilitation programs over the past two decades, the nation's bridge systems received a grade of C (ASCE 2001) which indicates that a lot of improvement is still needed. While the most seriously deteriorated bridges would get top priority for restoration, the next tier of deteriorated bridges will have to compete for the scarce resource.

An assessment of the overall condition of an infrastructure system requires the interpretation of observed data. Oftentimes, these data are inter-related, for example, a pair of parameters may inversely influencing each other, but they individually may impact the structures in the same

---

† Structural Engineer
‡ Associate Professor and Coordinator

manner as other parameters. Engineers have devised various models to accommodate the types of structural system they have to evaluate. Most of these models treat the observed data as deterministic attributes. Some would include the uncertainty of the observed quantities and implemented some forms of probabilistic analysis such as Bayesian method. Few models would also consider the subjective nature of the data. In those cases, fuzzy logic was implemented (for example, Chou and Yuan 1993, Wong *et al.* 1999).

Neural network can be considered as a set of resources in analysis toolkit (Eberhart and Dobbins 1990). It has an ability to make reasonable responses even when presented with incomplete, noisy or previously unseen input (Vanluchene 1990, Fwa and Chan 1993). Neural network differs from expert systems which simulate the function of a brain through a rule-based procedure. Neural network, on the other hand, is analogous to a biological neural system both in structure and functionality (Wassermann 1989). It develops its analysis algorithm through the examples that are fed to the network (Hecht-Nielson 1990).

In engineering applications, ocean engineering perhaps owns the biggest share. Under the autopsy of IEEE, it held its first conference on neural networks for ocean engineering (IEEE 1991). Studies included the use of neural networks for controlling underwater vehicles when the dynamics are not completely known (Venugopal *et al.* 1992), and application of neural networks to oceanic environments, particularly with reference to tracking underwater objects (Fa-Long *et al.* 1992, Silven 1992). In the chemical engineering area, neural network was used for fault diagnosis of chemical processes (Fan *et al.* 1993, Hoskins *et al.* 1991, Watanabe *et al.* 1989). Neural network was used to determine pore pressure for petroleum engineering (Accarain and Desbrandes 1993). In nuclear engineering, neural network was combined with expert system to identify abnormal events in nuclear power plants (Cheon and Chang 1993, Ohga and Seki 1993).

In civil engineering, neural network was used in water and wastewater treatment plant operation (Boger 1992), for optimum markup estimation in construction engineering and management (Moselhi *et al.* 1991), for priority rating of highway maintenance needs (Fwa and Chan 1993), for developing constitutive modeling of concrete (Sankarasubramanian and Rajasekaran 1996), and for effects of admixture on alkali-silica concrete (Li *et al*. 2000). A good volume of studies were focused on the damage assessment (Zhao 1998, Feng and Bahng 1999, Marwala 2000, Masri *et al.* 2000), structural control under vibraton such as earthquake (Liut *et al.* 1999, Kim *et al.* 2000, Hung *et al.* 2000), and detection of change of structural system for health monitoring purpose (Masri *et al*. 2000, Loh *et al*. 2000). More recently, neural network was implemented with response surface method to determine failure probabilities (Sasaki 2001) and to attempt to determine the reliability of large structural systems (Cabral and Katafygiotis 2001).

Another development that has enhanced the applicability of neural network is the computer technology, both in hardware and in software. Neural network can even be performed in pre-window generations of PCs. The rapid improvement of PCs and PC based software such as the *Neural Network Toolbox* which runs with *MATLAB* by The Math Works, Inc. have made the neural network analysis an attractive artificial intelligence tool in the decision making process.

Based on the results of using neural network in the engineering applications, it is interesting to explore further in civil engineering applications. Hence, the objective of this paper is to <u>illustrate</u> the concept of neural network in a bridge inspection problem. The goal is to <u>demonstrate</u> the potential of implementing neural network in future bridge inspection programs. In this paper, actual bridge inspection data collected by the Tennessee Department of Transportation (TDOT) during the period of 1979 to 1983 were used for illustration. Several neural network models for evaluating the observed data were developed. Discussion on their error distribution is presented.

2. Neural network concept

In its simplest form, a neural network can be classified as a function approximator. Given a set of input variables ($x_1,\ldots, x_i$) and one or more output variables ($y_1,\ldots, y_i$), a neural network finds the relationship or connection between the input and output. These connections are in the form of parameters called weights ($w_1,\ldots, w_i$) and biases (b) (Masters 1993). Eq. (1) shows a linear relationship between one output variable and a set of input variables.

$$\boldsymbol{f}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i) = \boldsymbol{x}_1\boldsymbol{w}_1 + \boldsymbol{x}_2\boldsymbol{w}_2 + \cdots + \boldsymbol{x}_i\boldsymbol{w}_i + \boldsymbol{b} \tag{1}$$

Nonlinear relationships between input and output are approximated by applying transfer functions to the value of Eq. (1). Two of the most common transfer functions employed in neural networks are the log-sigmoid function and the hyperbolic tangent function (see Eqs. 2 and 3, respectively).

$$\boldsymbol{f}(\boldsymbol{x}) = \frac{1}{1 + \boldsymbol{e}^{-x}} \tag{2}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \frac{\boldsymbol{e}^{x} - \boldsymbol{e}^{-x}}{\boldsymbol{e}^{x} + \boldsymbol{e}^{-x}} \tag{3}$$

The log-sigmoid function produces numbers between zero and positive one, while the hyperbolic tangent function produces numbers between negative one and positive one. Both functions produce a wide range of output values when input ranges from about negative two to positive two. Outside this range, the functions only produce values close to their respective output limits.

3. Structure of a neural network

The basic building block of neural networks is the neuron. Each neuron contains a set of weights and biases and an associated transfer function. The weights and biases are applied to the inputs, as indicated in Eq. (1), and further processing is performed by the transfer function. The output produced by a neuron can then be used as input by other neurons.

The arrangement of neurons within a neural network is commonly called the network architecture. One of the most popular architectures is the multi-layer feed forward network (Masters 1993). The general structure of this model typically consists of one or more layers of intermediate neurons or hidden neurons and a final layer of output neurons. The network input is processed by hidden neurons whose resulting output can be accepted as input by subsequent hidden neuron layers or the final output layer.

The difference between neuron layers lies in the number of neurons and the associated transfer function. Neurons within the same layer commonly employ the same transfer function. Fig. 1 shows two output variables (y_1, y_2) approximated by processing four input variables (x_1, ..., x_4) through a three-layer feed forward network. A multi-layer feed forward network consisting of at least one hidden layer with non-linear transfer functions has been proven to be a universal approximator

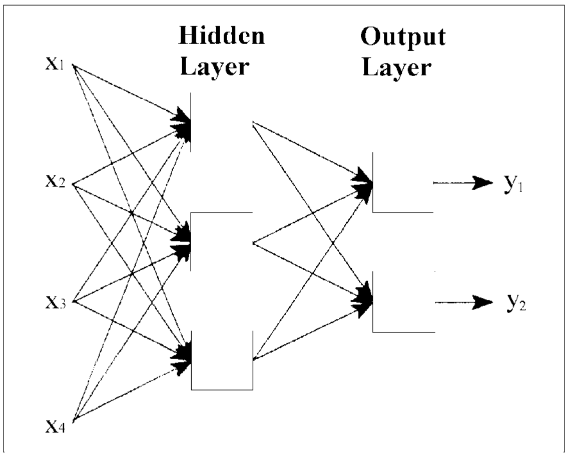

Fig. 1 Three-layer feed forward network structure

(Cybenko 1989, Funahashi 1989, Haykin 1994, Hornik *et al.* 1989). Therefore, this type of network is appropriate for performing function approximation, pattern association and pattern classification (Beale and Demuth 1992). Consequently, the multi-layer feed forward network model was implemented in this study.

