Comparative analysis of using artificial neural networks (ANN) and gene expression programming (GEP) in backcalculation of pavement layer thickness

Mehmet Saltan \(^a\) & Serdal Terzi \(^b\)

\(^a\)Civil Engineering Department, Engineering and Architectural Faculty, \(^b\)Structural Education Department, Technical Education Faculty, Suleyman Demirel University, 32260 Isparta, Turkey

Received 21 August 2003; accepted 27 October 2004

Pavement deflection data are often used to evaluate a pavement’s structural condition non-destructively. It is essential not only to evaluate the structural integrity of an existing pavement but also to have accurate information on pavement surface condition in order to establish a reasonable pavement rehabilitation design system. Pavement layers are characterized by their elastic moduli estimated from surface deflections through backcalculation. Backcalculating the pavement layer moduli is a well-accepted procedure for the evaluation of the structural capacity of pavements. The ultimate aim of the backcalculation process from non-destructive testing (NDT) results is to estimate the pavement material properties. Using backcalculation analysis, flexible pavement layer thicknesses together with in-situ material properties can be backcalculated from the measured field data through appropriate analysis techniques. In this study, artificial neural networks (ANN) and gene expression programming (GEP) are used in backcalculating the pavement layer thickness from deflections measured on the surface of the flexible pavements. Experimental deflection data groups from NDT are used to show the capability of the ANN and GEP approaches in backcalculating the pavement layer thickness and compared each other. These approaches can be easily and realistically performed to solve the optimization problems which do not have a formulation or function about the solution.

IPC Code: E01C 9/10

Highway pavements are generally constructed in the form of flexible pavements. Flexible pavements are layered systems with better materials on top and inferior materials at the bottom. Starting from the top, the pavement consists of wearing course, base and sub-base layers. The base material may be a bituminous mix or a granular material, depending on the number of heavy vehicles on the considered section of the road. However, local and cheaper materials can be used as a sub-base layer on top of the subgrade. Repeated application of vehicle loads, weather conditions and other factors decrease the serviceability of the pavement. For this reason, a maintenance program should be set up to decide when and where to carry out maintenance works. The most difficult aspect is to determine the remaining life of the pavement. In order to determine the remaining life, the pavement should be analyzed structurally with material properties for each layer being elastic modulus, Poisson’s ratio and thickness of layer.

Non-destructive testing (NDT) and backcalculating pavement layer moduli are well-accepted procedures for the evaluation of the structural capacity of pavements.\(^3\) NDT enables the use of a mechanistic approach for pavement design and rehabilitation because in-situ material properties may be backcalculated from the measured field data through appropriate analysis techniques\(^2\). In order to backcalculate reliable moduli, it is essential to accomplish several deflection tests at different locations along a highway section having relating uniform layer thicknesses\(^1\). But flexible pavement layer thicknesses must also be known to get realistic results. Layer thicknesses can be obtained by coring the flexible pavement. But it is important that non-destructive tests are carried out on flexible pavements for preventing to be damaged. Among non-destructive deflection measurement methods, commercially available devices are the Dynaflect, Road Rater and Falling Weight Deflectometer (FWD). FWD is commonly used in many countries.

In recent years, one of the most important and promising research fields has been “Heuristics from Nature”, an area utilizing some analogies with natural or social systems and using them to derive non-deterministic heuristic methods and to obtain very good results. Artificial neural networks (ANN) and genetic algorithms (GA) are among the heuristic methods.
Artificial neural networks method is widely used in a variety of practical tasks from process monitoring, fall diagnosis and adaptive human interference to natural events and artificial intelligence such as computers. They are very important in control system applications because of their universal mapping characteristics and learning ability. ANN process can be considered as a black-box modelling with a set of input factors and output variables which are a result of input factors treatment through a systematic neural network. The first appearance of ANN concept in the literature is due to McCullough and Pitts who suggested the cell model. In such a model, ANNs are exemplified as a set of logical statements. Later on, many researchers concentrated their attention on the learning ability of human and its modelling which can be accounted as the pioneering work on ANNs. However, actual leaps in the ANN development appeared towards 1980 through various researches. ANN architecture includes many interconnected neurons or processing elements with familiar characteristics such as inputs, synaptic strengths, activation, output and bias.

