Artificial intelligence for modeling load-settlement response of axially loaded (steel) driven piles

Application de l'intelligence artificielle à la modélisation de la courbe effort-tassement des pieux battus (en acier) soumis à un chargement axial

Shahin M.A.
Department of Civil Engineering, Curtin University, Perth WA, Australia

ABSTRACT: The design of pile foundations requires good estimation of the pile load-carrying capacity and settlement. Design for bearing capacity and design for settlement have been traditionally carried out separately. However, soil resistance and settlement are influenced by each other and the design of pile foundations should thus consider the bearing capacity and settlement in-separately. This requires the full load-settlement behavior of piles to be well predicted. However, it is well known that the actual load-settlement behavior of pile foundations can only be obtained by load tests carried out in-situ, which are expensive and time-consuming. In this paper, artificial intelligence (AI) using the recurrent neural networks (RNN) is used to develop a prediction model that can resemble the full load-settlement response of steel driven piles subjected to axial loading. The developed RNN model is calibrated and validated using several in-situ full-scale pile load tests, as well as cone penetration test (CPT) data. The results indicate that the RNN model has the ability to predict well the load-settlement response of axially loaded steel driven piles and can thus be used by geotechnical engineers for routine design practice.

RÉSUMÉ: Le dimensionnement des fondations sur pieux nécessite une estimation précise de la capacité portante et du tassement d'un pieu. Traditionnellement, la détermination de la capacité portante et du tassement d'un pieu est effectuée de manière séparée. Cependant, la résistance du sol et le tassement du pieu sont interdépendants. Ainsi, le dimensionnement des fondations sur pieux devrait considérer de manière simultanée la capacité portante et le tassement du pieu. Ceci nécessite une bonne prédiction de la courbe effort-tassement du pieu. Cependant, il est bien connu que la courbe effort-tassement du pieu ne peut être obtenue que par des essais de chargement du pieu in-situ, et qui sont coûteux et consommateurs en temps. Dans cet article, l'intelligence artificielle (IA) utilisant les réseaux de neurones récurrents (RNN) est utilisée pour développer un modèle de prédiction qui simule la courbe effort-tassement des pieux en acier soumis à un chargement axial à partir des essais in-situ. Le modèle RNN développé est calibré et validé en utilisant plusieurs résultats d'essais de chargement de pieux in-situ, ainsi que des résultats d'essais pénétrométriques (CPT). Les résultats obtenus indiquent que le modèle RNN a la capacité de prédire avec précision la courbe effort-tassement d'un pieu en acier chargé axialement et il peut ainsi être utilisé dans la pratique par les géotechniciens.

KEYWORDS: artificial intelligence, recurrent neural networks, pile foundations, load-settlement, modeling.

1 INTRODUCTION

Bearing capacity and settlement are the two main criteria that govern the design process of pile foundations so that safety and serviceability requirements are achieved. Design for bearing capacity is carried out by determining the allowable pile load, which is obtained by dividing the ultimate pile load by an assumed factor of safety. Design for settlement, on the other hand, consists of obtaining the amount of settlement that occurs when the allowable load is applied to the pile, causing the soil to consolidate or compress. Design for bearing capacity and design for settlement have been traditionally carried out separately. However, Fellenius (1988) stated that: "*The allowable load on the pile should be governed by a combined appraoch considering soil resistance and settlement inseparately acting together and each influencing the value of the other*". In addition, there is a strong argument regarding the definition of the ultimate pile load and many methods have been proposed in the litearture, some result in interpreted ultimate loads that greatly depend on judgement and the shape of the load-settlement curve (1980). Consequenlty, for design purposes, the full load-settlement response of piles needs to be well predicted and simulated; the designer can thus decide the ultimate load and comply with the srevieability requirement.

Good prediction of the full load-settlement response of pile foundations needs thorough understanding of the load transfer along the pile length, which is complex, indeterminate and difficult to quantify (Reese et al. 2006). The actual load-settlement response of pile foundations can only be obtained by carrying out load tests in-situ, which is expensive and time-consuming. On the other hand, the load-settlement response of pile foundations can be estimated using many methods available in the literature. However, due to many complexities, available methods, by necessity, simplify the problem by incorporating several assumptions associated with the factors that affect the pile behavior. Therefore, most existing methods failed to achieve consistent success in relation to the predictions of pile capacity and corresponding settlement. In this respect, the artificial intelligence (AI) can be efficient as they can resemble the in-situ full-scale pile load tests without the need for any assumptions or simplifications. AI is a data mining statistical technique that has proved its potential in many applications in geotechncial engineering (see Shahin et al. 2009).

