We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the
TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Study for Application of Artificial Neural Networks in Geotechnical Problems

Hyun Il Park
Samsung C&T
Korea of Republic

1. Introduction

The geotechnical engineering properties of soil exhibit varied and uncertain behaviour due to the complex and imprecise physical processes associated with the formation of these materials (Jaksa, 1995). This is in contrast to most other civil engineering materials, such as steel, concrete and timber, which exhibit far greater homogeneity and isotropy. In order to cope with the complexity of geotechnical behaviour, and the spatial variability of these materials, traditional forms of engineering design models are justifiably simplified. Moreover, geotechnical engineers face a great amount of uncertainties. Some sources of uncertainty are inherent soil variability, loading effects, time effects, construction effects, human error, and errors in soil boring, sampling, in-situ and laboratory testing, and characterization of the shear strength and stiffness of soils.

Although developing an analytical or empirical model is feasible in some simplified situations, most manufacturing processes are complex, and therefore, models that are less general, more practical, and less expensive than the analytical models are of interest. An important advantage of using Artificial Neural Network (ANN) over regression in process modeling is its capacity in dealing with multiple outputs or responses while each regression model is able to deal with only one response. Another major advantage for developing NN process models is that they do not depend on simplified assumptions such as linear behavior or production heuristics. Neural networks possess a number of attractive properties for modeling a complex mechanical behavior or a system: universal function approximation capability, resistance to noisy or missing data, accommodation of multiple nonlinear variables for unknown interactions, and good generalization capability.

Since the early 1990s, ANN has been increasingly employed as an effective tool in geotechnical engineering, including: **constitutive modelling** (Agrawal et al., 1994; Gribb & Gribb, 1994; Penumadu et al., 1994; Ellis et al., 1995; Millar & Calderbank, 1995; Ghaboussi & Sidarta 1998; Zhu et al., 1998; Sidarta & Ghaboussi, 1998; Najjar & Ali, 1999; Penumadu & Zhao, 1999); **geo-material properities** (Goh, 1995; Ellis et al., 1995; Najjar et al., 1996; Najjar and Basheer, 1996; Romero & Pamukcu, 1996; Ozer et al., 2008; Park et al., 2009; Park & Kim, 2010; Park & Lee, 2010; **Bearing capacity of pile** (Chan et al., 1995; Goh, 1996; Bea et al., 1999; Goh et al., 2005; Teh et al., 1997; Lee & Lee, 1996; Abu-Kiefa, 1998; Nawari et al., 1999; Das & Basudhar, 2006, Park & Cho, 2010); **slope stability** (Ni et al., 1995; Neaupane and Achet, 2004; Ferentinou & Sakellariou, 2007; Zhao, 2007; Cho, 2009); **liquefaction** (Agrawal

et al., 1997; Ali & Najjar, 1998; Najjar & Ali, 1998; Ural & Saka, 1998; Juang and Chen, 1999; Goh, 2002; Javadi et al., 2006; Kim & Kim, 2006); **shallow foundations** (Sivakugan et al., 1998; Provenzano et al., 2004; Shahin et al., 2005); and **tunnels and underground openings** (Lee & Sterling, 1992; Moon et al., 1995; Shi, 2000; Yoo & Kim, 2007). For example, the behavior of pile foundations installed in soils is considerably complicated, uncertain, and not yet entirely understood (Baik, 2002). This fact has encouraged many researchers to apply the ANN technique to the prediction of the behavior of foundations such as, modeling the axial and lateral load capacities of deep foundations. Constitutive modeling of soil behavior plays an important role in dealing with issues related to soil mechanics and foundation engineering. Over the past three decades many researchers devoted enormous effort collectively to model soil behavior. However, proposed constitutive models based on elasticity and plasticity theories have limited capability to simulate properly the behavior of soils. This is attributed to reasons associated with the formulation complexity, idealization of soil behavior, and excessive empirical parameters. In this regard, many ANNs have been proposed as a reliable and practical alternative to model the constitutive behavior of soils. Geotechnical properties soils are controlled by factors such as mineralogy; stress history; void ratio; pore water pressure, and the interactions of these factors are difficult to establish solely by traditional statistical methods due to their interdependence. Based on the application of ANNs, methodologies have been developed for estimating several soil properties, including the compression index, shear strength, permeability, soil compaction, lateral earth pressure, and others.

The performance and computational complexity of NNs are mainly based on network architecture, which generally depends on the determination of input, output and hidden layers and number of neurons in each layer. The number of layers and neurons in each layer affect the complexity of NN architecture. NN architectures are discussed at length in several research works (Hecht-Nelson,1987; Bounds et al., 1988; Lawrence & Fredrickson, 1988; Cybenko, 1989; Marchandani & Cao, 1989; Fahlman & Lebiere, 1990; Lawrence, 1994; Goh, 1995; Swingler, 1996; Öztütk, 2003). Nevertheless, there is no clear framework to select the optimum NN architecture and its parameters. Structural design of NN involves the determination of layers and neurons in each layer and selection of training algorithm. In general, parameters of NN architecture are determined by trial and error approach such that the number of neurons in input layer, number of hidden layers, number of neurons in hidden layers and number of neurons in output layer are found using several repeated runs of the system.

The main objective of this chapter is to provide a brief overview of the operation of ANN models, the area, the areas of geotechnical engineering to which ANNs have been applied, and highlights and discusses four important issues which require further attention in the future. The chapter is divided into seven major parts. The first part reviews the background for application of ANN methodology to getechnical engineering. In the second part, an introduction to basic neural network architectures is followed. In the third part, methodologies for designing appropriate network architectures and practical guidelines on finding optimum structure of neural network are shortly discussed. The forth part is the application section, which summarizes the completed applicable work in geotechnical engineering problems and mathematical calculation of an ANN model is illustrated in the fifth part. In the sixth part of this chapter, in order to investigate further research directions of ANNs in geotechnical engineering, author's latest issues of researches related to ANNs are reviewed and then the conclusion is followed in the seventh part.

2. Oververw of the Artificial Neural Network

2.1 The concept of artificial neuron

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites* (See Fig. 1). The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurones. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes. An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.
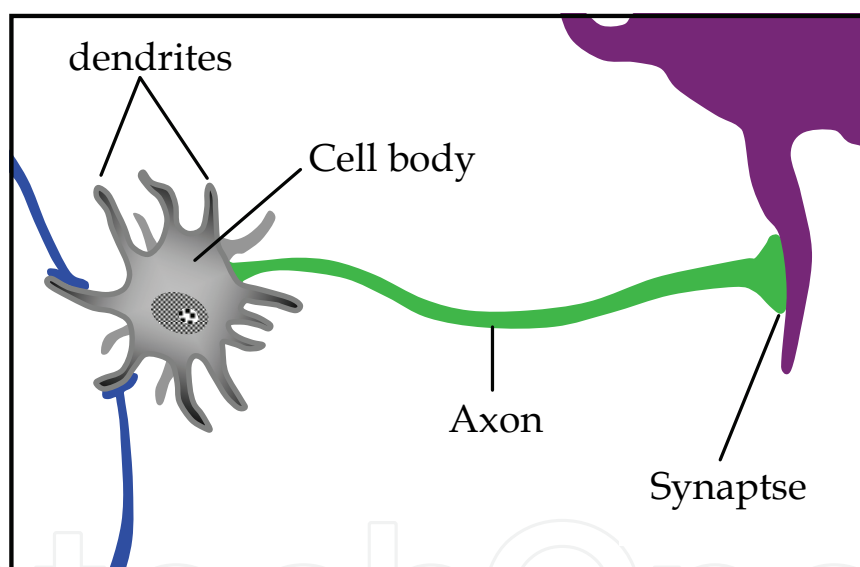

Fig. 1. Biological neuron

2.2 Mathematical modeling of artificial neuron

A *neuron* is an information-processing unit that is fundamental to the peration of a neural network. As shown in Fig. 2, we may identify three basic elements of the neuron model. A set of *synapses*, each of which is characterized by a *weight* or *strength* of its own. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}. It is important to make a note of the manner in which the subscripts of the synaptic weight w_{kj} are written. The first subscript refers to the neuron in question and the second subscript refers to the input end of the synapse to which the weight refers. The weight w_{kj} is positive if the associated synapse is excitatory; it is negative if the synapse is inhibitory. An *adder* for summing the input signals, weighted by the respective synapses of the neuron. An *activation function* for limiting the amplitude of the output of a neuron. The

activation function is also referred to in the literature as a *squashing function* in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0, 1] or alternatively [-1, 1]. The model of a neuron also includes an externally applied bias (threshold) $w_{k0} = b_k$ that has the effect of lowering or increasing the net input of the activation function. In matrix form, we may describe a neuron *k* by writing the following matrix.

$$v_k = \begin{bmatrix} w_{k0} & w_{k1} & \cdots & w_{kp} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_p \end{bmatrix} = w_k^T x \qquad (1)$$
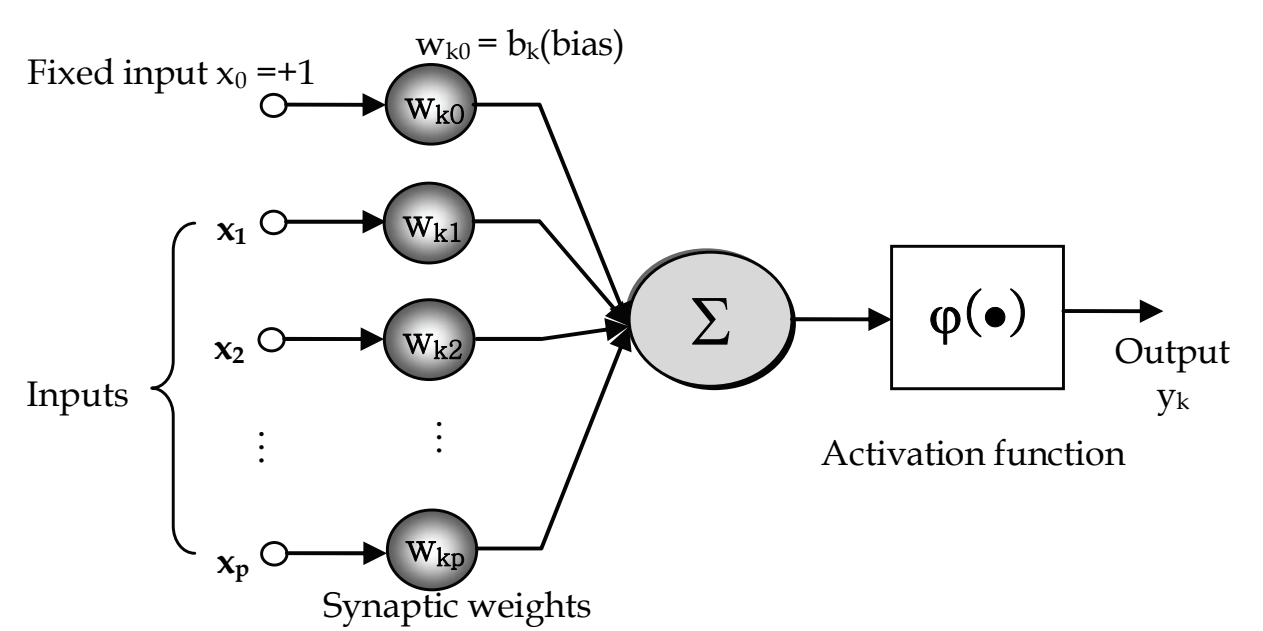

Fig. 2. Basic elements of an artificial neuron

2.3 Activation function

In this section, three of the most common activation functions are presented. An activation function performs a mathematical operation on the output. More sophisticated activation functions can also be utilized depending upon the type of problem to be solved by the network. As is known, a linear function satisfies the superposition concept. The function is shown in Fig. 3(a). The mathematical equation for the above linear function can be written as

$$Y = f(u) = \alpha.u \qquad (2)$$

where α is the slope of the linear function. If the slope α is 1, then the linear activation function is called the identity function. The output (y) of identity function is equal to input function (u). Although this function might appear to be a trivial case, nevertheless it is very useful in some cases such as the last stage of a multilayer neural network.

