EJGE

State of the Art of Artificial Neural Networks in Geotechnical Engineering

Mohamed A. Shahin
Lecturer, Department. of Civil Engineering, Curtin University of Technology, Perth, WA 6845, Australia
m.shahin@curtin.edu.au

Mark B. Jaksa
Associate Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide, Adelaide, SA 5005, Australia
mark.jaksa@adelaide.edu.au

Holger R. Maier
Professor, School of Civil, Environmental and Mining Engineering, University of Adelaide, Adelaide, SA 5005, Australia
holger.maier@adelaide.edu.au

ABSTRACT

Over the last few years, artificial neural networks (ANNs) have been used successfully for modeling almost all aspects of geotechnical engineering problems. Whilst ANNs provide a great deal of promise, they suffer from a number of shortcomings such as knowledge extraction, extrapolation and uncertainty. This paper presents a state-of-the-art examination of ANNs in geotechnical engineering and provides insights into the modeling issues of ANNs. The paper also discusses current research directions of ANNs that need further attention in the future.

KEYWORDS: artificial neural networks; artificial intelligence; geotechnical engineering.

INTRODUCTION

Artificial neural networks (ANNs) are a form of artificial intelligence which attempt to mimic the function of the human brain and nervous system. ANNs learn from data examples presented to them in order to capture the subtle functional relationships among the data even if the underlying relationships are unknown or the physical meaning is difficult to explain. This is in contrast to most traditional empirical and statistical methods, which need prior knowledge about the nature of the relationships among the data. ANNs are thus well suited to modeling the complex behavior of most geotechnical engineering materials which, by their very nature, exhibit

extreme variability. This modeling capability, as well as the ability to learn from experience, have given ANNs superiority over most traditional modeling methods since there is no need for making assumptions about what the underlying rules that govern the problem in hand could be.

Since the early 1990s, ANNs have been applied successfully to almost every problem in geotechnical engineering. The literature reveals that ANNs have been used extensively for predicting the axial and lateral load capacities in compression and uplift of pile foundations (Abu-Kiefa 1998; Ahmad et al. 2007; Chan et al. 1995; Das and Basudhar 2006; Goh 1994a; Goh 1995a; Goh 1996b; Hanna et al. 2004; Lee and Lee 1996; Nawari et al. 1999; Rahman et al. 2001; Shahin 2008; Teh et al. 1997), drilled shafts (Goh et al. 2005; Shahin and Jaksa 2008) and ground anchors (Rahman et al. 2001; Shahin and Jaksa 2004; Shahin and Jaksa 2005a; Shahin and Jaksa 2005b; Shahin and Jaksa 2006).

Classical constitutive modeling based on the elasticity and plasticity theories is unable to properly simulate the behavior of geomaterials for reasons pertaining to formulation complexity, idealization of material behavior and excessive empirical parameters (Adeli 2001). In this regard, many researchers (Basheer 1998; Basheer 2000; Basheer 2002; Basheer and Najjar 1998; Ellis et al. 1992; Ellis et al. 1995; Fu et al. 2007; Ghaboussi and Sidarta 1998; Habibagahi and Bamdad 2003; Haj-Ali et al. 2001; Hashash et al. 2004; Lefik and Schrefler 2003; Najjar and Ali 1999; Najjar et al. 1999; Najjar and Huang 2007; Penumadu and Chameau 1997; Penumadu and Zhao 1999; Romo et al. 2001; Shahin and Indraratna 2006; Sidarta and Ghaboussi 1998; Tutumluer and Seyhan 1998; Zhu et al. 1998a; Zhu et al. 1998b; Zhu et al. 1996) proposed neural networks as a reliable and practical alternative to modeling the constitutive monotonic and hysteretic behavior of geomaterials.

Geotechnical properties of soils are controlled by factors such as mineralogy; fabric; and pore water, and the interactions of these factors are difficult to establish solely by traditional statistical methods due to their interdependence (Yang and Rosenbaum 2002). Based on the application of ANNs, methodologies have been developed for estimating several soil properties including the pre-consolidation pressure (Celik and Tan 2005), shear strength and stress history (Kurup and Dudani 2002; Lee et al. 2003; Penumadu et al. 1994; Yang and Rosenbaum 2002), swell pressure (Erzin 2007; Najjar et al. 1996a), compaction and permeability (Agrawal et al. 1994; Goh 1995b; Gribb and Gribb 1994; Najjar et al. 1996b; Sinha and Wang 2008), soil classification (Cal 1995) and soil density (Goh 1995b).

Liquefaction during earthquakes is one of the very dangerous ground failure phenomena that cause a large amount of damage to most civil engineering structures. Although the liquefaction mechanism is well known, the prediction of the value of liquefaction induced displacements is very complex and not entirely understood (Baziar and Ghorbani 2005). This has attracted many researchers (Agrawal et al. 1997; Ali and Najjar 1998; Baziar and Ghorbani 2005; Goh 2002; Goh 1994b; Goh 1996a; Goh et al. 1995; Hanna et al. 2007; Javadi et al. 2006; Juang and Chen 1999; Kim and Kim 2006; Najjar and Ali 1998; Ural and Saka 1998; Young-Su and Byung-Tak 2006) to investigate the applicability of ANNs for predicting liquefaction.

The problem of predicting the settlement of shallow foundations, especially on cohesionless soils, is very complex, uncertain and not yet entirely understood. This fact has encouraged some researchers (Chen et al. 2006; Shahin et al. 2002a; Shahin et al. 2003a; Shahin et al. 2004a; Shahin et al. 2005a; Shahin et al. 2005b; Shahin et al. 2002b; Shahin et al. 2003b; Shahin et al. 2003c; Shahin et al. 2003d; Sivakugan et al. 1998) to apply the ANN technique to settlement prediction. The problem of estimating the bearing capacity of shallow foundations by ANNs has also been investigated by Padminin et al. (2008) and Provenzano et al. (2004).

Other applications of ANNs in geotechnical engineering include retaining walls (Goh et al. 1995; Kung et al. 2007), dams (Kim and Kim 2008), blasting (Lu 2005), mining (Rankine and

Sivakugan 2005; Singh and Singh 2005), geoenvironmental engineering (Shang et al. 2004), rock mechanics (Gokceoglu et al. 2004), site characterisation (Basheer et al. 1996; Najjar and Basheer 1996; Rizzo and Dougherty 1994; Rizzo et al. 1996; Zhou and Wu 1994), tunnels and underground openings (Benardos and Kaliampakos 2004; Lee and Sterling 1992; Moon et al. 1995; Neaupane and Achet 2004; Shi et al. 1998; Shi 2000; Yoo and Kim 2007) and slope stability (Ferentinou and Sakellariou 2007; Goh and Kulhawy 2003; Mayoraz and Vulliet 2002; Neaupane and Achet 2004; Ni et al. 1996; Zhao 2008). The interested reader is referred to Shahin et al. (2001) where the pre 2001 papers are reviewed in some detail.

OVERVIEW OF ARTIFICIAL NEURAL NETWORKS

Many authors have described the structure and operation of ANNs (e.g. Fausett 1994; Zurada 1992). ANNs consist of a number of artificial neurons variously known as 'processing elements' (PEs), 'nodes' or 'units'. For multilayer perceptrons (MLPs), which is the most commonly used ANNs in geotechnical engineering, processing elements in are usually arranged in layers: an input layer, an output layer and one or more intermediate layers called hidden layers (Fig. 1).

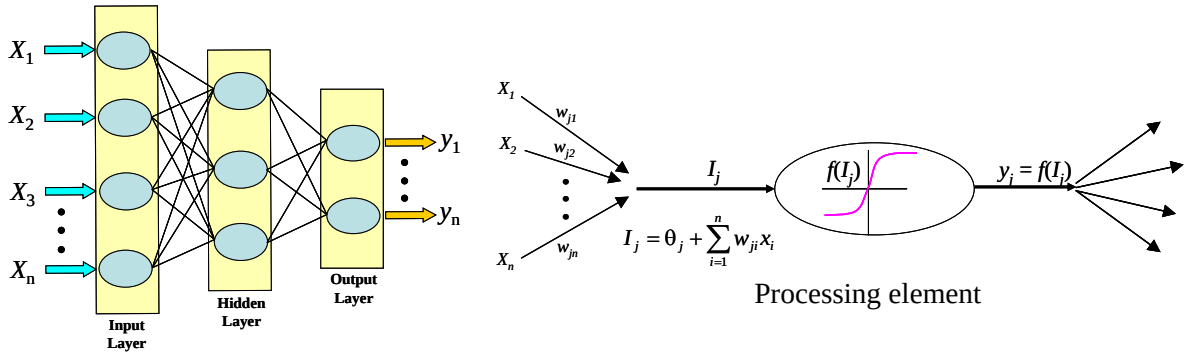


Figure 1: Typical structure and operation of ANNs

Each processing element in a specific layer is fully or partially connected to many other processing elements via weighted connections. The scalar weights determine the strength of the connections between interconnected neurons. A zero weight refers to no connection between two neurons and a negative weight refers to a prohibitive relationship. From many other processing elements, an individual processing element receives its weighted inputs, which are summed and a bias unit or threshold is added or subtracted. The bias unit is used to scale the input to a useful range to improve the convergence properties of the neural network. The result of this combined summation is passed through a transfer function (e.g. logistic sigmoid or hyperbolic tangent) to produce the output of the processing element. For node *j*, this process is summarized in Equations 1 and 2 and illustrated in Fig. 1.

$$I_j = \theta_j + \sum_{i=1}^{n} w_{ji} x_i \quad \text{summation} \tag{1}$$

$$y_j = f(I_j) \quad \text{transfer} \tag{2}$$

where

I_j = the activation level of node j;

w_{ji} = the connection weight between nodes j and i;

x_i = the input from node i, $i = 0, 1, \ldots, n$;

θ_j = the bias or threshold for node j;

y_j = the output of node j; and

$f(.)$ = the transfer function.