4. Neural network design

The process of designing a neural network consists of two phases: training and testing. During training the network changes the weights and biases to learn an input/output relationship. After training is completed, testing is performed by processing new input through the network and comparing the desired network output to the actual output. When a network is presented with new input, it will tend to produce output similar to output associated with similar input. This behavior is called generalization (Beale and Demuth 1992). The design process is complete when the actual output is judged acceptable.

Training is the principal phase of neural network design. Small initial weights and biases are chosen randomly and training data (containing inputs and target output) are processed repeatedly through the network. Training is completed when the error between the target network output and the actual output is acceptable. For neural network design, the error is commonly measured in terms of sum-squared error (SSE). At the end of each training cycle or epoch, the SSE is calculated and the weights and biases are updated based on the derivative of the error.

4.1 Selection of training data

Training data consist of actual input and output. Ideally, neural networks are trained using only a fraction of the total available data. A necessary property of the training data is that they must represent the total population. Training data should also include the maximum and minimum input and output values; a network will not generalize well when asked to extrapolate the input/output beyond the training data range. Based on these criteria, selection of training data will greatly depend on the nature of the total population. As a result, there is no optimum size for the training set that should be used for all neural networks.

4.2 Scaling of input data

The use of nonlinear transfer functions is very important. Unfortunately, the log-sigmoid and hyperbolic tangent functions have a limited useful input value range for fast training. Within a range of approximately negative two to positive two, the gradients (or slopes) of the functions are large. Outside of this range, the gradient is close to zero and the transfer functions will only produce values close to their respective output limits. During training, the changes in weights and biases depend on the value of the gradient. Consequently, when a small gradient exists, training slows considerably. Large internal activations, which can be caused by input with large magnitudes, are mapped to small gradients in both the log-sigmoid and hyperbolic tangent functions, stopping training almost completely. This type of occurrence is known as premature saturation. Scaling the input within a specific range prevents premature saturation and results in faster training.

Two methods of scaling that are most frequently used are linear scaling and Z-score scaling. The linear method simply scales the input between +0.1 and +0.9. The Z-score method centers all the data around zero by subtracting off the mean and then scales the data to a unit variance. Z-score scaling assumes that the variables have an approximately normal distribution (Masters 1993).

The scaling parameters that are calculated for the training input data (slope and intercept for linear scaling and mean and standard deviation for Z-score scaling) must also be applied to input processed by the network after training. These parameters have a direct effect on training since the weights and biases are determined based on scaled input versus actual input. Therefore, consistent correlation between different sets of input is maintained by using a consistent set of scaling parameters.

4.3 Training rule for updating weights and biases

For this study, all networks were trained using Levenberg-Marquardt optimization. This method typically converges to an answer more rapidly than other conventional methods such as back-propagation. It uses the reliable convergence of gradient descent, when far from the minimum error and the rapid convergence of the Gauss-Newton method, when close to the minimum error.

The Levenberg-Marquardt rule for updating weights and biases is given by Eq. (4)

$$\Delta \boldsymbol{w}(\text{or } \Delta \boldsymbol{b}) = -\,(\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J} + \boldsymbol{\mu I})^{-1}\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{e} \tag{4}$$

in which $\Delta\boldsymbol{w}$ is the weight change, $\Delta\boldsymbol{b}$ is the bias change, $\boldsymbol{J}$ is the Jacobian matrix of derivatives of each error to each weight (or bias), μ is a scalar, $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{e}$ is an error vector. The variable μ determines whether $\Delta\boldsymbol{w}$ and $\Delta\boldsymbol{b}$ are calculated according to gradient descent or the Gauss-Newton method.

The variable μ is modified as the minimum error is approached. When far from the minimum, μ gets large and the $\boldsymbol{J}^T\boldsymbol{J}$ term becomes negligible. The calculation progresses according to $-\mu^{-1}\boldsymbol{J}^T\boldsymbol{e}$, which is gradient descent. When close to the minimum, μ is reduced and the calculation progresses according to $-(\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T\boldsymbol{e}$, which is the Gauss-Newton method (Levenberg 1944).

4.4 Overfitting

Overfitting is a common problem in neural network design. It can be caused by using an excessive number of hidden neurons or using training data that are not representative of the total population. Overfitting occurs when a network learns the specific details of the training data instead of the general shape of the total population function. Overfitting of training data results in poor generalization of new data.

With traditional design procedures, overfitting is usually identified during the testing phase. Cross-validation provides a method for identifying overfitting during the training phase. It consists of simultaneous testing of a separate data set while training is performed on the training set; the separate data set is called the cross-validation set.

Overfitting occurs when the cross-validation error reaches a minimum and starts to increase. This point can be identified by the number of training cycles or the SSE. Subsequent training can then be stopped when the minimum error is reached (Haykin 1994).

5. Evaluation and rating of bridges

5.1 General

The evaluation of bridges in the United States is generally performed by state government agencies such as departments of transportation. As a result, the methods for determining overall bridge ratings can be varied among agencies. Typically, bridges are evaluated based on the condition of key components such as girders, bracing, bearing devices, deck, expansion joint devices, and so forth. In many instances these components are rated on a numerical scale such as a range from zero to nine (zero indicating the worst condition and nine indicating the best condition).

After data are collected and key components are rated, a bridge is given an overall rating which can indicate level of structural adequacy, need for repair or final life expectancy. These ratings can then be used to prioritize the allocation of funds for rehabilitation or re-building. For bridges in which all the key components are either in excellent condition or very poor condition, an overall rating is quite obvious and is easily determined. For these situations, ratings are fairly consistent among engineers. But a large percentage of bridges usually contain only a few components that are in poor condition while the remaining components are in fair to excellent condition. The interaction of these components with the structure's overall performance is usually very complex. Neural networks have the capability of incorporating such varied significance of key components during its training process.

5.2 Structure inventory and appraisal

The U.S. Department of Transportation (DOT) provides a bridge evaluation guide for use by the States. The data collected via this guide is used for Federal reporting requirements. They include reports for the Highway Bridge Replacement and Rehabilitation Program and the National Bridge Inspection Program. This guide also provides a means of standardizing bridge evaluation across the entire country (U.S. DOT/Federal Highway Administration 1979).

A total of ninety inventory items are recorded for each bridge. These items are grouped under the categories of Identification, Classification, Structure Data, Condition, Appraisal and Proposed Improvements. Of the ninety items, only nineteen are actually used to determine the overall condition of a bridge. Table 1 lists the nineteen items under their corresponding categories.

