Transaction A: Civil Engineering
Vol. 16, No. 1, pp. 98–105
© Sharif University of Technology, February 2009

Effect of Asphalt Content on the Marshall Stability of Asphalt Concrete Using Artificial Neural Networks

M. Saffarzadeh[1,*] and A. Heidaripanah[1]

Abstract. *The Marshall Stability of asphalt concrete is one of the most important parameters in mix design and quality control. This property depends on many factors such as gradation, percentage of crushed aggregates, asphalt content and construction quality. In this research, the variation of Marshall Stability with asphalt content is simulated using Artificial Neural Networks (ANNs) with a Levenberg-Marquardt Back Propagation (LMBP) training algorithm. The percentage of crushed aggregates; the percentage passing through sieve numbers 200, 50, 30, 8, 4 and 1/2 inch, and the percentage of asphalt content are considered as network inputs and Marshall Stability as the network output. In the first stage, the maximum generalization ability of each network with a specified number of neurons in the hidden layer is determined. Comparing these maximum values reveals that the network with 8 neurons in the hidden layer has the maximum generalization ability. In the second stage, the variation of Marshall Stability with asphalt content is simulated by applying a sensitivity analysis to the network with the maximum generalization ability. This simulation is in good agreement with theory.*

Keywords: *Marshall Stability; Asphalt concrete; Backpropagation; Sensitivity analysis; Mix design.*

INTRODUCTION

The application of ANNs to various fields of pavement engineering, such as pavement design, mix design and the prediction of long-term pavement performance, is in progress. Owusu–Ababio [1] modeled the skid resistance of flexible pavements using ANNs and compared neural networks and the regression method. Eldin and Senouci [2] developed an ANN model to predict the condition rating of rigid pavements. Owusu–Ababio [3] investigated the effect of neural network topology on flexible pavement cracking prediction. Jidong, Jian and Manjriker [4] developed an ANN model to forecast the pavement condition rating.

ANNs are valuable computational tools that are increasingly being used to simulate complex problems as an alternative to use more traditional techniques. Ceylan et al. [5] employed ANNs as pavement structural analysis tools for the rapid and accurate prediction of critical responses and deflection profiles of flexible pavements subjected to typical highway loadings.

In another successful application, Meier et al. [6] trained back propagation ANNs as surrogates for ELP analysis in a computer program for back calculating a pavement layer module and realized a 42 times increase in processing speed. Similar ANN applications were also reported by Meier and Rix [7], Guncunski and Krstic [8], Khazanovich and Roesler [9] and Kim [10].

The research NCHRP 1-37A project team has used ANNs as a rapid and powerful tool to analyze rigid pavements in the AASHTO 2002 design package. Yu, Darter and Khazanovich [11] provided the neural networks to compute stresses at the critical locations in Jointed Plain Concrete Pavements (JPCP).

In this paper, a variation of Marshall Stability with asphalt content is simulated by ANN, which uses LMBP training algorithms. To achieve this, some artificial neural networks are designed to estimate

1. *Department of Civil Engineering, Tarbiat Modares University, P.O. Box 14155-4838, Tehran, Iran.*
*. *Corresponding author. E-mail: saffar_m@modares.ac.ir*

Received 3 July 2006; received in revised form 20 October 2007; accepted 4 November 2007

Marshall Stability, based upon effective factors. The variation of Marshall Stability with asphalt content is then simulated by performing a sensitivity analysis. MATLAB 7 has been used as the main software package in this research.

DATA ACQUISITION

In order to collect the required data needed to design the networks and evaluate their generalization abilities, a database of 110 asphalt concrete specimens are taken from the road surface courses before compaction by a standard practice for sampling the bitumen paving (ASTM D979-89) method. All specimens are asphalt concrete with 0-19 mm and 0-25 mm gradation. The bitumen type is asphalt cement with a penetration grade of 60/70. These data have been extracted from the top layer of the following projects: Kerman-Mahan, Bardsir-Negar-Baft, Baghin-Rafsanjan and Kerman-Zarand roads.

All the Marshall Stability and extraction tests have been done at the Soil Mechanics Laboratory of Kerman province. By performing Marshall Stability (ASTM D1559) and extraction (ASTM D2172) tests, the following parameters of specimens were specified: Marshall Stability, bitumen content, gradation curve and percentage of crushed aggregates.

