MultiCraft

International Journal of Engineering, Science and Technology
Vol. 2, No. 2, 2010, pp. 83-92

INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND TECHNOLOGY
www.ijest-ng.com



# Effect of training algorithms on neural networks aided pavement diagnosis

**Kasthurirangan Gopalakrishnan**

*Department of Civil, Construction, and Environmental Engineering, 354 Town Engineering Bldg., Iowa State University, Ames, IA 50011-3232, USA*
*e-mail: rangan@iastate.edu, Phone: (515) 294-3044 Fax: (515) 294-8216*

**Abstract**

Routine pavement maintenance necessitates present structural diagnosis and condition evaluation of pavements by employing non-destructive test equipment such as the Falling Weight Deflectometer (FWD). FWD testing of pavements involves measuring time-domain surface deflections resulting from applied impulse loading on the pavement. Through inverse analysis of FWD deflection data, the stiffness parameters of the individual pavement layers are, in general, determined using iterative optimization routines. In recent years, Neural Networks (NN) aided inverse analysis has emerged as a successful alternative for predicting pavement layer moduli from FWD deflection data in real-time. Especially, the use of Finite Element (FE) based pavement modeling results for training the NN aided inverse analysis is considered to be accurate in realistically characterizing the non-linear stress-sensitive response of underlying pavement layers in real-time. Efficient NN learning algorithms have been developed and proposed to determine the weights of the network, according to the data of the computational task to be performed. In this paper, the effect of training algorithms on the NN aided inversion process is analyzed and discussed.

*Keywords:* Neural networks; Non-destructive testing; Inverse analysis; Finite element; Flexible pavement.

## 1. Introduction

Roads deteriorate over time due to accumulated damage from vehicular traffic loading as well as environmental effects. Damage assessment is routinely carried out by engineers from public works, road authorities, or other public bodies that build and maintain the road system. According to recent statistics (ASCE 2009), poor road conditions cost U.S. motorists more than billions of dollars a year in repairs and operating costs. Thus, to improve the performance and serviceability of pavements, to reduce the cost to taxpayers, and to schedule proactive pavement repair and maintenance activities, consistent, cost-effective, and accurate monitoring of pavement is necessary.

A major portion of the road networks in the United States are reported to be composed of flexible pavement (Highway Statistics 2000). Flexible pavements are multi-layered, heterogeneous structures that are designed to “flex” under repeated traffic loading. A typical flexible pavement structure consists of the surface course (typically Hot-Mix Asphalt) at the top, underlying base and subbase (optional) courses (typically unbound granular material), and a subgrade (typically soil) at the bottom. In the field, Non-Destructive Testing (NDT) of in-service pavements using a Falling Weight Deflectometer (FWD) equipment is carried out to measure the deflection response of the pavement structure to applied dynamic load that simulates a moving wheel (see Figure 1). Using FWD test results, back-calculation of in-situ pavement layer moduli from measured deflections (inverse analysis) is carried out. The backcalculated pavement layer moduli are indicators of pavement layer condition as well as necessary inputs for mechanistic pavement structural analysis and thus remaining life calculations (Lytton, 1989; Ullidtz and Coetzee, 1995).

Most of the conventional commercial backcalculation programs involve Multi-Layer Elastic Theory (MLET) in their forward calculation routines and assume that pavement materials are linear-elastic, homogenous, and isotropic resulting in unrealistic backcalculated pavement layer moduli. Several research studies have shown that pavement geomaterials used in the underlying pavement layers exhibit non-linear, stress-sensitive behavior under repeated traffic loading. Unbound aggregates used in pavement base/sub-base course exhibit stress-hardening and fine-grained soils show stress-softening-type behavior (Brown and Pappin, 1981; Thompson and Elliot, 1985; Garg *et al*., 1998). Research studies have shown that non-linear analysis using Finite Element (FE) based approach increases the precision of the forward model (Gopalakrishnan *et al*., 2010).


Figure 1. Non-destructive testing of pavements using Falling Weight Deflectometer (FWD)

Recent research studies have focused on the development of Neural Networks (NN) based flexible pavement analysis models trained using FE solutions database to predict critical pavement responses and layer moduli (Ceylan, Guclu, Bayrak, and Gopalakrishnan 2007; Gopalakrishnan and Thompson 2004). The objective of this paper is to study the effect of different training algorithms on the performance of FE-NN backcalculation models which has not been done in previous studies.

**2. Neural networks aided inversion of pavement surface deflections**

NNs are parallel connectionist structures constructed to simulate the working network of neurons in the human brain. They attempt to achieve superior performance via dense interconnection of non-linear computational elements operating in parallel and arranged in a pattern reminiscent of a biological neural network. The perceptrons or processing elements and interconnections are the two primary elements which make up a neural network. A single perceptron is mathematically represented as follows (Hicks and Monismith 1971):

$$y_k = \varphi(v_j) = \varphi\left(\sum_{i=1}^{n} x_i w_{ij} - b_j\right) \tag{1}$$

where $x_i$ is input signal, $w_{ij}$ is synaptic weight, $b_j$ is bias value, $v_j$ is activation potential, $\varphi()$ is activation function, $y_k$ output signal, $n$ is the number of neurons for previous layer, and $k$ is the index of processing neuron.