Overall condition is measured in terms of a percentage called the Sufficiency Rating (SR). An entirely acceptable bridge has a SR of 100% and 0% represents an entirely deficient bridge. The Sufficient Ratings is composed of four variables each of which represents a rating category (see Eq. 5).

$$\boldsymbol{SR} = \boldsymbol{S}_1 + \boldsymbol{S}_2 + \boldsymbol{S}_3 - \boldsymbol{S}_4 \qquad (0\% \le \boldsymbol{SR} \le 100\%) \tag{5}$$

Where:

- S_1 represents Structural Adequacy and Safety and ranges from 0% to 55%
- S_2 represents Serviceability and Functional Obsoleteness and ranges from 0% to 30%

Table 1 Bridge rating inventory items

Item Number	Item Title	Evaluation Method	SR Correlation Coefficient
Category: Identification			
12	Designated Defense Highway No.	Precise	0.3460
19	Detour Length	Precise	–0.1950
Category: Structure Data			
28	Lanes On Structure	Precise	0.1387
29	Average Daily Traffic	Precise	0.2259
32	Approach Roadway Width	Precise	0.1143
36	Traffic Safety Features	Subjective	0.5156
43	Structure Type	Precise	0.1204
51	Bridge Roadway Width	Precise	0.3754
53	Vertical Clearance Over Deck	Precise	0.0806
Category: Condition			
58	Deck	Subjective	0.4653
59	Superstructure	Subjective	0.6676
60	Substructure	Subjective	0.5652
62	Culvert & Retaining Walls	Subjective	–0.2061
66	Inventory Rating	Subjective	0.8275
Category: Appraisal			
67	Structure Condition	Subjective	0.7209
68	Deck Geometry	Subjective	0.5074
69	Underclearance-Vertical & Lateral	Subjective	–0.2793
71	Waterway Adequacy	Subjective	0.4626
72	Approach Roadway Alignment	Subjective	0.4085

Table 2 Bridge rating variables and inventory item cross-reference

Variable	Item Number Used												
S1	59	60	62	66									
S2	12	28	29	32	43	51	53	58	67	68	69	71	72
S3	12	19	29										
S4	19	36	43										

• S_3 represents Essential Public Use and ranges from 0% to 15%
• S_4 represents Special Reductions and ranges from 0% to 13%

The variables are calculated based on a structured mathematical formulation of the nineteen inventory items. Table 2 lists the items that are used by each variable. Eleven of the nineteen inventory items are rated subjectively. The remaining eight items consist of precise data such as structure dimensions and material type. Table 1 identifies the subjective items. Ten of the subjective items fall under the categories of Condition and Appraisal and the remaining item (Traffic Safety Features) falls under Structure Data. A sample calculation of Sufficient Ratings using one set of observed bridge conditions is shown in the Appendix. As can be seen from the sample calculations, the procedure was very tedious. The logic of the mathematical formulations was not obvious.

5.3 Tennessee Department of Transportation bridge rating data

Bridge rating data used in this study were collected by the Tennessee Department of Transportation. A total of 447 samples were obtained between November, 1979 and September, 1983 with Sufficiency Ratings ranging from 2% to 99.8%. The sample mean and standard deviation of the Sufficiency Ratings are 67.99% and 21.68%, respectively. Fig. 2 shows the Sufficiency Ratings in ascending order. A complete summary of the bridge ratings can be found in Molina (1996).

Correlation coefficients were calculated to determine the influence each inventory item has on the Sufficient Ratings (Table 1). The correlation coefficient is a measure of the degree of linear

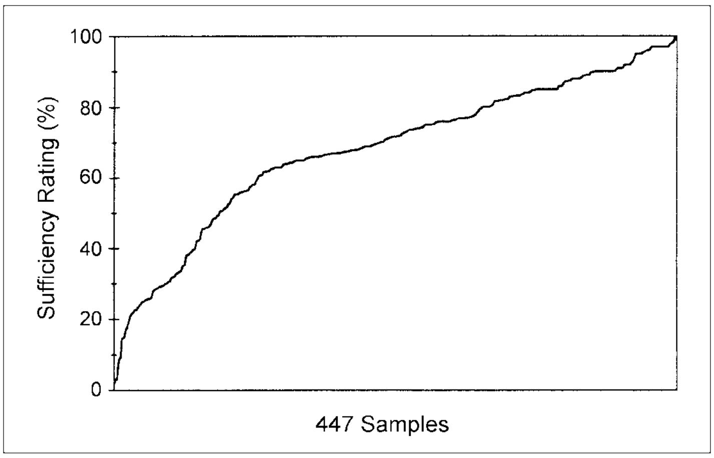

Fig. 2 Tennessee Department of Transportation bridge rating data

relationship between two variables. For this case, the coefficients with the largest magnitude are related to items which are subjective. This suggests subjective ratings have an important impact on bridge evaluation. In this paper, the subjectivity of the information would not be treated differently from the objective data. The impact of subjectivity on the analysis could be assessed using fuzzy neural network which is beyond the scope of this study.

6. Neural network architectures of bridge rating data

6.1 Training parameters

Output data conversion

The Sufficiency Ratings (SR) are calculated in terms of percentage, with the minimum of 0% and maximum of 100%. The log-sigmoid transfer function has a limited output range between zero and positive one. To take advantage of this range, log-sigmoid functions were used in the output layer and the SR values were converted to decimal equivalents. This guarantees that a neural network will predict the SR within the minimum and maximum values.

Scaling of input data

Scaling is necessary to prevent premature saturation during training. All the inventory items are recorded as positive numbers with values greater than or equal to zero. Therefore, all inputs were scaled between +0.1 and +0.9.

Transfer functions

The use of non-linear transfer functions is essential for modelling nonlinear systems. For this study, all networks were designed as multi-layer with at least one hidden layer. Hyperbolic tangent functions were used in the first hidden layer. Log-sigmoid functions were used in the subsequent hidden layers and the output layer to ensure output values lie between zero and positive one.

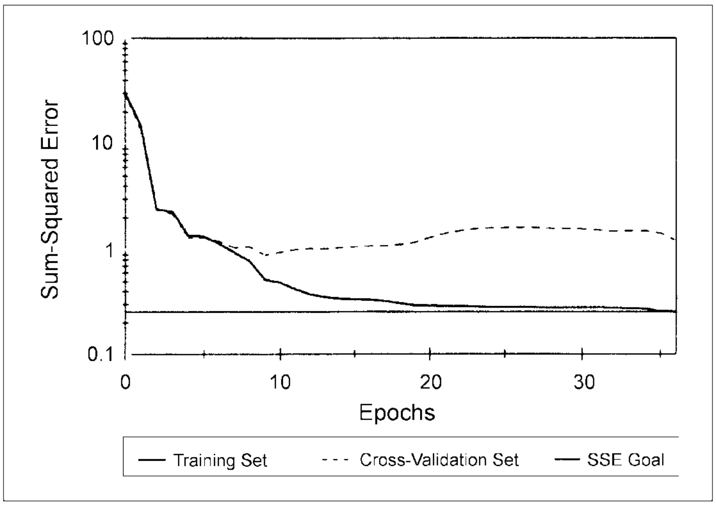

Fig. 3 Training progress for design #1, 4N network

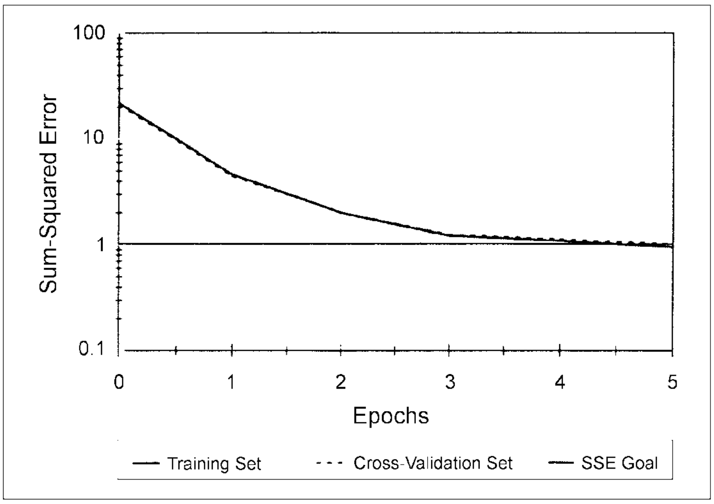

Fig. 4 Training progress for design #4, 4N network

Matlab Neural Network Toolbox

The neural network analyses were performed using the software *MATLAB* and the *Neural Network Toolbox* from Math Works, Inc. During the training stage, the weights and biases for each neuron in each layer were determined for each input parameter as described in the "Neural Network Design" section. The weights and biases were modified at the end of each epoch (cycle) until the sum-of-squared-error (SSE) has reached a desired level. The training was performed using the command TRAINLM.