In this paper, the feasibility of using one of the most commonly used AI techniques, i.e. recurrent neural networks (RNN), is used for modeling the load-settlement response of steel driven piles subjected to axial loading. To facilitate the use of the developed RNN model for routine design by practitioners, the model is translated into an executable program that is made available for interested readers upon request.

2. OVERVIEW OF RECURRENT NEURAL NETWORKS

The type of neural networks used in this study are multilayer perceptrons (MLPs) that are trained with the back-propagation algorithm (Rumelhart et al. 1986). A comprehensive description of backpropagation MLPs is beyond the scope of this paper but can be found in Fausett (1994). The typical MLP consists of a number of processing elements or nodes that are arranged in layers: an input layer; an output layer; and one or more intermediate layers called hidden layers. Each processing element in a specific layer is linked to the processing element of the other layers via weighted connections. The input from each processing element in the previous layer is multiplied by an adjustable connection weight. The weighted inputs are summed at each processing element, and a threshold value (or bias) is either added or subtracted. The combined input is then passed through a nonlinear transfer function (e.g. sigmoidal or tanh function) to produce the output of the processing element. The output of one processing element provides the input to the processing elements in the next layer. The propagation of information in MLPs starts at the input layer, where the network is presented with a pattern of measured input data and the corresponding measured outputs. The outputs of the network are compared with the measured outputs, and an error is calculated. This error is used with a learning rule to adjust the connection weights to minimize the prediction error. The above procedure is repeated with presentation of new input and output data until some stopping criterion is met. Using the above procedure, the network can obtain a set of weights that produces input-output mapping with the smallest possible error. This process is called "training" or "learning", which once has been successful, the performance of the trained model has to be verified using an independent validation set.

In simulations of the typical non-linear response of pile load-settlement curves, the current state of load and settlement governs the next state of load and settlement; thus, a recurrent neural network (RNN) is recommended. A recurrent neural network proposed by Jordan (1986) implies an extension of the MLPs with current-state units, which are processing elements that remember past activity (i.e. memory units). The neural network then has two sets of input neurons: plan units and current-state units (Figure 1). At the beginning of the training process, the first pattern of input data is presented to the plan units while the current-state units are set to zero. As mentioned earlier, the training proceeds, and the first output pattern of the network is produced. This output is copied back to the current-state units for the next input pattern of data.

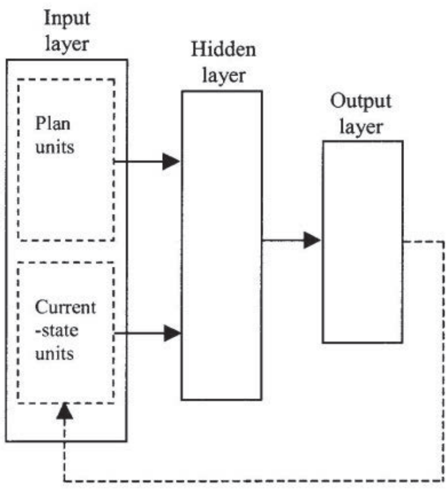

Figure 1. Schematic diagram of the recurrent neural network.

3. DEVELOPMENT OF NEURAL NETWORK MODEL

In this work, the RNN model was developed with the computer-based software package Neuroshell 2, release 4.2 (Ward 2007). The data used to calibrate and validate the model were obtained from the literature and included a series of 23 in-situ full-scale load-settlement tests reported by Eslami (1996). The tests were conducted on sites of different soil types and geotechnical conditions, ranging from cohesive clays to cohesionless sands. The pile load tests include compression and tension loading conducted on steel driven piles of different shapes (i.e., circular with closed toe and H-pile with open toe). The piles ranged in diameter between 273 and 660 mm with embedment lengths between 9.2 and 34.3 m.

3.1 *Model inputs and outputs*

Six factors affecting the capacity of driven piles were presented to the plan units of the RNN as potential model input variables (Figure 2). These include the pile diameter, D (the equivalent diameter is rather used in case of H-pile as: pile perimeter/π), embedment length, L, weighted average cone point resistance over pile tip failure zone, $\bar{q}_{c-tip}$, weighted average sleeve friction over pile tip failure zone, $\bar{f}_{s-tip}$, weighted average cone point resistance over pile embedment length, $\bar{q}_{c-shaft}$, and weighted average sleeve friction over pile embedment length, $\bar{f}_{s-shaft}$. The current state units of the neural network were represented by three input variables: the axial strain, $\varepsilon_{a,i}$, (= pile settlement/pile diameter), the axial strain increment, $\Delta\varepsilon_{a,i}$, and pile load, Q_i. The single model output variable is the pile load at the next state of loading, Q_{i+1}.