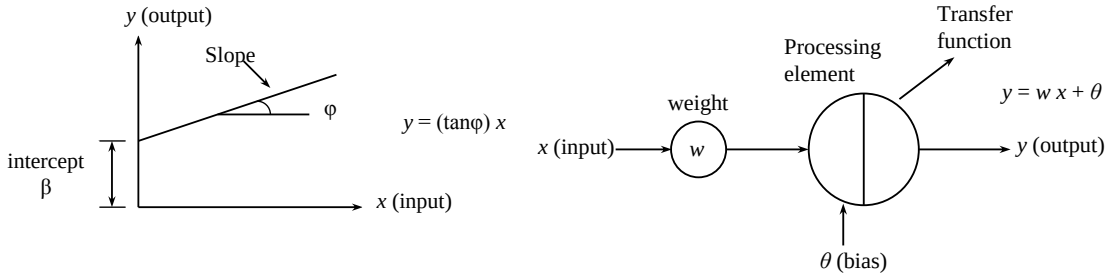
The propagation of information in MLPs starts at the input layer where the input data are presented. The inputs are weighted and received by each node in the next layer. The weighted inputs are then summed and passed through a transfer function to produce the nodal output, which is weighted and passed to processing elements in the next layer. The network adjusts its weights on presentation of a set of training data and uses a learning rule until it can find a set of weights that will produce the input-output mapping that has the smallest possible error. The above process is known as 'learning' or 'training'.

Learning in ANNs is usually divided into supervised and unsupervised (Masters 1993). In supervised learning, the network is presented with a historical set of model inputs and the corresponding (desired) outputs. The actual output of the network is compared with the desired output and an error is calculated. This error is used to adjust the connection weights between the model inputs and outputs to reduce the error between the historical outputs and those predicted by the ANN. In unsupervised learning, the network is only presented with the input stimuli and there are no desired outputs. The network itself adjusts the connection weights according to the input values. The idea of training in unsupervised networks is to cluster the input records into classes of similar features.

ANNs can be categorized on the basis of two major criteria: (i) the learning rule used and (ii) the connections between processing elements. Based on learning rules, ANNs, as mentioned above, can be divided into supervised and unsupervised networks. Based on connections between processing elements, ANNs can be divided into feed-forward and feedback networks. In feed-forward networks, the connections between the processing elements are in the forward direction only, whereas, in feedback networks, connections between processing elements are in both the forward and backward directions.

The ANN modeling philosophy is similar to a number of conventional statistical models in the sense that both are attempting to capture the relationship between a historical set of model inputs and corresponding outputs. For example, suppose a set of x-values and corresponding y-values in 2 dimensional space, where $y = f(x)$. The objective is to find the unknown function f, which relates the input variable x to the output variable y. In a linear regression model, the function f can be obtained by changing the slope $\tan\varphi$ and intercept β of the straight line in Fig. 2(a), so that the error between the actual outputs and outputs of the straight line is minimized. The same principle is used in ANN models. ANNs can form the simple linear regression model by having one input, one output, no hidden layer nodes and a linear transfer function (Fig. 2(b)). The connection weight w in the ANN model is equivalent to the slope $\tan\varphi$ and the threshold θ is equivalent to the intercept β, in the linear regression model. ANNs adjust their weights by

repeatedly presenting examples of the model inputs and outputs in order to minimize an error function between the historical outputs and the outputs predicted by the ANN model.



(a) Linear regression model

(b) ANN representation of the linear regression model

Figure 2: Linear regression versus ANN models

If the relationship between x and y is non-linear, regression analysis can only be successfully applied if prior knowledge of the nature of the non-linearity exists. On the contrary, this prior knowledge of the nature of the non-linearity is not required for ANN models. In the ANN model, the degree of non-linearity can be also changed easily by changing the transfer function and the number of hidden layer nodes. In the real world, it is likely that complex and highly non-linear problems are encountered. In such situations, traditional regression analysis is inadequate (Gardner and Dorling 1998). In contrast, ANNs can be used to deal with this complexity by changing the transfer function or network structure, and the type of non-linearity can be changed by varying the number of hidden layers and the number of nodes in each layer. In addition, ANN models can be upgraded from univariate to multivariate by increasing the number of input nodes.

MODELING ISSUES IN ARTIFICAL NEURAL NETWORKS

In order to improve performance, ANN models need to be developed in a systematic manner. Such an approach needs to address major factors such as the determination of adequate model inputs, data division and pre-processing, the choice of suitable network architecture, careful selection of some internal parameters that control the optimization method, the stopping criteria and model validation. These factors are explained and discussed below.

Determination of Model Inputs

An important step in developing ANN models is to select the model input variables that have the most significant impact on model performance (Faraway and Chatfield 1998). A good subset of input variables can substantially improve model performance. Presenting as large a number of input variables as possible to ANN models usually increases network size (Maier and Dandy 2000), resulting in a decrease in processing speed and a reduction in the efficiency of the network (Lachtermacher and Fuller 1994). A number of techniques have been suggested in the literature to assist with the selection of input variables. An approach that is usually utilized in the field of geotechnical engineering is that appropriate input variables can be selected in advance

based on a priori knowledge (Maier and Dandy 2000). Another approach used by some researchers (Goh 1994b; Najjar et al. 1996b; Ural and Saka 1998) is to train many neural networks with different combinations of input variables and to select the network that has the best performance. A step-wise technique described by Maier and Dandy (2000) can also be used in which separate networks are trained, each using only one of the available variables as model inputs. The network that performs the best is then retained, combining the variable that results in the best performance with each of the remaining variables. This process is repeated for an increasing number of input variables, until the addition of additional variables results in no improvement in model performance. Another useful approach is to employ a genetic algorithm to search for the best sets of input variables (NeuralWare 1997). For each possible set of input variables chosen by the genetic algorithm, a neural network is trained and used to rank different subsets of possible inputs. A set of input variables derives its fitness from the model error obtained based on those variables. The adaptive spline modeling of observation data (ASMOD) algorithm proposed by Kavli (1993) is also a useful technique that can be used for developing parsimonious neurofuzzy networks by automatically selecting a combination of model input variables that have the most significant impact on the outputs.

A potential shortcoming of the above approaches is that they are model-based. In other words, the determination as to whether a parameter input is significant or not is dependent on the error of a trained model, which is not only a function of the inputs, but also model structure and calibration. This can potentially obscure the impact of different model inputs. In order to overcome this limitation, model-free approaches can be utilized, which use linear dependence measures, such as correlation, or non-linear measures of dependence, such as mutual information, to obtain the significant model inputs prior to developing the ANN models (e.g. Bowden et al. 2005; May et al. 2008).

Division of Data

As described earlier, supervised ANNs are similar to conventional statistical models in the sense that model parameters (e.g. connection weights) are adjusted in the model calibration phase (training) so as to minimize the error between model outputs and the corresponding measured values for a particular data set (the training set). ANNs perform best when they do not extrapolate beyond the range of the data used for calibration (Flood and Kartam 1994; Minns and Hall 1996; Tokar and Johnson 1999). Therefore, the purpose of ANNs is to non-linearly interpolate (generalize) in high-dimensional space between the data used for calibration. Unlike conventional statistical models, ANN models generally have a large number of model parameters (connection weights) and can therefore overfit the training data, especially if the training data are noisy. In other words, if the number of degrees of freedom of the model is large compared with the number of data points used for calibration, the model might no longer fit the general trend, as desired, but might learn the idiosyncrasies of the particular data points used for calibration leading to '*memorization*', rather than '*generalization*'. Consequently, a separate validation set is needed to ensure that the model can generalize within the range of the data used for calibration. It is common practice to divide the available data into two subsets; a training set, to construct the neural network model, and an independent validation set to estimate the model performance in a deployed environment (Maier and Dandy 2000; Twomey and Smith 1997). Usually, two-thirds of the data are suggested for model training and one-third for validation (Hammerstrom 1993). A modification of the above data division method is cross-validation (Stone 1974) in which the data are divided into three sets: training, testing and validation. The training set is used to adjust the

connection weights, whereas the testing set is used to check the performance of the model at various stages of training and to determine when to stop training to avoid over-fitting. The validation set is used to estimate the performance of the trained network in the deployed environment. In an attempt to find the optimal proportion of the data to use for training, testing and validation, Shahin et al. (2004b) investigated the impact of the proportion of data used in various subsets on ANN model performance for a case study of settlement prediction of shallow foundations and found that there is no clear relationship between the proportion of data for training, testing and validation and model performance, however, they found that the best result was obtained when 20% of the data were used for validation and the remaining data were divided into 70% for training and 30% for testing.

In many situations, the available data are small enough to be solely devoted to model training and collecting any more data for validation is difficult. In this situation, the *leave-k-out* method can be used (Masters 1993) which involves holding back a small fraction of the data for validation and using the rest of the data for training. After training, the performance of the trained network has to be estimated with the aid of the validation set. A different small subset of data is held back and the network is trained and tested again. This process is repeated many times with different subsets until an optimal model can be obtained from the use of all of the available data.