Multilayer perceptrons (MLPs), frequently referred as multi-layer feedforward neural networks, consist of an input layer, one or more hidden layer, and an output layer and they have numerous applications in the engineering domain (Murali *et al*., 2010). Learning in a MLP is an unconstrained optimization problem, which is subject to the minimization of a global error function depending on the synaptic weights of the network. For a given training data consisting of input-output vectors, values of synaptic weights in a MLP are iteratively updated by a learning algorithm to approximate the target behavior. This update process is usually performed by backpropagating the error signal layer by layer and adapting synaptic weights with respect to the magnitude of error signal (Goktepe *et al*., 2004). Rumelhart *et al*., (1986) presented the first backpropagation learning algorithm (referred to as gradient-descent backpropagation) for use with MLP structures, which is typically employed in the development of FE-NN backcalculation models.

In the backpropagation algorithm, the error energy used for monitoring the progress toward convergence is the generalized value of all errors that is calculated by the least-squares formulation and represented by a Mean Squared Error (MSE) as follows (Haykin 1999):

$$MSE = \frac{1}{MP}\sum_{1}^{P}\sum_{k=1}^{M}\left(d_k - y_k\right)^2 \tag{2}$$

where $y_k$ are the actual outputs and $d_k$ are the desired outputs; M is the number of neurons in the output layer and P represents the total number of training patterns. The objective of this research is to study the effect of various training algorithms on the prediction performance of FE-NN backcalculation models.

A FE-NN backcalculation procedure was developed and implemented in MATLAB® to approximate the FWD backcalculation function. Using the synthetic training and testing dataset generated using a 2-D axisymmetric pavement FE program considering the nonlinear stress-dependent geomaterial characterization, the NN was trained to learn the relation between the synthetically generated surface deflection basins (inputs) and the pavement layer moduli (outputs).

A generic three-layer flexible pavement structure consisting of Asphalt Concrete (AC) surface layer, unbound aggregate base layer, and subgrade layer was modeled using the FE program. The top surface AC layer was characterized as a linear elastic material with Young's Modulus, $E_{AC}$, and Poisson ratio, ν. The K-θ model (Hicks and Monismith 1971)was used as the stress-dependent resilient modulus model for the unbound aggregate layer.

$$E_R = K\theta^n \tag{3}$$

where $E_R$ is resilient modulus, $\theta = \sigma_1 + \sigma_2 + \sigma_3 = \sigma_1 + 2\sigma_3$ = bulk stress, and K and n are multiple regression constants obtained from repeated load triaxial test data on granular materials. Previous research studies have shown that K and n model parameters can be correlated to characterize the non-linear stress dependent behavior with only one model parameter.

Fine-grained subgrade soils were modeled using the commonly used bi-linear resilient modulus model (Thompson and Elliot 1985):

$$\begin{aligned} E_R &= E_{Ri} + K_1.(\sigma_d - \sigma_{di}) \quad for\, \sigma_d < \sigma_{di} \\ E_R &= E_{Ri} + K_2.(\sigma_d - \sigma_{di}) \quad for\, \sigma_d > \sigma_{di} \end{aligned} \tag{4}$$

where $E_{Ri}$ is the breakpoint resilient modulus, $\sigma_d$ is the breakpoint deviator stress ($\sigma_d = \sigma_1 - \sigma_3$), $\sigma_{di}$ is the breakpoint deviator stress, and $K_1$ and $K_2$ are statistically determined coefficients from laboratory tests. As indicated by Thompson and Elliot (1985), the value of the resilient modulus at the breakpoint in the bilinear curve, $E_{Ri}$, can be used to classify fine-grained soils as being soft, medium or stiff. The $E_{Ri}$ is the main input for subgrade soils in ILLI-PAVE. The bilinear model parameters were set to default values. Also, the Asphalt Institute's Thickness Design Manual MS-1 recommends $E_{Ri}$ as the subgrade modulus input for ELP analysis.

Thus, asphalt concrete modulus, $E_{AC}$, granular base K-θ model parameter $K$, and the subgrade break-point resilient moduli, $E_{Ri}$, were used as the layer stiffness inputs for all the FE runs. The 40-kN wheel load was applied as a uniform pressure of 552 kPa over a circular area of radius 150 mm simulating the FWD loading. A comprehensive FE synthetic database was generated by varying the AC layer thickness (in the range of 75 to 700 mm), aggregate base layer thickness (in the range of 100 to 550 mm), $E_{AC}$ (in the range of 6.9 to 41.5 GPa), K (in the range of 21 to 82 MPa), and $E_{Ri}$ (in the range of 7 to 105 MPa) for NN training and testing. A total of 3,000 and 1,000 independent datasets, respectively, was used for training and testing the FE-NN backcalculation model. Two hidden layers were found to be sufficient in solving a problem of this size. Therefore the architecture was reduced to a four-layer feedforward network. Before discussing the effect of different training algorithms on FE-NN aided inversion of pavement deflection data for prediction of non-linear pavement layer stiffnesses, a brief review of the different training algorithms is presented first.