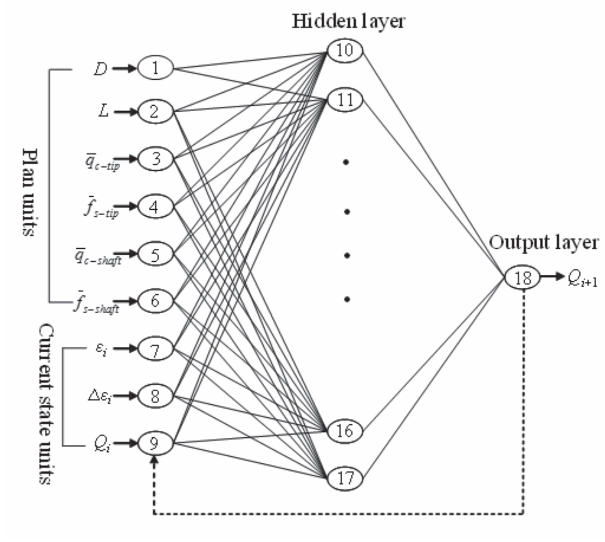

Figure 2. Architecture of the developed recurrent neural network.

In this study, an axial strain increment that increases by 0.05% was used, in which $\Delta\varepsilon_a$ = (0.1, 0.15, 0.2, …, 1.0, 1.05, 1.1, …) were utilized. As recommended by Penumadu and Zhao (1999), using varying strain increment values results in good modeling capability without the need for a large size training data. Because the data points needed for the RNN model development were not recorded at the above strain increments in the original pile load-settlement tests, the load-settlement curves were digitized to obtain the required data points. This was carried out using Microcal Origin version 6.0 (Microcal 1999) and then implementing the cubic spline interpolation (Press et al. 1992). A range between 14 to 28 training patterns was used in representing a single pile load-settlement test, depending on the maximum strain values available for each test.

It should be noted that the following aspects were applied to the input and output variables used in the RNN model:

- The pile tip failure zone over which $\bar{q}_{c-tip}$ and $\bar{f}_{s-tip}$ were calculated is taken in accordance with Eslami (1996), in which the influence zone extends to 4 D below and 8 D above pile toe when the pile toe is located in

- nonhomogeneous soil of dense strata with a weak layer above. Also, in non-homogeneous soil, when the pile toe is located in weak strata with a dense layer above, the influence zone extends to 4 *D* below and 2 *D* above pile toe. In homogeneous soil, however, the influence zone extends to 4 *D* below and 4 *D* above pile toe.
- Both measurements of cone point resistance and sleeve friction are incorporated as model inputs. This allows the soil type (classification) to be implicitly considered in the RNN model.
- Several CPT tests used in this work include mechanical rather than electric CPT data and thus, it was necessary to convert the mechanical CPT readings into equivalent electric CPT values as the electric CPT is the one that is commonly used at present. This is carried out for the cone point resistance using the following correlation proposed by Kulhawy and Mayne (1990):

$$\left(\frac{q_c}{p_a}\right)_{Electric} = 0.47\left(\frac{q_c}{p_a}\right)^{1.19}_{Mechanical} \quad (1)$$

- For the cone sleeve friction, the mechanical cone gives higher reading than the electric cone in all soils with a ratio in sands of about 2, and 2.5–3.5 for clays (Kulhawy and Mayne 1990). In the current work, a ratio of 2 is used for sands and 3 for clays.

3.2 *Data division and preprocessing*

The next step in the development of the RNN model is dividing the available data into their subsets. In this work, the data were randomly divided into two sets: a training set for model calibration and an independent validation set for model verification. In total, 20 in-situ pile load tests were used for model training and 3 tests for model validation. A summary of the tests used in the training and validation sets is not given due to the lack of space. Once the available data are divided into their subsets, the input and output variables are preprocessed; in this step the variables were scaled between 0.0 and 1.0 to eliminate their dimensions and to ensure that all variables receive equal attention during training.