BACKPROPAGATION

The backpropagation algorithm, like the Least Mean Square (LMS) learning rule, is an algorithm of Steepest Decent (SD). This algorithm is actually a generalization of the least mean square method to the multi layered network with a non linear function.

During the training process in this method, a series of appropriate behavioral models, which are desirable to the network, is presented as follows [12]:

$$\mathbf{p}_1, \mathbf{t}_1, \mathbf{p}_2, \mathbf{t}_2, \cdots, \mathbf{p}_q, \mathbf{t}_q, \tag{1}$$

where:

$\mathbf{p}_q =$ target vector,
$\mathbf{t}_q =$ input vector.

When an input ($\mathbf{p}$) is applied to the network, the corresponding output ($\mathbf{a}$) of the network is compared to the corresponding target ($\mathbf{t}$). The learning rule is then used to adjust the weights and biases of the network, in order to move the network outputs closer to the targets, so that the Mean Squared Error (MSE) is minimized as follows:

$$\text{MSE} = \frac{\sum_{i=1}^{n}(t_i - a_i)^2}{n}, \tag{2}$$

where:

MSE = Mean Square Error,
t_i = ith element of target vector $\mathbf{t}$,
a_i = ith element of output vector $\mathbf{a}$,
n = number of output layer neurons, which is equal to the number of target vector elements.

The error function is defined as follows:

$$F(\mathbf{x}) = E[\mathbf{e}^2] = E[(\mathbf{t} - \mathbf{a})^2], \tag{3}$$

where:

$\mathbf{x}$ = weight and biases vector,
$\mathbf{e}$ = error vector,
$\mathbf{t}$ = target vector
$\mathbf{a}$ = output vector.

If the network has more than one input, then Equation 3 is generalized to:

$$\text{F}(\mathbf{x}) = \text{E}[\mathbf{e}^T\mathbf{e}] = E[(\mathbf{t} - \mathbf{a})^T][(\mathbf{t} - \mathbf{a})], \tag{4}$$

where:

$\mathbf{x}$ = weight and biases vector,
$\mathbf{e}$ = error vector,
$\mathbf{t}$ = target vector,
$\mathbf{a}$ = output vector.

Like the LMS rule, the mean squared error is estimated as:

$$\hat{F}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T(\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k)\mathbf{e}(k), \tag{5}$$

where errors in step k substitute the error estimation in Equation 4. The SD algorithm for MSE is as follows:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha\frac{\partial\hat{F}}{\partial w_{i,j}^m}, \tag{6}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha\frac{\partial\hat{F}}{\partial b_i^m}, \tag{7}$$

where:

$w_{i,j}^m$ = weight of the ith neuron of the mth layer that received the output of the jth neuron of the $m-1$th layer,
b_i^m = bias of the the ith neuron of the mth layer,
α = learning rate.

Through the training process, network parameters (weights and biases) change in such a manner that the network performance index or training error (MSE) is optimized. The optimization techniques are: Steepest Descent (SD), Newton Method (NM) and Conjugate Gradient (CG).

All the above methods are based on iteration. The algorithm starts with an initial value for x, like x_0 and the new value of x is obtained using the following equations [12]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k, \tag{8}$$

$$\triangle \mathbf{x}_k = (\mathbf{x}_{k+1} - \mathbf{x}_k) = \alpha_k \mathbf{p}_k, \tag{9}$$

where:

$\mathbf{x}_k$ = vector of variables (weight and biases) at step k,
$\mathbf{x}_{k+1}$ = updated vector of variables at step $k+1$,
α_k = step size,
$\mathbf{p}_k$ = search direction vector (a positive value).

Levenberg - Marquardt Backpropagation Algorithm

The LMBP algorithm is a variation of the Newton method, which was designed for minimizing functions that are the sum of squares of other nonlinear functions. This technique is formed as follows [12]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k), \tag{10}$$

$$\Delta \mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}^{-1} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k), \tag{11}$$

where:

$\mathbf{J}$ = Jacobian matrice,
$\mathbf{I}$ = unique matrice,
$\mathbf{V}$ = error vector,
$\nabla F(\mathbf{x})$ = gradient of F.