## 3. Neural networks training algorithms

The ability to 'learn' the mapping between inputs and outputs is one of the main advantages that make the NNs so attractive. Efficient learning algorithms have been developed and proposed to determine the weights of the network, according to the data of the computational task to be performed. The learning ability of the NNs makes them suitable for unknown and non-linear problem structures such as pattern recognition, medical diagnosis, time series prediction, and others.

Considerable research has been carried out to accelerate the convergence of learning algorithms which can be broadly classified into two categories: (1) development of ad-hoc heuristic techniques which include such ideas as varying the learning rate, using momentum and rescaling variables; (2) development of standard numerical optimization techniques. The three types of numerical optimization techniques commonly used for NN training include the conjugate gradient algorithms, quasi-Newton algorithms, and the Levenberg-Marquardt algorithm (Hagan and Menhaj 1994).

Although numerous training algorithms appear in recent neural network literature, it is difficult to know which algorithm works best, in terms of convergence speed and accuracy, for a given problem. A number of factors, including the complexity of the problem, the number of datasets used in training, the number of weights and biases in the network, the error goal, and whether the NN is used for function approximation or classification, etc., seem to have influence (Coskun and Yildrim 2003). The following sub-sections briefly describe the various NN training algorithms considered in this study.

### *3.1Gradient Descent Backpropagation (GD)*

The gradient descent backpropagation training algorithm is based on minimizing the mean square error between the network's output and the desired output. Once the network's error has decreased to the specified threshold level, the network is said to have converged and is considered to be trained. The GD backpropagation algorithm updates synaptic weights and biases along the negative gradient of the error energy function (MATLAB Toolbox, User's Guide, 2010).

*3.2 Gradient Descent with Momentum Backpropagation (GDM)*

This is a batch steepest descent training algorithm that often provides faster convergence. Momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low-pass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network may get stuck in a shallow local minimum. With momentum a network can slide through such a minimum (MATLAB Toolbox, User's Guide, 2010).

*3.3 Gradient Descent with Adaptive Learning Rate Backpropagation (GDA)*

In the standard steepest descent (GD) backpropagation algorithm, the learning rate parameter is held constant throughout training. However, the performance of the algorithm is very sensitive to the proper setting of the learning rate parameter. For this reason, the GDA algorithm was developed to allow the learning rate parameter to be adaptive to keep the learning step size as large as possible while keeping learning stable. In GDA algorithm, the optimal value of the learning rate parameter changes with the gradient's trajectory on the error surface (MATLAB Toolbox, User's Guide, 2010).

*3.4 Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation (GDX)*

The GDX training algorithm combines adaptive learning rate with momentum training. It is similar to GDA algorithm except that it has a momentum coefficient as an additional training parameter. Thus, the weight vector update is carried out the same way as in GDM except that a varying learning rate is used as in GDA (MATLAB Toolbox, User's Guide, 2010).

*3.5 Resilient Backpropagation (RP)*

To overcome the inherent disadvantages of pure gradient-descent, the RP training algorithm performs a local adaptation of the weight-updates according to the behavior of the error function. In contrast to other adaptive techniques, the effect of the RP adaptation process is not affected by the size of the derivative, but only by the temporal behavior of its sign leading to an efficient and transparent adaptation process. Resilient Backpropagation is generally much faster than the standard steepest descent algorithm although it requires only a modest increase in memory requirements (MATLAB Toolbox, User's Guide, 2010).

*3.6 Conjugate Gradient Backpropagation with Fletcher-Reeves Update (CGF)*

The basic backpropagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient), the direction in which the cost function is decreasing most rapidly. Although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions (MATLAB Toolbox, User's Guide, 2010).

All the conjugate gradient algorithms start out by searching in the steepest descent direction (negative of the gradient) on the first iteration. A line search is then performed to determine the optimal distance to move along the current search direction. Then the next search direction is determined so that it is conjugate to previous search directions. The general procedure for determining the new search direction is to combine the new steepest descent direction with the previous search direction. The various versions of the conjugate gradient algorithm are distinguished by the manner in which the constant associated with the determination of new search direction is computed. For the Fletcher-Reeves update, this constant is computed as the ratio of the norm squared of the current gradient to the norm squared of the previous gradient (MATLAB Toolbox, User's Guide, 2010).

*3.7 Conjugate Gradient Backpropagation with Polak-Ribiére Update (CGP)*

For the Polak-Ribiére update, the constant associated with the determination of new search direction is computed as the inner product of the previous change in the gradient with the current gradient divided by the norm squared of the previous gradient. The storage requirements for Polak-Ribiére (four vectors) are slightly larger than for Fletcher-Reeves (three vectors) (MATLAB Toolbox, User's Guide, 2010).

*3.8 Conjugate Gradient Backpropagation with Powell-Beale Restarts (CGB)*

For all conjugate gradient algorithms, the search direction is periodically reset to the negative of the gradient. The standard reset point occurs when the number of iterations is equal to the number of network parameters (weights and biases), but there are other reset methods that can improve the efficiency of training. One such reset method is the Powell-Beale Restarts. This technique restarts if there is very little orthogonality left between the current gradient and the previous gradient. The storage requirements for the Powell-Beale algorithm (six vectors) are slightly larger than for Polak-Ribiére (four vectors) (MATLAB Toolbox, User's Guide, 2010).