In the majority of ANN applications in geotechnical engineering, the data are divided into their subsets on an arbitrary basis. However, recent studies have found that the way the data are divided can have a significant impact on the results obtained (Tokar and Johnson 1999). As ANNs have difficulty extrapolating beyond the range of the data used for calibration, in order to develop the best ANN model, given the available data, all of the patterns that are contained in the data need to be included in the calibration set. For example, if the available data contain extreme data points that were excluded from the calibration data set, the model cannot be expected to perform well, as the validation data will test the model's extrapolation ability, and not its interpolation ability. If all of the patterns that are contained in the available data are contained in the calibration set, the toughest evaluation of the generalization ability of the model is if all the patterns (and not just a subset) are contained in the validation data. In addition, if cross-validation is used as the stopping criterion, the results obtained using the testing set have to be representative of those obtained using the training set, as the testing set is used to decide when to stop training or, for example, which model architecture or learning rate is optimal. Consequently, the statistical properties (e.g. mean and standard deviation) of the various data subsets (e.g. training, testing and validation) need to be similar to ensure that each subset represents the same statistical population (Masters 1993). If this is not the case, it may be difficult to judge the validity of ANN models (Maier and Dandy 2000).

This fact has been recognized for some time (ASCE 2000; Maier and Dandy 2000; Masters 1993), and several studies have used ad-hoc methods to ensure that the data used for calibration and validation have the same statistical properties (Braddock et al. 1998; Campolo et al. 1999; Ray and Klindworth 2000; Tokar and Johnson 1999). Masters (1993) strongly confirms the above strategy of data division as he says "*if our training set is not representative of the data on which the network will be tested, we will be wasting our time*". However, it was not until a few years ago that systematic approaches for data division have been proposed in the literature. Bowden et al. (2002) used a genetic algorithm to minimize the difference between the means and standard deviations of the data in the training, testing and validation sets. While this approach ensures that the statistical properties of the various data subsets are similar, there is still a need to choose which proportion of the data to use for training, testing and validation. Kocjancic and

Zupan (2000) and Bowden et al. (2002) used a self-organizing map (SOM) to cluster high-dimensional input and output data in two-dimensional space and divided the available data so that values from each cluster are represented in the various data subsets. This ensures that data in the different subsets are representative of each other and has the additional advantage that there is no need to decide what percentage of the data to use for training, testing and validation. The major shortcoming of this approach is that there are no guidelines for determining the optimum size and shape of the SOM (Cai et al. 1994; Giraudel and Lek 2001). This has the potential to have a significant impact on the results obtained, as the underlying assumption of the approach is that the data points in one cluster provide the same information in high-dimensional space. However, if the SOM is too small, there may be significant intra-cluster variation. Conversely, if the map is too large, too many clusters may contain single data points, making it difficult to choose representative subsets. To overcome the problem of determining the optimum size of clusters associated with using SOMs, Shahin et al. (2004b) have introduced a data division approach that utilizes a fuzzy clustering technique so that data division can be carried out in a systematic manner.

Data Pre-processing

Once the available data have been divided into their subsets (i.e. training, testing and validation), it is important to pre-process the data in a suitable form before they are applied to the ANN. Data pre-processing is necessary to ensure all variables receive equal attention during the training process (Maier and Dandy 2000). Moreover, pre-processing usually speeds up the learning process. Pre-processing can be in the form of data scaling, normalization and transformation (Masters 1993). Scaling the output data is essential, as they have to be commensurate with the limits of the transfer functions used in the output layer (e.g. between –1.0 to 1.0 for the tanh transfer function and 0.0 to 1.0 for the sigmoid transfer function). Scaling the input data is not necessary but it is almost always recommended (Masters 1993). In some cases, the input data need to be normally distributed in order to obtain optimal results (Fortin et al. 1997). However, Burke and Ignizio (1992) stated that the probability distribution of the input data does not have to be known. Transforming the input data into some known forms (e.g. linear, log, exponential, etc.) may be helpful to improve ANN performance. However, empirical trials (Faraway and Chatfield 1998) showed that the model fits were the same, regardless of whether raw or transformed data were used.

Determination of Model Architecture

Determining the network architecture is one of the most important and difficult tasks in ANN model development. It requires the selection of the optimum number of layers and the number of nodes in each of these. There is no unified approach for determination of an optimal ANN architecture. It is generally achieved by fixing the number of layers and choosing the number of nodes in each layer. For MLPs, there are always two layers representing the input and output variables in any neural network. It has been shown that one hidden layer is sufficient to approximate any continuous function provided that sufficient connection weights are given (Cybenko 1989; Hornik et al. 1989). Hecht-Nielsen (1989) provided a proof that a single hidden layer of neurons, operating a sigmoidal activation function, is sufficient to model any solution surface of practical interest. To the contrary, Flood (1991) stated that there are many solution surfaces that are extremely difficult to model using a sigmoidal network using one hidden layer. In addition, some researchers (Flood and Kartam 1994; Ripley 1996; Sarle 1994) stated that the

use of more than one hidden layer provides the flexibility needed to model complex functions in many situations. Lapedes and Farber (1988) provided more practical proof that two hidden layers are sufficient, and according to Chester (1990), the first hidden layer is used to extract the local features of the input patterns while the second hidden layer is useful to extract the global features of the training patterns. However, Masters (1993) stated that using more than one hidden layer often slows the training process dramatically and increases the chance of getting trapped in local minima.

The number of nodes in the input and output layers is restricted by the number of model inputs and outputs, respectively. There is no direct and precise way of determining the best number of nodes in each hidden layer. A trial-and-error procedure, which is generally used in geotechnical engineering to determine the number and connectivity of the hidden layer nodes, can be used. It has been shown in the literature (Maren et al. 1990; Masters 1993; Rojas 1996) that neural networks with a large number of free parameters (connection weights) are more subject to overfitting and poor generalization. Consequently, keeping the number of hidden nodes to a minimum, provided that satisfactory performance is achieved, is always better, as it: (a) reduces the computational time needed for training; (b) helps the network achieve better generalization performance; (c) helps avoid the problem of overfitting and (d) allows the trained network to be analyzed more easily. For single hidden layer networks, there are a number of rules-of-thumb to obtain the best number of hidden layer nodes. One approach is to assume the number of hidden nodes to be 75% of the number of input units (Salchenberger et al. 1992). Another approach suggests that the number of hidden nodes should be between the average and the sum of the nodes in the input and output layers (Berke and Hajela 1991). A third approach is to fix an upper bound and work back from this bound. Hecht-Nielsen (1987) and Caudill (1988) suggested that the upper limit of the number of hidden nodes in a single layer network may be taken as $(2I+1)$, where I is the number of inputs. The best approach found by Nawari et al. (1999) was to start with a small number of nodes and to slightly increase the number until no significant improvement in model performance is achieved. Yu (1992) showed that the error surface of a network with one hidden layer and $(I–1)$ hidden nodes has no local minima. For networks with two hidden layers, the *geometric pyramid rule* described by Nawari et al. (1999) can be used. The notion behind this method is that the number of nodes in each layer follows a geometric progression of a pyramid shape, in which the number of nodes decreases from the input layer towards the output layer. Kudrycki (1988) found empirically that the optimum ratio of the first to second hidden layer nodes is 3:1, even for high dimensional inputs.

Another way of determining the optimal number of hidden nodes that can result in good model generalization and avoid overfitting is to relate the number of hidden nodes to the number of available training samples (Maier and Dandy 2000). Masters (1993) stated "*the only way to prevent the network from learning unique characteristics of the training set, to the detriment of learning universal characteristics, is to flood it with so many examples that it cannot possibly learn all of their idiosyncrasies*". There are a number of rules-of-thumb that have been suggested in the literature to relate the training samples to the number of connection weights. For instance, Rogers and Dowla (1994) suggested that the number of weights should not exceed the number of training samples. Masters (1993) stated that the required minimum ratio of the number of training samples to the number of connection weights should be 2 and, the minimum ratio of the optimum training sample size to the number of connection weights should be 4. Hush and Horne (1993) suggested that this ratio should be 10. Amari et al. (1997) demonstrated that if this ratio is at least 30, overfitting does not occur.

A number of systematic approaches have also been proposed to obtain automatically the optimal network architecture. The *adaptive method of architecture determination*, suggested by Ghaboussi and Sidarta (1998), is an example of the automatic methods for obtaining the optimal network architecture that suggests starting with an arbitrary, but small, number of nodes in the hidden layers. During training, and as the network approaches its capacity, new nodes are added to the hidden layers, and new connection weights are generated. Training is continued immediately after the new hidden nodes are added to allow the new connection weights to acquire the portion of the knowledge base which was not stored in the old connection weights. For this process to be achieved, some training is carried out with the new modified connection weights only, while the old connection weights are frozen. Additional cycles of training are then carried out where all the connection weights are allowed to change. The above steps are repeated and new hidden nodes are added as needed to the end of the training process, in which the appropriate network architecture is automatically determined. Kingston et al. (2008) showed that Bayesian approaches can be used to determine the optimal number of hidden nodes by using Bayes' factors in conjunction with an examination of the correlation structure between connection weights. *Pruning* is another automatic approach to determine the optimal number of hidden nodes. One such technique proposed by Karnin (1990) starts training a network that is relatively large and later reduces the size of the network by removing the unnecessary hidden nodes. *Genetic algorithms* provide evolutionary alternatives to obtain an optimal neural network architecture that have been used successfully in many situations (Miller et al. 1989). The *adaptive spline modeling of observation data* (ASMOD) (Kavli 1993) algorithm is an automatic method for obtaining the optimal architecture of B-spline neurofuzzy networks, as shown previously.