This algorithm has a very useful feature. When μ_k is increased, it approaches the steepest descent algorithm with a small learning rate [12].

$$\mathbf{x}_{k+1} \cong \mathbf{x}_k - \frac{1}{\mu_k} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) = \mathbf{x}_k - \frac{1}{2\mu_k} \nabla F(\mathbf{x}). \tag{12}$$

While μ_k is decreased to zero, the algorithm becomes Gauss-Newton. The algorithm begins with μ_k set to some small value (e.g., μ_k =0.01). If a step does not yield a smaller value for $F(x)$, then, the step is repeated with μ_k multiplied μ-inc, a factor which is greater than one (e.g., $\upsilon = 10$). Eventually $F(x)$ should decrease, since a small step is taken in the direction of the steepest descent. If a step does produce a smaller value for $F(x)$, then μ_k is divided by υ, or multiplied μ-dec, a factor which is smaller than the one for the next step, so that the algorithm will approach Gauss-Newton, which should provide faster convergence. The algorithm provides a nice compromise between the speed of Newton's method and guaranteed convergence of the steepest descent [12,13].

DEVELOPING ARTIFICIAL NEURAL NETWORK MODEL

The Marshall Stability of an asphalt concrete mixture depends on a variety of criteria, including the properties, gradation of aggregates and asphalt type. To develop the model, the percent aggregate passing through sieve numbers 200, 50, 30, 8, 4, and 1/2 inch, percent of crushed aggregates and asphalt content, are selected as input to the network and the Marshall Stability is selected as output. Therefore, the number of input layer neurons is eight and the output layer neuron is one. The input variables, output variable and the way that they change are tabulated in Table 1 [14].

The tangent sigmoid transfer function is selected for hidden layer neurons and the linear transfer function for the output layer. The inputs and outputs are normalized between -1 and 1 to improve the performance of the networks.

In order to investigate the optimum number of hidden layer neurons, the network generalization ability is assessed, based on different training errors for each network, with a specified number of hidden layer neurons.

Training and Testing

In this section, the optimum number of hidden layer neurons is determined, based on 85 data for training and 25 data to assess the generalization ability of the networks. In this study, 3, 5, 8 and 10 neurons are adopted for the hidden layer in the networks. In order to determine the optimum number of neurons to be used in the network, and the maximum generalization ability of each network with a specified number of hidden layer neurons, the following procedure is used:

1. Evaluation of generalization ability, based on the training error for each network, using a specified number of neurons in the hidden layer and determination of maximum network generalization ability;

Table 1. Network inputs and output and their ranges.

Network Inputs	**Range (%)**
Agg. passing #200	2.5-10
Agg. passing #50	8-19
Agg. passing #30	12-31
Agg. passing #8	31-54
Agg. passing #4	48-77
Agg. passing 1/2 inch	86-100
Crushed agg. (%)	50-92
Asphalt content	3.53-5.82
Network Output	**Range (K)**
Marshall Stability	866-1661

2. Determination of an optimum number of neurons in the hidden layer, based on comparing maximum generalization abilities.

Optimum Number of Hidden Layer Neurons

The training of the network with 3 neurons in the hidden layer is depicted in Figure 1. Based on this figure, the dashed line indicates the simulation error for new data versus the training cycles and the solid line indicates the training error or performance (MSE) versus the training cycles (epochs). Based on this figure, the variation of the generalization ability of the network can be assessed.

Based on Figure 1, the maximum generalization ability of the network occurs in the initial training cycles, so the training rate of the networks must be minimized. Therefore, the training parameters of the networks, μ-inc and μ-dec, are selected as close to 1.0 as possible. In the following sections, a comparative study is made, expressing the variation of the generalization ability of the networks versus the training errors, using 3, 5, 8 and 10 neurons in the hidden layer.

Network Testing Curves

In this section, the data required to assess the maximum generalization ability of the networks are tabulated in Tables 2 to 5 and curves 2 to 5. In Tables 2 to 5, the first column from the left shows the name of the network; the second, third and forth columns indicate μ, μ-inc and μ-dec, the latter two parameters of which are selected as close to 1 as possible to minimize the training rate in some networks. The fifth column shows the training error, at which the network training stops, which is named "Goal". The sixth column shows the network training error or performance, which is derived based on Equation 2. The last column shows the relative coefficient, which is determined based on performing linear regression between simulated values for new data and the targets. In order to assess the variation of the generalization ability of the network, a curve representing R versus MSE is expressed for each network with a specified number of neurons in the hidden layer.

All network names in the first column from the right of Tables 2 to 5 begin with R, which is an arbitrary letter. The letter T expresses the use of the Tangent sigmoid transfer function in the hidden layer. The number after T shows the number of neurons in the hidden layer, P is the first letter of the Purelin or linear transfer function, which is used in the output layer, and the number after P is an arbitrary number.