*3.9 Scaled Conjugate Gradient Backpropagation (SCG)*

The three conjugate gradient algorithms discussed so far require a line search at each iteration, which is computationally expensive, since it requires that the network response to all training inputs be computed several times for each search. The SCG training algorithm was developed to avoid this time-consuming line search, thus significantly reducing the number of computations performed in each iteration, although it may require more iterations to converge than the other conjugate gradient algorithms. The storage requirements for the SCG algorithm are about the same as those of CGF (MATLAB Toolbox, User's Guide, 2010).

*3.10 BFGS Quasi-Newton Backpropagation (BFGS)*

Newton's method is an alternative to the conjugate gradient methods for fast optimization. The Broyden–Fletcher–Golfarb–Shanno (BFGS) algorithm is one of the most popular of the quasi-Newton algorithms (Haykin 1999). The basic step of Newton's method is to form the Hessian Matrix (second derivatives). This method often converges faster than conjugate gradient methods but it is complex and expensive to compute the Hessian Matrix for feedforward neural networks. For smaller networks, however, BFGS can be an efficient training function (MATLAB Toolbox, User's Guide, 2010).

*3.11 One Step Secant Backpropagation (OSS)*

Since the BFGS algorithm requires more storage and computation in each iteration than the conjugate gradient algorithms, there is need for a secant approximation with smaller storage and computation requirements. The OSS training algorithm requires less storage and computation per epoch than the BFGS algorithm. It requires slightly more storage and computation per epoch than the conjugate gradient algorithms. Thus, the OSS method can be considered a compromise between full quasi-Newton algorithms and conjugate gradient algorithms (MATLAB Toolbox, User's Guide, 2010).

*3.12 Levenberg-Marquardt Backpropagation (LM)*

The LM second-order numerical optimization technique combines the advantages of Gauss–Newton and steepest descent algorithms. While this method has better convergence properties than the conventional backpropagation method, it requires $O(N^2)$ storage and calculations of order $O(N^2)$ where N is the total number of weights in an MLP backpropagation. The LM training algorithm is considered to be very efficient when training networks which have up to a few hundred weights. Although the computational requirements are much higher for each iteration of the LM training algorithm, this is more than made up for by the increased efficiency. This is especially true when high precision is required (MATLAB Toolbox, User's Guide, 2010).

*3.13 Bayesian Regularization Backpropagation (BR)*

The BR training algorithm is considered as one of the best approaches to overcome the over-fitting tendencies of NNs so that their prediction accuracies for unseen data can be further enhanced. This approach minimizes the over-fitting problem by taking into account the goodness-of-fit as well as the network architecture. The BR network training function updates the weight and bias values according to Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. This process is called Bayesian regularization (MATLAB Toolbox, User's Guide, 2010).

## 4. Effect of Training Algorithms on NN Aided Inversion of Deflection Data

Twelve different training algorithms discussed in the previous section were considered in evaluating their effect on the prediction performance of FE-NN based inversion of pavement deflection data. The network architecture was set to 8 input neurons (six surface deflections at equal radial offsets, and AC surface and base layer thicknesses), two hidden layers with 60 hidden neurons each and one output layer (layer modulus). The determination of optimum number of hidden neurons and hidden layers for the pavement modulus backcalculation problem considered in this study is discussed elsewhere (Ceylan, Guclu, Bayrak, and Gopalakrishnan 2007). Two different architectures, one for AC modulus, $E_{AC}$, and one for subgrade break-point non-linear resilient modulus, $E_{Ri}$, were used. To enable easy comparison, the size of the networks and the learning parameters were held constant in studying the effect of different training algorithms on prediction accuracies of FE-NN inversion models. The effect of gradient-descent backpropagation training algorithms (GD, GDM, GDA, GDX, and RP) on the training performance of FE-NN $E_{AC}$ prediction model is captured in Figure 2.

While training the networks, a set of validation vectors were used to stop training early if further training on the primary vectors will hurt generalization to the validation vectors. The effect of conjugate gradient training algorithms (CGF, CGP, CGB, and SCG) on the training performances of FE-NN $E_{AC}$ prediction model is displayed in Figure 3. Similar results showing the effect of other numerical optimization algorithms (BFGS, OSS, LM, and BR) on the training performance of FE-NN $E_{AC}$ prediction model are shown in Figure 4. It is seen that the learning curves of gradient-descent backpropagation algorithms and conjugate gradient training algorithms have different trends.