The TRAINLM command (subroutine) will give the final weights and biases developed for the given set of data. The user can also graphically display the SSE at the end of each epoch so that

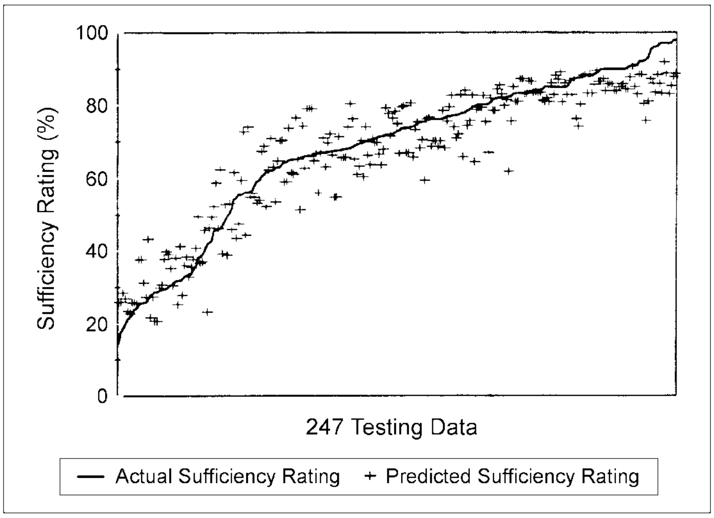

Fig. 5 Testing results for design #3, 4N network

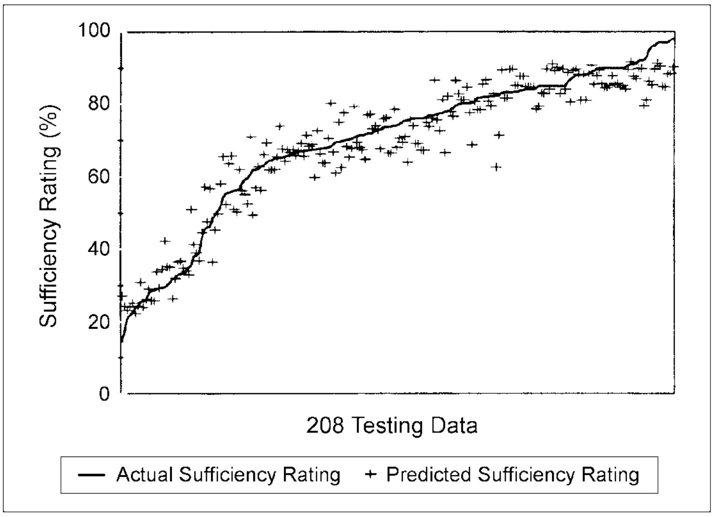

Fig. 6 Testing results for design #6, 3N-2N-1N network

one can see the rate of convergence (for example, the solid line in Figs. 3 and 4). In addition, if cross-validation is performed, the SSE obtained while validating the data can be displayed along side with the SSE from training. This dual display of SSE can assist the user in determining whether overfitting may have occurred (for example, dashed line in Figs. 3 and 4).

Once the user is satisfied with the network architecture, he/she can use the design and the weights and biases obtained to test a new set of data. A Neural Network Toolbox command, SIMULM, was used. The subroutine can display the predicted output based on the network design and compared that with the observed output (for example, Figs. 5 and 6).

6.2 Neural network design #1

Two hundred, among 447, samples were selected for training. The training set included twenty-five samples that contained the minimum and maximum input and output values. The remaining 175 samples were selected randomly using a random number generator. Another set of two hundred samples were selected randomly for use as a cross-validation set. The cross-validation samples were selected from the same 447 samples as the training data. Hence, there is a possibility some of the data used in the training were also used in the cross-validation. The sum-squared error goal was calculated based on 5% error per output (SSE = Σ $(y_i * 0.05)^2$). For the first training set, the SSE goal equaled 0.2557 (see Table 3 for summary of training parameters).

Three different feed forward networks were developed for training. The first contained one hidden layer, the second contained two hidden layers and the third contained three hidden layers. Each network was trained with the minimum number of neurons required to reach the SSE goal. A small number of neurons were initially used for training and the amount was gradually increased until training was successful. Training for specific network architectures was considered unsuccessful after three trials. Since the initial weights and biases are chosen randomly, training results will vary from one trial to the next. Therefore, more than one trial is necessary to consider training unsuccessful (Masters 1993).

Table 3 Neural network design training parameters

Design No.	No. of Training Samples*	No. of Cross-Validation Samples	Sample Selection Method	Average Error Goal Per Sample (%)	Sum-Squared Error Goal	No. of Input Variables
1	200	200	Random	5	0.2557	19
2	300	300	Random	5	0.3822	19
3	200	200	Random	10	1.0227	19
4	200	200	Random	10	1.0227	6
5	200	200	Random	10	1.0227	10
6	239	239	Random	10	1.1840	19

Table 4 Summary of weights and biases for design #1, 4N network

W1	Neuron 1							
	parameters 1-7	0.0750	–0.8252	–0.3926	–0.4516	–0.3143	1.3170	–0.3189
	parameters 8-14	3.7782	0.0130	–0.7813	2.3862	1.2478	–0.9853	0.2628
	parameters 15-19	–0.2433	0.0356	0.8877	0.6689	0.2205		
	Neuron 2							
	parameters 1-7	0.1269	17.7217	–0.4723	1.3557	12.0253	–0.7096	1.0225
	parameters 8-14	–2.7018	–1.6007	–0.6574	0.1395	–1.6516	–0.0666	–2.5651
	parameters 15-19	1.7437	–1.4927	–0.7330	–2.2481	–0.3932		
	Neuron 3							
	parameters 1-7	–1.4432	1.2041	–1.2799	–2.0408	–0.5515	–0.2109	–1.7934
	parameters 8-14	0.4401	–2.5060	–0.2630	–0.0348	1.8159	–0.6753	0.7071
	parameters 15-19	0.6673	2.4907	0.7385	–1.5172	0.3972		
	Neuron 4							
	parameters 1-7	0.5836	–2.5046	1.1372	–0.0836	–1.9460	0.2542	–0.6374
	parameters 8-14	–2.8846	–0.1319	0.0930	0.0988	0.5567	–0.9700	–1.4028
	parameters 15-19	–0.9652	0.4570	0.3688	0.9697	–0.2884		
B1	Neurons 1-4	–0.8690	3.5097	–1.8338	0.2301			
W2	Neurons 1-4	2.8359	–1.5350	3.3615	–8.6206			
B2		–5.5020						

The first network contained four neurons in the hidden layer (notation = 4N) and reached the error goal after thirty-six epochs. The second network contained three neurons in the first hidden layer and two neurons in the second hidden layer (notation = 3N-2N) and reached the error goal after twenty-four epochs. The third network contained three neurons in the first hidden layer, two neurons in the second hidden layer and one neuron in the third hidden layer (notation = 3N-2N-1N). This network reached the error goal after eighty-three epochs. Fig. 3 shows the SSE versus epochs for the 4N network. This figure is typical for the other two networks. Table 4 summarizes the final weights and biases for the 4-neurons hidden layer and the single output layer. The weight W1 is a 4×19 matrix where each row represents the weight for each neuron in the hidden layer and each column represents each of the 19 parameters used in the assessment. The bias B1 is a 4×1 vector which represents the bias for each neuron. The W2 vector represents the weight for each neuron from the hidden layer used to yield the single output. The value B2 represents the bias for the output.