3.3 *Network architecture and internal parameters*

Following the data division and the preprocessing, the optimum model architecture (i.e., the number of hidden layers and the corresponding number of hidden nodes) must be determined. It should be noted that a network with one hidden layer can approximate any continuous function if sufficient connection weights are used (Hornik et al. 1989). Therefore, one hidden layer was used in the current study. The optimal number of hidden nodes was obtained by a trial-and-error approach in which the network was trained with a set of random initial weights and a fixed learning rate of 0.1; a momentum term of 0.1; a tanh transfer function in the hidden layer nodes; and a sigmoidal transfer function in the output layer nodes. The following number of hidden layer nodes were then utilized: 2, 4, 6, …, and (2*I*+1), where *I* is the number of input variables. It should be noted that (2*I*+1) is the upper limit for the number of hidden layer nodes needed to map any continuous function for a network with *I* inputs, as discussed by Caudill (1988). To obtain the optimum number of hidden layer nodes, it is important to strike a balance between having sufficient free parameters (connection weights) to enable representation of the function to be approximated and not having too many, so as to avoid overtraining (Shahin and Indraratna 2006).

To determine the criterion that should be used to terminate the training process, the normalized mean squared error between the actual and predicted values of all outputs over all patterns is monitored until no significant improvement in the error occurs. This was achieved at approximately 10,000 training cycles (epochs). Figure 3 shows the impact of the number of hidden layer nodes on the performance of the RNN model. It can be seen that the RNN model improves with increasing numbers of hidden layer nodes; however, there is little additional impact on the predictive ability of the model beyond 8 hidden layer nodes. Figure 3 also shows that the network with 19 hidden layer nodes has the lowest prediction error; however, the network with 8 hidden nodes can be considered optimal: its prediction error is not far from that of the network with 19 hidden nodes, and it has fewer connection weights and is thus less complex. As a result of training, the optimal network produced 9 × 8 weights and 8 bias values connecting the input layer to the hidden layer and 8 × 8 weights and one bias value connecting the hidden layer to the output layer.

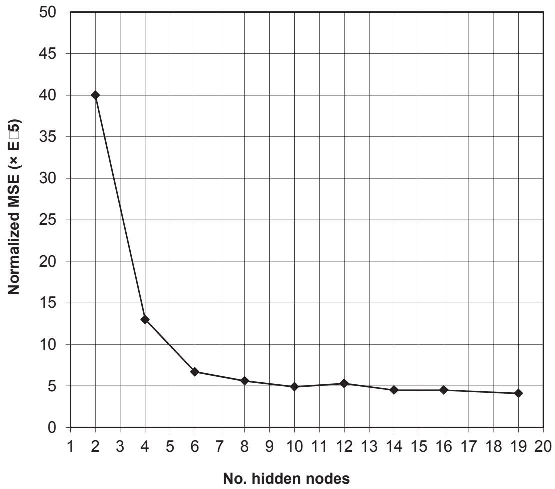

Figure 3. Effect of number of hidden nodes on RNN performance.

3.4 *Model performance and validation*

The performance of the optimum RNN model in the training and validations sets is given numerically in Table 1. It can be seen that three different standard performance measures are used, including the coefficient of correlation, *r*, the coefficient of determination (or efficiency), R^2, and the mean absolute error, MAE. The formulas of these three measures are as follows:

$$r = \frac{\sum_{i=1}^{N}(O_i - \overline{O})(P_i - \overline{P})}{\sqrt{\sum_{i=1}^{N}(O_i - \overline{O})^2 \sum_{i=1}^{N}(P_i - \overline{P})^2}} \quad (2)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(O_i - P_i)^2}{\sum_{i=1}^{N}(O_i - \overline{O})^2} \quad (3)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|O_i - P_i\right| \quad (4)$$

where N is the number of data points presented to the model; O_i and P_i are the observed and predicted outputs, respectively; and $\overline{O}$ and $\overline{P}$ are the mean of the predicted and observed outputs, respectively.

The coefficient of correlation, *r*, is a measure that is used to determine the relative correlation between the predicted and observed outputs. However, *r* sometimes may not necessarily indicate better model performance due to the tendency of the model to deviate toward higher or lower values, particularly when the data range is very wide and most of the data are distributed about their mean (Das and Sivakugan 2010). Consequently, the coefficient of determination, R^2, is used as it can give unbiased estimate and may be a better measure for model performance. The MAE eliminates the emphasis given to large errors, and is a desirable measure when the data evaluated are smooth or continuous. The performance measures in Table 1 indicate that the optimum RNN model performs well and has good prediction accuracy in both the training and validation sets. Table 1 also indicates that the RNN model has consistent performance on the validation set with that obtained on the training set.

Table 1. Performance results of the optimal RNN model.