Cascade-Correlation (Fahlman and Lebiere 1990) is another automatic method to obtain the optimal architecture of ANNs. Cascade-Correlation is a constructive method that can be characterized by the following steps (Fahlman and Lebiere 1990). The neural network is initially trained using Fahlman's quickprop (Fahlman 1988) algorithm without hidden nodes and with direct connection between the input layer and the output layer. Hidden nodes are added randomly one or a few at a time. New hidden nodes receive connections from all previously established hidden nodes as well as from the original inputs. At the time new hidden nodes are added to the network, their connections with the inputs are frozen and only their output connections are trained using the quickprop algorithm. This process is stopped when the model performance shows no further improvement. Consequently, the architecture of ANN networks using Cascade-Correlation is that the input nodes are connected to the output nodes and the hidden nodes are connected to the input and output nodes as well as other previously established hidden nodes. The constructive nature of the Cascade-Correlation method means that the way in which the hidden nodes are connected results in the addition of a new single-node layer to the network each time a new node is added. This is designed to result in the smallest network that can adequately map the design input-output relationship, which has a number of advantages, including improved generalization ability (Castellano et al. 1997) and higher processing speed (Bebis and Georgiopoulos 1994). It should be noted that Masters (1993) has argued that the automatic approaches for obtaining optimal network architectures can be easily abused, as they do not directly address the problem of overfitting.

Model Optimization (Training)

As mentioned previously, the process of optimizing the connection weights is known as 'training' or 'learning'. This is equivalent to the parameter estimation phase in conventional

statistical models (Maier and Dandy 2000). The aim is to find a global solution to what is typically a highly non-linear optimization problem (White 1989). The method most commonly used for finding the optimum weight combination of feed-forward MLP neural networks is the back-propagation algorithm (Rumelhart et al. 1986) which is based on first-order gradient descent. The use of global optimization methods, such as simulated annealing and genetic algorithms, have also been proposed (Hassoun 1995). The advantage of these methods is that they have the ability to escape local minima in the error surface and, thus, produce optimal or near optimal solutions. However, they also have a slow convergence rate. Ultimately, the model performance criteria, which are problem specific, will dictate which training algorithm is most appropriate. If training speed is not a major concern, there is no reason why the back-propagation algorithm cannot be used successfully (Breiman 1994). On the other hand, as mentioned previously, the weights of B-spline neurofuzzy networks are generally updated using the *Least Mean Squared* or *Normalized Least Mean Squared* learning rules (Brown and Harris 1994).

Stopping Criteria

Stopping criteria are used to decide when to stop the training process. They determine whether the model has been optimally or sub-optimally trained (Maier and Dandy 2000). Many approaches can be used to determine when to stop training. Training can be stopped: after the presentation of a fixed number of training records; when the training error reaches a sufficiently small value; or when no or slight changes in the training error occur. However, the above examples of stopping criteria may lead to the model stopping prematurely or over-training. As mentioned previously, the *cross-validation* technique (Stone 1974) is an approach that can be used to overcome such problems. It is considered to be the most valuable tool to ensure over-fitting does not occur (Smith 1993). Amari et al. (1997) suggested that there are clear benefits in using cross-validation when limited data are available, as is the case for many real-life case studies. The benefits of cross-validation are discussed further in Hassoun (1995). As mentioned previously, the cross-validation technique requires that the data be divided into three sets; training, testing and validation. The training set is used to adjust the connection weights. The testing set measures the ability of the model to generalize, and the performance of the model using this set is checked at many stages of the training process. Training is stopped when the error of the testing set starts to increase. The testing set is also used to determine the optimum number of hidden layer nodes and the optimum values of the internal parameters (learning rate, momentum term and initial weights). The validation set is used to assess model performance once training has been accomplished. A number of different stopping criteria (e.g. Bayesian Information Criterion, Akaike's Information Criterion and Final Prediction Error) can also be used, as mentioned previously. Unlike cross-validation, these stopping criteria require the data be divided into only two sets; a training set, to construct the model; and an independent validation set, to test the validity of the model in the deployed environment. The basic notion of these stopping criteria is that model performance should balance model complexity with the amount of training data and model error.

Model Validation

Once the training phase of the model has been successfully accomplished, the performance of the trained model should be validated. The purpose of the model validation phase is to ensure that the model has the ability to generalize within the limits set by the training data in a robust fashion, rather than simply having memorized the input-output relationships that are

contained in the training data. The approach that is generally adopted in the literature to achieve this is to test the performance of trained ANNs on an independent validation set, which has not been used as part of the model building process. If such performance is adequate, the model is deemed to be able to generalize and is considered to be robust.

The coefficient of correlation, *r*, the root mean squared error, RMSE, and the mean absolute error, MAE, are the main criteria that are often used to evaluate the prediction performance of ANN models. The coefficient of correlation is a measure that is used to determine the relative correlation and the goodness-of-fit between the predicted and observed data. Smith (1986) suggested the following guide for values of $|r|$ between 0.0 and 1.0:

$|r| \geq 0.8$ strong correlation exists between two sets of variables;

$0.2 < |r| < 0.8$ correlation exists between the two sets of variables; and

$|r| \leq 0.2$ weak correlation exists between the two sets of variables.

The RMSE is the most popular measure of error and has the advantage that large errors receive much greater attention than small errors (Hecht-Nielsen 1990). In contrast with RMSE, MAE eliminates the emphasis given to large errors. Both RMSE and MAE are desirable when the evaluated output data are smooth or continuous (Twomey and Smith 1997).

Kingston et al. (2005b) stated that if "*ANNs are to become more widely accepted and reach their full potential…, they should not only provide a good fit to the calibration and validation data, but the predictions* should *also be plausible in terms of the relationship modeled and robust under a wide range of conditions*." and that "*while ANNs validated against error alone may produce accurate predictions for situations similar to those contained in the training data, they may not be robust under different conditions unless the relationship by which the data were generated has been adequately estimated*". This is in agreement with the investigation into the robustness of ANNs carried out by Shahin et al. (2005c) for a case study of predicting the settlement of shallow foundations on granular soils. Shahin et al. (2005c) found that good performance of ANN models on the data used for model calibration and validation does not guarantee that the models will perform well in a robust fashion over a range of data similar to those used in the model calibration phase. For this reason, Shahin et al. (2005c) proposed a method to test the robustness of the predictive ability of ANN models by carrying out a sensitivity analysis to investigate the response of ANN model outputs to changes in its inputs. The robustness of the model can be determined by examining how well model predictions are in agreement with the known underlying physical processes of the problem in hand over a range of inputs. In addition, Shahin et al. (2005c) also advised that the connection weights be examined as part of the interpretation of ANN model behavior, as suggested by Garson (1991). On the other hand, Kingston et al. (2005b) adopted the *connection weigh approach* of Olden et al. (2004) for a case study in hydrological modeling in order to assess the relationship modeled by the ANNs. They concluded that this approach provided the best overall methodology for quantifying ANN input importance in comparison to other commonly used methods, though with a few limitations.

FUTURE DIRECTIONS IN UTILIZATION OF ARTIFICIAL NEURAL NETWORKS

Despite the good performance of ANNs in many situations, they suffer from a number of shortcomings including the limited ability to extract knowledge from trained neural networks, inability to extrapolate beyond the range of the data used for model training and dealing with uncertainty. These issues have received recent attention and require further research in the future. Each of these is briefly treated below.

Knowledge Extraction

ANNs are often referred to as "black boxes" due to their lack of transparency as they do not consider nor explain the underlying physical processes. This is because the knowledge extracted by ANNs is stored in a set of weights that are difficult to interpret properly. This issue has been addressed by many researchers with respect to hydrological engineering. For example, Jain et al. (2004) examined whether or not the physical processes in a watershed were inherent in a trained ANN rainfall-runoff model by assessing the strengths of the relationships between the distributed components of the ANN model, in terms of the responses from the hidden nodes, and the deterministic components of the hydrological process, computed from a conceptual rainfall runoff model, along with the observed input variables, using correlation coefficients and scatter plots. They concluded that the trained ANN, in fact, captured different components of the physical process and a careful examination of the distributed information contained in the trained ANN can inform one about the nature of the physical processes captured by various components of the ANN model. Sudheer (2005) performed perturbation analysis to assess the influence of each individual input variable on the output variable and found it to be an effective means of identifying the underlying physical process inherent in the trained ANN. Olden et al. (2004), Sudheer and Jain (2004) and Kingston et al. (2006) also addressed this issue of model transparency and knowledge extraction.

In the context of geotechnical engineering, Shahin et al. (2002a) and Shahin and Jaksa (2005b) expressed the results of the trained ANNs in the form of relatively straightforward equations. This was possible due to the relatively small number of input and output variables, and hidden nodes. More recently, Feng et al. (2006), Javadi et al. (2006), Johari et al. (2006), Narendara et al. (2006) and Rezania and Javadi (2007), using genetic programming techniques, and Pal (2006), Goh and Goh (2007), Samui (2008) and Zhao (2008), using support vector machines, obtained greatly simplified formulae for some geotechnical engineering problems that further enhanced the transparency of the ANN models. Neurofuzzy applications are another means of knowledge extraction that facilitate model transparency. Examples of such applications in geotechnical engineering include Ni et al. (1996), Shahin et al. (2003a; 2003c; 2003d), Gokceoglu et al. (2004), Provenzano et al. (2004), Shahin et al. (2005b) and Padmini et al. (2008).

Extrapolation

It is generally accepted that ANNs perform best when they do not extrapolate beyond the range of the data used for calibration (Flood and Kartam, 1994; Minns and Hall, 1996; Tokar and Johnson, 1999). Whilst this is not unlike other models, it is nevertheless an important limitation of ANNs, as it restricts their usefulness and applicability. Extreme value prediction is of

particular concern in several areas of civil engineering, such as hydrological engineering, when floods are forecast, as well as in geotechnical engineering when, for example, liquefaction potential and the stability of slopes are assessed. Sudheer et al. (2003) highlighted this issue and proposed a methodology, based on the Wilson-Hilferty transformation, for enabling ANN models to predict extreme values with respect to peak river flows. Their methodology yielded superior predictions when compared with those obtained from an ANN model using untransformed data.