As shown Fig. 3(b), sigmoidal(S shape) function is the most common nonlinear type of the activation used to construct the neural networks. It is mathematically well behaved, differentiable and strictly increasing function. A sigmoidal transfer function can be written in the following form:

$$f(x) = \frac{1}{1+e^{-\alpha x}} \ , \ 0 \leq f(x) \leq 1 \tag{3}$$

where α is the shape parameter of the sigmoid function. By varying this parameter, different shapes of the function can be obtained as illustrated in Fig. 3(b). This function is continuous and differentiable.
Tangent sigmoidal function is described by the following mathematical form:

$$f(x) = \frac{2}{1+e^{-\alpha x}} - 1 \ , \ -1 \leq f(x) \leq +1 \tag{4}$$
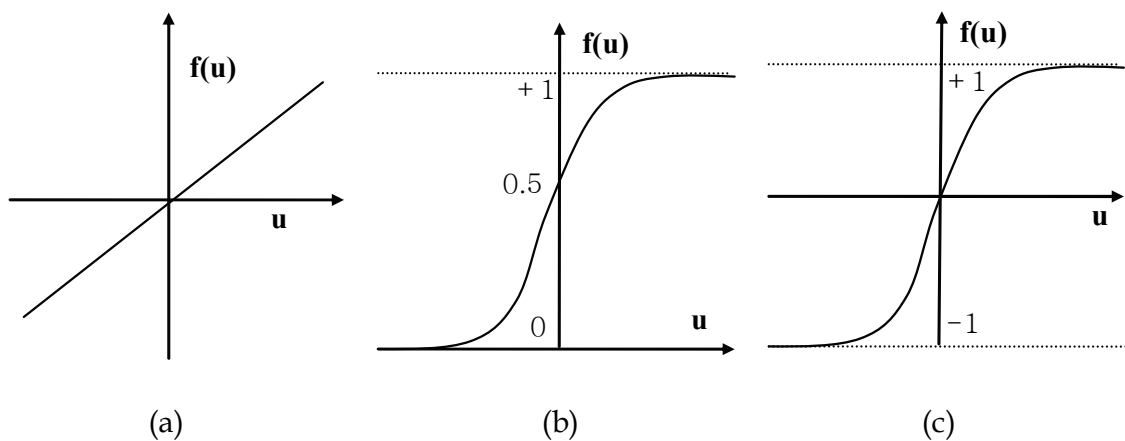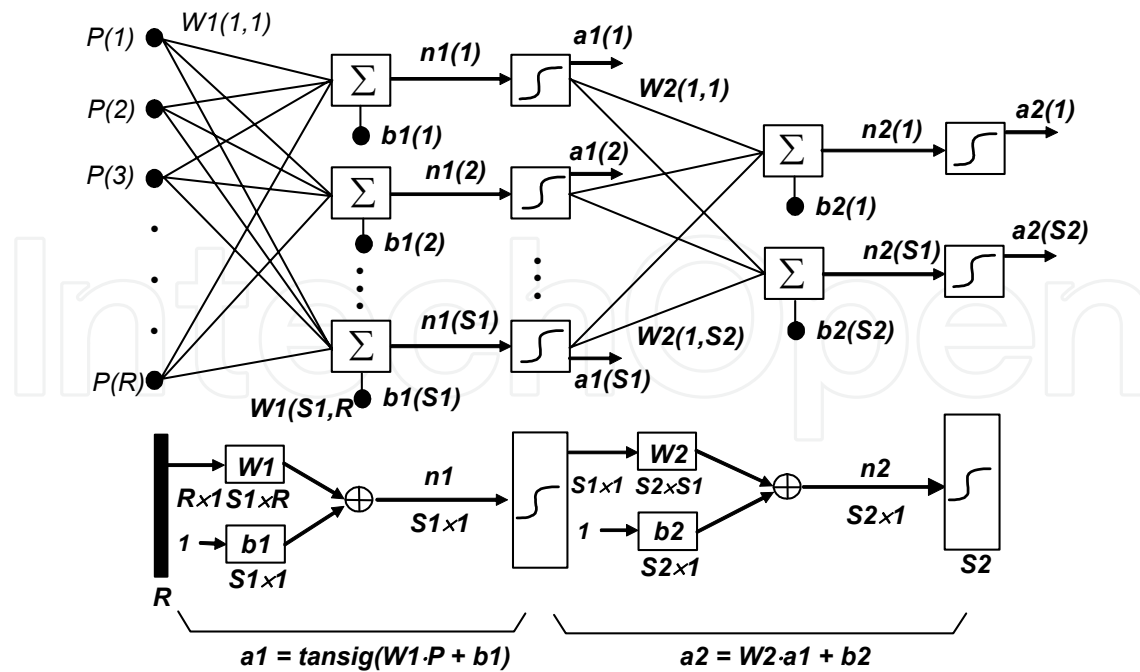

(a)   (b)   (c)

Fig. 3. Activation Function.

2.4 Multilayered Neural Network

The source nodes in the *input layer* of the network supply respective elements of the activation pattern (input vector), which constitute the input signals applied to the neurons (computation nodes) in the second layer (i.e. the first *hidden layer*). The output signals of the second layer are used as inputs to the third layer, and so on for the rest of the network. Typically, the neurons in each layer of the network have as their inputs the output signals of the preceding layer only. The set of output signals of the neurons in the *output layer* of the network constitutes the overall response of the network to the activation pattern supplied by the source nodes in the input layer. The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units (see Fig. 4). The activity of the input units represents the raw information that is fed into the network. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

R = No. of input parameter; S1 = No. of hidden nodes; S2 = No. of output nodes

Fig. 4. Example of Multilayer neural network

2.4 Back-propagation

Backpropagation algorithm (BP) is the most widely used search technique for training neural networks. Information in an ANN is stored in the connection weights which can be thought of as the memory of the system. The purpose of BP training is to change iteratively the weights between the neurons in a direction that minimizes the error E, defined as the squared difference between the desired and the actual outcomes of the output nodes, summed over training patterns (training dataset) and the output neurons. The algorithm uses a sample-by-sample updating rule for adjusting connection weights in the network. In one algorithm iteration, a training sample is presented to the network. The signal is then fed in a forward manner through the network until the network output is obtained. The error between the actual and desired network outputs is calculated and used to adjust the connection weights. Basically, the adjustment procedure, derived from a gradient descent method, is used to reduce the error magnitude. The procedure is firstly applied to the connection weights in the output layer, followed by the connection weights in the hidden layer next to output layer. This adjustment is continued backward through to network until connection weights in the first hidden layer are reached. The iteration is completed after all connection weights in the network have been adjusted. Rumelhart, Hinton, and Williams (1986) popularized the use of BP for learning internal representation in neural networks. Despite their popularity, BP has the drawback of converging to an optimal solution slowly when the gradient search technique is applied. That is, a BP using the gradient search technique has two serious disadvantages: the gradient search technique converges to an optimal solution with inconsistent and unpredictable performance for some applications and when trapped into some local areas, the gradient search technique performs poorly in getting a globally optimal solution. The most major problem during the training process of the neural network is the possible overfitting of training data. That is, during a certain

training period, the network no longer improves its ability to solve the problem. In this case, the training stopped in a local minimum, leading to ineffective results and indicating a poor fit of the model. In order to attempt to prevent these disadvantages, researchers have modified the basic algorithm to try to escape local optima and find the global solution. Numerous modifications have been implemented in order to overcome this problem.

Over-fitting problem or poor generalization capability happens when a neural network over learns during the training period. As a result, such a too well-trained model may not perform well on unseen data set due to its lack of generalization capability. Several approaches have been suggested in literature to overcome this problem. The first method is an early learning stopping mechanism in which the training process is concluded as soon as the overtraining signal appears. The signal can be observed when the prediction accuracy of the trained network applied to a test set, at that stage of training period, gets worsened. The second approach is the Bayesian Regularization. This approach minimizes the over-fitting problem by taking into account the goodness-of-fit as well as the network architecture. Early stopping approach requires the data set to be divided into three subsets: training, test, and verification sets. The training and the verification sets are the norm in all model training processes. The test set is used to test the trend of the prediction accuracy of the model trained at some stages of the training process. At much later stages of training process, the prediction accuracy of the model may start worsening for the test set. This is the stage when the model should cease to be trained to overcome the over-fitting problem. The Bayesian Regularization approach involves modifying the usually used objective function, such as the mean sum of squared network errors (MSE) The modification aims to improve the model's generalization capability. The objective function in Eq. (5) is expanded with the addition of a term, $w E$ which is the sum of squares of the network weights:

$$F = \beta E d + \alpha E w \qquad (5)$$

where the α and β are parameters which are to be optimized in Bayesian framework of MacKay (1992a; 1992b). It is assumed that the weights and biases of the network are random variables following Gaussian distributions and the parameters are related to the unknown variances associated with these distributions.

3. Designing the structure of Artificial Neural Network

Structural design of NN involves the determination of layers and neurons in each layer and selection of training algorithm. The selection of only effective input parameters to the NN is one of the most difficult processes since: (1) there may be interdependencies and redundancies between parameters, (2) sometimes it is better to omit some parameters to reduce the total number of input parameters, and therefore computational complexity of the problem and topology of the network, and (3) NN is usually applied to problems where there is no strong knowledge about the relations between input and output, and therefore it is not clear which of the input parameters are most useful. Moreover, other design parameters of NN architecture, such as the number of neurons in input layer, number of hidden layers, number of neurons in hidden layers and number of neurons in output layer, are found using several repeated runs of the system based on trial and error method. There is no clear framework to select the optimum NN architecture and its parameters (Chung and Kusiak, 1994; Kusiak and Lee, 1996). Nevertheless, some research work has contributed to determine the number of hidden layers, the number of neurons in each layer, selecting the learning rate parameter, and others.

3.1 Determining the number of hidden layers

Determining the number of hidden layers and the number of neurons in each hidden layer is a considerable task. The number of hidden layers is usually determined first and is a critical step. The number of hidden layers required depends on the complexity of the relationship between the input parameters and the output value. Most problems only require one hidden layer, and if relationship between the inputs and output is linear the network does not need a additional hidden layer at all. It is unlikely that any practical problem will require more than two hidden layers(THL). Cybenko (1989) and Bounds et al. (1988) suggested that one hidden layer (OHL) is enough to classify input patterns into different group.

Chester (1990) argued that a THL should perform better than an OHL network. More than one hidden layer can be useful in certain architectures, such as cascade correlation (Fahlman & Lebiere, 1990) and others. A simple explanation for why larger networks can sometimes provide improved training and lower generalization error is that the extra degrees of freedom can aid convergence; that is, the addition of extra parameters can decrease the chance of becoming stuck in local minima or on "plateaus". The most commonly used training methods for back-propagation networks are based on gradient descent; that is, error is reduced until a minimum is reached, whether it be a global or local minimum. However, there isn't clear theory to tell how many hidden units are needed to approximate any given function. If only one input availavle, one sees no advantages in using more than one hidden layer. But things get much more complicated when two or more inputs are given. The rule of thumb in deciding the number of hidden layers is normally to start with OHL (Lawrence, 1994). If OHL does not train well, then try to increase the number of neurons. Adding more hidden layers should be the last option.

3.2 Determining the number of hidden neurons

The choice of hidden neuron size is problem-dependent. For example, any network that requires data compression must have a hidden layer smaller than the input layer (Swingler, 1996). A conservative approach is to select a number between the number of input neurons and the number of output neurons. It can be seen that the general wisdom concerning selection of initial number of hidden neurons is somewhat conflicting. A good rule

Formula	Comments
$h = 2i + 1$	Hecht-Nelson (1987) used Kolmogorov's theorem which any function of I variavles may be represented by the superposition of set of 2i+1 univariate functions-to derive the upper bound for the required number of hidden neurons.
$h = (i + o) / 2$ $\frac{N}{10} - i - o \leq h \leq \frac{N}{2} - i - o$	Lawrence and Fredrickson (1988) suggested that a best estimation for the number of hidden neurons is half the sum of inputs and outputs. Moreover, they proposed the range of number of hidden neurons.
$h = i \log_2 P$	Marchandani and Cao (1989) proposed a equation for best number of hidden neurons

*. h = the number of hidden neurons, i = the number of input neurons, o = the number of output neurons.

Table 1. Rule of thumbs to select the number of neurons in hidden layer

of thumb is to start with the number of hidden neurons equal to half of the number of input neurons and then either add neurons if the training error remains above the training error tolerance, or reduce neurons if the training error quickly drops to the training error tolerance.

3.3 Determining the number of training data

In order to train the neural network well, the number of data set must be carefully decided. An over fitted model could approximate the training data well but generalize poorly to the validation data set. On the other hand, an underfitted model would generalize to the validation data set well but approximate the training data poorly. To avoid over fitting and underfitting is to determine the best number of training observations. No general guidelines are available to achieve this. However, Lawrence and Fredrickson (1988) suggested the following rule of thumb.

$$2\,(i + h + o) \leq N \leq 10((i + h + o) \tag{6}$$

4. ANN applications in geotechnical engineering
4.1 Constitutive Modelling of geo-materials

During the past decades, increasing interest has been shown in the development of a satisfactory formulation for the stress–strain relationships of geo-materials that incorporates a concise statement of nonlinearity, inelasticity and stress dependency based on a set of assumptions and proposed failure criteria. In spite of the considerable complexities of these constitutive models, and due to an inadequate understanding of the mechanisms and all factors involved, it is not possible to capture the complete material response along all complex stress paths and densities. Furthermore, the degree of complexity of these constitutive models (in many cases) inhibits their incorporation into general purpose numerical codes, thus restricting their usefulness in engineering practice (Shin and Pande, 2000). On the other hands, for the convenience of practical in engineering, the model seems to be established simple enough. In the process of establishing the model, the conventional method oversimplifies the soil mechanic behavior. When simplifying the model, parameters have been artificially lessened and only a few of them could be applied in setting up the soil constitutive model while the remaining large number of test data is neglected. Eventually, the model will be poor.

Unlike conventional constitutive models, it needs no prior knowledge, or any constants and/or assumptions about the deformation characteristics of the geo-materials. Other powerful attributes of ANN models are their flexibility and adaptivity, which play an important role in material modeling (Ghaboussi & Sidarta 1998). When a new set of experimental results cannot be reproduced by conventional models, a new constitutive model or a set of new constitutive equations, needs to be developed. However, trained ANN models can be further trained with the new data set to gain the required additional information needed to reproduce the new experimental results. These features ascertain the ANN model to be an objective model that can truly represent natural neural connections among variables, rather than a subjective model, which assumes variables obeying a set of predefined relations (Zhu et al., 1998). So far, ANNs have been applied to the constitutive modeling of rocks, clays, sands, gravels and other geo-materials (Zhu et al., 1998; Millar & Calderbank, 1995; Penumadu et al., 1994; Ellis et al., 1995; Penumadu & Zhao, 1999; Najjar & Ali, 1999)

Ghaboussi and co-workers originally proposed an NN-based framework for constitutive modeling in geomechanics (Ghaboussi & Sidarta, 1998; Sidarta & Ghaboussi, 1998). They

introduced a concept of nested adaptive NNs, which considers the nested structure of the material test data, e.g. dimensionality, stress path dependency or drainage conditions. By means of the finite element (FE) method and the autoprogressive training algorithm proposed in (Ghaboussi et al., 1998), they trained NNs with experimental nonuniform triaxial test data, in order to capture and reproduce the non-linear response of the soil without conventional concepts of the theory of plasticity. In addition, further research proved that the NN-constitutive models can be successfully embedded within the FE codes to compute the consistent tangent stiffness matrix (Shin and Pande, 2000; Hashash et al., 2004). Hashash et al. (2004) demonstrated that a tangent stiffness matrix can be derived from the NN-based material models, using the explicit formulation represented by network parameters. However, the main drawback of the NN-constitutive models is that it is valid only for a specific material for which a new NN has to be adopted each time. Moreover, a material model loses its 'flexibility', which is inherent in the case of conventional models and which is controlled by parameters explicitly describing concepts of plasticity, such as yield surface, flow rule and hardening law.