In every case, the cross-validation error increased before training was completed, suggesting overfitting of the training data. Training and cross-validation error deviated at an SSE equal to approximately 1.0. This is equivalent to 10% error per output. Since the minimum number of neurons was used for successful training, overfitting could have occurred due to insufficient training data. Therefore, the training set size for the second network design was increased to three hundred samples.

6.3 Neural network design #2

For the second design, three hundred samples were selected for training. The training set included the twenty-five samples that contained the minimum and maximum input and output values. The remaining 275 samples were selected randomly. The SSE goal increased to 0.3822. Another set of three hundred samples were selected randomly for use as a cross-validation set. The cross-validation samples were selected from the same 447 samples as the training data (see Table 3). The same network architectures used in the first design were used on the second training set. Only the 4N network was trained successfully, reaching the error goal after thirty-two epochs.

The training results for the second design were consistent with those for the first. Overfitting of the training data occurred. The training and cross-validation error deviated after reaching an SSE of approximately 10% error per output. These results suggested that the best predictions that could be made by the neural networks were within 10% of the actual data. Therefore, the remaining networks were designed based on an SSE of 10% error per output using the two hundred sample training set.

6.4 Neural network design #3

The third design used the training and cross-validation sets selected from the first design (200 samples for training and 200 samples for cross-validation). Based on a 10% error per output, the sum-squared error was 1.0227 (see Table 3). The same architectures as neural network designs #1 and #2 (4N, 3N-2N, and 3N-2N-1N) were repeated with a new sum-squared error goal and trained successfully in each case. Fig. 4 shows the training progress for the 4N network, which is typical of the other two networks.

The results showed no overfitting of training data. In every case, the cross-validation error decreased consistently as training progressed. This further suggested that an average of 10% error per output was the optimum prediction that could be made by the neural networks.

The three networks (4N, 3N-2N, and 3N-2N-1N) were tested against the remaining data that were not used for training (247 samples). Fig. 5 shows a typical comparison between actual Sufficient Ratings and predicted Sufficient Ratings. Table 5 summarizes the maximum error and error distribution for each network.

6.5 Neural network design #4

The fourth design used the same training and cross-validation sets selected for the previous design. But in this case, the number of input variables was reduced to six (see Table 3). These variables were selected based on the magnitude of their Sufficient Ratings correlation coefficient.

Table 5 Testing summary for designs #3 to #6

Neural Network	Network Architecture	Maximum % Error	Percent of Test Samples with Error less than or equal to:				
			10%	20%	30%	40%	50%
Design #3	4N	78.08	67.21	88.26	94.74	97.17	98.79
	3N-2N	71.58	65.99	85.43	94.74	97.57	97.57
	3N-2N-1N	95.15	68.42	91.50	96.36	97.57	**99.60**
	Regression	109.35	60.73	86.23	94.74	96.36	97.98
Design #4	4N	**58.03**	40.32	85.43	92.31	96.76	99.19
	3N-2N	61.75	59.51	82.19	91.90	95.55	97.98
	3N-2N-1N	64.05	63.97	82.59	93.52	96.76	98.79
	Regression	106.63	51.01	80.97	91.09	95.95	98.38
Design #5	4N	69.11	54.25	81.78	91.50	98.38	98.79
	3N-2N	59.30	55.47	77.33	89.88	94.74	98.38
	3N-2N-1N	64.80	56.68	81.38	91.90	95.55	98.38
	Regression	104.48	50.20	79.76	91.09	95.55	97.98
Design #6	3N-2N-1N	86.28	**77.88**	**94.71**	**98.08**	**98.56**	99.52
	Regression	103.96	66.35	89.42	97.60	**98.56**	99.04

All variables with a correlation coefficient greater than 0.5 were selected. They include inventory items 36, 59, 60 and 66 through 68 (see Table 1). The 4N, 3N-2N and 3N-2N-1N networks were used for training to an SSE goal of 1.0227 and another multi-variate linear regression analysis was performed on the neural network training data.

All networks trained successfully with no significant overfitting and test results indicated that the network prediction was better than regression (see Table 5). The networks predicted the data with a maximum error of 64.05% (for 3N-2N-1N). But the error distribution for these networks was larger by comparison to the networks from the third design.

6.6 Neural network design #5

The fifth design also considered a reduced input variable set (less than 19). Variables with the top ten Sufficient Ratings correlation coefficients were selected. The four additional variables include inventory items 51, 58, 71, and 72 (see Table 1). The 4N, 3N-2N and 3N-2N-1N networks were used for training to an SSE goal of 1.0227 and another linear regression analysis was performed on the neural network training data.

All networks trained successfully with no significant overfitting. The networks predicted the data with a maximum error of 64.80% (for 3N-2N-1N). But, like the fourth design, the error distribution was larger by comparison to the networks from the third design (see Table 5).

6.7 Neural network design #6

The previous testing results showed that the 3N-2N-1N network from the third design produced the best general results (see Table 5). The 3N-2N-1N network of design #3 predicted 68.42% of the

Table 6 Summary of weights and biases for design #6, 3N-2N-1N network

W1	Neuron 1 (L1-1)							
	parameters 1-7	–0.1294	–0.9770	–0.6522	–0.6348	–0.5051	0.7708	0.3091
	parameters 8-14	0.7995	–0.2056	0.0697	–0.2048	0.1031	0.1965	0.0035
	parameters 15-19	–0.0412	1.0425	–0.6296	–0.1825	0.3416		
	Neuron 2 (L1-2)							
	parameters 1-7	–0.0153	0.0286	–0.1157	0.1262	–0.1728	–0.3490	–0.1961
	parameters 8-14	1.2323	0.1364	–0.1284	0.6195	0.3541	0.1981	1.0481
	parameters 15-19	0.3025	–0.2754	0.3720	0.1241	–0.0569		
	Neuron 3 (L1-3)							
	parameters 1-7	0.1986	0.0387	–0.3023	0.1204	–0.3867	–0.2711	0.1912
	parameters 8-14	–0.0238	–0.4375	–0.4934	–0.0012	–0.1978	–0.2336	0.0270
	parameters 15-19	0.2592	–0.2262	–0.5401	–0.2987	–0.4292		
B1	L1-1 to L1-3	–0.4509	–1.1344	–0.3325				
W2	L2-1							
	L1-1 to L1-3	–0.9720	–1.9342	–0.2661				
	L2-2							
	L1-1 to L1-3	1.2949	1.8391	0.2554				
B2	L2-1 to L2-2	0.0525	–0.3832					
W3	L2-1 to L2-2	2.9939	–3.2816					
B3	L3-1	0.0150						
W4	L3-1	–5.3655						
B4	L4-1	2.9954						

test samples within 10% error. A total of 78 samples were predicted with an error greater than 10%. This sample output has a mean Sufficient Ratings of 59.84%. The total population mean Sufficient Ratings was 67.99%. The difference in mean Sufficient Ratings suggested that the network had difficulty predicting small Sufficient Ratings values.