Comparing the maximum relative coefficients in Figures 2 to 5 shows that the maximum generalization ability is achieved for the RT8P4 network with 8 neurons in the hidden layer ($R = 0.768$), so the optimum value for the hidden layer neurons is selected to be 8. Based on the investigations made in this paper, an increase in the number of hidden layer neurons to

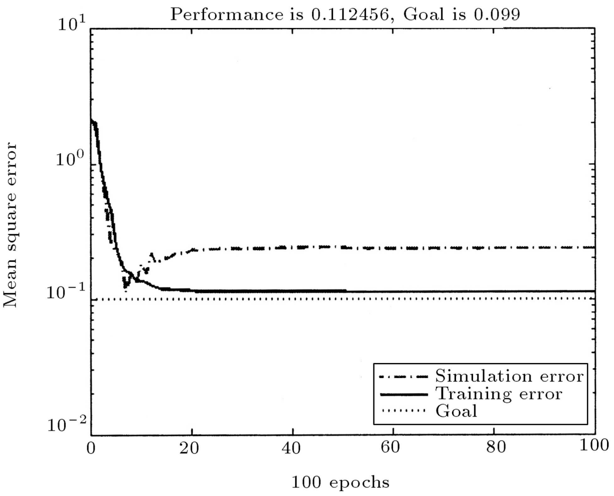

Figure 1. Variation of training and simulation error versus training cycles.

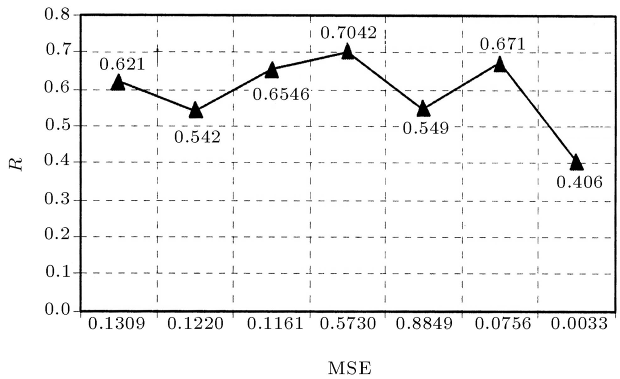

Figure 2. Variation of simulation ability (R) with training error (MSE) for the networks with 3 neurons in the hidden layer.

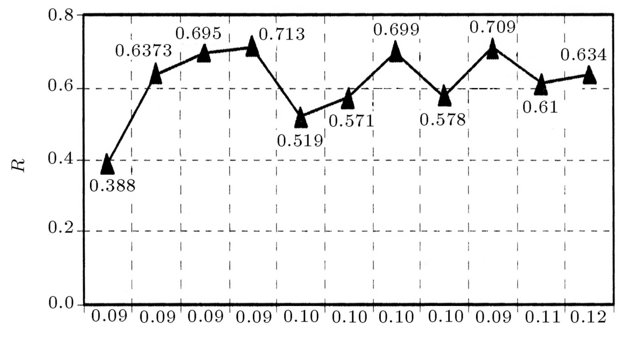

Figure 3. Variation of simulation ability (R) with training error (MSE) for the networks with 6 neurons in the hidden layer.

Table 2. Simulation results for the networks with 3 neurons in the hidden layer.

Network	μ	μ_inc	μ_dec	Goal	MSE	R
RT3P11	1.5	1.5	0.5	0.14	0.141081	0.561
RT3P10	1.5	1.5	0.5	0.1	0.130937	0.621
RT3P3	1.5	1.5	0.5	0.1	0.122038	0.542
RT3P4	1.5	1.5	0.5	0.11	0.116123	0.6546
RT3P5	**1.5**	**1.5**	**0.5**	**0.07**	**0.5730**	**0.7042**
RT3P6	0.001	10	0.1	0.09	0.08849	0.549
RT3P7	1.5	1.5	0.5	0.099	0.0756395	0.671
RT3P8	2	1.01	0.98	0.9	0.00331591	0.406

Table 3. Simulation results for the networks with 6 neurons in the hidden layer.