4.2 Properties of geo-materials

In geotechnical engineering, empirical relationships are often used to estimate certain engineering properties of soils. Using data from extensive laboratory or field testing, these correlations are usually derived with the aid of statistical methods. The relationships between soil parameters are clearly complex, but the degree of interaction enables a degree of statistical correlation to be established, suggesting the promise of a potential for estimation. Developing engineering correlations between various soil parameters is an issue discussed by Goh (1995). Goh used neural networks to model the correlation between the relative density and the cone resistance from cone penetration test (CPT), for both normally consolidated and over-consolidated sands. Laboratory data, based on calibration chamber tests, were used to successfully train and test the neural network model.

The neural network model used soil parameters as inputs and the compression index as a single output(Ozer et al., 2008; Park & Lee, 2010). The ANN models was found to give higher coefficients of correlation than empirical equations for the training and testing data, respectively, which indicated that the neural network was successful in modelling the complex relationship between the compression index and the other soil parameters. Many other studies have successfully used ANNs for modelling soil properties. Ellis et al. (1995) developed an ANN model for sands based on grain size distribution and stress history. Najjar et al. (1996) showed that neural network-based models can be used to accurately assess soil swelling, and that neural network models can provide significant improvements in prediction accuracy over statistical models. Romero and Pamukcu (1996) showed that neural networks are able to effectively characterise and estimate the shear modulus of granular materials. Agrawal et al. (1994); Gribb and Gribb (1994) and Najjar and Basheer (1996) all used neural network approaches for estimating the permeability of clay liners. Park et al. (2010) used ANN models to develop an empirical model for the resilient modulus of subgrade soils and subbase materials from basic material properties and in-situ conditions related to stresses.

Park and Kim (2010a) proposed an ANN model to predict the unconfined compressive strength of reinforced lightweight soil (RLS). RLS consisting of dredged soil, cement, air-foam, and waste fishing net is considered to be an eco-friendly backfilling material in construction because it provides a means to recycle both dredged soil and waste fishing net.

Several series of laboratory tests were performed to investigate the unconfined compressive strength of RLS in various mixing ratios. It may be difficult to find an optimum mixing ratio of RLS considering the design criteria and the construction's situation using the limited test results because the unconfined compressive strength is complicatedly influenced by various mixing ratios of admixtures. As a result, in order to expedite the field application of reinforced lightweight soil, an appropriate prediction method is needed. However, since the strength of RLS is strongly influenced by the mixing ratio of each admixture (i.e., cement, water, air foam, and waste fishing net), it is difficult to empirically formulate a mathematical relationship between the strength and the admixture content of the composite materials. An ANN model that predict the strength of RLS at a given mixing ratio was developed using experimental test results performed on various mixing admixture contents.
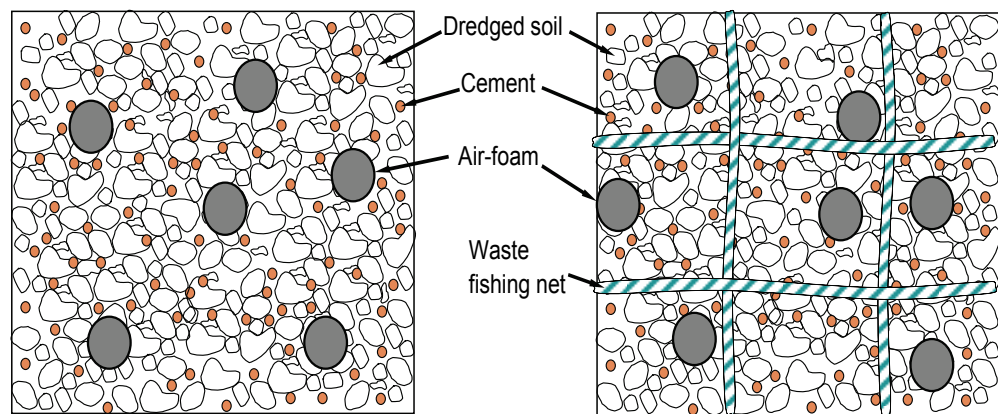

Fig. 5. Schematic diagram of (a) unreinforced and (b) reinforced light-weight soil (Park & Kim, 2010)

As shown in Fig.6(a) the proposed NN model has four nodes in the input layer, four nodes in the hidden layer, and one node in the output layer Fig. 6(a). Fig. 6(b) shows the relationship between the output targets (measured values) and predicted values obtained through the training and testing process. the model shows very good correlation to the

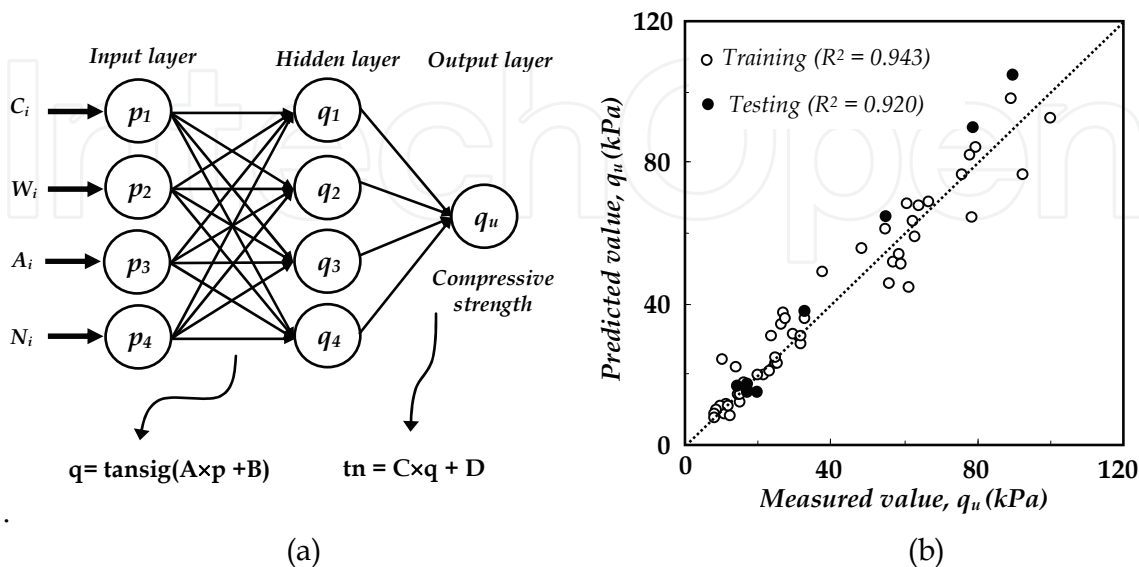(a)   (b)

Fig. 6. Architecture for the developed artificial neural network (Park & Kim, 2010)

training and testing data. As shown in Fig. 7, the developed ANN model is able to obtain the complex behaviors between the compressive strength of RLS and the mixing ratios of admixtures. It has been proven that NN is well suited to modeling the complex behavior of most geo-materials which, by their very nature, exhibit extreme variability.
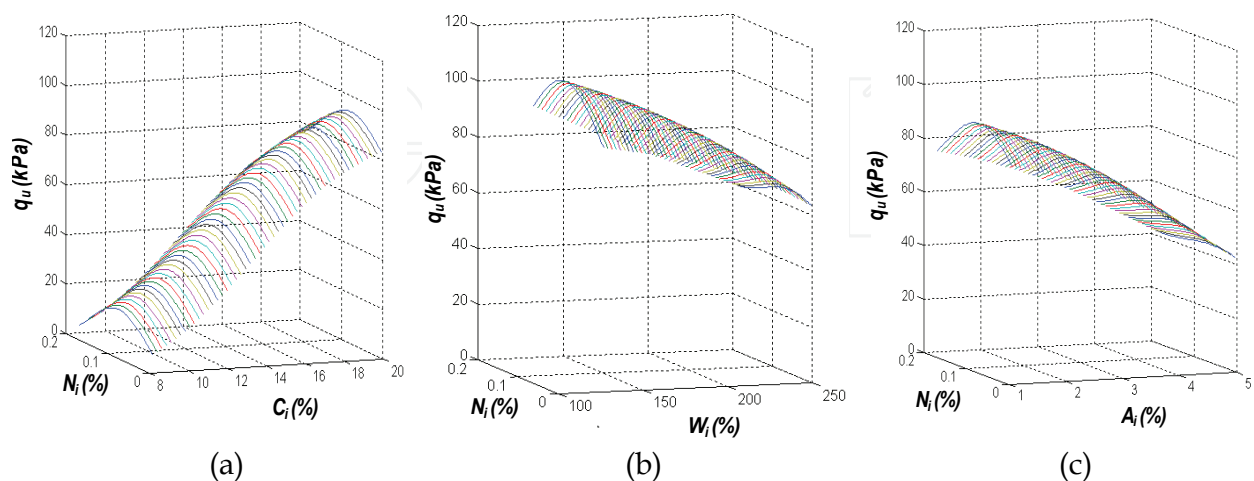

Fig. 7. The unconfined compressive strength with variation of input parameters (Park & Kim, 2010)

4.3 Pile capacity

Design of axial loaded pile can be done be solving equations of static equilibrium whereas design of lateral loaded piles requires solution of nonlinear differential equations (Poulos & Davis, 1980). Other semi-empirical methods used for lateral load capacity of piles are due to Hansen (1961), Broms (1964) and Meyerhof (1976). Although numerous investigations have been performed over the years to predict the behavior and capacity of piles, the mechanisms are not yet entirely understood. Predicting pile capacity is a difficult task because there are a large number of parameters affecting the capacity which have complex relationships with each other. It is extremely difficult to develop appropriate relationships between various essential parameters, including the soil condition, pile type, driving condition, time effect, and others. Baik (2002) illustrated that these factors include the soil condition (type of soil, density, shear strength, etc.), information related to the piles' shape (diameter, penetration depth, whether the tip of pile is open-ended or closed-ended, etc.), and other information (driving method, driving energy, set-up effect, etc.). Although many methods predicting pile resistance have been presented, they did not appropriately consider the various parameters that affect pile resistance. The main criticism of these methods is that they oversimplify the complicated mechanism of pile resistance, and the soil characteristics, type of pile, and information on driving conditions are not properly taken into account.

Hence, ANN models could be an alternate approach for the above case. Goh (1995) used back propagation neural network (BPNN) to predict the skin friction of pile in clay. Goh (1995; 1996) observed that ultimate load capacity of driven timber, pre-cast concrete and steel piles in cohesionless soils using ANN was found to outperform the methods like Engineering News formula, the Hiley formula and the Janbu formula. Chan et al. (1995) and Teh et al. (1997) found that the static pile capacity predicted by using neural network have

excellent agreement with the same obtained by using the commercially available computer code CAPWAP (GRL, 1972). Lee and Lee (1996) used neural networks to predict the ultimate bearing capacity of piles based on model and in situ pile load test results. Abu-Kiefa (1998) used a generalized regression neural network (GRNN), which is a type of probabilistic neural network to predict the pile load capacity considering separately the tip, the shaft and total load capacity of piles driven in cohesionless soils. Nawari et al. (1999) have used neural networks for prediction of axial load capacity of steel H-piles, steel piles and pre-stressed and reinforced concrete piles using both BPNN and GRNN. They also predicted the top settlement of drill shaft due to lateral load based on in situ testing.

Park and Cho (2010) applied an artificial neural network (ANN) to predict the resistance of driven piles in dynamic load tests. They collected 165 data sets for driven piles at various construction sites in Korea. Predictions on the tip, shaft, and total pile resistance were made for piles with available corresponding measurements of such values. The results indicate that the ANN model serves as a reliable and simple predictive tool to appropriately consider various essential parameters for predicting the resistance of driven piles. The proposed neural network model has seven nodes in the input layer, eight nodes in the hidden layer, and three nodes in the output layer (Fig. 8). In order to find an appropriate combination of transfer functions providing good correlation in training and testing stage, various combinations using log-sigmoid, tan-sigmoid and linear was applied to hidden layer and output layer. The combination of transfer functions applied to the hidden layer and output layer neurons are tan-sigmoid ($2/(1+e^{-2n})-1$) and linear, respectively.
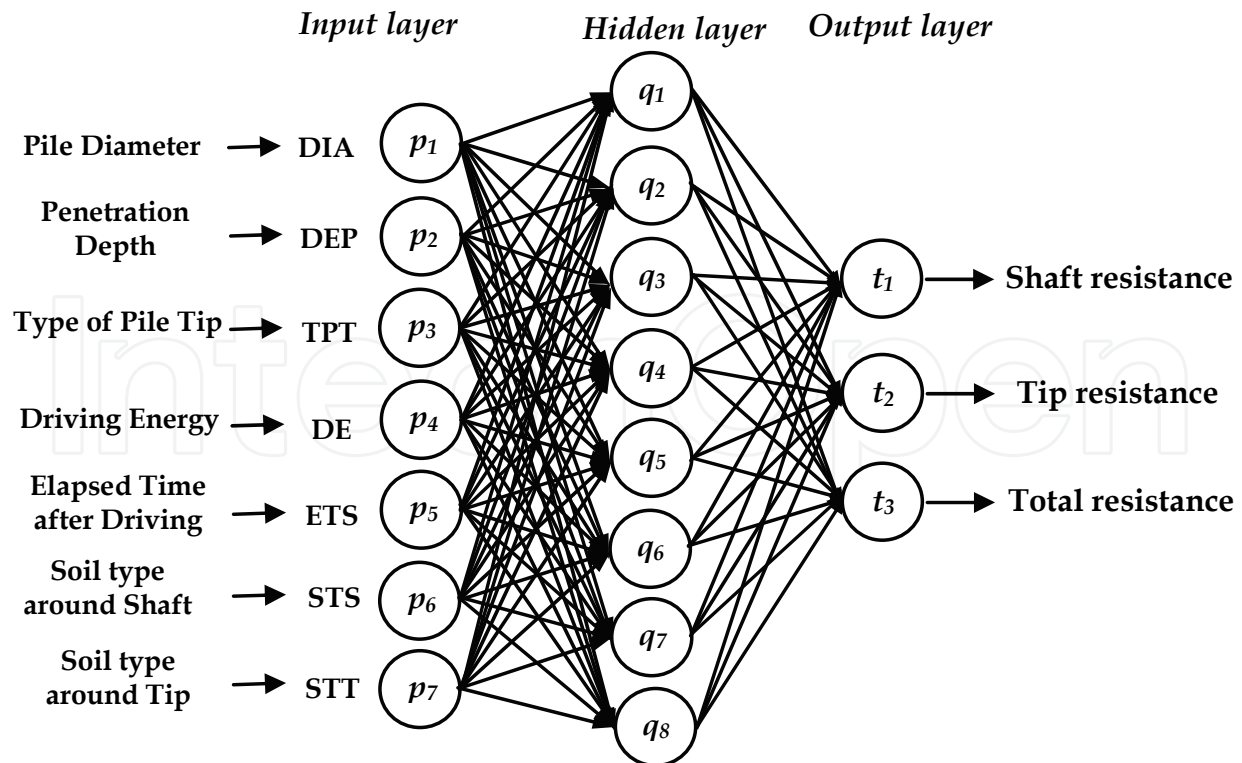