For the sixth design, the 3N-2N-1N network was retrained with a new data set that included all nineteen input variables. The new training set consisted of the original two hundred sample set and one half of the samples previously predicted with more than 10% error (39 samples). This increased the SSE goal to 1.1840. The thirty-nine additional samples were selected randomly with a mean Sufficient Ratings of approximately 59.84% (see Table 3). The network trained successfully with no indication of overfitting. Testing of the network improved with 77.88% of the test samples predicted within 10% error, increased from 68.42% for design #3.

The final weights and biases for each of the neurons in the 3 hidden layers and the single output layer are shown in Table 6. The readers are referred to the discussion in “Neural network design #1" section for definition of W’s, B’s on the weights and biases shown in Table 6. The value followed W’s and B’s represents the hidden layer number. For example, W3 is the weights for the 3rd hidden layer and W4 is for the output layer. The notation Li-j represents the j-th neuron of the i-th hidden layer. The last hidden layer represents the output layer. In Table 5, the bold face value shown in the “Maximum % Error” column represents the lowest maximum percent error among all designs. The bold face values shown in other columns represent the highest percentage of test samples within the given percentage error among all designs. As can be seen from the table that design #6 yielded the best results. Fig. 6 shows the comparison between actual Sufficient Ratings and predicted Sufficient Ratings.

6.8 Summary of results

The study showed that for the neural networks to be trained successfully without overfitting, a target error goal should be set at 10% per output. This was evidenced by the overfitting of designs #1 and #2 with target error goal of 5% per output, and the successful training of designs #3 to #6 (with target error goal of 10% per output). Design #2 demonstrated that the overfitting was not due to insufficient number of data. When the number of data increased from 200 to 300, the overfitting still occurred.

Testing results from the study showed that the networks with more than one hidden layer tended to produce better results. The networks with 3-hidden layers performed the best. However, based on the error distribution of the testing samples, the difference among the 3 networks was insignificant. Hence, any one of the three networks (4N, 3N-2N, and 3N-2N-1N) proposed here would provide an adequate assessment of the bridge conditions.

7. Conclusions

A study was performed to assess the potential advantages and benefits of using neural networks for evaluating infrastructure systems. Bridge rating data were used to illustrate the method. The results from this study show that neural networks can model data relationships nicely. The results also show that the neural network model yielded desirable output without the intermediate step of classifying the inventory items into the four categories as required in the Sufficient Ratings computations (see sample Sufficient Ratings calculations in Appendix). Furthermore, the logic associated with the mathematical formulations used to determine the Sufficient Ratings was not obvious.

An advantage provided by neural networks is its great design versatility. Network architecture can be easily modified by using different transfer functions and changing the number of hidden neurons and hidden neuron layers. This makes it possible to develop a prediction model suited for a specific type of problem. Networks can be developed for bridges made from a specific material (concrete, steel, timber) or for evaluating specific types of construction (girder, box beam, truss, arch, suspension).

The availability of many PC-based neural network programs makes the use of neural networks simple and affordable. The *Neural Network Toolbox* (which runs with *MATLAB*) provides several different training algorithms that can be used to design a variety of neural network architectures. Its flexibility also enables the user to customize existing training algorithms and create new ones (Beale and Demuth 1992).

The results of this study show that there is a great potential for using neural networks for infrastructure system evaluation. Other infrastructure systems that may benefit from neural network applications include roads and highways, power plants, airports, railroads, dams and sewer systems. Consistency in future evaluations is just one of the benefits of using neural networks. Their versatility also provides many advantages over other traditional methods of data analysis such as regression analysis and expert systems. The continuing refinement of computer technology also makes neural networks an increasingly attractive tool.

In addition, unlike the regression analysis, expert system, and even the Sufficient Ratings formula, neural network architectures can be developed to incorporate the uncertainties associated with the input parameters through probability and fuzzy set theory. Math Works Inc. has already developed a toolbox called *Fuzzy Neural Network Toolbox* to be used with *MATHLAB* for analyzing subjective data like those identified in Table 1.

Acknowledgements

The authors are grateful to the financial support by the National Science Foundation under Grant No. CMS-9402049 and to the support of the grant's program officer Dr. Ken P. Chong. The authors also appreciate the helpful and constructed comments offered by the reviewers.

References

Accarain, P., and Desbrandes, R. (1993), "Neuro-computing helps pore pressure determination", *Petroleum Engineer Int.*, **65**, February, 39-42.

ASCE (2001), "The 2001 Report Card for Americas Infrastructure", *American Society of Civil Engineers*, http://www.asce.org/reportcard/index.cfm.

Beale, M., and Demuth, H. (1992), *Neural Network Toolbox*, The MathWorks, Inc., Natick, MA.

Boger, Z. (1992), "Application of neural networks to water and wastewater treatment plant operation", *ISA Transactions*, **31**(1), 25-33.

Cabral, S.V., and Katafygiotis, L.S. (2001), "Neural network based response surface method and adaptive importance sampling for reliability analysis of large structural systems", *Proc. of 8th* ICOSSAR'01, Newport Beach, CA, USA, (in print).

Cheon, S.W., and Chang, S.H. (1993), "Application of neural networks to a connectionist expert system for transient identification in nuclear power plants", *Nuclear Technology*, **102**, May, 177-191.

Chou, K.C., and Yuan, J. (1993), "Fuzzy-Bayesian approach to reliability of existing structures", *J. Struct. Eng.*, ASCE, **119**(11), November, 3276-3290.

Cybenko, G. (1989), "Approximation by superposition of a sigmoidal function", *Mathematics of Control, Signals, and Systems*, **2**, 303-314.

Eberhart, R.C., and Dobbins, R.W. (1990), *Neural Network PC Tools: A Practical Guide*, Academic Press, San Diego, CA.

Fa-Long, L., Zheng, B., and Xiao-Peng, Z. (1992), "Real-time implementation of 'propagator' bearing estimation algorithm by use of neural network", *IEEE J. Oceanic Eng.*, **17**(4), October, 320-325.

Fan, J.Y., Nikolaou, M., and White, R.E. (1993), "An approach to fault diagnosis of chemical processes via neural networks", *AIChE J.*, **39**, January, 82-88.

Feng, M.Q., and Bahng, E.Y. (1999), "Damage assessment of jacketed RC columns using vibration tests", *J. Struct. Eng.*, ASCE, **125**(3), March, 265-271.

Funahashi, K. (1989), "On the approximate realization of continuous mappings by neural network", *Neural Networks*, **2**, 183-192.

Fwa, T.F., and Chan, W.T. (1993), "Priority rating of highway maintenance needs by neural networks", *J. Transp. Eng.*, ASCE, **119**(3), May/June, 419-432.

Haykin, S. (1994), *Neural Networks*, Macmillan College Publishing Company, Inc., Englewood Cliffs, NJ.

Hect-Nielsen, R. (1990), *Neurocomputing*, Addison-Wesley Publishing Co., Reading, MA.

Hornik, K., Stinchcombe, M., and White, H. (1989), "Multilayer feedforward network are universal approximators" *Neural Networks*, **2**, 359-366.