Everybody agrees that, by and large, evolution relies on genetic variation coupled with some kind of selection and, in fact, all evolutionary algorithms explore these fundamental processes. In all evolutionary algorithms, an evolutionary epoch or run starts with an initial population. Epoch is maximum number of trials for both ANN and genetic algorithm. Initial populations, though, are generated in many different ways, and the performance and the costs (in terms of CPU time) of different algorithms depend greatly on the characteristics of initial populations. The simplest and less time consuming population is the totally random initial population. However, few evolutionary algorithms are able to use this kind of initial population due not only to structural constraints but also to the kind of genetic operators available to create genetic modification. The initial populations of gene expression programming (GEP) are totally random and consist of the linear genomes of the individuals of the population.

GAs belong to a class of probabilistic search methods that strike excellent balance between exploration and exploitation of the search space. It is different from random algorithms, as it combines elements of directed and stochastic search methods. It has been successfully applied to optimization problems. But, in this study, GEP is used as training algorithm. Then the problem is solved using ANN and results of ANN and GEP solutions are compared and examined.

**Backcalculation of Pavement Layer Thickness**

Backcalculation generally refers to an iterative procedure whereby the layer properties of the pavement model are adjusted until the computed deflections under a given load agree with the corresponding measured values. NDT and backcalculation processes are well-accepted procedures for the evaluation of structural capacity and pavement layer thickness.

Measurement of an impulse deflection wave by the FWD appears to have emerged as the coming method of structural pavement evaluation. A weight of known magnitude is dropped from different heights, creating various levels of impulse loads. The pavement structure responds by a dynamic wave of deflections which spreads outward from the centre under the load. The peaks of this deflection wave are measured at several points by sensors called geophones. One of the sensors is placed in the centre, accessible through a hole in the disk, and the others at various distances outside the disk. The outer sensors are placed on the pavement surface by lowering a boom. The measured deflections generated by the FWD test load represent a deflection bowl or basin such as it may occur under a passing wheel load of corresponding magnitude and speed and of similar distribution area of tire contact pressure.

The ultimate aim of the backcalculation process from NDT results is to estimate the pavement material properties and layer thicknesses. The backcalculation procedure finds the set of parameters corresponding to the best fit to the measured deflection bowls. It is important to obtain the layer thicknesses through in-situ deflection test data equally non-destructively. Maximum precision is needed from the backcalculation procedures, and more realistic models will reduce the size of systematic errors. This will make it possible to predict the remaining life of a pavement realistically in the field immediately after it has been tested.

**FWD Testing Device**

In order to simulate the truck loading on the pavement, a circular mass is dropped from a certain height on the pavement. The height is adjusted according to the desired load level. Underneath the
circular plate a rubber pad is mounted to prevent shock loading. Seven geophones are generally mounted on the trailer (the number of geophones can change). When the vertical load is applied on the pavement, the geophones collect the deflection data.

Benkelman beam and dynaflect which are most commonly used devices in the developing countries, give the information about underneath the centre of circular mass (i.e. these devices give one deflection data in each measurement) whereas the FWD gives the information about other six points (or more points) which are away from the circular plate. Therefore, the effect of the wheel loading can also be seen in other points.

There are many types of FWDs which can apply similar loading. The time of loading varies between 0.025 and 0.030 s; the applied loads vary between 6.7-156 kN\textsuperscript{10-12}. The loading time of 0.030 s represents duration of a load pulse produced by a wheel moving at a speed of 30 km/h. ±0.023 mm deviations can be seen from the FWD measurements\textsuperscript{13}. Typically, 200-300 FWD measurements can be made in a day.

Artificial Neural Networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between these elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in Fig. 1. The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target output pairs are used to train a network. Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Sometimes, incremental training is referred to as “on line” or “adaptive” training. Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems. Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings.