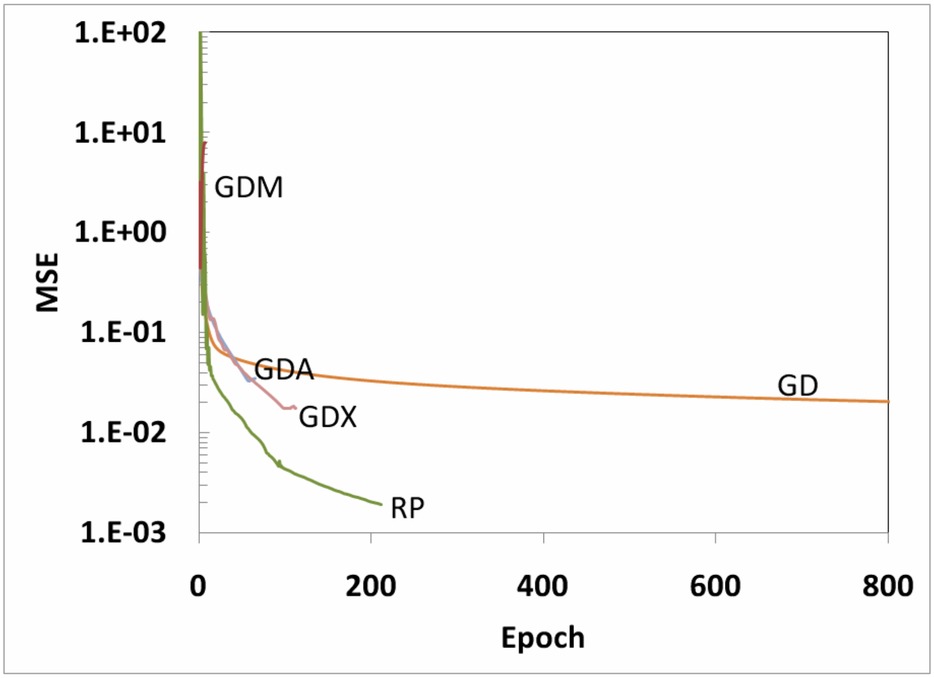


Figure 2. Effect of gradient-descent backpropagation learning algorithms on training performance of FE-NN AC modulus prediction model.

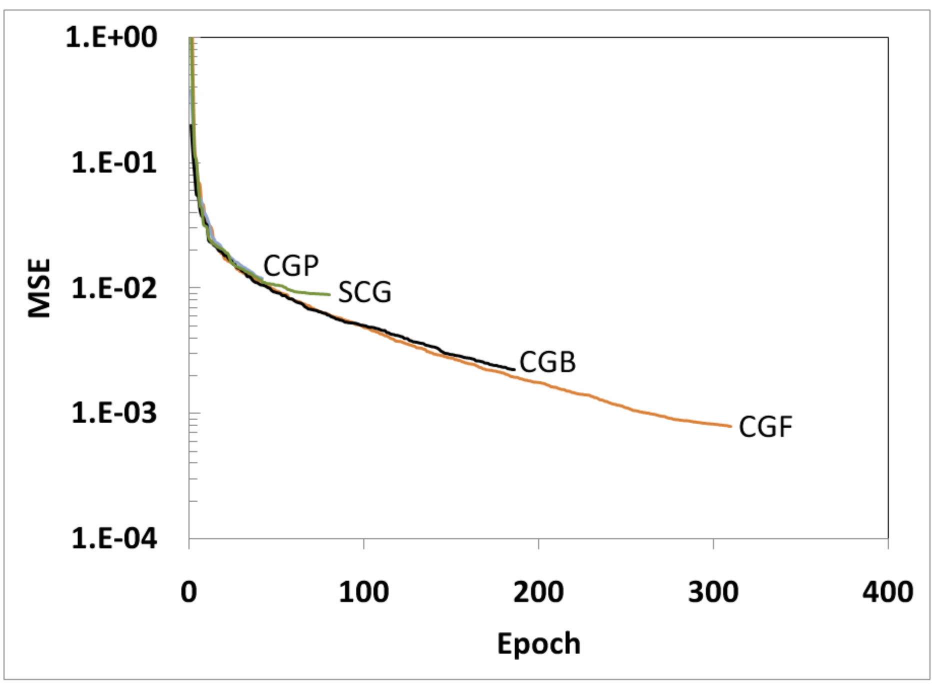


Figure 3. Effect of conjugate gradient backpropagation learning algorithms on training performance of FE-NN AC modulus prediction model