Hoskin, J.C., Kaliyur, K.M., and Himmelblau, D.M. (1991), “Fault diagnosis in complex chemical plants using artificial neural networks”, *AIChE J.*, **37**(1), January, 137-141.

Hung, S-L, Kao, C.Y., and Lee, J.C. (2000), “Active pulse structural control using artificial neural networks”, *J. Eng. Mech.*, ASCE, **126**(8), Aug., 839-849.

IEEE (1991), *First IEEE Conference on Ocean Engineering (CNNOE)*, IEEE Press, NY.

Kim, J-T., Jung H-J., and Lee, I-W. (2000), “Optimal structural control using neural networks”, *J. Eng. Mech.*, ASCE, **126**(2), Feb., 201-205.

Levenberg, K. (1944), “A method for the solution of certain non-linear problems in least squares”, *Quart. J. Appl. Math.*, **2**, 164-168.

Li, Z., Mu, B., and Peng, J. (2000), “Alkali-silica reaction of concrete with admixtures”, *J. Eng. Mech.*, ASCE, **126**(3), March, 243-249.

Liut, D.A., Matheu, E.E., Singh, M.P., and Mook, D.T. (1999), “Neural network control of building structures by a force-matching training scheme”, *Earthq. Eng. and Struct. Dyn.*, **28**(12), 1601-1620.

Loh, C-H., Lin, C-Y., and Huang, C-C. (2000), “Time domain identification of frames under earthquake loadings”, *J. Eng. Mech.*, ASCE, **126**(7), July, 693-703.

Marwala, T. (2000), “Damage identification using committee of neural networks”, *J. Eng. Mech.*, ASCE, **126**(1), Jan., 43-50.

Masri, S.F., Smyth, A.W., Chassiakos, A.K., Caughey, T.K. and Hunter, N.F. (2000) “Application of neural networks for detection of changes in nonlinear systems”, *J. Eng. Mech.*, ASCE, **126**(7), July, 666-676.

Masters, T. (1993), *Practical Neural Network Recipes in C++*, Academic Press, Inc., San Diego, CA.

Molina, A.V. (1996), “Evaluation of infrastructure systems using neural networks”, Master Thesis, The University of Tennessee, Knoxville, TN, 95.

Moselhi, O., Hegazy, T., and Fazio, P. (1991) “Neural networks as tools in construction”, *J. Const. Eng. and Man.*, ASCE, **117**(4), December, 606-624.

Ohga, Y., and Seki, H. (1993), “Abnormal event identification in nuclear power plants using a neural network and knowledge processing”, *Nuclear Technology*, **101**, February, 159-167.

Sankarasubramanian, G., and Rajasekaran, S. (1996), “Constitutive modeling of concrete using a new failure criterion”, *Comp. and Struct.*, **58**(5), March, 1003-1014.

Sasaki, T. (2001), “A neural network based response surface approach for computing failure probabilities”, *Proc. of 8th ICOSSAR’01*, Newport Beach, CA, USA, (in print).

Silven, S. (1992), “A neural approach to the assignment algorithm for multiple-target tracking”, *IEEE J. Oceanic Eng.*, **17**(4), October, 326-332.

U.S. Department of Transportation/Federal Highway Administration (1979) *Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges*, Design & Inspection Branch Bridge Division, Washington, D.C.

Vanluchene, R.D., and Sun, R. (1990), “Neural networks in structural engineering”, *Microcomputers in Civil Engineering*, **9**(3), Sept., 207-215.

Venugopal, K.P., Sudhakar, R., and Pandya, A.S. (1992), “On-line learning control of autonomous underwater vehicles using feedforward neural networks”, *IEEE J. of Oceanic Eng.*, **17**, October, 308-319.

Wassermann, P.D. (1989), *Neural Computing: Theory and Practice*, Van Nostrand Reinhold, NY.

Watanabe, K., Matsuura, I., and Abe, M. (1989), “Incipient fault diagnosis of chemical processes via artificial neural networks”, *AIChE J.*, **35**, November, 1803-1812.

Wong, F.S., Chou, K.C., and Yao, J.T.P. (1999), “Civil engineering including earthquake engineering”, Chapter 6 of *Practical Applications of Fuzzy Technologies*, H-J Zimmermann editor, **7** of *The Handbooks of Fuzzy Sets Series*, Kluwer Academic Publisher, 207-246.

Zhao, J., Ivan, J.N., and DeWolf, J.T. (1998), “Structural damage detection using artificial neural networks”, *J. Infrastructure Systems*, **4**(3), Sept., 93-101.

Appendix: sufficient ratings calculations

The Sufficient Ratings *SR* as defined in Eq. (5) and repeated here is composed of 4 categories, S_1, S_2, S_3, and S_4.

$$SR = S_1 + S_2 + S_3 - S_4 \tag{5}$$

The Sufficient Ratings has a value between 0% and 100%. The 4 categories are defined as:

S_1 = structural adequacy and safety and has a value ranges from 0% to 55%

S_2 = serviceability and functional obsoleteness and has a value ranges from 0% to 30%

S_3 = essential public use and has a value ranges from 0% to 15%

S_4 = special reductions and has a values ranges from 0% to 13%

The following sections describe the formulation used to determine the values for each of these 4 categories. Note that the items used to determine SR are listed in Table 1.

Table A.1 Determination of values for *A* through *D* in % used in Eq. (A.1)

Item 59	Item 60	*A*	*B*	*C*	*D*
0-2	0-2	55	0	0	0
	3	55	40	0	0
	4	55	0	25	0
	5-9	55	0	0	0
3	0-2	55	40	0	0
	3	0	40	0	0
	4	0	40	25	0
	5-9	0	40	0	0
4	0-2	55	0	25	0
	3	0	40	25	0
	4	0	0	25	0
	5-9	0	0	25	0
5	0-2	55	0	0	0
	3	0	40	0	0
	4	0	0	25	0
	5-9	0	0	0	10
6-9	0-2	55	0	0	0
	3	0	40	0	0
	4	0	0	25	0
	5	0	0	0	10
	6-9	0	0	0	0

Structural adequacy and safety, S_1

The item numbers used for this category are 59, 60, 62 and 66. The value for S_1 is defined as

$$S_1 = 55 - (A + B + C + D + E + F + G + H + I) \tag{A.1}$$

and S_1 shall not be less than 0% nor greater than 55%. Values for *A* through *D* can be determined from Table A.1. Values for *E* through *H* are determined according to Table A.2. Values for *E* through *H* equal zero if either item 59 or item 60 is not equal to “NA”. The value for *I* is defined as

$$I = 0.2778(36–AIT)^{1.5} \tag{A.2}$$

Table A.2 Determination of values for E through H in % used in Eq. (A.1)

Item 59 = Item 60 = NA and Item 62 =	E	F	G	H
0-2	55	0	0	0
3	0	40	0	0
4	0	0	25	0
5	0	0	0	10
6-9	0	0	0	0

Table A.3 *AIT* Factors

First Digit of Item 66	1	2	3	4	5	6	9
AIT Factors	1.56	1.00	1.56	1.01	0.77	0.67	1.00

where AIT = the adjusted inventory tonnage = the value from the second and third digits of item 66 times a AIT factor. The AIT factor depends on the value from the first digit of item 66 and is given in Table A.3. Note that if $(36\text{-}AIT)^{1.5} \leq 0$, then $I = 0$. I ranges from 0% to 55%.