Uncertainty

Finally, a further limitation of ANNs is that the uncertainty in the predictions generated is seldom quantified (Maier and Dandy, 2000). Failure to account for such uncertainty makes it impossible to assess the quality of ANN predictions, which severely limits their efficacy. In an effort to address this, a few researchers have applied Bayesian techniques to ANN training (e.g. Buntine and Weigend 1991; Kingston et al. 2005a; Kingston et al. 2008; MacKay 1992) in the context of hydrological engineering; and Goh et al. 2005 with respect to geotechnical engineering). Goh et al. (2005) observed that the integration of the Bayesian framework into the backpropagation algorithm enhanced the neural network prediction capabilities and provided assessment of the confidence associated with the network predictions. Shahin et al. (2005a; 2005b) also incorporated uncertainty in the ANN process by developing a series of design charts expressing the reliability of settlement predictions for shallow foundations on cohesionless soils. Research to date has demonstrated the value of Bayesian neural networks, although further work is needed in the area of geotechnical engineering.

DISCUSSION AND CONCLUSIONS

In the field of geotechnical engineering, it is possible to encounter some types of problems that are very complex and not well understood. In this regard, ANNs provide several advantages over more conventional computing techniques. For most traditional mathematical models, the lack of physical understanding is usually supplemented by either simplifying the problem or incorporating several assumptions into the models. Mathematical models also rely on assuming the structure of the model in advance, which may be sub-optimal. Consequently, many mathematical models fail to simulate the complex behaviour of most geotechnical engineering problems. In contrast, ANNs are a data driven approach in which the model can be trained on input-output data pairs to determine the structure and parameters of the model. In this case, there is no need to either simplify the problem or incorporate any assumptions. Moreover, ANNs can always be updated to obtain better results by presenting new training examples as new data become available. These factors combine to make ANNs a powerful modelling tool in geotechnical engineering.

Despite the success of ANNs in geotechnical engineering and other disciplines, they suffer from some shortcomings that need further attention in the future including model transparency and knowledge extraction, extrapolation and uncertainty. Together, improvements in these issues will greatly enhance the usefulness of ANN models with respect to geotechnical engineering applications. Until such an improvement is achived, the authors agree with Flood and Kartam (1994) that neural networks for the time being might be treated as a complement to conventional computing techniques rather than as an alternative, or may be used as a quick check on solutions developed by more time-consuming and in-depth analyses.

REFERENCES

1. Abu-Kiefa, M. A. (1998). "General regression neural networks for driven piles in cohesionless soils." Journal of Geotechnical & Geoenviromental Engineering, ASCE, 124(12), 1177-1185.

2. Adeli, H. (2001). "Neural networks in civil engineering: 1989-2000." Computer-Aided Civil and Infrastructure Engineering, 16(2), 126-142.

3. Agrawal, G., Chameau, J. A., and Bourdeau, P. L. (1997). "Assessing the liquefaction susceptibility at a site based on information from penetration testing." Artificial neural networks for civil engineers: Fundamentals and applications, N. Kartam, I. Flood, and J. H. Garrett, eds., New York, 185-214.

4. Agrawal, G., Weeraratne, S., and Khilnani, K. (1994). "Estimating clay liner and cover permeability using computational neural networks." Proceedings of the 1st Congress on Computing in Civil Engineering, Washington.

5. Ahmad, I., El Naggar, H., and Kahn, A. N. (2007). "Artificial neural network application to estimate kinematic soil pile interaction response parameters." Soil Dynamics and Earthquake Engineering, 27(9), 892-905.

6. Ali, H. E., and Najjar, Y. M. (1998). "Neuronet-based approach for assessing liquefaction potential of soils." Transportation Research Record No. 1633, 3-8.

7. Amari, S. I., Murata, N., Muller, K. R., Finke, M., and Yang, H. H. (1997). "Asympotic statistical theory of overtraining and cross-validation." IEEE Transactions on Neural Networks, 8(5), 985-996.

8. ASCE. (2000). "Artificial neural networks in hydrology. I: Preliminary concepts." J. Hydrologic Engineering, 5(2), 115-123.

9. Basheer, I. A. (1998). "Neuromechanistic-based modeling and simulation of constitutive behaviour of fine-grained soils," PhD Thesis, Kansas State University, Manhattan, KS.

10. Basheer, I. A. (2000). "Selection of methodology for neural network modeling of constitutive hysteresis behavior of soils." Computer-Aided Civil and Infrastructure Engineering, 15(6), 445-463.

11. Basheer, I. A. (2002). "Stress-strain behavior of geomaterials in loading reversal simulated by time-delay neural networks." Journal of Materials in Civil Engineering, 14(3), 270-273.

12. Basheer, I. A., and Najjar, Y. M. (1998). "Modeling cyclic constitutive behavior by neural networks: Theoretical and real data." Proceedings of the 12th Engineering Mechanics Conference, La Jolla, California, 952-955.

13. Basheer, I. A., Reddi, L. N., and Najjar, Y. M. (1996). "Site characterization by neuronets: An application to the landfill sitting problem." Ground Water, 34, 610-617.

14. Baziar, M. H., and Ghorbani, A. (2005). "Evaluation of lateral spreading using artificial neural networks." Soil Dynamics and Earthquake Engineering, 25(1), 1-9.

15. Bebis, G., and Georgiopoulos, M. (1994). "Feed-forward neural networks: Why network size is so important." IEEE Potentials, October/November, 27-31.

16. Benardos, A. G., and Kaliampakos, D. C. (2004). "Modeling TBM performance with artificial neural networks." Tunneling and Underground Space Technology, 19(6), 597-605.

17. Berke, L., and Hajela, P. (1991). "Application of neural networks in structural optimization." NATO/AGARD Advanced Study Institute, 23(I-II), 731-745.

18. Bowden, G. J., Dandy, G. C., and Maier, H. R. (2005). "Input determination for neural network models in water resources applications: Part 1 - Background and methodology." Journal of Hydrology, 301(1-4), 75-92.

19. Bowden, G. J., Maier, H. R., and Dandy, G. C. (2002). "Optimal division of data for neural network models in water resources applications." Water Resources Research, 38(2), 2.1-2.11.

20. Braddock, R. D., Kremmer, M. L., and Sanzogni, L. (1998). "Feed-forward artificial neural network model for forecasting rainfall run-off." Environmetrics, 9, 419-432.

21. Breiman, L. (1994). "Comment on 'Neural networks: A review from a statistical by B. Cheng and D. M. Titterington." Statistical Science, 9(1), 38-42.

22. Brown, M., and Harris, C. J. (1994). Neurofuzzy adaptive modeling and control, Prentice-Hall, Englewood Cliffs, New Jersey.

23. Buntine, W. L., and Weigend, A. S. (1991). "Bayesian back-propagation." Complex Systems, 5, 603-643.

24. Burke, L. I., and Ignizio, J. P. (1992). "Neural networks and operations research: An overview." Computer and Operations Research, 19(3/4), 179-189.

25. Cai, S., Toral, H., Qiu, J., and Archer, J. S. (1994). "Neural network based objective flow regime identification in air-water two phase flow." Canadian Journal of Chemical Engineering, 72, 440-445.

26. Cal, Y. (1995). "Soil classification by neural-network." Advances in Engineering Software, 22(2), 95-97.

27. Campolo, M., Soldati, A., and Andreussi, P. (1999). "Forecasting river flow rate during low-flow periods using neural networks." Water Resources Research, 35(11), 3547-3552.

28. Castellano, G., Fanelli, A. M., and Pelillo, M. (1997). "An iterative pruning algorithm for feed-forward neural networks." IEEE Transactions on Neural Networks, 8(3), 519-531.

29. Caudill, M. (1988). "Neural networks primer, Part III." AI Expert, 3(6), 53-59.

30. Celik, S., and Tan, O. (2005). "Determination of pre-consolidation pressure with artificial neural network." Civil Engineering and Environmental Systems, 22(4), 217-231.

31. Chan, W. T., Chow, Y. K., and Liu, L. F. (1995). "Neural network: An alternative to pile driving formulas." Computers and Geotechnics, 17, 135-156.

32. Chen, Y., Azzam, R., and Zhang, F. (2006). "The displacement computation and construction pre-control of a foundation pit in Shanghai utilizing FEM and intelligent methods." Geotechnical and Geological Engineering, 24(6), 1781-1801.

33. Chester, D. L. (1990). "Why two hidden layers are better than one." International Joint Conference on Neural Networks, 1, 265-268.

34. Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function." Mathematics of Control, Signals, and Systems, 3, 303-314.

35. Das, S. K., and Basudhar, P. K. (2006). "Undrained lateral load capacity of piles in clay using artificial neural network." Computers and Geotechnics, 33(8), 454-459.

36. Ellis, G. W., Yao, C., and Zhao, R. (1992). "Neural network modeling of the mechanical behavior of sand." Proceedings of the Engineering Mechanics, 421-424.

37. Ellis, G. W., Yao, C., Zhao, R., and Penumadu, D. (1995). "Stress-strain modeling of sands using artificial neural networks." Journal of Geotechnical Engineering, ASCE, 121(5), 429-435.

38. Erzin, Y. (2007). "Artificial neural networks approach for swell pressure versus soil suction behavior." Canadian Geotechnical Journal, 44(10), 1215-1223.