Artificial neurons

An artificial neuron (AN) is a model of an actual neuron. A typical AN is shown in Fig. 2. Input signals are represented by $x_0, x_1, x_2, ..., x_n$. Each input is readjusted by a weight ($w_{ij}$) similar to the synaptic function in a biological neuron\textsuperscript{14}. These weights may be negative or positive depending on the response of the electrical signals.

The sum of the weights of the inputs ($I_j$) and activation function ($y_j$) are given in Eqs (1) and (2), respectively.

\begin{align*}
I_j &= \sum_{i=1}^{n} w_{ij} x_i \quad \ldots (1) \\
y_j &= \phi(I_j) \quad \ldots (2)
\end{align*}

A logistic function is generally used as an activation function [Eq. (3)]. Other forms of activation function such as threshold can also be used. However, in order to introduce non-linearity into the

\[ \frac{1}{1+e^{-\beta x}} \]

Fig. 1—Basic principle of artificial neural networks\textsuperscript{13}

Fig. 2—A typical artificial neuron
neural network a logistic function has to be used. The range of the activation function is between 0 and 1 or –0.5 and 0.5 (Fig. 3). \( \alpha \) characterizes the shape of the activation function. When \( \alpha \) has a small value, the slope of the function is lower than with a larger value.

\[
\phi(I) = \frac{1}{1 + e^{-\alpha I}} \quad \cdots (3)
\]

\( \alpha \) is a coefficient.

Modeling with artificial neural network

The ANN modeling consists of two steps: The first step is to train the network; the second step is to test the network with data, which are not used for training. The processing of adaptation of the weights is called “learning”\(^15\). During the training stage the network uses the inductive-learning principle to learn from a set of examples called the “training set”\(^16\). Learning methods can be classified as supervised and unsupervised learning. In supervised learning, for each input neuron there is always an output neuron. However, for unsupervised learning it is enough only to have input neurons.

A backpropagation algorithm is generally used for training. All the input and output data were normalized between zero and one. Initially, each weight is assigned randomly. The weights are then updated after each iteration according to the equations given below:

\[
w_{hj}(t+1) = w_{hj}(t) + \Delta w_{hj}(t+1) \quad \cdots (4)
\]

\[
w_{ih}(t+1) = w_{ih}(t) + \Delta w_{ih}(t+1) \quad \cdots (5)
\]

where \( w_{hj}(t+1) \) and \( w_{ih}(t+1) \) are interconnection weights between hidden neurons and output neurons and between input neurons and hidden neurons at time \( (t+1) \) respectively.

Based on the gradient descent method the weights can be updated as given below.

\[
\Delta w_{hj}(t+1) = \eta e_{j} o_{h} + \alpha w_{hj}(t) \quad \cdots (6)
\]

\[
\Delta w_{ih}(t+1) = \eta o_{j}(1-o_{h}) \sum e_{j} w_{hj}(t) + \alpha \Delta w_{hj}(t) \quad \cdots (7)
\]

where \( e_{j} \) is the error in output neuron \( j \) in the output layer at time \( (t+1) \) (between the desired and the actual outputs), \( o_{j} \) and \( o_{h} \) are the outputs of output neuron \( j \) and hidden neuron \( h \) at time \( (t+1) \) respectively.

\( \eta = \) training rate \((0<\eta<1)\)

\( \alpha = \) momentum term \((0<\alpha<1)\)

The adjustment of weights continues iteratively until the difference between the present and the previous output is in the range of specified square error limit.

After determining the weight for each connection in the ANN, data which were not used during the training, can be tested to check the performance of the model. For this purpose, some spreadsheet programs can be used or the program, which is used for backpropagation, can also be used.

The selected network has a feed-forward structure. Feed-forward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with non-linear transfer functions allow the network to learn non-linear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range from –1 to +1.

Formulation of the used model has been obtained from formulations of selected functions (i.e. summation and activation) used in the ANN model and weights of neurons. By changing the architecture of ANN and the functions, different formulations can be obtained\(^17\). Deflection values which give the minimum surface layer thickness can be estimated by using the formulas obtained from ANN in genetic algorithms as objective function\(^18\).