Data sets	Performance measures		
	r	R^2	*MAE* (kN)
Training	0.998	0.996	34
Validation	0.994	0.988	38

The performance of the optimum RNN model in the training and testing sets is further investigated graphically, as shown in Figures 4 and 5. It should be noted that, for brevity, only five of the most appropriate simulation results in the training set are given in Figure 4. These five simulations are chosen because they reflect the entire range of the in-situ pile load-settlement tests used in this study. As can be seen in Figures 4 and 5, excellent agreement between the actual pile load tests and the RNN model predictions is obtained, in both the training and validation sets. The nonlinear relationships of the load-settlement response are well predicted, and the results demonstrate that the RNN model has a strong capability to simulate the behavior of steel driven piles.

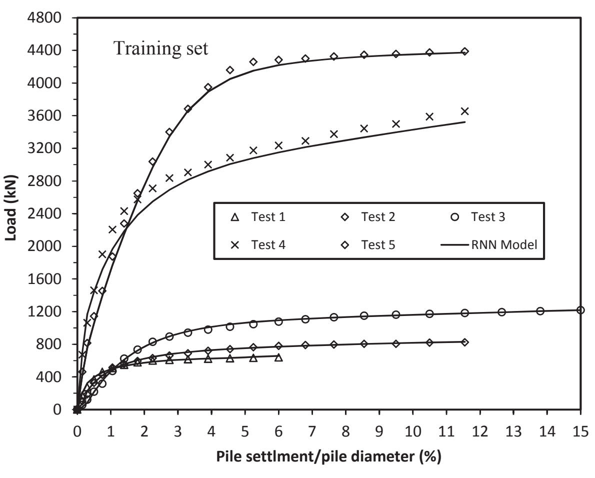

Figure 4. Some simulation results of RNN model in the training set.

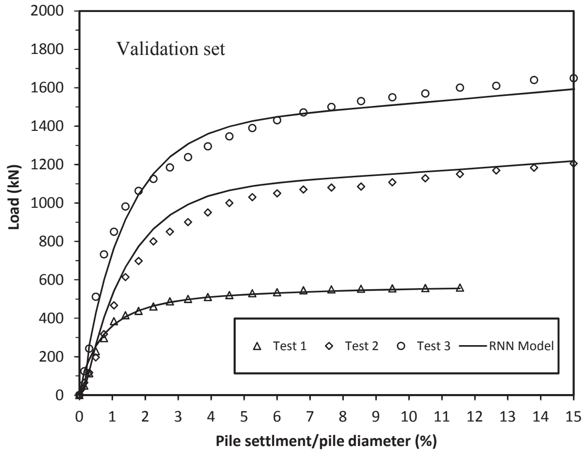

Figure 5. Simulation results of RNN model in the validation set.

4. CONCLUSION

This work presented in this paper has used a series of full-scale in-situ pile load-settlement tests and CPT data collected from the literature to develop a recurrent neural network (RNN)-based model for simulating the load-settlement response of steel driven piles. The results indicate that the RNN model was capable of simulating the behavior of steel driven piles reasonably well. The graphical comparison of the load-settlement curves between the RNN model and experiements showed an excellement agreement and indicates that the RNN model can capture the highly non-linear load-settlement response of steel driven piles. To facilitae the use of the developed RNN model, it is translated into C++ code and executable program, which are made available upon request.

5. REFERENCES

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

Das S.K. and Sivakugan N. 2010. Discussion of: intelligent computing for modeling axial capacity of pile foundations. *Canadian Geotechnical Journal* 47, 928-930.

Eslami A. 1996. Bearing capacity of piles from cone penetration test data. PhD Thesis, University of Ottawa, Ottawa, Ontario.

Fausett L.V. 1994. *Fundamentals neural networks: Architecture, algorithms, and applications*, Prentice-Hall, Englewood Cliffs, NJ.

Fellenius B.H. 1980. The analysis of results from routine pile load tests. *Ground Engineering* 13 (6), 19-31.

Fellenius B.H. 1988. Unified design of piles and pile groups. *Transportation Research Record* 1169, 75-81.

Hornik K., Stinchcombe M. and White H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359-366.

Jordan M.I. 1986. Attractor dynamics and parallesim in a connectionist sequential machine. *Proceedings of the 8th Annual Conference of the Cognitive science Society*, Amherst, MA, 531-546.

Reese L.C., Isenhower W.M. and Wang S.T. 2006. *Analysis and design of shallow and deep foundations*, John Wiley & Sons, New Jersey.

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

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

Shahin M.A., Jaksa M.B. and Maier H.R. 2009. Recent advances and future challenges for artificial neural systems in geotechncial engineering applications. *Journal of Advances in Artificial Neural Systems*, doi: 10.1155/2009/308239.

Ward. 2007. *NeuroShell 2 Relaese 4.2*, Ward Systems Group, Mass.