39. Fahlman, S. E. (1988). "Fast learning variations on back-propagation: An empirical study." Proceedings of the 1988 Connectionist Models Summer School, Pittsburgh, 38-51.

40. Fahlman, S. E., and Lebiere, C. (1990). "The cascade-correlation learning architecture." Advances in Neural Information Processing Systems 2, D. S. Touretzky, ed., Morgan Kaufmann, San Mateo, California, 524-532.

41. Faraway, J., and Chatfield, C. (1998). "Time series forecasting with neural networks: A comparative study using the airline data." Applied Statistics, 47(2), 231-250.

42. Fausett, L. V. (1994). Fundamentals neural networks: Architecture, algorithms, and applications, Prentice-Hall, Englewood Cliffs, New Jersey.

43. Feng, X. T., Chen, B., Yang, C., Zhou, H., and Ding, X. (2006). "Identification of visco-elastic models for rocks using genetic programming coupled with the modified particle swarm optimization algorithm." International Journal of Rock Mechanics and Mining Sciences, 43(5), 789-801.

44. Ferentinou, M. D., and Sakellariou, M. G. (2007). "Computational intelligence tools for the prediction of slope performance." Computers and Geotechnics, 34(5), 362-384.

45. Flood, I. (1991). "A Gaussian-based neural network architecture and complementary training algorithm." Proceedings of the International Joint Conference on Neural Networks, New York, 171-176.

46. Flood, I., and Kartam, N. (1994). "Neural networks in civil engineering I: Principles and understanding." Journal of Computing in Civil Engineering, 8(2), 131-148.

47. Fortin, V., Ouarda, T. B. M. J., and Bobee, B. (1997). "Comment on 'The use of artificial neural networks for the prediction of water quality parameters' by H. R. Maier and G. C. Dandy." Water Resources Research, 33(10), 2423-22424.

48. Fu, Q., Hashash, Y. M. A., Hung, S., and Ghaboussi, J. (2007). "Integration of laboratory testing and constitutive modeling of soils." Computers and Geotechnics, 34(5), 330-345.

49. Gardner, M. W., and Dorling, S. R. (1998). "Artificial neural networks (the multilayer perceptron) - A review of applications in the atmospheric sciences." Atmospheric Environment, 32(14/15), 2627-2636.

50. Garson, G. D. (1991). "Interpreting neural-network connection weights." AI Expert, 6(7), 47-51.

51. Ghaboussi, J., and Sidarta, D. E. (1998). "New nested adaptive neural networks (NANN) for constitutive modeling." Computers and Geotechnics, 22(1), 29-52.

52. Giraudel, J. L., and Lek, S. (2001). "A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination." Ecological Modeling, 146, 329-339.

53. Goh, A. T. (2002). "Probabilistic neural network for evaluating seismic liquefaction potential." Canadian Geotechnical Journal, 39(1), 219-232.

54. Goh, A. T., and Goh, S. H. (2007). "Support vector machines: Their use in geotechnical engineering as illustrated using seismic liquefaction data." Computers and Geotechnics, 34(5), 410-421.

55. Goh, A. T., Kulhawy, F. H., and Chua, C. G. (2005). "Bayesian neural network analysis of undrained side resistance of drilled shafts." Journal of Geotechnical and Geoenvironmental Engineering, 131(1), 84-93.

56. Goh, A. T. C. (1994a). "Nonlinear modeling in geotechnical engineering using neural networks." Australian Civil Engineering Transactions, CE36(4), 293-297.

57. Goh, A. T. C. (1994b). "Seismic liquefaction potential assessed by neural network." Journal of Geotechnical & Geoenvironmental Engineering, ASCE, 120(9), 1467-1480.

58. Goh, A. T. C. (1995a). "Empirical design in geotechnics using neural networks." Geotechnique, 45(4), 709-714.

59. Goh, A. T. C. (1995b). "Modeling soil correlations using neural networks." Journal of Computing in Civil Engineering, ASCE, 9(4), 275-278.

60. Goh, A. T. C. (1996a). "Neural-network modeling of CPT seismic liquefaction data." Journal of Geotechnical Engineering, ASCE, 122(1), 70-73.

61. Goh, A. T. C. (1996b). "Pile driving records reanalyzed using neural networks." Journal of Geotechnical Engineering, ASCE, 122(6), 492-495.

62. Goh, A. T. C., and Kulhawy, F. H. (2003). "Neural network approach to model the limit state surface for reliability analysis." Canadian Geotechnical Journal, 40(1235-1244), 6.

63. Goh, A. T. C., Wong, K. S., and Broms, B. B. (1995). "Estimation of lateral wall movements in braced excavation using neural networks." Canadian Geotechnical Journal, 32, 1059-1064.

64. Gokceoglu, C., Yesilnacar, E., Sonmez, H., and Kayabasi, A. (2004). "A neuro-fuzzy model for modulus of deformation of jointed rock masses." Computers and Geotechncis, 31(5), 375-383.

65. Gribb, M. M., and Gribb, G. W. (1994). "Use of neural networks for hydraulic conductivity determination in unsaturated soil." Proceedings of the 2nd International Conference on Ground Water Ecology, Bethesda, 155-163.

66. Habibagahi, G., and Bamdad, A. (2003). "A neural network framework for mechanical behavior of unsaturated soils." Canadian Geotechnical Journal, 40(3), 684-693.

67. Haj-Ali, R., Pecknold, D. A., Ghaboussi, J., and Voyiadjis, G. Z. (2001). "Simulated micromechanical models using artificial neural networks." Journal of Engineering Mechanics, 127(7), 730-738.

68. Hammerstrom, D. (1993). "Working with neural networks." IEEE Spectrum, 30(7), 46-53.

69. Hanna, A. M., Morcous, G., and Helmy, M. (2004). "Efficiency of pile groups installed in cohesionless soil using artificial neural networks." Canadian Geotechnical Journal, 41(6), 1241-1249.

70. Hanna, A. M., Ural, D., and Saygili, G. (2007). "Neural network model for liquefaction potential in soil deposits using Turkey and Taiwan earthquake data." Soil Dynamics and Earthquake Engineering, 27(6), 521-540.

71. Hashash, Y. M. A., Jung, S., and Ghaboussi, J. (2004). "Numerical implementation of a neural network based material model in finite element analysis." International Journal for Numerical Methods in Engineering, 59(7), 989-1005.

72. Hassoun, M. H. (1995). Fundamentals of artificial neural networks, MIT Press, Cambridge.

73. Hecht-Nielsen, R. (1987). "Kolmogrov's mapping neural network existence theorem." Proceedings of the 1st IEEE International Joint Conference on Neural Networks, San Diego, CA, III-11-III-14.

74. Hecht-Nielsen, R. (1989). "Theory of the back-propagation neural network." Proceedings of the International Joint Conference on Neural Networks, Washington, DC, 593-605.

75. Hecht-Nielsen, R. (1990). Neurocomputing, Addison-Wesely Publishing Company, Reading, MA.

76. Hornik, K., Stinchcombe, M., and White, H. (1989). "Multilayer feed-forward networks are universal approximators." Neural Networks, 2, 359-366.

77. Hush, D. R., and Horne, B. G. (1993). "Progress in supervised neural networks." IEEE SP Magazine, 10(1), 8-39.

78. Jain, A., Sudheer, K. P., and Srinivasulu, S. (2004). "Identification of physical processes inherent in artificial neural network rainfall runoff models." Hydrological Processes, 18(3), 571-581.

79. Javadi, A., Rezania, M., and Mousavi, N. M. (2006). "Evaluation of liquefaction induced lateral displacements using genetic programming." Computers and Geotechnics, 33(4-5), 222-233.

80. Johari, A., Habibagahi, G., and Ghahramani, A. (2006). "Prediction of soil-water characteristic curve using genetic programming." Journal of Geotechnical and Geoenvironmental Engineering, 132(5), 661-665.

81. Juang, C. H., and Chen, C. J. (1999). "CPT-based liquefaction evaluation using artificial neural networks." Computer-Aided Civil and Infrastructure Engineering, 14(3), 221-229.

82. Karnin, E. D. (1990). "A simple procedure for pruning back-propagation trained neural networks." IEEE Transactions on Neural Networks, 1(2), 239-242.

83. Kavli, T. (1993). "ASMOD - An algorithm for adaptive spline modeling of observation data." International Journal of Control, 58(4), 947-967.

84. Kim, Y., and Kim, B. (2006). "Use of artificial neural networks in the prediction of liquefaction resistance of sands." Journal of Geotechnical and Geoenvironmental Engineering, 132(11), 1502-1504.

85. Kim, Y., and Kim, B. (2008). "Prediction of relative crest settlement of concrete-faced rockfill dams analyzed using an artificial neural network model." Computers and Geotechnics, 35(3), 313-322.

86. Kingston, G. B., Lambert, M. F., and Maier, H. R. (2005a). "Bayesian parameter estimation applied to artificial neural networks used for hydrological modeling." Water Resources Research, 41(W12409), doi:10.1029/2005WR004152.

87. Kingston, G. B., Maier, H. R., and Lambert, M. F. (2005b). "Calibration and validation of neural networks to ensure physically plausible hydrological modeling." Journal of Hydrology, 314(2005), 158-176.

88. Kingston, G. B., Maier, H. R., and Lambert, M. F. (2006). "A probabilistic method to assist knowledge extraction from artificial neural networks used for hyrdological prediction." Mathematical and Computer Modeling, 44(5-6), 499-512.

89. Kingston, G. B., Maier, H. R., and Lambert, M. F. (2008). "Bayesian model selection applied to artificial neural networks used for water resources modeling." Water Resources Research, 44(W04419), doi:10.1029/2007WR006155.