Genetic algorithms

The fundamental unit of information is in living systems in the gene. In general, a gene is defined as a portion of a chromosome that determines or affects a
single character or phenotype (visible property), for example, eye colour. It comprises a segment of deoxyribonucleic acid (DNA), commonly packaged into structures called chromosomes. This genetic information is capable of producing a functional product which is most a protein.

Genetic algorithm (GA) is inspired by the mechanism of natural selection where stronger individuals are likely the winners in a competing environment. Here, GA uses a direct analogy of such natural evolution. Through the genetic evolution method, an optimal solution can be found and represented by the final winner of the genetic game.

GA presumes that the potential of an individual problem is an individual and can be represented by a set of parameters. These parameters are regarded as the genes of a chromosome and can be structured by a string of values in binary form. A positive value, generally known as a fitness value, is used to reflect the degree of “goodness” of chromosome for the problem which would be highly related with its objective value.

Pragmatic researchers see evolution’s remarkable power as something to be emulated rather than envied. Natural selection eliminates one of the greatest hurdles in software design: specifying in advance all the features of a problem and the actions a program should take to deal with them. By harnessing the mechanism of evolution, researchers may be able to “breed” programs that solve problems even when no person can fully understand their structure. Indeed, these so-called genetic algorithms have already demonstrated the ability to make breakthroughs in the design of such complex systems as jet engines.

Genetic algorithms make it possible to explore a far greater range of potential solutions to a problem than do conventional programs. Furthermore, as researchers probe the natural selection of programs under controlled and well-understood conditions, the practical results they achieve may yield some insight into the details of how life and intelligence evolved in natural world.

**Basic steps of genetic algorithms**

Given a way or a method of encoding solution of a problem into the form of chromosomes and given an evaluation function that returns a measurement of the cost value of any chromosome in the context of the problem, a GA consists of the following steps (see Fig. 4).

1. Initialize a population of chromosomes.
2. Evaluate each chromosome in the population.
3. Create new chromosomes by mating current chromosomes; apply mutation and recombination as the parent chromosomes mate.
4. Delete members of population to make room for new chromosomes.
5. Evaluate the new chromosomes and insert them into the population.
6. If stopping criterion is satisfied, then stop and return the best chromosome; otherwise, go to step 3.

**Gene expression programming**

The phenotype of GEP individuals consists of the same kind of ramified structures used in genetic programming. However, these complex entities are encoded in simpler, linear structures of fixed length—the chromosomes. Thus, there are two main players in GEP: the chromosomes and the ramified structures or expression trees (ETs), the latter being the expression of the genetic information encoded in the former. Fig. 5 shows an example of ETs.

As in nature, the process of information decoding is called translation. And this translation implies obviously a kind of code and a set of rules. The genetic code is very simple: a one-to-one relationship between the symbols of the chromosome and the functions or terminals they represent. The rules are also very simple: they determine the spatial organization of the functions and terminals in the ETs and the type of interaction between sub-ETs in multi-genic systems. In GEP there are therefore two
languages: the language of the genes and the language of ETs. However, thanks to the simple rules that determine the structure of ETs and their interactions, it is possible to infer immediately the phenotype given the sequence of a gene, and vice versa. This bilingual and unequivocal system is called Karva language. Fig. 6 shows an example of Karva language.22

Results and Discussion

Backcalculation of surface layer thickness with ANN

Setting up a finite element mesh and iteration procedure for backcalculation takes long time. The ANN procedure will reduce the required computation time significantly. A typical flexible pavement, as can be seen in Fig. 7, was chosen for this study. Range of the surface layer thickness is determined as 4-10 cm for this study.

The architecture for the model has seven neurons, a hidden layer with fourteen neurons, and an output neuron (Fig. 8). A learning rate of 0.001 was chosen and maximum number of trials was limited with 10000 after training.