Network	μ	μ_inc	μ_dec	Goal	MSE	R
RT6P1	2	0.98	1.01	0.099	0.0866831	0.388
RT6P7	2	0.9	1.1	0.4	0.09186	0.6373
RT6P3	2	0.9	1.1	0.97	0.0933005	0.695
RT6P4	2	0.9	1.1	0.097	0.0941276	0.713
RT6P6	2	0.9	1.1	0.098	0.095434	0.519
RT6P2	2	0.9	1.1	0.099	0.0972685	0.571
RT6P5	2	0.9	1.1	0.098	0.975381	0.699
RT6P8	**2**	**0.9**	**1.1**	**0.1**	**0.0983244**	**0.709**
RT6P9	2	0.9	1.1	0.11	0.105976	0.61
RT6P10	2	0.9	1.1	0.11	0.119323	0.634

Table 4. Simulation results for the networks with 8 neurons in the hidden layer.

Network	μ	μ_inc	μ_dec	Goal	MSE	R
RT8P2	2	0.98	1.01	0.08	0.0789997	0.48
RT8P3	2	0.98	1.01	0.09	0.085589	0.7561
RT8P8	2	0.98	1.01	0.091	0.0869996	0.684
RT8P9	2	0.98	1.01	0.091	0.0880123	0.586
RT8P4	**2**	**0.98**	**1.01**	**0.09**	**0.0890116**	**0.768**
RT8P7	2	0.98	1.01	0.092	0.0904684	0.726
RT8P6	2	0.98	1.01	0.1	0.096805	0.578
RT8P5	2	0.98	1.01	0.095	0.0941677	0.737
RT8P10	**0.001**	**0.1**	**10**	**0**	**3.709 e-29**	**0.0109**

Table 5. Simulation results for the networks with 10 neurons in the hidden layer.

Network	μ	μ_inc	μ_dec	Goal	MSE	R
RT10P1	1.5	1.01	0.99	0	0.0640925	0.585
RT10P4	1.5	1.01	0.99	0.08	0.07732	0.488
RT10P5	1.5	1.01	0.99	0.08	0.087095	0.68
RT10P3	**1.5**	**1.01**	**0.99**	**0.08**	**0.09095**	**0.743**
RT10P7	1.5	1.01	0.99	0.093	0.0926634	0.65
RT10P2	1.5	1.01	0.99	0.093	0.0983409	0.683
RT10P6	1.5	1.01	0.99	0.08	0.101197	0.683

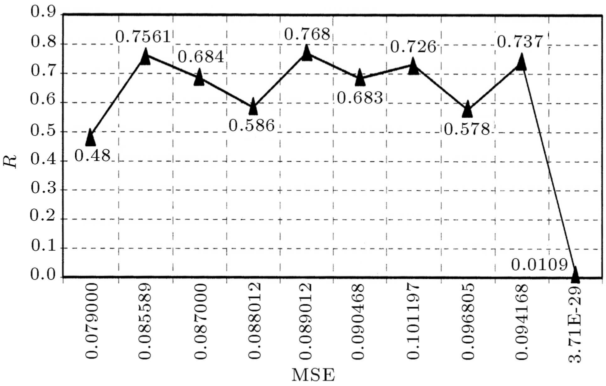

Figure 4. Variation of simulation ability (R) with training error (MSE) for the networks with 8 neurons in the hidden layer.

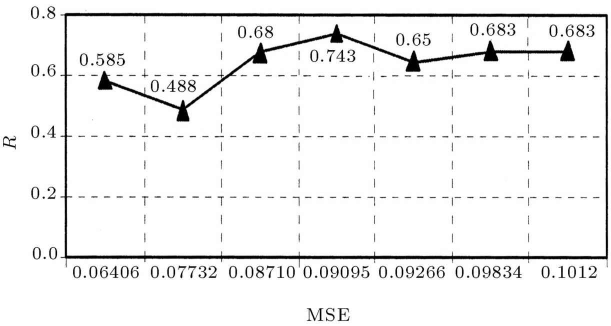

Figure 5. Variation of simulation ability (R) with training error (MSE) for the networks with 10 neurons in the hidden layer.

more than 8 has a negligible effect on generalization ability. The results show that the generalization ability of the networks is very sensitive to the training error, so determination of maximum generalization ability requires the designing and training of various networks.

On the other hand, in spite of reducing the training rate of the network, stopping the network training around the goal performance is difficult and requires iterations of weight initializing and retraining of the networks.

SENSITIVITY ANALYSES

The Marshall Stability variation with asphalt content, based on a sensitivity analysis for the developed ANN model, is performed in this section. For asphalt concrete specimens with various asphalt content, all network inputs except the asphalt content are considered constant. While the asphalt content is considered variable, the Marshall Stability of the specimens with various asphalt content is derived from the RT8P4 network, based on simulation. The results are shown in Table 6 and depicted in Figure 6.