90. Kocjancic, R., and Zupan, J. (2000). "Modeling of the river flow rate: The influence of the training set selection." Chemometric and Intelligent Laboratory Systems, 54, 21-34.

91. Kudrycki, T. P. (1988). "Neural network implementation of a medical diagnosis expert system," MS thesis, College of Engineering, University of Cincinnati.

92. Kung, G. T., Hsiao, E. C., Schuster, M., and Juang, C. H. (2007). "A neural network approach to estimating deflection of diaphram walls caused by excavation in clays." Computers and Geotechnics, 34(5), 385-396.

93. Kurup, P. U., and Dudani, N. K. (2002). "Neural network for profiling stress history of clays from PCPT data." 2002, 128(7), 569-579.

94. Lachtermacher, G., and Fuller, J. D. (1994). "Back-propagation in hydrological time series forecasting." Stochastic and Statistical Methods in Hydrology and Environmental Engineering, K. W. Hipel, A. I. McLeod, U. S. Panu, and V. P. Singh, eds., Kluwer Academic, Dordrecht.

95. Lapedes, A., and Farber, R. (1988). "How neural networks work." Neural Information Processing Systems, American Institute of Physics, 442-456.

96. Lee, C., and Sterling, R. (1992). "Identifying probable failure modes for underground openings using a neural network." International Journal of Rock Mechanics and Mining Science & Geomechanics Abstracts, 29(1), 49-67.

97. Lee, I. M., and Lee, J. H. (1996). "Prediction of pile bearing capacity using artificial neural networks." Computers and Geotechnics, 18(3), 189-200.

98. Lee, S. J., Lee, S. R., and Kim, Y. S. (2003). "An approach to estimate unsaturated shear strength using artificial neural network and hyperbolic formulation." Computers and Geotechnics, 30(6), 489-503.

99. Lefik, M., and Schrefler, B. A. (2003). "Artificial neural network as an incremental non-linear constitutive model for a finite element code." Computer Methods in Applied Mechanics and Engineering, 192(31-32), 3265-3283.

100. Lu, Y. (2005). "Underground blast induced ground shock and its modeling using artificial neural network." Computers and Geotechnics, 32(3), 164-178.

101. MacKay, G. (1992). "A practical Bayesian framework for back-propagation networks." Neural Computation, 4, 448-472.

102. Maier, H. R., and Dandy, G. C. (2000). "Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications." Environmental Modeling & Software, 15(2000), 101-124.

103. Maren, A., Harston, C., and Pap, R. (1990). Handbook of neural computing applications, Academic Press, San Diego, California.

104. Masters, T. (1993). Practical neural network recipes in C++, Academic Press, San Diego, California.

105. May, R. J., Maier, H. R., Dandy, G. C., and Fernando, T. M. (2008). "Non-linear variable selection for artificial neural networks using partial mutual information." Environmental Modeling and Software, doi:10.1016/j.envsoft.2008.03.007.

106. Mayoraz, F., and Vulliet, L. (2002). "Neural networks for slope movement prediction." The International Journal of Geomechanics, 2(2), 153-173.

107. Miller, G. F., Todd, P. M., and Hedge, S. U. (1989). "Designing neural networks using genetic algorithms." Proceedings of the 3rd International Conference on Genetic Algorithms, Arlington, 379-384.

108. Minns, A. W., and Hall, M. J. (1996). "Artificial neural networks as rainfall-runoff models." Hydrological Sciences Journal, 41(3), 399-417.

109. Moon, H. K., Na, S. M., and Lee, C. W. (1995). "Artificial neural-network integrated with expert-system for preliminary design of tunnels and slopes." Proceedings of the 8th International Congress on Rock Mechanics, Rotterdam: Balkema, 901-905.

110. Najjar, Y. M., and Ali, H. E. (1998). "CPT-based liquefaction potential assessment: A neuronet approach." ASCE Geotechnical Special Publication, 1, 542-553.

111. Najjar, Y. M., and Ali, H. E. (1999). "Simulating the stress-strain behavior of Nevada sand by ANN." Proceedings of the 5th U.S. National Congress on Computational Mechanics (USACM), Boulder, Colorado.

112. Najjar, Y. M., Ali, H. E., and Basheer, I. A. (1999). "On the use of neuronets for simulating the stress-strain behavior of soils." Proceedings of the 7th International Symposium on Numerical Models in Geomechanics, Graz, Austria, 657-662.

113. Najjar, Y. M., and Basheer, I. A. (1996). "Neural network approach for site characterization and uncertainty prediction." ASCE Geotechnical Special Publication, 58(1), 134-148.

114. Najjar, Y. M., Basheer, I. A., and McReynolds, R. (1996a). "Neural modeling of Kansan soil swelling." Transportation Research Record, No. 1526, 14-19.

115. Najjar, Y. M., Basheer, I. A., and Naouss, W. A. (1996b). "On the identification of compaction characteristics by neuronets." Computers and Geotechnics, 18(3), 167-187.

116. Najjar, Y. M., and Huang, C. (2007). "Simulating the stress-strain behavior of Georgia kaolin via recurrent neuronet approach." Computers and Geotechnics, 34(5), 346-362.

117. Narendara, B. S., Sivapullaiah, P. V., Suresh, S., and Omkar, S. N. (2006). "Prediction of unconfined compressive strength of soft grounds using computational intelligence techniques: A comparative study." computers and Geotechnics, 33(3), 196-208.

118. Nawari, N. O., Liang, R., and Nusairat, J. (1999). "Artificial intelligence techniques for the design and analysis of deep foundations." Electronic Journal of Geotechnical Engineering, http://geotech.civeng.okstate.edu/ejge/ppr9909.

119. Neaupane, K., and Achet, S. (2004). "Some applications of a back-propagation neural network in geo-engineering." Environmental Geology, 45(4), 567-575.

120. NeuralWare. (1997). NeuralWorks Predict Release 2.1, NeuralWare Inc., Pittsburgh.

121. Ni, S. H., Lu, P. C., and Juang, C. H. (1996). "A fuzzy neural network approach to evaluation of slope failure potential." Journal of Microcomputers in Civil Engineering, 11, 59-66.

122. Olden, J. D., Joy, M. K., and Death, R. G. (2004). "An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data." Ecological Modeling, 178(3-4), 389-397.

123. Padmini, D., Ilamparuthi, K., and Sudheer, K. P. (2008). "Ultimate bearing capacity prediction of shallow foundations on cohesionless soils using neurofuzzy models." Computers and Geotechnics, 35(1), 33-46.

124. Pal, M. (2006). "Support vector machine-based modeling of seismic liquefaction potential." International Journal for Numerical and Analytical Methods in Geomechanics, 30(10), 983-996.

125. Penumadu, D., and Chameau, J.-L. (1997). "Geo-material modeling using artificial neural networks." Artificial neural networks for civil engineers: fundamentals and applications, N. Kartam, I. Flood, and J. Garrett, eds., ASCE, New York, 160-184.

126. Penumadu, D., Jin-Nan, L., Chameau, J.-L., and Arumugam, S. (1994). "Rate dependent behavior of clays using neural networks." Proceedings of the 13th Conference of the International Society of Soil Mechanics & Foundation Engineering, New Delhi, 1445-1448.

127. Penumadu, D., and Zhao, R. (1999). "Triaxial compression behavior of sand and gravel using artificial neural networks (ANN)." Computers and Geotechnics, 24(3), 207-230.

128. Provenzano, P., Ferlisi, S., and Musso, A. (2004). "Interpretation of a model footing response through an adaptive neural fuzzy inference system." Computers and Geotechnics, 31(3), 251-266.

129. Rahman, M. S., Wang, J., Deng, W., and Carter, J. P. (2001). "A neural network model for the uplift capacity of suction cassions." Computers and Geotechnics, 28(4), 269-287.

130. Rankine, R., and Sivakugan, N. (2005). "Prediction of paste backfill performance using artificial neural networks." Proceedings of the 16th International Society for Soil Mechanics and Foundation Engineering Osaka, Japan, 1107-1110.

131. Ray, C., and Klindworth, K. K. (2000). "Neural networks for agrichemical vulnerability assessment of rural private wells." Journal of Hydrologic Engineering, 5(2), 162-171.

132. Rezania, M., and Javadi, A. (2007). "A new genetic programming model for predicting settlement of shallow foundations." Canadian Geotechnical Journal, 44(12), 1462-1472.

133. Ripley, B. D. (1996). Pattern recognition and neural networks, Cambridge University Press, Cambridge.

134. Rizzo, D. M., and Dougherty, D. E. (1994). "Application of artificial neural networks for site characterization using hard and soft information." Proceedings of the 10th International Conference on Computational Methods in Water Resources, Dordrecht: Kluwer Academic, 793-799.

135. Rizzo, D. M., Lillys, T. P., and Dougherty, D. E. (1996). "Comparisons of site characterization methods using mixed data." ASCE Geotechnical Special Publication, 58(1), 157-179.

136. Rogers, L. L., and Dowla, F. U. (1994). "Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling." Water Resources Research, 30(2), 457-481.

137. Rojas, R. (1996). Neural networks: A systematic introduction, Springer-Verlag, Berlin.

138. Romo, M., Garcia, S. R., Mendoza, M. J., and Taboada-Urtuzuastegui, V. (2001). "Recurrent and Constructive-Algorithm Networks for Sand Behavior Modelling." The International Journal of Geomechanics, 1(4), 371-387.

139. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning internal representation by error propagation." Parallel Distributed Processing, D. E. Rumelhart and J. L. McClelland, eds., MIT Press, Cambridge.