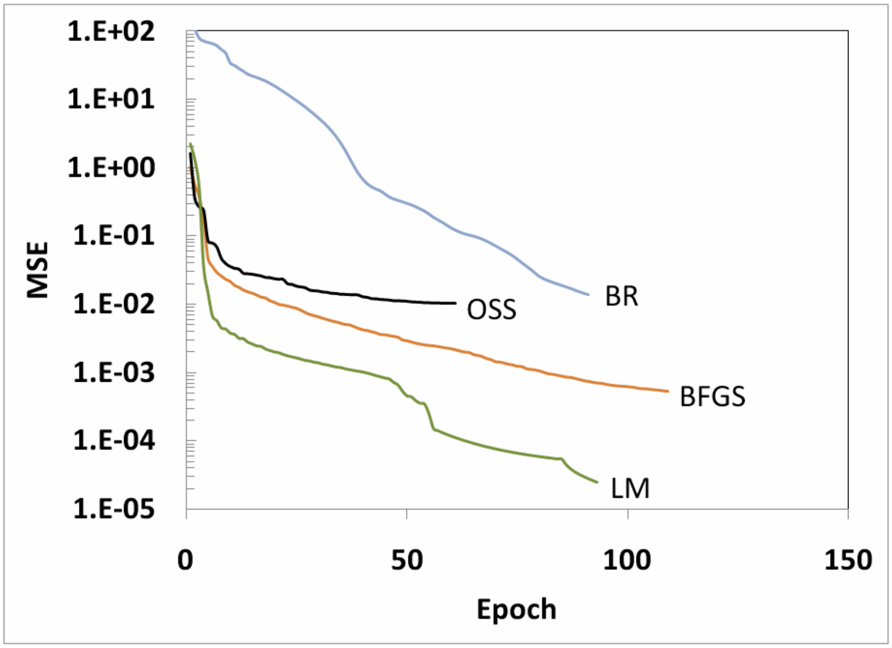


Figure 4. Effect of quasi-Newton and Levenberg-Marquardt backpropagation learning algorithms on training performance of FE-NN AC modulus prediction model

Table 1. $R^2$ values with Various NN Training Algorithms

| Algorithm | Modulus | Training | Testing |
|---|---|---|---|
| GD | $E_{AC}$ | 0.804 | 0.795 |
| | $E_{Ri}$ | 0.829 | 0.822 |
| GDM | $E_{AC}$ | 0.067 | 0.131 |
| | $E_{Ri}$ | 0.115 | 0.078 |
| GDA | $E_{AC}$ | 0.631 | 0.637 |
| | $E_{Ri}$ | 0.696 | 0.699 |
| GDX | $E_{AC}$ | 0.824 | 0.815 |
| | $E_{Ri}$ | 0.839 | 0.829 |
| RP | $E_{AC}$ | 0.978 | 0.972 |
| | $E_{Ri}$ | 0.229 | 0.248 |
| CGF | $E_{AC}$ | 0.992 | 0.990 |
| | $E_{Ri}$ | 0.978 | 0.977 |
| CGP | $E_{AC}$ | 0.864 | 0.861 |
| | $E_{Ri}$ | 0.973 | 0.973 |
| CGB | $E_{AC}$ | 0.978 | 0.975 |
| | $E_{Ri}$ | 0.985 | 0.984 |
| SCG | $E_{AC}$ | 0.915 | 0.906 |
| | $E_{Ri}$ | 0.963 | 0.964 |
| BFGS | $E_{AC}$ | 0.995 | 0.994 |
| | $E_{Ri}$ | 0.987 | 0.987 |
| OSS | $E_{AC}$ | 0.893 | 0.892 |
| | $E_{Ri}$ | 0.980 | 0.980 |
| LM | $E_{AC}$ | 0.999 | 0.998 |
| | $E_{Ri}$ | 0.999 | 0.998 |
| BR | $E_{AC}$ | 1.000 | 1.000 |
| | $E_{Ri}$ | 0.998 | 0.999 |

In Figure 2-4, the training performances of neural networks are displayed in terms of Mean Squared Error (MSE) for normalized input-output values. Table 1 summarizes the prediction performance results using different training algorithms in terms of coefficient of determination ($R^2$) values obtained with the training and testing datasets.

Both in terms of training and testing, the LM algorithm seems to provide the best performance. It was also one of the fastest algorithms in terms of speed of convergence. This result is also in agreement with the results of a previous research study which evaluated the role of learning algorithms in NN based backcalculation models trained with elastic layered pavement analysis program generated synthetic datasets (Goktepe, Agar, and Lav 2006).

Using the Bayesian Regularization (BR) approach, the performance is further enhanced although over-fitting tendency of the network for this problem is likely to be minimal considering the relatively large number of datasets used in training the network. The performance of FE-NN model trained with quasi-Newton BFGS algorithm is almost as good (with respect to the coefficient of determination) as with the LM algorithm, although the BFGS algorithm took considerably longer time for training. The prediction performance of the FE-NN models using LM and BFGS training algorithms on the 1,000 independent test datasets is displayed in Figs. 5 and 6. These results indicate that the LM training algorithm could be used quite reliably with backpropagation NN for predicting pavement layer modulus from FWD surface deflection basins. In terms of Average Absolute Errors (AAEs), the NN-based backcalculation models trained with LM learning algorithm successfully predicted pavement layer moduli values with an overall AAE value of less than 1.5%.

The adoption of NN-based approach can result in both a drastic reduction in computation time and a simplification of the backcalculation approach from the viewpoint of a pavement designer/analyst. Rapid prediction ability of the NN models, capable of analyzing 100,000 FWD deflection profiles in less than a second, provide a tremendous advantage to the pavement engineers by allowing them to nondestructively assess the condition of the transportation infrastructure system in real time, including when the FWD testing is conducted in the field.