Fig. 8. Architecture of the artificial neural network model (Park & Cho, 2010)

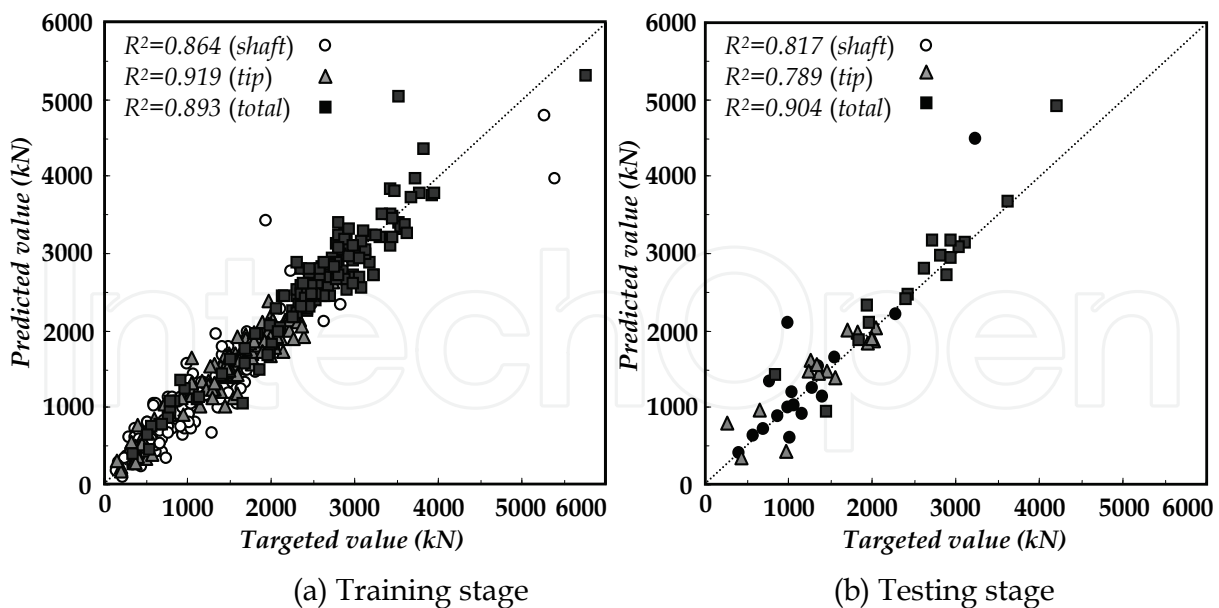(a) Training stage   (b) Testing stage

Fig. 9. Comparison of predicted and measured pile resistance (Park and Cho, 2010)

4.4 Slope stability

Slope stability is important because slope failures or landslides can lead to the loss of life and property. Slope failures are complex natural phenomena that constitute a serious natural hazard in many countries. Limited data and unclearly defined problems often complicate the study of landslides (Nieuwenhuis 1991). To prevent or mitigate the landslide damage, slope-stability analyses and stabilization require an understanding and evaluation of the processes that govern the behavior of the slopes. The factor of safety based on an appropriate geotechnical model as an index of stability, is required in order to evaluate slope stability. Black-box models, based on the Artificial Neural Networks (ANNs), currently attract many researchers studying slope instability, owing to their successful performance in modeling non-linear multivariate problems (Ni et al., 1995; Neaupane & Achet, 2004; Sakellariou & Ferentinou, 2005; Cho, 2009; Wang et al., 2005). Many variables are involved in slope stability evaluation and the calculation of the factor of safety requires geometrical data, physical data on the geologic materials and their shear-strength parameters (cohesion and angle of internal friction), information on pore-water pressures, etc. To evaluate slope instability, the complexity of the slope system requires employment of new methods that are efficient in predicting this nonlinear characteristic of natural landslides.

5. Practical mathematical formulation of ANN

5.1 Mathematical formulation

Training a neural network is conducted by presenting a series of example patterns for associated input and output values. Initially, when a network is created, the connection weights and biases are set to random values. The performance of an ANN model is measured in terms of an error criterion between the target output and the calculated output. The output calculated at the end of each feed-forward computation is compared with the target output to estimate the mean-squared error, as shown in Eq. (7)

$$E = \sum_{i=1}^{Num} (T_i - t_i)^2 \qquad (7)$$

where, Num = number of target data, T_i = ith target output, t_i = ith calculated output, respectively.

An algorithm called back-propagation is then used to adjust the weights and biases until the mean-squared error is minimized. The network is trained by repeating this process several times. Once the ANN is trained, the prediction mode simply consists of propagating the data through the network, giving immediate results. In this study, the training data sets (inputs and target outputs) were normalized according to Eq. (8). Processing of the training data was performed so that the processed data were in the range of -1 to +1. The output of the network was trained to produce outputs in the range of -1 to +1, and we converted these outputs back into the same units used for the original targets.

$$\mathbf{pn} = 2\,(\,\mathbf{p} - \min \mathbf{p}\,)\,/\,(\,\max \mathbf{p} - \min \mathbf{p}\,) - 1 \,,\, \mathbf{tn} = 2\,(\,\mathbf{t} - \min \mathbf{t}\,)\,/\,(\,\max \mathbf{t} - \min \mathbf{t}\,) - 1 \qquad (8)$$

where $\mathbf{p}$ = a matrix of input vectors; $\mathbf{t}$ = a matrix of target output vectors; $\mathbf{pn}$ = a matrix of normalized input vectors; $\mathbf{tn}$ = a matrix of normalized target output vectors; $\max \mathbf{p}$ = a vector containing the maximum values of the original input; $\min \mathbf{p}$ = a vector containing the minimum value of the original input; $\max \mathbf{t}$ = a vector containing the maximum value of the target output; and $\min \mathbf{t}$ = a vector containing the minimum value of the target output. The normalized data were then used to train the neural network to obtain the final connection weights. The data from the output neuron have to be post-processed to convert it back into non-normalized units as shown in Eq. (9).

$$\mathbf{t} = 0.5 \cdot (\mathbf{tn} + 1) \cdot (\max \mathbf{t} - \min \mathbf{t}) + \min \mathbf{t} \qquad (9)$$

The normalized output is then obtained by propagating the normalized input vector through the network as follows:

$$\mathbf{tn} = \mathbf{W2} \times \log sig\,(\mathbf{W1} \times \mathbf{pn} + \mathbf{B1}) + \mathbf{B2} \qquad (10)$$

where $\mathbf{W1}$ = a weight matrix representing connection weights between the input layer neurons and the hidden layer; $\mathbf{B1}$ = a weight matrix representing connection weights between the hidden layer neurons and the output neuron; $\mathbf{W2}$ = a bias vector for the hidden layer neurons; and $\mathbf{B2}$ = a bias for the output neuron. The log-sigmoid function $\log sig$ is defined in Eq. (3).

The output $\mathbf{t}$ is then obtained using Eq. (9) and (10):

$$\mathbf{t} = 0.5 \cdot (\,\mathbf{W2} \times \log sig\,(\,\mathbf{W1} \times \mathbf{pn} + \mathbf{B1}\,) + \mathbf{B2} + 1\,) \cdot (\max \mathbf{t} - \min \mathbf{t}) + \min \mathbf{t} \qquad (11)$$

where the transfer function in the hidden layer is the log-sigmoid activation function $a = 1/(1 - e^{-n})$, and the transfer function in the output layer is the linear function $a = n$.

5.2 Example calculating pile resistance using ANN model(Park and Cho, 2010)

The proposed neural network model has seven nodes in the input layer, eight nodes in the hidden layer, and three nodes in the output layer (Fig. 8). In this study, the soil types near the tip and shaft of pile were classified as shown in Table 2. Weight matrix and bias vector used in the ANN model are summarized in Table 3.

Classification of soil	Value
Clay	1
Silt – Clay	2
Silt	3
Sand – Clay	4
Sand – Silt	5
Fine Sand	6
Sand	7
Sand – Gravel	8

Table 2. Classification according to soil types near the shaft and the tip of pile

W1	0.910	-1.070	-3.323	1.594	0.376	-1.196	-2.252	B1	1.189
	-0.785	0.189	-1.658	-0.106	0.133	1.922	-0.266		0.169
	2.505	0.625	-1.354	-0.422	-4.459	-0.615	1.252		-1.676
	2.871	2.612	-1.622	-0.413	-4.854	0.259	0.277		-0.712
	1.397	2.235	0.354	-0.972	0.194	-1.625	-2.250		-0.889
	0.227	4.302	-2.049	-0.753	0.391	1.649	-1.787		2.777
	-0.153	-0.506	-0.284	-3.868	-0.795	-1.434	1.386		-3.926
	0.058	-4.905	-0.370	0.882	-0.158	-0.712	-3.116		1.408
W2	1.510	-0.472	-3.371	3.190	0.110	-1.474	-0.079	-1.192	B2	0.598
	-0.417	-3.524	3.203	-2.910	-3.145	3.588	-0.768	1.880		-0.899
	1.230	-2.128	-1.662	1.631	-1.397	0.317	-0.441	-0.231		0.543

*. Matrix W1 (8×7), B1 (8×1), W2 (3×8), and B2 (3×1) is used in Eq. (9).

Table 3. Weight matrix and bias vector for ANN Model

The input vector **p** is selected obtained given as follows:

$$p = \begin{bmatrix} DIA \\ DEP \\ TPT \\ DE \\ ETS \\ STS \\ STT \end{bmatrix} = \begin{bmatrix} 0.508 \\ 9.6 \\ 0 \\ 36.3 \\ 31 \\ 3 \\ 3 \end{bmatrix}$$

The normalized input vector **pn** could be calculated using eq. (8) and **min p** and **max p** vectors are given in Table 4.

$$pn = \begin{bmatrix} 0.396 \\ -1.0 \\ -1.0 \\ -0.473 \\ 0.442 \\ 0 \\ -0.429 \end{bmatrix}$$

	Input parameters							Output values		
	DIA (m)	DEP (m)	TPT	DE (kN·m)	ETS (day)	STS	STT	Shaft (kN)	Tip (kN)	Total (kN)
Max.	0.273	0	9.6	1.3	0	1	1	154	158	360
Min	0.610	1	42.8	102.0	43	5	8	5401	2742	6126

*. For the type of pile tip(TPT), 0 represents a closed-ended tip and 1 represents an open-ended one.

Table 4. Maxiimum and minimum values of input parameters and output values

The normalized output could be calculated by propagating the normalized input vector as follows.

$$
A \times pn + B =
\begin{bmatrix}
0.910 & -1.07 & -3.323 & 1.594 & 0.376 & -1.196 & -2.252 \\
-0.785 & -0.189 & -1.658 & -0.106 & 0.133 & 1.922 & -0.266 \\
2.505 & 0.625 & -1.354 & -0.422 & -4.459 & -0.615 & 1.252 \\
2.871 & 2.612 & -1.622 & -0.413 & -4.854 & 0.259 & 0.277 \\
1.397 & 2.235 & 0.354 & -0.972 & 0.194 & -1.625 & -2.250 \\
0.227 & 4.302 & -2.049 & -0.753 & 0.391 & 1.649 & -1.787 \\
-0.153 & -0.506 & -0.284 & -3.868 & -0.795 & -1.434 & 1.386 \\
0.058 & -4.905 & -0.370 & 0.882 & -0.158 & -0.712 & -3.116
\end{bmatrix}
\times
\begin{bmatrix}
0.396 \\
-1.0 \\
-1.0 \\
-0.473 \\
0.442 \\
0 \\
-0.429
\end{bmatrix}
+
\begin{bmatrix}
1.189 \\
0.169 \\
-1.676 \\
-0.712 \\
-0.889 \\
2.777 \\
-3.926 \\
1.408
\end{bmatrix}
=
\begin{bmatrix}
6.321 \\
1.550 \\
-2.262 \\
-2.633 \\
-1.415 \\
1.908 \\
-2.314 \\
7.554
\end{bmatrix}
$$

$$
\log sig(A \times pn + B) =
\begin{bmatrix}
0.998 \\
0.825 \\
0.094 \\
0.067 \\
0.196 \\
0.871 \\
0.090 \\
1.000
\end{bmatrix}
$$

$$
tn = C \times \log sig(A \times pn + B) + D =
\begin{bmatrix}
1.510 & -0.472 & -3.371 & 3.190 & 0.110 & -1.474 & -0.079 & -1.192 \\
-0.417 & -3.524 & 3.203 & -2.910 & -3.145 & 3.588 & -0.768 & 1.880 \\
1.230 & -2.128 & -1.662 & 1.631 & -1.397 & 0.317 & -0.441 & -0.231
\end{bmatrix}
\times
\begin{bmatrix}
0.998 \\
0.825 \\
0.094 \\
0.067 \\
0.196 \\
0.871 \\
0.090 \\
1.000
\end{bmatrix}
+
\begin{bmatrix}
0.598 \\
-0.899 \\
0.543
\end{bmatrix}
=
\begin{bmatrix}
-0.848 \\
0.205 \\
-0.299
\end{bmatrix}
$$