140. Salchenberger, L. M., Cinar, E. M., and Lash, N. A. (1992). "Neural networks: A new tool for predicting thrift failures." Decision Science, 23, 899-916.

141. Samui, P. (2008). "Prediction of friction capacity of driven piles in clay using the support vector machine." Canadian Geotechnical Journal, 45(2), 288-296.

142. Sarle, W. S. (1994). "Neural networks and statistical models." Proceedings of the 19th Annual SAS Users Group International Conference, Cary, NC: SAS Institute, 1538-1550.

143. Shahin, M. A. (2008). "Modeling axial capacity of pile foundations by intelligent computing." Proceedings of the BGA International Conference on Foundations, Dundee (Scotland), in press.

144. Shahin, M. A., and Indraratna, B. (2006). "Modeling the mechanical behavior of railway ballast using artificial neural networks." Canadian Geotechnical Journal, 43(1), 1144-1152.

145. Shahin, M. A., and Jaksa, M. B. (2004). "Probabilistic assessment of the uncertainty associated with the pullout capacity of marquee ground anchors." Proceedings of the 9th Australia New Zealand Conference on Geomechanics, Auckland, In Press.

146. Shahin, M. A., and Jaksa, M. B. (2005a). "Modeling the pullout capacity of marquee ground anchors using neurofuzzy technique." Proceedings of the International Journal of Modeling and Simulation, MODSIM 2005, Melbourne, Australia, 66-72.

147. Shahin, M. A., and Jaksa, M. B. (2005b). "Neural network prediction of pullout capacity of marquee ground anchors." Computers and Geotechnics, 32(3), 153-163.

148. Shahin, M. A., and Jaksa, M. B. (2006). "Pullout capacity of small ground anchors by direct cone penetration test methods and neural methods." Canadian Geotechnical Journal, 43(6), 626-637.

149. Shahin, M. A., and Jaksa, M. B. (2008). "Intelligent computing for predicting axial capacity of drilled shafts." International Foundation Congress and Equipment Expo '09, Florida, abstract submitted.

150. Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2001). "Artificial neural network applications in geotechnical engineering." Australian Geomechanics, 36(1), 49-62.

151. Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2002a). "Artificial neural network-based settlement prediction formula for shallow foundations on granular soils." Australian Geomechanics, 37(4), 45-52.

152. Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2003a). "Neurofuzzy networks applied to settlement of shallow foundations on granular soils." Proceedings of the 9th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP9, San Francisco, Millpress, Rotterdam, 1379-1383.

153. Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2004a). "Application of neural networks in foundation engineering." International e-Conference on Modern Trends in Foundation Engineering: Geotechnical Challenges & Solutions, http://www.civil.iitm.ac.in.

154. Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2005a). "Neural network based stochastic design charts for settlement prediction." Canadian Geotechnical Journal, 42(1), 110-120.

155. Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2005b). "Stochastic simulation of settlement of shallow foundations based on a deterministic neural network model." Proceedings of the International Congress on Modeling and Simulation, MODSIM 2005, Melbourne (Australia), 73-78.

156. Shahin, M. A., Maier, H. R., and Jaksa, M. B. (2002b). "Predicting settlement of shallow foundations using neural networks." Journal of Geotechnical & Geoenvironmental Engineering, ASCE, 128(9), 785-793.

157. Shahin, M. A., Maier, H. R., and Jaksa, M. B. (2003b). "Closure to: Predicting settlement of shallow foundations on cohesionless soils using neural networks." Journal of Geotechnical & Geoenvironmental Engineering, ASCE, 128(9), 785-793. of the International Congress on Modeling and Simulation, MODSIM 2003, Townsville, Queensland, 1886-1891.

159. Shahin, M. A., Maier, H. R., and Jaksa, M. B. (2003d). "Settlement prediction of shallow foundations on granular soils using B-spline neurofuzzy models." Computers and Geotechnics, 30(8), 637-647.

160. Shahin, M. A., Maier, H. R., and Jaksa, M. B. (2004b). "Data division for developing neural networks applied to geotechnical engineering." Journal of Computing in Civil Engineering, ASCE, 18(2), 105-114.

161. Shahin, M. A., Maier, H. R., and Jaksa, M. B. (2005c). "Investigation into the robustness of artificial neural network models for a case study in civil engineering." Proceedings of the International Congress on Modeling and Simulation, MODSIM 2005, Melbourne (Australia), 79-83.

162. Shang, J. Q., Ding, W., Rowe, R. K., and Josic, L. (2004). "Detecting heavy metal contamination in soil using complex permittivity and artificial neural networks." Canadian Geotechnical Journal, 41(6), 1054-1067.

163. Shi, J., Ortigao, J. A. R., and Bai, J. (1998). "Modular neural networks for predicting settlement during tunneling." Journal of Geotechnical & Geoenvironmental Engineering, ASCE, 124(5), 389-395.

164. Shi, J. J. (2000). "Reducing prediction error by transforming input data for neural networks." Journal of Computing in Civil Engineering, ASCE, 14(2), 109-116.

165. Sidarta, D. E., and Ghaboussi, J. (1998). "Constitutive modeling of geomaterials from non-uniform material tests." Computers & Geomechanics, 22(10), 53-71.

166. Singh, T. N., and Singh, V. (2005). "An intelligent approach to prediction and control ground vibration in mines." Geotechnical and Geological Engineering, 23(3), 249-262.

167. Sinha, S. K., and Wang, M. C. (2008). "Artificial neural network prediction models for soil compaction and permeability." Geotechnical Engineering Journal, 26(1), 47-64.

168. Sivakugan, N., Eckersley, J. D., and Li, H. (1998). "Settlement predictions using neural networks." Australian Civil Engineering Transactions, CE40, 49-52.

169. Smith, G. N. (1986). Probability and statistics in civil engineering: An introduction, Collins, London.

170. Smith, M. (1993). Neural networks for statistical modeling, Van Nostrand Reinhold, New York.

171. Stone, M. (1974). "Cross-validatory choice and assessment of statistical predictions." Journal of Royal Statistical Society, B 36, 111-147.

172. Sudheer, K. P. (2005). "knowledge extraction from trained neural network river flow models." Journal of Hydologic Engineering, 10(4), 264-269.

173. Sudheer, K. P., and Jain, A. (2004). "Explaining the internal behavior of artificial neural network river flow models." Hydrological Processes, 18(4), 833-844.

174. Sudheer, K. P., Nayak, P. C., and Ramasastri, K. S. (2003). "Improving peak flow estimates in artificial neural network river flow models." Hydrological Processes, 17, 677-686.

158. Shahin, M. A., Maier, H. R., and Jaksa, M. B. (2003c). "Neural and neurofuzzy techniques applied to modeling settlement of shallow foundations on granular soils." Proceedings

175. Teh, C. I., Wong, K. S., Goh, A. T. C., and Jaritngam, S. (1997). "Prediction of pile capacity using neural networks." Journal of Computing in Civil Engineering, ASCE, 11(2), 129-138.

176. Tokar, S. A., and Johnson, P. A. (1999). "Rainfall-runoff modeling using artificial neural networks." Journal of Hydrologic Engineering, 4(3), 232-239.

177. Tutumluer, E., and Seyhan, U. (1998). "Neural network modeling of anisotropic aggregate behavior from repeated load triaxial tests." Transportation Research Record 1615, National Research Council, Washington, DC.

178. Twomey, J. M., and Smith, A. E. (1997). "Validation and verification." Artificial neural networks for civil engineers: Fundamentals and applications, N. Kartam, I. Flood, and J. H. Garrett, eds., ASCE, New York, 44-64.

179. Ural, D. N., and Saka, H. (1998). "Liquefaction assessment by neural networks." Electronic Journal of Geotechnical Engineering, http://www.ejge.com/Ppr9803/ Ppr9803.htm

180. White, H. (1989). "Learning in artificial neural networks: A statistical perspective." Neural Computation, 1, 425-464.

181. Yang, Y., and Rosenbaum, M. S. (2002). "The artificial neural network as a tool for assessing geotechnical properties." Geotechnical Engineering Journal, 20(2), 149-168.

182. Yoo, C., and Kim, J. (2007). "Tunneling performance prediction using an integrated GIS and neural network." Computers and Geotechnics, 34(1), 19-30.

183. Young-Su, K., and Byung-Tak, K. (2006). "Use of artificial neural networks in the prediction of liquefaction resistance of sands." Journal of Geotechnical and Geoenvironmental Engineering, 132(11), 1502-1504.

184. Yu, X.-H. (1992). "Can back-propagation error surface not have local minima." IEEE Transaction on Neural Networks, 3, 1019-1021.

185. Zhao, H. (2008). "Slope reliability analysis using a support vector machine." Computers and Geotechnics, 35(3), 459-467.

186. Zhou, Y., and Wu, X. (1994). "Use of neural networks in the analysis and interpretation of site investigation data." Computer and Geotechnics, 16, 105-122.

187. Zhu, J. H., Zaman, M. M., and Anderson, S. A. (1998a). "Modeling of soil behavior with a recurrent neural network." Canadian Geotechnical Journal, 35(5), 858-872.

188. Zhu, J. H., Zaman, M. M., and Anderson, S. A. (1998b). "Modeling of shearing behaviour of a residual soil with recurrent neural network." International Journal of Numerical and Analytical Methods in Geomechanics, 22(8), 671-687.

189. Zhu, J. H., Zaman, M. M., and Trafalis, T. B. (1996). "Prediction of shear stress-strain behavior of soil with recurrent neural network." Intelligent Engineering Systems Through Artificial Neural Networks, 6, 809-814.

190. Zurada, J. M. (1992). Introduction to artificial neural systems, West Publishing Company, St. Paul.