Based on Figure 6, any increase in the percentage of asphalt content of the specimens increases Marshall

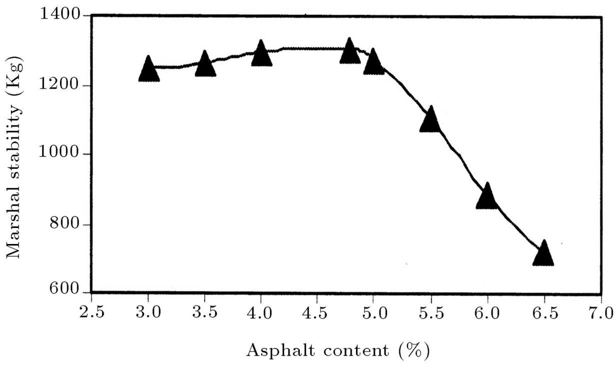

Figure 6. Simulation of Marshall Stability variation with asphalt content.

Stability as it reaches a maximum content, after which any increase in asphalt content leads to a decrease in Marshall Stability.

Based on this theory, the Marshall Stability reaches its maximum value at optimum asphalt content, at which the internal friction angle of the aggregates is maximum, so, any increase in asphalt content decreases the friction angle, which leads to lower values for Marshall Stability.

CONCLUSIONS AND RECOMMENDATIONS

In this paper, an ANN model is calibrated to assess the effect of asphalt content variation on Marshall Stability, using data from 110 Marshall Stability and 110 extraction tests. Eighty five data sets have been used for network training and 25 data sets used for testing the network generalization abilities. The following conclusions are made:

1. Based on Figures 2 to 5 and Tables 2 to 5, for a network with a specified number of neurons in the hidden layer, by decreasing training error, the generalization ability of the network increases as it reaches its maximum value. Then, the network overfits and the generalization ability decreases considerably. As seen in Figure 4 and Table 4, for an RT8P9 network with 8, when the training error (MSE) decreases to 3.709e-29, the simulation ability decreases to a very low volume of 0.109.
2. The generalization ability of all networks is sensitive to training error. As shown in Figures 2 to 5 and Tables 2 to 5, a very small change in training error (MSE) may cause a large variation in simulation ability (R). Therefore, in spite of reducing the training rate, stopping the training around the goal performance is difficult and requires the initializing and retraining of the networks several times over.
3. Based on comparing the maximum simulation abilities from Figures 2 to 5, the maximum gener-

Table 6. Variations of Marshall Stability of asphalt concrete specimen with various percentage of asphalt content.

Sieve No. 200	Sieve No. 50	Sieve No. 30	Sieve No. 8	Sieve No. 4	1/2 Inch	Crushed Aggregates (%)	Asphalt Content (%)	Marshall Stability (kg)
5	13	23	43	66	93	81	3	1248.4
5	13	23	43	66	93	81	3.5	1265.8
5	13	23	43	66	93	81	4	1298.7
5	13	23	43	66	93	81	4.79	1308.8
5	13	23	43	66	93	81	5	1275.1
5	13	23	43	66	93	81	5.5	1108.8
5	13	23	43	66	93	81	6	888.65
5	13	23	43	66	93	81	6.5	727.26

alization ability ($R = 0.768$) is achieved for the RT8P4 network with 8 neurons in the hidden layer. Comparing the maximum network generalization abilities in Figures 2 to 5 shows that, by increasing the number of neurons in the hidden layer up to 8, the network generalization ability increases.

4. Based on the developed ANN model and due to compatibility of the variation trend of the Marshal Stability of asphalt concrete with asphalt content, it is possible to calibrate the sensitivity analysis curve (Figure 6) for practical use by increasing the number of training data for the ANN model.

For future study, it is recommended to verify the sensitivity analysis results. For this purpose, the Marshall Stability of the specimens with the same gradation and percentage of crushed aggregates and various asphalt contents are determined by performing Marshall Stability tests. By comparing these values with simulation results, the relative coefficient can be determined.

The effect of other factors, such as aggregate type, aging and number of training data, on maximum network generalization ability and sensitivity analysis results, can further be investigated.