The normalized output **tn** could be translated to real Pile resistance values using Eq. (9).

t=0.5·(tn+1)·(max t−min t)+min t=

$$
0.5 \cdot \left(\begin{bmatrix} -0.848 \\ 0.205 \\ -0.299 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right) \times \left(\begin{bmatrix} 5401 \\ 2742 \\ 6126 \end{bmatrix} - \begin{bmatrix} 154 \\ 158 \\ 360 \end{bmatrix} \right) + \begin{bmatrix} 154 \\ 158 \\ 360 \end{bmatrix} = \begin{bmatrix} 543.7 \\ 1715.1 \\ 2258.8 \end{bmatrix}
$$

Measured values for shaft, tip and total resistance of pile are 529.7, 1785.4 and 2315.2 kN and predicted values using ANN model are 543.7, 1715.1 and 2258.8 kN, respectively

6. Advances in ANN technology

6.1 Automatic design of ANN structure
6.1.1 Overviews
Neural network (NN), also called artificial neural system, is an information processing technique which is developed to simulate the functions of a human brain. Although ANN is an effective algorithm for solving complex engineering problems, only few approaches are available to design the network and most of them rely on iterative procedures. The design of network architecture mainly consists of the network layers, number of neurons of each layer, the transfer functions between layers, and the appropriate selections of a training algorithm. Especially, there are some kinds of input variables and values in which some of them may not carry important information to define the relationship between the input and output. These values can be ignored for the sake of solution convergence and efficiency, even sometimes at the cost of losing some input information. This provides smaller network models, which may be more desirable because of computational resource requirements and generalization capability. Therefore, the present study applies GA to select only effective inputs of network to decrease the time required to design smaller network and to reduce the computational complexity of problems. GA is used to find the best combination of only effective input parameters to provide a solution with less computational process.

To make an ANN more efficient, the computational complexity of ANN should be reduced. The computational complexity of network are generally affected by the number of neurons in each layer. And the network performs poorly as the model become larger and more complex. Although the design methodology of structure of ANN was described in the chapter three, the structure of ANN have to be designed by the trial and error approach, which runs repeatedly to find the network architecture. There is no general framework for the selection of the optimum ANN architecture and its parameters.

Genetic Algorithm (GA) is a very effective approach in solving problems from a wide range of applications, which is difficult to solve with traditional techniques. GA works by repeatedly modifying a population of artificial structures through the application of genetic operators (Goldberg, 1989). There have been a large number of applications of the GA for the NN especially for the evaluation of the weights and the architecture as a search engine to improve the convergence speed of network. Yu and Liang (2001) presented a hybrid approach involving ANN and GA to solve job-shop scheduling problem. The computational ability of the hybrid approach, ANN's computability and GA's searching efficiency, is strong enough to deal with complex scheduling problems.

Park & Kim (2011) proposed the hybrid design method based on ANN and GA. In their approach, a trained NN was employed to model the complex relationships among the parameters related to the geotechnical problems, whereas GA was applied to determine a set of optimal architecture of NN including input parameters, number of hidden layer and each layer's neuron, combination of transfer function between layers. The hybrid approach involving ANN and GA was developed and implemented. It consists of two unit: an NN prediction unit and a GA optimization unit. As shown in Fig. 10, their procedure can be summarized as follows:

1. First, an initial population, which contains a number of sets including information about the structure of ANN, is randomly generated. Then the individuals stored in it are fed into a NN-based prediction unit.
2. The predicted quality measures, which related to objective function, are used to indicate the fitness of the individuals. Evaluate the fitness of each individual according to the rank-based fitness.
3. Based on the fitness, select individuals and place them in the mating pool according to the rank-based fitness assignment and stochastic universal sampling.
4. Do crossover and mutation to the current population to create new individuals.
5. Insert a number of new random individuals replacing old individuals in the current population randomly. Make sure that the inserted individuals did not replace the best individual in the population.
6. Evaluate the fitness of each individual.
7. Steps 3–6 are called a generation, and they are repeated until a certain stop criterion is met. Typical stop criteria in a genetic algorithm run include a predefined maximum number of generations or an error smaller than a predefined value. In our genetic algorithm, maximum number of generations is used.
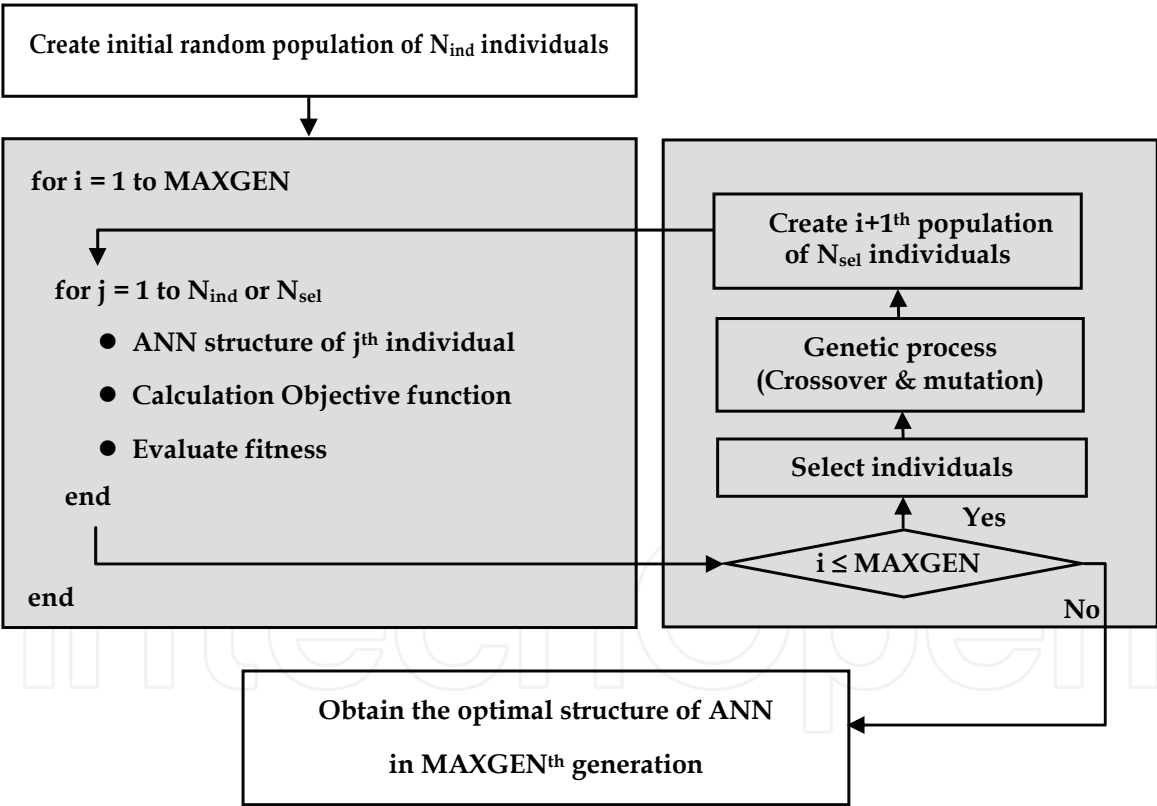

Fig. 10. Schematic flow chart of determination of optimal structure of ANN (Park & Kim, 2011)

6.1.2 Creation of initial population

The hybrid ANN-GA approach starts with the generation of an initial population, which contains a predefined number of chromosomes (strings). Each chromosome is composed of binary strings that include the design information of ANN's structure. For example, in case of design condition given in Table 5, a chromosome created is presented in Fig. 11.

parameters	values
Total number of input variables, N_{ini}	7
Maximum number of hidden layer, N_{HL}	2
Maximum node number in hidden layer, N_{HN} = 15	15
Transfer functions which can be used between layers	linear function, sigmoid function, tangent-sigmoid function

Table 5. An Example of design information to determine the structure of ANN

Fig. 11. Design information about the structure of ANN included in chromosome (Park & Kim, 2011)

- Node number of input layer, N_{in} = 6
- Number of hidden layer, N_{hl} = 1(in case of 0, N_{hl} = 1 and in case of 1, N_{hl} = 2)
- Number of Node of hidden layer, N_{hn} = $2^3 \times 0 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 = 5$
- Information of transfer function : Determination of the combination of transfer functions using five binary strings

This chromosome is composed of the eighteen binary strings. First seven binary strings in the chromosome include the information about the selection of input parameters. Six binary strings deal with the input variables used for the network architecture, with the 0 code indicating that a variable that cannot be used and with the 1 code indicating that a variable can be used. There are seven input variables, in this chromosome; seven binary strings present that the first six inputs should be kept, and the last two inputs removed. One Hidden layer was selected and five node was applied to the hidden layer. The information about transfer function is included in the other five binary strings. For example, a population of q individuals can be created as follows:

$$
\begin{aligned}
P^1 &= 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \\
P^2 &= 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1 \\
\vdots &= \qquad\quad \vdots \qquad\quad \vdots \qquad\quad \vdots \\
P^q &= 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0
\end{aligned}
\tag{11}
$$

6.1.3 Genetic operation

GA is an optimization procedure that operates on sets of design variables. Each set is called a string and it defines a potential. Each string consists of a series of characters representing the values of the discrete design variables for a particular solution. The fitness of each string is the measurement of the performance of the design variables as defined by the objective function.

In its simplest form, a genetic algorithm consists of three operations: (1) reproduction, (2) crossover, and (3) mutation (Goldberg, 1989). Each of these operations is described below.

The reproduction operation is the basic engine of Darwinian natural selection by the survival of the fittest. The reproduction process promotes the information stored in strings with good fitness values to survive into the next generation. The next generation of offspring strings is developed from the selected pairs of parent strings exposed to the application of explorative operators such as crossover and mutation.

Crossover is a procedure in which a selected parent string is broken into segments, some of which are exchanged with corresponding segments of another parent string. In this manner, the crossover operation creates variations in the solutions population by producing new solution strings that consist of parts taken from a selected parent string.

Fig. 12. Genetic process using crossover (Park & Kim, 2011)

The mutation operation is introduced as an insurance policy to enforce diversity in a population. It introduces random changes in the solution population by exploring the possibility of creating and passing features that are nonexistent in both parent strings to the offsprings. Without an operator of this type, some possibly important regions of the search space may never be explored.

6.1.4 Definition of objective function

The objective function for each individual is computed by Eq. 12. The objective function of the ith individual, ObjV(i) is composed of the error function, E_i, calculated as the difference between measured values and predicted values, and the penalty function, P_i, calculated on the basis of the complexity of structure of ANN. The complex structure of an ANN model increases the probability that the value of the error function will decrease, but generality is more likely to decrease due to overfitting. Therefore, the penalty function, P_i, is included in the objective function to control the decrease of generality.

$$ObjV(i) = E_i + P = \left(\sum_{k=1}^{N_{mea}} \frac{|T_k - t_k|}{T_{\max}} \right) / N_{mea} + \alpha \cdot \left(\frac{N_n^i / N_{\max} + CW^i / CW_{\max}}{2} \right) \qquad (12)$$

where $\alpha = 0.01$; N_{mea} = the total number of measured data; T_{max} = the maximum value among measured values; T_k = kth measured value; and t_k = kth predicted value; N^i_n = total number of nodes used in the ith chromosome; N_{max} = the maximum number of nodes that can be applied to the structure of ANN in this study; CW^i = total number of connections used in the ith chromosome; and CW_{max} = the maximum number of connections that can be applied to the structure of ANN in this study.

6.2 Example analysis

The developed methodology was estimated through it's application to the geotechnical problem which ANN was used. The optimal ANN model obtained through opmization process based the developed GA-NN method was compared with the ANN model obtained in basis of researcher's experiance. Rahman et al. (2001) develoved an ANN model to predict the uplift capacity of suction caissons which are frequently used for the anchorage of large compliant offshore structures. The uplift capacity of the suction caissons is a critical issue in these applications. the developed neural network model has five nodes in the input layer, ten nodes in the hidden layer, and one nodes in the output layer. The five input parameters to the neural network model are the aspect ration of caisson (L/d), the undrained shear strength of the caly soil in which the caisson is installed (s_u), the relative depth of the lug to which the caisson forces is applied (D/L), the angle that the chain force makes with the horizontal (θ), and the loading rate defined with respect ot the soil permeability (T_k). the transfer functions applied to the hidden layer and output layer neurons are tan-sigmoid and log-sigmoid functions, respectively.
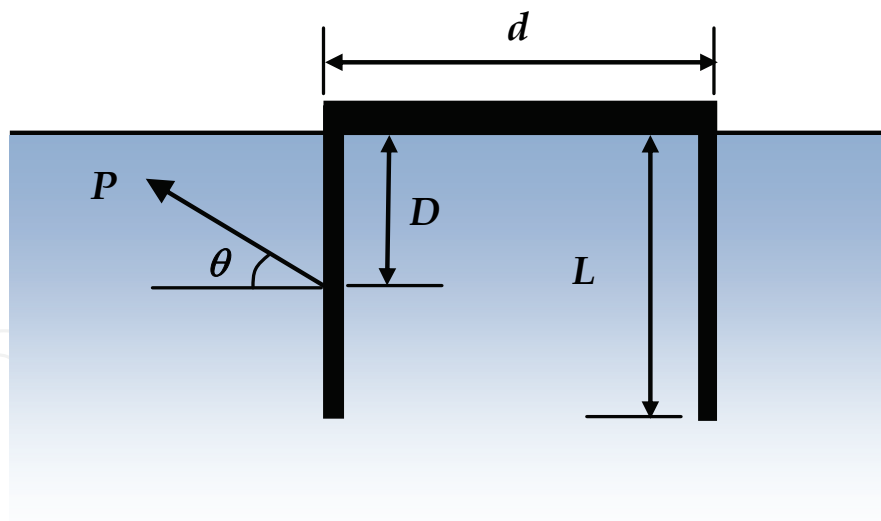

Fig. 13. Description for suction cassion

Design information for the application of GA-NN method is given in Table 6. Through the optimization process using the developed method, the optimal structure of ANN model is obtained in Table 7. Three input variables, D/L, T_k, and θ was removed through the optimization based GA-NN method. The optimized number of hidden node was decreased compared with Rahman et al. (2001)'s model. the transfer functions of the hidden layer and output layer were obtained as tan-sigmoid and linear functions, respectively.