The data set contained 114 samples for the first model. 95 out of 114 samples were chosen randomly as training data, and the remaining 19 samples were selected as simulating data (approximately 20%). The training data set used on training process at selected architecture. After training, testing data group was simulated. Further, regression values between desired and estimating data were determined for flexible pavement surface layer thickness value. Fig. 9 shows the training performance of the model. The performance in this figure shows the Mean Square Error (MSE). The mean square error (MSE) is defined and used in order to decide about the best model as

\[
\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (E_{i(\text{Real})} - E_{i(\text{ANN})})^2
\]

where \( n \) is the number of observed data, \( E_{i(\text{Real})} \) and \( E_{i(\text{ANN})} \) are surface layer thickness and ANN prediction result, respectively.

Backcalculation of surface layer thickness with GEP

In order to backcalculate the layer thickness of wearing course, seven deflection measurement points were used as input. 95 data sets were used for training, and 19 data sets were used for testing. Table 1 shows the structure of the model. This model can be seen on Fig. 10 as schematically. At the end of

<table>
<thead>
<tr>
<th>Property</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of chromosomes</td>
<td>50</td>
</tr>
<tr>
<td>Number of genes</td>
<td>8</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.044</td>
</tr>
<tr>
<td>One-point recombination rate</td>
<td>0.3</td>
</tr>
<tr>
<td>Two-point recombination rate</td>
<td>0.3</td>
</tr>
<tr>
<td>Gene recombination rate</td>
<td>0.1</td>
</tr>
<tr>
<td>Gene transposition rate</td>
<td>0.1</td>
</tr>
</tbody>
</table>
10000 generation, best fitness was found as 772.8 and regression coefficient was found as 0.76. Fig. 11 shows the regression curve of the model.

**Comparison of results of GEP with ANN**

After ANN and GEP calculations were performed, results obtained from ANN and GEP approaches were compared with measured values for both training and test group data sets. Fig. 12 shows the comparison between measured values and the results of ANN and GEP. It can be seen that ANN results are close to measured values while GEP results are not. Namely, GEP results have lower regression value.

A similar graphic was also obtained for randomly chosen data group for testing period. It can be seen that the GEP results show a good match with training data set (see Fig. 13). Therefore, it can be concluded that ANN gives more realistic results. As a summary of this study: (i) both model can solve the problem
using data set without using any pre-assumption; (ii)
ANN gives higher regression coefficient than GEP;
and (iii) consequently, mathematical formulations
are obtained using ANN and GEP approaches. But, ANN
gives “input numbers × neuron numbers of hidden
layer × output numbers × 2 (summation function and
activation function) formulas (7×14×1×2=196
formulas for this study), while a simple formula is
obtained using GEP. The formula obtained from GEP
is shown in Eq. (9).

\[
F = (d_i - d_s) + \left( \frac{d_i}{d_s} \right) + \left( \frac{d_i - d_s}{d_i} \right) + \left( \frac{1}{(d_i + d_s)} \right) + \left( \frac{d_i - d_s}{d_i} \right) - d_s \times d_s \\
+ d_s + \left( \frac{d_i - (d_i + d_s)}{d_i} \right) \times d_s \\
+ \left( d_i - (d_i + d_s) \right) \times \left( \frac{d_i - d_s}{d_i} \right) + (d_i \times d_s)
\]  

... (9)

where \( F \) is surface layer thickness; \( d_i \) is the deflection
value in sensor \( i \).

Conclusions

In the present study, two models have been
presented for determining flexible pavement surface
layer thickness. The first model used ANN approach.
For the second model, GEP was selected as estimating
method. Results show that wearing course thickness
of flexible pavement regression values of the first
model is better than that of the second model. When
ANN and GEP results are close to each other, GEP
approach can be selected in order to obtain only one
formula. As ANN gives more realistic results than
GEP, ANN is convenient to be used for solution
although it gives more formulas which are long and
complex. Some models used for this type of problems
are based on some simplifying assumptions that
cannot reflect the reality. Solutions of the problems
which do not have a formula or function about the
solution can be easily and realistically performed
using these approaches presented here. This new
methodology can help the highway agency in
estimating flexible pavement layer thickness values
using a backcalculation process from deflection
measurements.

References

1  Uzan J, Lytton R L, Germann F P, General procedure for
backcalculating layer moduli, Non-destructive testing of
pavements and backcalculation of moduli (ASTM STP 1026,
USA), 1989, 217-228.