NOMENCLATURE

MSE	Mean Square Error
t_i	The ith element of target vector $\mathbf{t}$
a_i	The ith element of output vector $\mathbf{a}$
n	number of neurons of the output layer, which are equal to the number of elements of the target vector
$\mathbf{a}$	output vector
b_i^m	bias of the ith neuron of the mth layer
$\mathbf{e}$	error vector
$\mathbf{I}$	unique matrice
$\mathbf{J}$	Jacobian matrice
$\mathbf{p}_k$	search direction vector (is a positive value)
$\mathbf{p}_q$	qth input vector
$\mathbf{t}$	target vector
$\mathbf{t}_q$	qth target vector
$w_{i,j}^m$	weight of the ith neuron of the mth layer that received the output of the jth neuron of the $m-1$th layer
$\mathbf{x}$	weight and biases vector
$\mathbf{x}_k$	vector of variables (weight and biases) at step k
$\mathbf{x}_{k+1}$	updated vector of variables at step $k+1$
α	learning rate
α_k	step size
$\nabla F(\mathbf{x})$	gradient of F

REFERENCES

1. Owusu-Ababio, S. "Modeling skid resistance for flexible pavements: A comparison between regression and neural network methods", *Transportation Research Record*, **1501**, TRB, National Research Council, Washington DC (1995).

2. Eldin Neil, N. and Senouci, A.B. "Condition rating of rigid pavements by neural networks", *Can. J. of Civ. Eng.*, **22**, pp. 861-870 (1995).

3. Owusu-Ababio, S. "Effect of neural network topology on flexible pavement Cracking Prediction", *Computer-Aided Civil and Infrastructure Engineering*, **13**(5), pp. 349-355 (1998).

4. Jidong, Y., Jian Jan, L. and Manjriker, G. "Application of neural network models for forecasting of pavement crack index and pavement condition rating", *Transportation Program Development of Civil & Environmental Engineering*, University of South Florida, a Project Sponsored By Florida Department of Transportation, USA (2003).

5. Ceylan, H., Guclu, A., Tutumluer, E., Thompson, M.R. and Gomez-Ramirez, F. "Neural network-based structural models for rapid analysis of flexible pavements with unbound aggregate layers", *Proc., 6th Int. Symposium on Pavements Unbound (UNBAR 6)*, Nottingham, England, pp. 139-147 (July 2004).

6. Meier, R.W., Alexander, D.R. and Freeman, R. "Using artificial neural networks as a forward approach to backcalculation", *Transportation Research Record*, **1570**, TRB, National Research Council, Washington DC, pp. 126-133 (1997).

7. Meier, R.W. and Rix, G.J. "Backcalculation of flexible pavement moduli from dynamic deflection basins using artificial neural networks", *Transportation Research Record*, **1473**, TRB, National Research Council, Washington DC, pp. 72-81 (1995).

8. Gucunski, N. and Krstic, V. "Backcalculation of pavement profiles from spectral analysis-of-surface waves test by neural networks using individual receiver spacing approach", *Transportation Research Record*, **1526**, TRB, National Research Council, Washington DC, pp. 6-13 (1996).

9. Khazanovich, L. and Roesler, J. "DIPLOBACK: Neural-network-based backcalculation program for composite pavement", *Transportation Research Record*, **1570**, TRB, National Research Council, Washington DC, pp. 143-150 (1997).

10. Kim, Y. and Kim, R.Y. "Prediction of layer moduli from falling weight deflectometer and surface wave measurements using artificial neural network", *Transportation Research Record*, **1639**, National Research Council, Washington DC, pp. 53-61 (1998).

11. Guide for Mechanistic-Empirical Design of New and Rehabilitated Pavement Guide for Mechanistic-Empirical Design of New and Rehabilitated Pavement Structures, Final Document, "Appendix KK: Transverse cracking of jointed plain concrete pavements (JPCP)", Prepared for *National Cooperative Highway Research Program*, Transportation Research Board, National Research Council, Submitted by ARA, Inc., ERES Division 505 West University Avenue Champaign, Illinois 61820 (2003).

12. Hagan, M., Demuth, H. and Beale, M. "Neural network design", *PWS Publishing Company*, 20 Park Plaza, Boston, MA 02116-4324 (1996).

13. MATLAB 7 "Neural network toolbox user's guide", *Math Works Inc.* (2004).

14. Heidaripanah, A. "Estimation of marshall stability of asphalt concrete using artificial neural networks", M.Sc Thesis, University of Kerman, Iran (2003).