Parameters		Values
GA paraemters	Number of initial population, N_{ind}	400
	Number of maximum generation, MAXGEN	40
	Number of seleced individuals for genetic process, N_{sel}	400×0.9 = 360
	Probability of mutation, P_{mut}	0.005
NN parameters	Maximum number of input node, IL_{max}	11
	Maximum number of hidden layer, HLmax	2
	Maximum node number in each hiddlayer, NH_{max}	16

Table 6. Design condition for application of the developed GA-NN method

Method	No of input node	No. of hidden node	Transfer function		R^2	
			I-H	H-O	Training	Testing
Traditional method	5	10	tansig	logsig	0.970	0.997
GA-NN	2	7	tansig	linear	0.984	0.982

*. I-H means transfer function connecting input layer to hidden layer, H-O means transfer function connecting hidden layer to output layer. Tansig and logsig means tangent-sigmoid and log-sigmoid function, respectively.

Table 7. Parameters of structure of ANN model obtained by each methods

In Fig. 14, the predictied uplift capacity of ANN model obtained by GA-NN method was compared with those of Rahman et al. (2001)'s ANN model. Even though three input variables were ommited in the prediction and also number of hidden node was decreased, it gave almost same correlation in traing and testing stage. the same the ANN model. It means that three input variable ommited in input layer couldn't affect to output value, uplift capacity in the data sets given by Rahman et al. (2001).
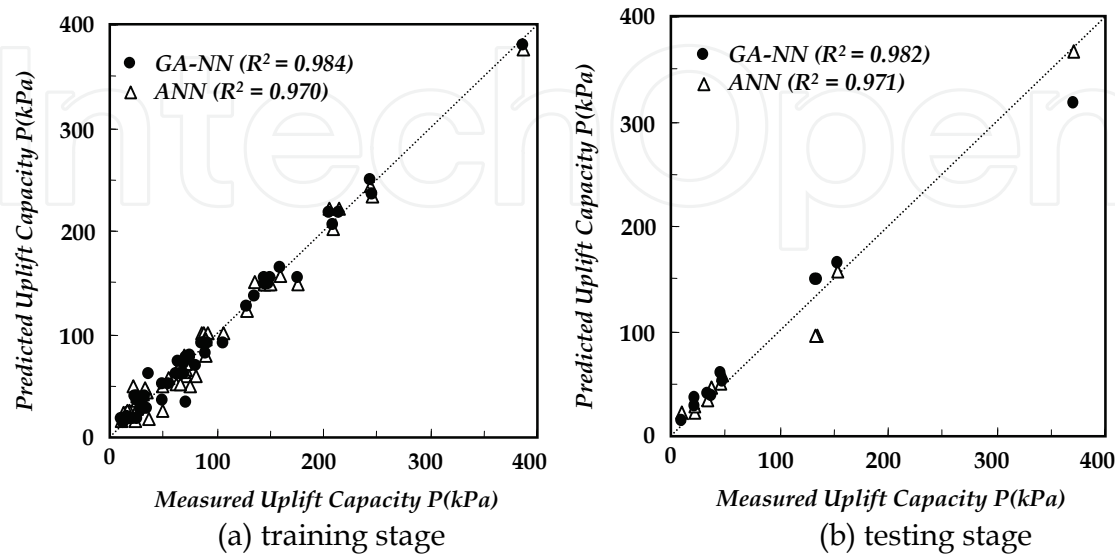

Fig. 14. Comparison of the uplift capacity predicted by each methods (Park & Kim, 2011)

In Fig. 15, the values of correlation coefficient, R² were obtained with variations of number of hidden node and transfer functions in the ANN model obtained by GA-NN method. The R² increased with the number of hidden nodes and then converged to a value after exceeding about seven node. In Eq. 11, Even though the value of error function doesn't decrease any more, the value of complexity fuction should be continually increased with increasing hidden node after seven node. It implies that if seven hidden node gives the minimum value of objective function in comparison of other hidden nodes.

Park & Kim (2011) suggested a hybrid NN/GA approach which is able to design optimal structure of ANN. The proposed approach combines the characteristics of GA and NN to overcome the shortcomings of NN structure design. The results of the proposed approach show that GA may enable the researchers to use NN more effectively and as an efficient tool for the solution of complex problems and reduces the risk of over designing the network architecture. The results of example showed that the performance of NN can be easily guaranteed with GA by selecting the optimal combination of input variables, number of hidden layer, node number of each hidden layer, and transfer functions between layers. GA reduces the complexity and over design of the network structure, as it helps to design smaller network architecture. Processing time of hybrid NN/GA for grouping parts can be decreased nearly to half of the preliminary NN-based approach. In summary, it is seen that GA enables to consider NN as an effective and efficient technique for the computationally complex type problems since it simultaneously reduces the computational complexity and enhances the prediction performance.
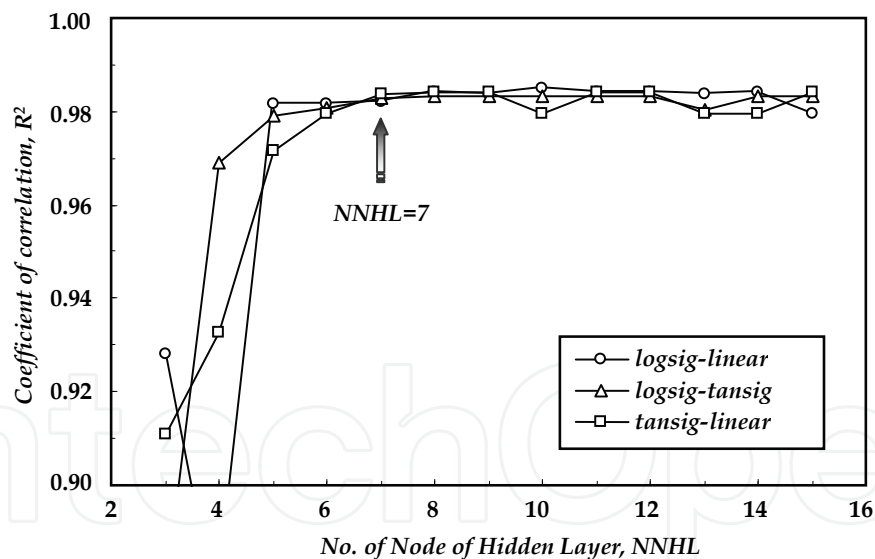

Fig. 15. The values of correlation coefficient with varing the design parameters of ANN model obtained by GA-NN method (Park & Kim, 2011)

6.2 Generalization of Neural Network using committee methodology
6.2.1 Generaliability of Neural Network

Over-training is the most serious problem in neural network training. The drawback is that such a network is quickly over-trained which means that the network error is driven to a small value for the training samples but will become large when new input is presented. This indicates that the network has memorized the training samples but is not able to

generalize to give reasonable answers on unseen input parameter combinations. As a result, such a too well-trained model may not perform well on unseen data set due to its lack of generalization capability. In this section, we focus on one particular problem with learning which is typical for neural networks: their generalization capabilities. Generalization is the ability to train with one data set and then successfully classify independent test sets. Although continued training will increase the training set accuracy, the danger exists that test set accuracy decreases after a certain point.

Approaches considered overcoming the over-fitting problems are early stopping, Bayesian Regularization approach, and others (Hirschen & Schäfer, 2006). One approach is to use early stopping, where the algorithm which minimizes the error function prevent it from doing so by stopping the algorithm at some point. In early stopping the available data is divided into a training, a validation and a test subset. The training set is used for training the network and updating the network weights. The validation subset is not used for training, yet the performance function indicates how the trained network responds to these samples. The validation error will normally decrease during the initial phase of training, as does the training set error. When the network begins to overfit the data, the error on the validation set will typically begin to increase. The test set is not used during the training, but utilized to compare different networks. If the response on the test set is too weak one may decide to restart the network training with a different division of data sets. The second approach is the Bayesian Regularization(MacKay, 1992a). This approach minimizes the over-fitting problem by taking into account the goodness-of-fit as well as the network architecture. The following is the short description about the Bayesian regularization. Typically, training aims to reduce the sum of squared errors $F = E_D$. However, regularization adds an additional term; i.e. the objective function becomes $F=\alpha \cdot E_D+\beta \cdot E_W$, where E_W is the sum of squares of the network weights, and α and β are objective function parameters. The relative size of the objective function parameters dictates the emphasis for training. If $\alpha \ll \beta$, then the training algorithm will drive the errors smaller. If $\alpha \ll \beta$ training will emphasize weight size reduction at the expense of network errors, thus producing a smoother network response (Foresee & Hagan, 1997).

Single multilayer perceptrons (MLPs), consisting of an input layer, a hidden layer and an output layer, trained by a back-propagation algorithm (e.g. Levengerg-Marquardt, see Hagan, Demuth & Beale 1996, pp. 12-19), have been the conventional method of choice for most practical applications over the last decade. However, single MLP, when repeatedly trained on the same patterns, tends to reach different minima of the objective function each time and hence give a different set of neuron weights, because the solution is not unique for noisy data, as in most geotechnical problems. Therefore, a common approach is to train many nets, and then select the one that yields the best generalization performance. Nevertheless, selecting the single best neural network is likely to result in loss of information. While one network reproduces the main patterns, the others may provide the details lost by the first. The aim should be to exploit, rather than lose, the information contained in a set of imperfect generalizers. This is the motivation for the committee neural network approach, where a number of individually trained networks are combined to improve accuracy and increase robustness. Reddy & Buch (2003), Das et al. (2001), Gopinath & Reddy (2000), and Reddy et al. (1995) developed the concept of committee neural networks in which a large number of networks are trained. Based on initial testing with data obtained from subjects not used in training, a few networks are recruited into a committee. A final evaluation of the committee is conducted with data obtained from subjects not used in training or in initial testing.

6.2.2 Overviews of Committee Neural Network (CNN)
The committee technique for neural networks has been used for engineering problems (Reddy & Buch, 2003; Das et al., 2001; Gopinath & Reddy, 2000; Reddy et al., 1995). It was observed that the committee provided good estimates by means of averaging the results of individual networks in the committee, when the individual errors are uncorrelated. In the committee technique, several multiple neural networks (Fig. 16) are constructed and each individual neural network is trained independently with different initial synaptic weights using the training patterns as

$$TP_1 = \left\{ (x_1, t_1) \right\}, \ \ TP_2 = \left\{ (x_2, t_2) \right\}, \ ..., \ TP_N = \left\{ (x_N, t_N) \right\} \tag{13}$$

where TP_i is a training patterns for the ith networks, and x_i and t_i are an input vector and target vector for the ith networks, respectively.

Fig. 16. Illustration of committee of networks (Kim & Park, 2011)

In Fig. 16, y_i is an output vector calculated from the ith networks. A mapping function $f_i(x_i)$ is determined from the ith networks based on the training patterns TP_i, and the error of this function can be calculated as

$$e_i(x_i) = d_i(x_i) - f_i(x_i) \tag{14}$$

where $d_i(x_i)$ is a desired function for the ith networks and is represented as $d_i(x_i) = E[t_i \mid x_i]$
The desired function for the committee of networks is determined as

$$d(x) = E[T \mid X] \tag{15}$$

where $X = \{(x_1, x_2, \ldots, x_N)\}$ and $T = \{(t_1, t_2, \ldots, t_N)\}$.
The committee mapping function can be represented as

$$f_{com}(X) = \sum_{i=1}^{N} \alpha_i f_i = \sum_{i=1}^{N} \alpha_i (d_i - e_i) = \sum_{i=1}^{N} \alpha_i d_i - \sum_{i=1}^{N} \alpha_i e_i = d - \sum_{i=1}^{N} \alpha_i e_i \tag{16}$$

where, α_i is a weighting factor for the ith networks, and $\Sigma \alpha_i = 1$. Therefore, the committee output can be calculated as Eq. (17), where the outputs from different neural networks were averaged as

$$y_{com} = \sum_{i=1}^{N} \alpha_i y_i \qquad (17)$$

The mean square error (MSE) of f_{com} can be calculated as

$$e\left[f_{COM}\right] = E\left[\left(d - f_{COM}\right)^2\right] = E\left[\left(\sum_{i=1}^{N} \alpha_i e_i\right)^2\right] = E\left[\sum_{i,j}^{N} \alpha_i \alpha_j e_i e_j\right] = \sum_{i,j}^{N} \alpha_i \alpha_j C_{ij} \qquad (18)$$

where C_{ij} is a correlation matrix as $C_{ij}=E[e_i e_j]$.

The local minima in determining the synaptic weights of a single MLP and the non-uniqueness of the solution due to the noise and a limited number of measurements may be resolved by employing the committee technique, which is a statistical approach averaging the outputs in the functional space.

6.2.3 Case study for CNN

Kim and Park (2010) examined the feasibility of committee neural network theory for the improvement of accuracy and consistency of the neural network model on the estimation of preconsolidation pressure from the field piezocone measurements. The validity of the committee technique was also examined through the comparison with a single NN model, an empirical and a theoretical model.

The case records from Chen (1994) are evaluated using neural network. A total of 119 case records are used for the training phase and 28 (randomly selected) for the testing phase. The proposed neural network model has four nodes in the input layer, seven nodes in the hidden layer, and one node in the output layer. In input layer, the total and effective overburden pressures σ_{vo}, σ'_{vo}, the cone tip resistance q_T, and pore pressure measurement behind the cone tip u_2 were selected as input variables.