Serviceability and functional obsoleteness, S_2

This category requires item numbers 12, 19, 28,, 29, 32, 51, 53, 58, 67, 68, 69, 71, and 72. The value for S_2 is computed as

$$S_2 = 30 - [J + (G + H) + I\,] \tag{A.3}$$

in which S_2 has a value between 0% and 30%; J is a rating reduction and it cannot exceed 13%; G and H are for width of roadway insufficiency where G and H combined cannot exceed 15%; and I is for vertical clearance insufficiency and it cannot exceed 2%. The values for these variables are determined in the following subsections.

Rating reductions, A through F

The rating reductions J is equal to the sum of values A through F which are determined according to Table A.4. In the Table, Column 1 gives the rating for each item number listed in columns 2 through 7. Columns 2 through 7 give the values in % for variables A through F depending on the item number listed in the first row. For example, if item 69 has a rating of 5, then D = 1%.

Table A.4 Values A through F in % for computing rating reductions J

Item values	Item 58, A	Item 67, B	Item 68, C	Item 69, D	Item 71, E	Item 72, F
≤ 3	5	4	4	4	4	4
4	3	2	2	2	2	2
5	1	1	1	1	1	1
others	0	0	0	0	0	0

Width of roadway insufficiency, G and H

The width of roadway insufficiency is composed of 2 variables, G and H. If the last 2 digits of item 43 are not 19 and the value of item 51 plus 2 ft. is less than the value of item 32, then G = 5%; otherwise, G = 0%. To determine the value for H, one needs to first compute X and Y where

$$X = \frac{\text{value of item 29}}{\text{first 2 digits of item 28}} \tag{A.4}$$

$$Y = \frac{\text{value of item 51}}{\text{first 2 digits of item 28}} \tag{A.5}$$

The value of H depends on the values of item 28, X and Y; and H is determined according to Table A.5.

Vertical clearance insufficiency, I

The vertical clearance insufficiency, I, has a value of either 0% or 2% according to the following:

1. If item 12 > 0, and item 53 < 1600, then I = 2%, otherwise, I = 0%.
2. If item 12 = 0, and item 53 < 1400, then I = 2%, otherwise, I = 0%

Essential public use, S_3

The category S_3 is defined as

Table A.5 Determination of value H in % for Eq. (A.3)

Item 28	X	Y	H
1		≤ 14	15
		$14 \leq Y < 18$	15(18−Y) ÷ 4
		≥ 18	0
2		≥ 16	0
		≥ 15	
		≥ 14	
		≥ 12	
≥ 2	>50	<9	15
	≤ 50	<9	7.5
	≤ 50	≥ 9	0
	$50 < X \leq 125$	<10	15
		$10 \leq Y < 13$	15(13−Y) ÷ 3
		≥ 13	0
	$125 < X \leq 375$	<11	15
		$11 \leq Y < 14$	15(14−Y) ÷ 3
		≥ 14	0
	$375 < X \leq 1350$	<12	15
		$12 \leq Y < 16$	15(16−Y) ÷ 4
		≥ 16	0
	>1350	<15	15
		$15 \leq Y < 16$	15(16−Y)
		≥ 16	0

$$S_3 = 15 - (A + B) \tag{A.6}$$

in which A shall not be less than 0% nor greater than 15%; and B is either 0% if item 12 = 0 or 2% if item 12 > 0. The value A is computed as

$$A = \frac{15(\text{value of item 29})(\text{value of item 19})}{200000K} \tag{A.7}$$

where

$$K = \frac{S_1 + S_2}{85} \tag{A.8}$$

Special reductions, S_4

The special reduction value S_4 applies only when the sum of the previous 3 categories, S_1, S_2, and S_3 is at least 50%; otherwise, S_4 = 0%. In addition to the values from the last 3 categories, S_4 also depends on the values from items 19, 36 and 43. Hence, when $(S_1 + S_2 + S_3) \geq 50\%$,

$$S_4 = A + B + C \tag{A.9}$$

in which

$$A = 5.205 \times 10^{-8} \text{ (value of item 19)}^4 \tag{A.10}$$

B = 5% if the second and third digits of item 43 are 10, 12 through 17 inclusive; and

$$C = \begin{cases} 1\% & \text{if 2 digits of item 36=0} \\ 2\% & \text{if 3 digits of item 36=0} \\ 3\% & \text{if 4 digits of item 36=0} \end{cases} \tag{A.11}$$

Sample calculations

The Sufficient Ratings calculations are illustrated using one of the 447 samples provided by TDOT. The ratings were recorded as shown in Table A.6.

Table A.6 Observed bridge ratings for items used in sufficient ratings calculations

Item #	Ratings	Item #	Ratings	Item #	Ratings	Item #	Ratings	Item #	Ratings
12	0	19	5	28	02	29	550	32	20
36	0000	43	102	51	20.6	53	unknown	58	5
59	4	60	5	62	NA	66	127	67	5
68	3	69	NA	71	6	72	6		

1. Structural adequacy and safety, S_1

Since item 66 = 227, *AIT* factor = 1.00 and *AIT* = 1.00*27 = 27. According to Eq. (A.2),

$$I = 0.2778(36–27)^{1.5} = 7.501\%$$

For item 59 = 4 and item 60 = 5, Tables A.1 and A.2 give C = 25% and the rest, A, B, and D through H, equal zero. This yields,

$$S_1 = 55 - (25 + 7.501) = 22.499\%$$

2. Serviceability and functional obsoleteness, S_2

When item 58 = 5, item 67 = 5, item 68 = 3, item 69 = NA, item 71 = 6 and item 72 = 6, these yield A = 1%, B = 1%, C = 4%, $D = E = F$ = 0%, respectively. The sum of these gives J = 6%.

Since last 2 digits of item 43 = 02 and (item 51 + 2) = 20.6 + 2 = 22.6 is greater than 20 (item 32), therefore, G = 0%.

With item 28 = 02, item 29 = 550, and item 51 = 20.6; X = 275 and Y = 10.3 (Eqs. A.4 and A.5). Using Table A.5, H becomes 15% and $G + H$ = 15% which is within the allowable maximum.

Since item 12 = 0 and item 53 = unknown, use I = 0%.

$$S_2 = 30 - [6 + (0 + 15) + 0] = 9\%$$

3. Essential public use, S_3

Item 12 = 0, hence B = 0.

$$K = \frac{22.499 + 9}{85} = 0.371 \qquad \text{yields} \qquad A = \frac{15(550)(5)}{200000(0.371)} = 0.557$$

The above gives S_3 = 15–(0.557 + 0) = 14.443.

4. Special reductions, S_4

Check to see if special reductions can be applied: 22.499 + 9 + 14.443 = 45.942% which is less than 50%. Therefore, S_4 = 0%, and the Sufficient Ratings SR = 45.942%.

Symbols

DOT = Department of Transportation
$\boldsymbol{I}$ = identity matrix
$\boldsymbol{J}$ = Jacobian matrix of derivatives of each error to each bias or weight
SR = Sufficiency Rating
SSE = sum-squared error
S_1 = Structural Adequacy and Safety
S_2 = Serviceability and Functional Obsoleteness
S_3 = Essential Public Use
S_4 = Special Reductions
b_i = bias
e = error vector
w_i = weight
x_i = input variable
y_i = output variable
$\Delta\boldsymbol{w}$ = weight change
$\Delta\boldsymbol{b}$ = bias change
μ = scalar