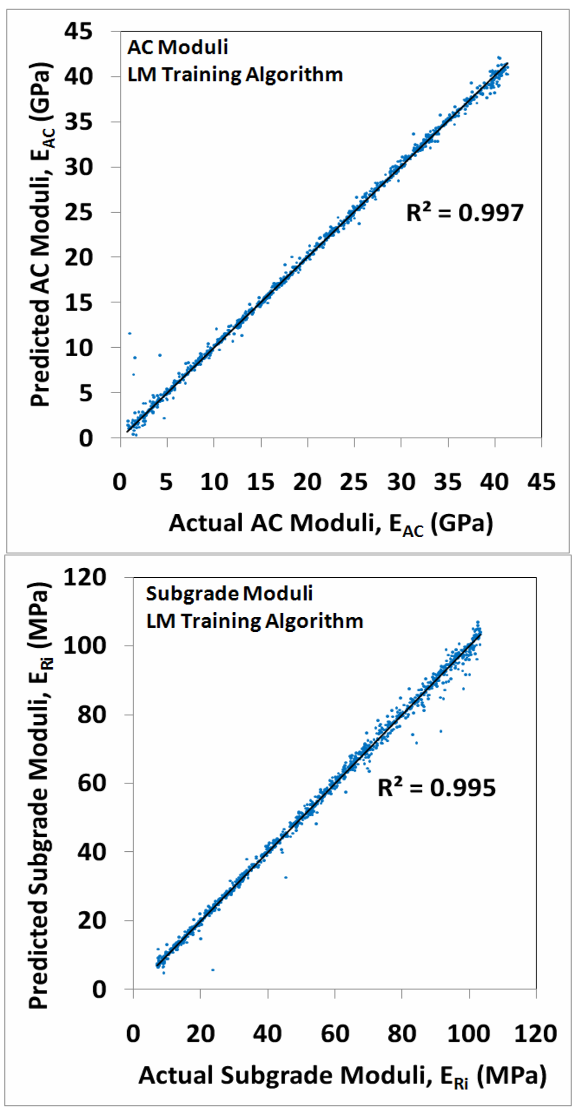


Figure 5. Neural networks AC and non-linear subgrade moduli prediction accuracies using Lavenberg-Marquardt (LM) training algorithm.

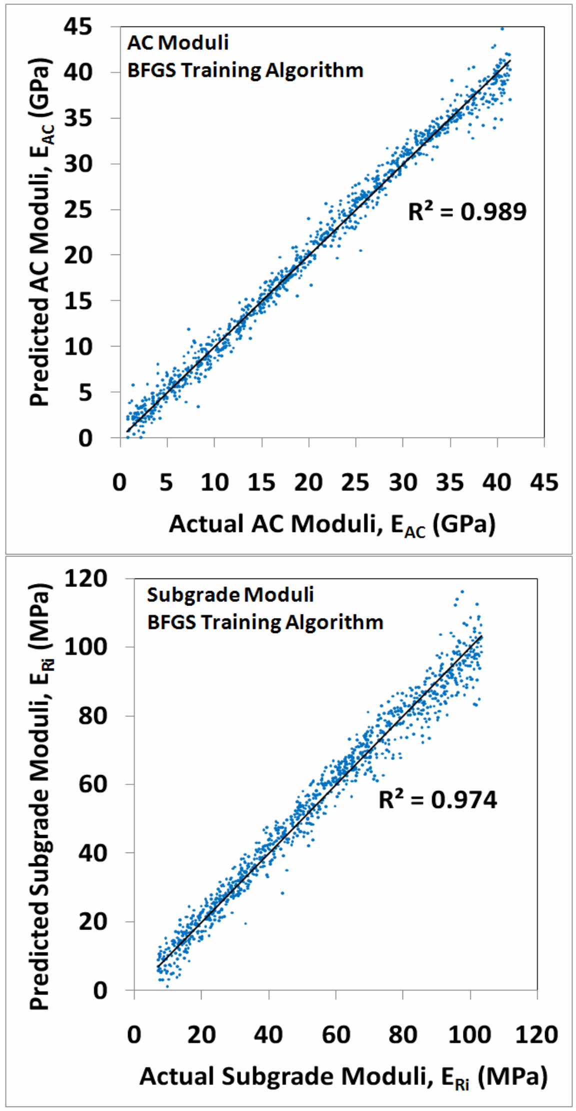


Figure 6. Neural networks AC and non-linear subgrade moduli prediction accuracies using BFGS training algorithm.

## 5. Conclusion

In the field, Non-Destructive Testing (NDT) of in-service pavements is routinely carried out using a Falling Weight Deflectometer (FWD) equipment to measure the surface deflection response of the pavement structure to applied dynamic load that simulates a moving wheel. Then, through an inversion procedure, referred to as backcalculation, the pavement layer stiffness properties are determined from the pavement deflection data using parameter identification routines. In recent years, hybrid Finite Element-Neural Networks (FE-NN) aided inverse analysis has emerged as a successful alternative for predicting non-linear pavement layer moduli from FWD deflection data. The effect of different training algorithms on the performance of FE-NN backcalculation models has been presented in this paper. The Lavenberg-Marquardt algorithm with Bayesian Regularization (BR) procedure was found to be the most successful algorithm for non-linear pavement layer moduli backcalculation using hybrid FE-NN inversion models. Elimination of the seed layer moduli selection step, combined with the integration of NN-based direct backcalculation approach, can be invaluable for the state and federal agencies for rapidly analyzing a large number of pavement deflection basins needed for routine pavement evaluation for both project-specific and network-level FWD testing.