In their study, twenty single neural networks were trained from the different initial weights and biases but with the same training patterns. Fig. 17(a) and (b) show the coefficients of determination between measured and predicted preconsolidation pressure using the piezocone test result from each of the 20 single NNs for the training data and testing data, respectively. As shown in Fig. 17(a), coefficients of determination for training data from each NN model show very similar accuracy i.e., coefficients of determination $R2$ are almost around 0.93. However, the prediction results for testing data from each NN model aren't as accurate as those of the training data. They significantly fluctuates i.e., they range from 0.84 to 0.94, even though they have the same structural characteristics. Therefore, if a single NN is to be used, the best model must be selected which gives the relatively highest coefficient of determination among various models, e.g., second NN among 20 neural networks, which gives the coefficients of determination of 0.93 and 0.94 in the training and testing phase, respectively. However, in reality, it is quite difficult to choose the best model among a number of candidate NNs.

Several committees of 20 NNs were constructed by changing the accumulated number n of NN in the committee to the equal weighting factor ($\alpha_i=1/n$). Prediction results of each committee are plotted in Fig. 18(a) and 18 (b) with respect to the increase of the accumulated number of NN for training data and testing data, respectively. As can be seen in Fig. 18 (a), the coefficients of determination of the committee neural network still increase with an increase of the number of accumulated NN in the committee for training data. Furthermore,

(a) training stage   (b) testing stage
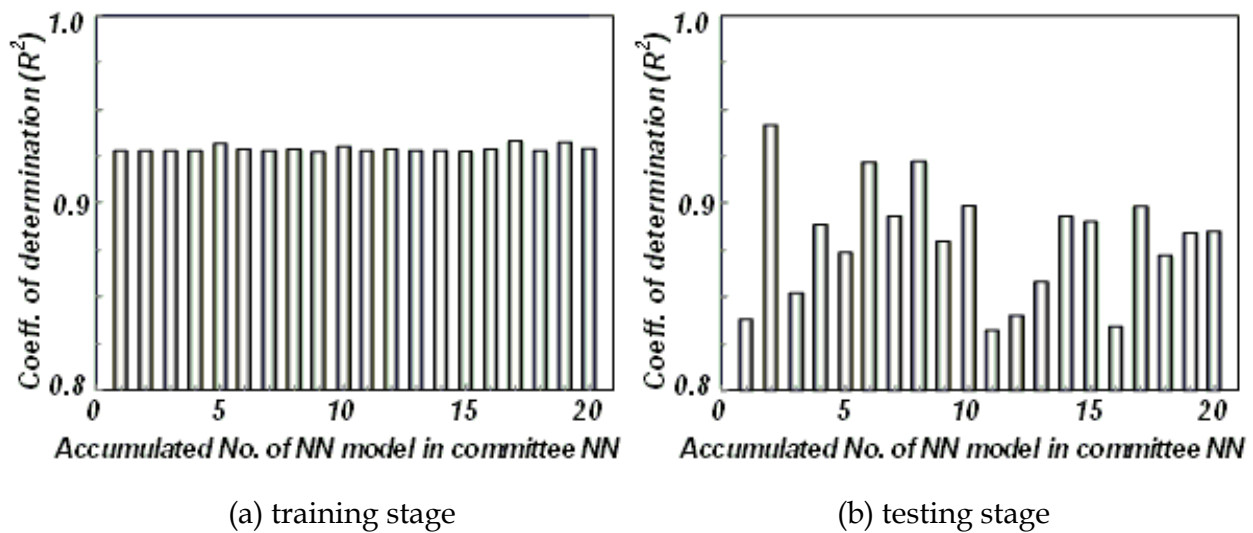
Fig. 17. Prediction performance of 20 MLPs which are optimized with different initial weights and biases by trial-and-error method (Kim & Park, 2010)

as shown in Fig. 18 (b) for testing data, even though the R^2 value of each single NN model shows severe variation, the R^2 values of CNNs don't show such a dramatic variation after accumulating two NN models in the committee. From these figures, it can be concluded that any single NN model still cannot avoid the variation on the prediction due to initial dependency of weight and bias. However, such variation can be eliminated by connecting those NNs with an appropriate weighting factor α_i as a committee neural network. Besides, by introducing Committee methodology, the conventional trial-and-error method for the optimization of the structure of a neural network can be used without any consideration of initial weight dependency and structural optimization. The authors observed that a committee neural network system is able to provide improved performance compared with a single optimal neural network. The committee technique has been found to be a very effective technique to improve the accuracy of the estimation of the preconsolidation pressure σ'_p.

The performance of NN has suffered because of its variation on the prediction of target value due to the localization of weight and bias during the optimization process on the structure. To overcome such problems of the single NN, in this study, structural optimization was carefully carried out by the trial-and-error method. Nevertheless, a single MLP, although it has successfully optimized structures, still cannot avoid the large variation on the prediction of preconsolidation pressure due to its initial weight dependency. Therefore, CNN is introduced to overcome the initial weight dependency of the single neural network model. Various committees of the single MLP were tested. It was found that if 8 single NNs, which have the same structure but have been trained with a different initial weight and bias, are accumulated in the committee with the same weighting factor α_i, any variation on the prediction of the preconsolidation pressure from the piezocone test result can be simply and successfully eliminated. A comparison of the prediction results of CNN with the theoretical and empirical method shows that CNN is significantly more precise and consistent than conventional statistical and theoretical methods.
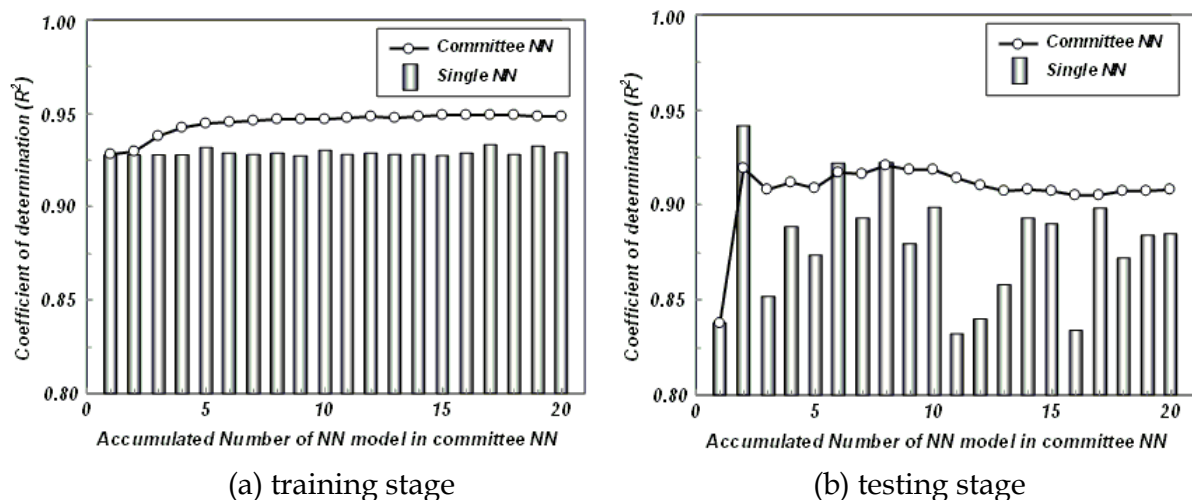
(a) training stage   (b) testing stage

Fig. 18. Improvement of estimation accuracy by accumulating the optimized single NNs in the committee (Kim & Park, 2010)

7. Conclusions

Artificial neural networks (ANNs) have been applied to various problem in geotechnical engineering. This include dams, earth retaining structures, environmental geotechnics, ground anchors, liquefaction, pile foundations, shallow foundations, slope stability, soil properties and behavior, site characterization, tunnels, underground openings, and other areas. In mathematical modeling to solve problem of above the geotechnical engineering area, the lack of understanding for complicated physical behavior is easily supplemented by either over-simplifying the problem or incorporating several assumptions into the model. Consequently, many mathematical models are apt to fail to simulate the complex behavior of geotechnical problems. In contrast, ANN methodology is based on the data alone in which the model can be trained on data sets to find the relationship between inputs and out values. There is no need to simplify the problem nor incorporate an any assumption. As geotechnical engineering exhibits extreme variability, ANNs are particularly amenable to modelling the complex behaviour of these materials and have generally demonstrated superior predictive performance when compared with traditional methods.

In science and engineering problems, there is still no clear procedure to design NN architecture. Therefore, this often causes over design or inefficient network structures especially in the case of complex problems. Although considerable research has been accounted in NN and GA applications, their use in optimal NN design is quite recent. Nevertheless, it is seen that GA enables to consider NN as an effective and efficient technique for the computationally complex type problems since it reduces the computational complexity and enhances the search performance.

In training of ANN model, over-fitting problem or poor generalization capability happens frequently when a neural network over learns during the training period. As a result, such a too well-trained model may not perform well on unseen data set due to its lack of generalization capability. Several approaches have been suggested in literature to overcome this problem. The author introduced the feasibility of committee neural network theory for the improvement of accuracy and consistency of the neural network model on the geotechnical probleme.

8. References

Abu-Kiefa, M. A. (1998). General regression neural networks for driven piles in cohesionless soils. *J. Geotech. Geoenv. Engrg., ASCE,* Vol.123, No.12, (December 1998), pp. 1177–1185, ISSN 1090-0241

Agrawal, G.; Weeraratne, S. & Khilnani, K. (1994). Estimating clay liner and cover permeability using computational neural networks, *Proc., First Congress on Computing in Civil Engrg.,* pp. 20-22, Washington, USA.

Agrawal, G.; Chameau, J.A. & Bourdeau, P.L. (1997). Assessing the liquefaction susceptibility at a site based on information from penetration testing. In *Artificial neural networks for civil engineers: fundamentals and applications*, N. Kartam, I. Flood, J.H. Garrett, (Ed.), 185-214, ASCE, ISBN 0784402256, New York, USA

Ali, H.E. & Najjar, Y.M. (1998). Neuronet-based approach for assessing liquefaction potential of soils, *Transportation Research Record*, No. 1633, (January 1998), pp. 3-8, ISSN 0361-1981.

Baik, K. (2002). Optimum Driving Method for Steel Pipe Piles in Sands, *J. of Civil Engineering*, Vol.22, No.1-C, pp. 45-55.

Bea, R.G.; Jin, Z.; Valle, C. & Ramos, R. (1999). "Evaluation of reliability of platform pile foundations." *J. Geotech. Geoenviron. Eng., ASCE,* Vol.125, No.8, (August 1999), pp. 696–704, ISSN 1090-0241

Bounds, D.G.; Lloyd, PJ.; Mathew, B.; and Waddell, G. (1988). A multilayer perceptron network for the diagnosis of low back pain, *Proc. of 2nd IEEE Annual Int'l Conf. on Neural Networks*, pp. 481-489, San Diego, NJ, USA, June 21-24, 1988,

Broms, B.B. (1964). Lateral resistance of piles in cohesive soils, *J. of Soil Mech. Found. Eng., ASCE,* Vol.90, No.2, (March 1964), pp. 27–63, ISSN 0038-0741

Chan, W.T.; Chow, Y.K. & Liu, L.F. (1995). Neural network: an alternative to pile driving formulas. *Comput Geotech,* Vol.17, No.2, pp. 135–156, ISSN 0266-352X

Chester, D.L. (1990). Why two hidden layers are better than one, *Proc. of 4th IEEE Annual Int'l Conf. on Neural Networks*, pp. 1.265-1.268, Washington, DC, NJ, USA, Jan. 15-19.

Cho, S.E. (2009). Probabilistic stability analyses of slopes using the ANN-based response surface, *Computers and Geotechnics*, Vol.36, pp. 787–797, ISSN 0266-352X

Chung, Y. & Kusiak, A. (1994). Grouping parts with a neural network, *Journal of Manufacturing Systems*, Vol.13, No.4, pp. 262-75

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function, *Mathematics of Control Signals and Systems*, Vol.2, No.4, pp. 303-314.

Das, A.; Reddy, N. P. & Narayanan, J. (2001). Hybrid Fuzzy Logic Committee Neural Networks for Recognition of Swallow Acceleration Signals, *Computer Methods and Programs in Biomedicine*, Vol.64, pp. 87-99.

Das, S.K. & Basudhar, P.K. (2006). Undrained lateral load capacity of piles in clay using artificial neural network, *Computers and Geotechnics*, Vol.33, pp. 454–459.

Ellis G.W.; Yao, C; Zha,o R. & Penumadu, D. (1995). Stress–strain modeling of sands using artificial neural networks, *J Geotech Eng,* Vol.121, No.5, pp. 429–435, ISSN 1089-3032.

Fahlman, S.E. & Lebiere, C. (1990). The cascade correlation learning architecture, In: *Advances in Neural Information Processing Systems*, H, D.S. Tounetzky, (Ed.), Morgan Kaufmann , San Mateo, CA, USA

Ferentinou, M.D. & Sakellariou, M.G. (2007). Computational intelligence tools for the prediction of slope performance, *Computers and Geotechnics*, Vol.34, No.5, pp. 362-384, ISSN 0266-352X
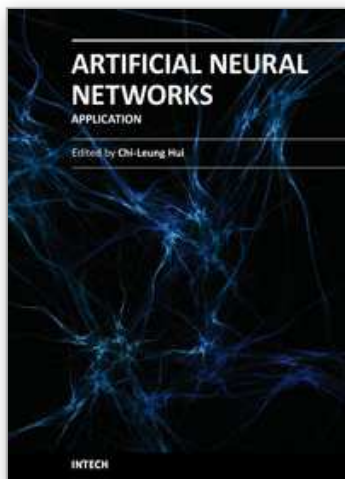
Foresee, F.D. & Hagan, M.T. (1997). Gauss–Newton approximation to Bayesian learning, *Proceedings of the International Joint Conference on Neural Networks,* Vol.3, pp. 1930–1935.

Ghaboussi J. & Sidarta, D.E. (1998). New nested adaptive neural networks (NANN) for constitutive modeling, *Computers and Geotechnics,* Vol.22, No.1, pp. 29–52, ISSN 0266-352X

Ghaboussi J.; Pecknold, D.A.; Zhang, M. & Haj-Ali, R.M. (1998). Autoprogressive training of neural network constitutive models, *International Journal for Numerical Methods in Engineering,* Vol. 42, pp. 105–126, ISSN 0029-5981

Goh, A.T.C. (1996). Pile driving records reanalyzed using neural networks, *J Geotech Engrg,* ASCE, Vol.122, No.6, pp. 492–495, ISSN 1938-6362

Goh, A.T.C. (2002). Probabilistic neural network for evaluating seismic liquefaction potential, Canadian Geotechnical Journal, Vol.39, pp. 219-232, ISSN 0008-3674.

Goh,A.T.C. (1995). Modeling soil correlations using neural networks. *J. Comput. Civil Engrg.,* ASCE, Vol.9, No.4, pp. 275–278, ISSN 1598-2351

Goh, A.T.C.; Kulhawy, F.H. & Chua, C.G. (2005). Bayesian neural network analysis of undrained side resistance of drilled shafts, *Journal of Geotechnical and Geoenvironmental Engineering,* Vol.131, No.1, pp. 84-93.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley, USA.

Gopinath, P. & Reddy, N.P. (2000). Toward Intelligent Web Monitoring Performance of Single Vs Committee Neural Networks, *2000 IEEE EMBS Conference on Information Technology Application in Biomedicine Proceedings*, pp. 179-182.

Gribb, M. M. & Gribb, G. W. (1994). Use of neural networks for hydraulic conductivity determination in unsaturated soil." *Proc., 2nd Int. Conf. Ground Water Ecology*, , Water Resources Assoc., pp. 155-163

GRL Associates, Inc. (1996). CAPWAP User Manual.

Hagan, M.T.; Demuth, B.H. & Beale, M. (1996). Neural Network Design, PWS Pub., USA

Hansen, B. (1961). The ultimate resistance of rigid piles against transversal force, Copenhagen: Danish Geotechnical Institute; 1961. Bulletin No. 12. p. 5–9.

Hashash, Y.M.; Jung, S. & Ghaboussi, J. (2004). Numerical implementation of a neural network based material model in finite element analysis, *International Journal for Numerical Methods in Engineering,* Vol.59, pp. 989–1005

Hecht-Nelson, R. (1987). Kolmogorov's mapping neural network existence theorem, *Proc. of 1st IEEE Annual Int'l Conf. on Neural Networks*, pp. III.11-111.14, San Diego, NJ, USA, June 21-24

Jaksa, M.B. (1995). The influence of spatial variability on the geotechncial design properties of a stiff, overconsolidated clay, PhD thesis, The University of Adelaide, Adelaide.

Javadi, A.A.; Rezania, M. & Mousavi Nezhad, M. (2006). Evaluation of liquefaction induced lateral displacements using genetic programming, *Computers and Geotechnics*, Vol. 33, 222-233, ISSN 0266-352X

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

Hirschen, K. & Schⓔfer, M. (2006) Bayesian regularization neural networks for optimizing fluid flow processes, *Comput. Methods Appl. Mech. Engrg.* Vol.195, pp. 481–500

Kim, Y.S. & Park, H.I. (2011). Committee Neural Network for Estimating Preconsolidation Pressure from Piezocone Test Result, *Engineering Computations*, Submitted

Kim, Y.S. & Kim, B.K. (2006). Use of artificial neural networks in the prediction of liquefaction resistance of sands, *Journal of Geotechnical and Geoenvironmental Engineering*, Vol.132, No.11, pp. 1502-1504.

Kusiak, A. & Lee, H. (1996). Neural computing based design of components for cellular manufacturing, *International Journal of Production Research*, Vol.34, No.7, pp. 1777-1790

Lawrence, J. (1994). *Introduction to Neural Networks: Design, Theory, and Applications*, 6th ed. Nevada City, CA: California Scientific Software.

Lawrence, J. & Fredrickson, J. (1998). *BrainMaker User's Guide and Reference Manual*, 7th Ed., Nevada City, CA: California Scientific Software.

Lee, C. & Sterling, R. (1992). Identifying probable failure modes for underground openings using a neural network, *Int. Journal of Rock Mechanics and Mining Science & Geomechanics Abstracts*, Vol.29, No. 1, pp. 49-67

Lee, I.M. & Lee, J.H. (1996). Prediction of pile bearing capacity using artificial neural networks, *Comput Geotech*, Vo.18, No.3, pp. 189–200, ISSN 0266-352X

MacKay, D.J.C. (1992a). Bayesian Interpolation, *Neural Computation*, Vol.4, No.3, pp. 415-447

MacKay DJC. (1992b). A practical bayesian framework for backpropagation networks. *Neural Computation,* Vol.4, No.3, pp. 448–472.

Marchandani, G. & Cao, W. (1989). On hidden nodes for neural nets, *IEEE Trans. on Circuits and Systems*, Vol.36, No.5, pp. 661-664

Meyerhof, G.G. (1976). Bearing capacity and settlement of pile foundations, *J Geotech Engrg*, ASCE, Vol.102, No.3, pp. 196–228.

Millar, D.L. & Calderbank, P.A. (1995). On the investigation of a multi-layer feedforward neural network model of rock deformability behavior, *International congress on rock mechanics*, pp. 933–938, Tokyo, Japan.

Moon, H.K.; Na, S.M. & Lee, C.W. (1995). Artificial neural-network integrated with expert-system for preliminary design of tunnels and slopes, *Proc. 8th Int. Congress on Rock Mechanics*, pp. 901-905, Balkema.

Najjar, Y.M.; Basheer, I.A. & McReynolds, R. (1996). Neural modeling of Kansan soil swelling, *Transportation Research Record No. 1526*, pp. 14-19

Najjar, Y.M. & Basheer, I.A. (1996). Utilizing computational neural networks for evaluating the permeability of compacted clay liners, *Geotechnical and Geological Engineering*, Vol.14, pp. 193-221.

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

Najjar, Y.M. & Ali, H.E. (1999). On the use of neuronets for simulating the stress–strain behavior of soils, *7th International symposium on numerical models in geomechanics*, pp. 657–662, Austria

Nawari N.O.; Liang, R. & Nusairat, J. (1999). Artificial intelligence techniques for the design and analysis of deep foundations, *Electron. J. Geotech. Eng.*, http://geotech.civeng.okstate.edu/ejge/ppr9909/ index.html.

Neaupane, K.M. & Achet, S.H. (2004). Use of backpropagation neural network for landslide monitoring: a case study in the higher Himalaya, *Engineering Geology*, Vol.74, pp. 213– 226

Ni, S.H.; Lu, P.C. & Juang, C.H. (1995). A fuzzy neural network approach to evaluation of slope failure potential, *Journal of Microcomputers in Civil Engineering*, Vol.11, pp. 59– 66.

Ozer, M.; Isik, N.S. & Orhan, M. (2008). Statistical and neural network assessment of the compression index of clay-bearing soils, *Bull Eng Geol Environ*, Vol.67, pp. 537–545 *Öztürk*, N. (2003). Use of genetic algorithm to design optimal neural network structure, *Engineering Computations*, Vol.20, No.8, pp. 979-997

Park, H.I. (2010). Development of neural network model to estimate the permeability coefficient of soils, *Marine Geosources and Geotechnology*, Accepted

Park, H.I.; Keon, G.C. & Lee, S.R. (2009). Prediction of Resilient Modulus of Granular Subgrade Soils and Subbase Materials Based on Artificial Neural Network, *Road Materials and Pavement Design,* Vol.10, No. 3, pp. 647- 665.

Park, H.I. & Cho, C.H. (2010). Neural Network Model for Predicting the Resistance of Driven Piles *Marine Geosources and Geotechnology*, In Press

Park, H.I. & Lee, S.R. (2010). Evaluation of the compression index of soils using an artificial neural network *Computers and Geotechnics*, Submitted

Park, H.I. & Kim, Y.T. (2010). Prediction of Strength of Reinforced Lightweight Soil Using an Artificial Neural Network, Engineering Computation, In press

Park, H.I. & Kim, Y.S. (2011). Evaluation of Geotechnical Parameters Based on Design of Optimal Neural Network Structure, *Computers and Geotechnics*, Submitted

Penumadu, D. & Zhao, R. (1999). Triaxial compression behavior of sand and gravel using artificial neural networks (ANN), *Comput Geotech*, Vol.24, pp. 207–30, ISSN 0266-352X

Penumadu, D.; Jin-Nan, L.; Chameau, J.L.; Arumugam, S. (1994). Rate dependent behavior of clays using neural networks, *Proceedings of the 13th conference of international society of soil mechanics and foundation engineering*, pp. 1445–1448, New Delhi

Poulos, H.G. & Davis, E.H. (1999). *Pile foundation analysis and design,* Wiley, New York

Provenzano, P.; Ferlisi, S. & Musso, A. (2004). Interpretation of a model footing response through an adaptive neural fuzzy inference system, *Computers and Geotechnics*, Vol.31, pp. 251-266

Rahman, M. S.; Wang, J.; Deng, W. & Carter, J. P. (2001). A Neural Network Model for the Uplift Capacity of Suction Caissions, *Computers and Geotechnics*, Vol.39, pp. 337-356.

Reddy, N. P.; Prabhu, D.; Palreddy, S.; Gupta, V.; Suryanarayanan, S., & Canilang, E.P. (1995), Redundant Neural Networks for Medical Diagnosis Diagnosis of Dysphagia, *Intelligent Systems through Artificial Neural Networks*, Vol.5, pp. 699-704.

Reddy, N.P. & Buch, O. (2003). Committee Neural Networks for Speaker Verification, *Computer Methods and Programs in Biomedicine*, Vol.72, pp. 109-115.

Romero, S. & Pamukcu, S. (1996). Characterization of granular meterial by low strain dynamic excitation and ANN, *Geotechnical Special Publication*, ASTM-ASCE, Vol.58, No.2, pp. 1134-1148.

Rumelhart, D.E.; Hinton, G. & Williams, R. (1986). Learning representation by back- 462 propagation errors. *Nature*, Vol.32, No.9, pp. 533–536.

Sakellariou, M.G. & Ferentinou, M.D. (2005). A study of slope stability prediction using neural networks, *Geotechnical and Geological Engineering* Vol.23, pp. 419–445

Shahin, M.A.; Jaksa, M.B. & Maier, H.R. (2005). Stochastic simulation of settlement prediction of shallow foundations based on a deterministic artificial neural network model, Proc. *Int. Congress on Modelling and Simulation*, MODSIM 2005, pp. 73-78, Melbourne (Australia)

Shi, J.J. (2000). Reducing prediciton error by transforming input data for neural networks, *Journal of Computing in Civil Engineering*, Vol.14, No.2, pp. 109-116.

Shin, H.S. & Pande, G.N. (2000). On self-learning finite element codes based on monitored response of structures, *Computers and Geotechnics,* Vol.27, pp. 161–178, ISSN 0266-352X

Sidarta, D.E. & Ghaboussi, J. (1998). Constitutive modeling of geomaterials from non-uniform material tests, *Computers and Geotechnics,* Vol.22, No.1, pp. 53–71, ISSN 0266-352X

Sivakugan, N.; Eckersley, J.D. & Li, H. (1998). Settlement predictions using neural networks, *Australian Civil Engineering Transactions*, CE40, pp. 49-52.

Swingler, K. (1996). *Applying Neural Networks: A Practical Guide.* San Francisco: Morgan Kaufmann Publishers.

Teh, C.I.; Wong, K.S.; Goh, A.T.C. & Jaritngam, S. (1997). Prediction of pile capacity using neural networks, *J Comput Civil Eng.*, ASCE, Vol.11, No.2, pp. 129–38

Ural, D.N. & Saka, H. (1998). Liquefaction assessment by neural networks, *Electronic Journal of Geotechnical Engineering*, http://geotech.civen.okstate.edu/ejge/ppr9803/index.html.

Wang, H.B.; Xu, W.Y. & Xu, R.C. (2005) Slope stability evaluation using Back Propagation Neural Networks, *Engineering Geology*, Vol.80, pp. 302– 315, ISSN 0013-7952

Yoo, C. & Kim, J.-M. (2007). Tunneling performance prediction using an integrated GIS and neural network, *Computers and Geotechnics*, Vol.34, pp. 19-30, ISSN 0266-352X

Yu, H. & Liang, W. (2001). Neural network and genetic algorithm based hybrid approach to expanded job-shop scheduling, *Computers and Industrial Engineering*, Vol. 39, pp. 337-356.

Zhao, H.-B. (2007). Slope reliability analysis using a support vector machine, *Computers and Geotechnics*, in press.

Zhu, J.H.; Zaman, M.M. & Anderson, A.A. (1998). Modeling of shearingbehavior of residual soil with recurrent neural network, *Int J Numer Anal Meth Geomech*, Vol.22, pp. 671–87

Artificial Neural Networks - Application

Edited by Dr. Chi Leung Patrick Hui

This book covers 27 articles in the applications of artificial neural networks (ANN) in various disciplines which includes business, chemical technology, computing, engineering, environmental science, science and nanotechnology. They modeled the ANN with verification in different areas. They demonstrated that the ANN is very useful model and the ANN could be applied in problem solving and machine learning. This book is suitable for all professionals and scientists in understanding how ANN is applied in various areas.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hyun Il Park (2011). Study for Application of Artificial Neural Networks in Geotechnical Problems, Artificial Neural Networks - Application, Dr. Chi Leung Patrick Hui (Ed.), ISBN: 978-953-307-188-6, InTech, Available from: http://www.intechopen.com/books/artificial-neural-networks-application/study-for-application-of-artificial-neural-networks-in-geotechnical-problems

INTECH
open science | open minds