Research studies have shown that to successfully backcalculate the pavement layer stiffness, or to predict the critical pavement responses (maximum stresses, strains and deflections), accurate layer thickness information is needed, especially at the FWD testing points. Future research efforts should focus on conducting sensitivity studies to determine the effect of pavement layer thickness on pavement performance data using the mechanistic-empirical based pavement design concepts. This will help to determine how much tolerance can be accommodated in assessing the pavement thickness by the means of NDT techniques and devices.

## References

ASCE. 2009 Report Card for America's Infrastructure. Available: http://www.infrastructurereportcard.org/sites/default/files/RC2009_full_report.pdf (accessed on Feb. 1, 2009) (URL).

Brown S. F. and Pappin J. W., 1981. Analysis of pavements with granular bases," *Transportation Research Record* 810, pp. 17-23.

Ceylan H., Guclu A., Bayrak M. B. and Gopalakrishnan K., 2007. Nondestructive evaluation of Iowa pavements-Phase I, CTRE, Iowa State University, Ames, IA, CTRE Project 04-177, Dec.

Coskun N. and Yildrim T., 2003. The effects of training algorithms in MLP network on image classification, in *Proc. Int. Joint Conf. on Neural Networks*, Vol. 2, pp. 1223-1226.

Garg N.E., Tutumluer E. and Thompson M. R., 1998. Structural modelling concepts for the design of airport pavements for heavy aircraft, in *Proc. Fifth International Conf. on the Bearing Capacity of Roads and Airfields*, Trondheim, Norway.

Goktepe A.B., Agar E., and Lav A. H., 2006. Role of learning algorithm in neural network-based backcalculation of flexible pavements, Technical Notes, *Journal of Computing in Civil Engineering*, Vol. 20, No. 5, pp. 370-373.

Gopalakrishnan K. and Thompson M.R., 2004. Backcalculation of airport flexible pavement non-linear moduli using artificial neural networks, in *Proc. 17th International FLAIRS Conference*, Miami Beach, Florida, May.

Gopalakrishnan, K., Ceylan, H. and Attoh-Okine, N. O. (Eds.), 2010. *Intelligent and Soft Computing in Infrastructure Systems: Recent Advances*, Springer, Inc., Germany.

Haykin S., 1999. *Neural networks: A comprehensive foundation*, Prentice-Hall, Inc., NJ, USA.

Hicks R. G. and Monismith C. L., 1971. Factors influencing the resilient properties of granular materials, *Transportation Research Record no. 345*, TRB, pp. 15-31.

Hagan M. T. and Menhaj M. B., 1994. Training feedforward networks with the Marquardt algorithm, *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, pp. 989-993.

Lytton R.L., 1989. Backcalculation of layer moduli, state of the art, In: NDT of pavements and backcalculation of moduli, A. J. Bush and G. Y. Baladi, editors., Vol. 1, *ASTM Special Technical Publication* (*STP*) 1026, pp. 7–38.

MATLAB Neural Network Toolbox™. User's Guide. Available: http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/ (accessed on Mar 3, 2010) (URL).

Murali, R. V., Puri, A. B., and Prabhakaran, G., 2010. Artificial neural networks based predictive model for worker assignment into virtual cells, *International Journal of Engineering, Science and Technology*, Vol. 2, No. 1, 2010, pp. 163-174.

Office of Highway Policy Information. 2004. Highway Statistics 2000, Federal Highway Administration, Washington, DC.

Rumelhart D.E., Hinton G. E., and Williams R. J., 1986. Learning internal representation by error propogation in Rumelhart, D.E. eds, *Parallel Distributed Processing*, MIT Press, Cambridge, MA, pp. 318-362.

Thompson M. R. and Elliot R. P., 1985. ILLI-PAVE based response algorithms for design of conventional flexible pavements, *Transportation Research Record* 1043, pp.50-57.

Ullidtz P. and Coetzee N. F., 1995. Analytical procedures in nondestructive testing pavement evaluation, *Transportation Research Record* 1482, pp. 61-66.

### Biographical notes

**Dr. Kasthurirangan Gopalakrishnan** is a Research Faculty in the Department of Civil, Construction and Environmental Engineering (CCEE) at Iowa State University, Ames, Iowa and is also affiliated with the Ames Lab, US Department of Energy and the Iowa Bioeconomy Institute. He obtained his Ph.D from University of Illinois at Urbana-Champaign in 2004 and M.S. from Louisiana State University in 2000. He is also a recipient of the Federal Highway Administration (FHWA) Dwight D. Eisenhower Graduate Research Fellowship. He has more than 100 peer-reviewed publications in the form of journal articles, articles in conference proceedings, and book chapters. He is also the lead editor of the following Springer book publications: (1) *Intelligent and Soft Computing in Infrastructure Systems Engineering: Recent Advances*; (2) *Sustainable and Resilient Critical Infrastructure Systems: Simulation, Modeling, and Intelligent Engineering*. His research interests include green energy and technology in civil infrastructure, sustainable systems engineering, nanotechnology and MEMS in transportation structures, knowledge discovery and intelligent data mining paradigms for civil engineering informatics.

Received April 2010
Accepted April 2010
Final acceptance in revised form April 2010