Comparison Between PLAXIS Output and Neural Network in the Guard Walls

Ali Mahbod[1], Abdolghafar Ghorbani Pour[2], Abdollah Tabaroei[3], Sina Mokhtar[2]

1- Department of Civil Engineering, Shahid Bahonar University, Kerman, Iran(Corresponding Author)
2- Department of Civil Engineering, Arak Branch ,Islamic Azad University , Arak ,Iran
3- Department of Civil, Science and Research Branch, Islamic Azad University, Tehran, Iran

(Presenter: Ali Mahbod)

Abstract

The purpose of this study to estimate the parameters of the soil and anchor by means of artificial neural network to determine its success in moving the walls of the guard. For this purpose, the artificial neural network application software Matlab was used. Six parameters have been chosen to anchor the soil and then examined the function of neural network for movement of the guard wall. Our finding suggest that With increasing soil mechanical parameters such as modulus of elasticity of soil and soil internal friction angle of the wall decreases the maximum amount of transformation and With increasing amounts of inhibitory Non tangly and transformation up to the wall decreases. using the artificial neural network in this article is cited can be made saving time and cust.

Key words: Artificial Neural Network, Excavation, PLAXIS 2D, Horizontal Displacement

1. Introduction

In many demanding engineering applications computationally inexpensive predictions, based on Met models can be used, rather than solving a set of mathematical equations analytically, or even numerically. Most methods are inspired by natural paradigms and therefore differ significantly from conventional mathematical approaches. Artificial neural networks (ANNs), fuzzy systems as well as evolutionary methods are the most popular. In general, artificial intelligence (AI) methods are used either in order to reduce the computational cost, or when the complexity and/or the size of the problem prohibits the use of conventional techniques (Lagaros,2006). Especially, ANNs have been widely-used in many fields of science and technology, as well as into an increasing number of various engineering applications (Wu,2006).

2. Methodology

A learning algorithm in order to achieve the correct response for each input vector that is given to the neural network. Previous estimates of the training set and a full investigation to obtain more information and a better approximation of the time the forecasts were reduced**.**

3. Ground Anchors Parameters

Pressure-injected ground anchors were used to support the wall. The ground anchor angle was selected to be 30˚ from the horizontal so the ground anchors would apply a significant downward load on the soldier beams. The anchors were installed by driving a closed-end, 8.89 cm casing into the ground. After the casing reached the desired depth, then the ground anchor tendon was inserted in the casing and the closure point driven off. Cement grout was pumped down the casing as the casing was extracted. The top row of anchors had a 5.48m unbounded length and the bottom row of anchors had a 4.57m unbounded length. A plastic tube was used as a bond breaker over the unbounded length. In 2d modeling the grout body(second part of anchor) modeled by Geogrid element, the unbonded lengh (first part of anchor) modeled by node to node anchor. Staged constructions was in 8 phases . The numerical modeling in this article has been done by PLAXIS 2D software and the selecting conditions is plane strain.

4. Artificial Neural Networks

Artificial neural networks (ANNs) are perhaps the most popular intelligent computational paradigms. An ANN consists of a number of units linked together and attempts to create a desired mapping between the input and the output data of a specific set. In order to achieve this goal a training set (D) is composed by input-target pairs D = [xm,tm], where m is the number of the pairs, x the input data and t the target pairs. A neural network architecture A consists of a specific number of layers, a number of neurons in each layer, and a suitable activation function. The input layer projects the data to the intermediate layer(s). Each intermediate or hidden layer passes the data to the next intermediate layer, while the final hidden layer projects the information to the output neurons. If a set of values w, corresponding to the weight factors, is assigned to the network then a mapping y(x;w; A) is defined between the inputs x and the outputs y. The quality of this mapping, with respect to the training set, is measured by an error function ED, defined as follows (Eq.1):

$$E_D\left(D \mid w, A\right) = \sum_m \frac{1}{2}\left(y(x^m; w, A) - t^m\right)^2 \tag{1}$$

A learning algorithm tries to determine the optimum values of w that minimize the value of ED in order to achieve the correct response for each input vector that is given to the neural network. The numerical minimization algorithms used for the ANN training generate a sequence of weight parameters w through an iterative procedure. To apply an algorithmic operator the starting weight parameters w are needed, which are subsequently updated as follows (Eq.2):

$$W^{new} = W^{old} - \alpha S^{m} (a^{m-1})^{T} \qquad (2)$$

Here is the learning rate α, S^{m} the sensitivity of the network layer m, $(a^{m-1})^{T}$ is the transposed matrix of output layer before.

$$S^{M} = \dot{F}^{M}(n^{M})(t-a) \qquad (3)$$

$$S^{m} = \dot{f}^{m}(n^{m})(W^{m+1})^{T} S^{m+1} \qquad m = M-1, \ldots, 2, 1 \qquad (4)$$

Where $\dot{F}^{M}(n^{M})$ derived from the conversion function, $(W^{m+1})^{T}$ transpose of weight matrix in the previous layer. (Eq.3)

Learning algorithms can be classified into local and global algorithms (MacKay,1992). Global algorithms use knowledge of the current state of the entire network, such as the direction of the overall weight update vector. For instance, in the widely-used back-propagation learning algorithm the gradient descent algorithm is used. In contrast, local adaptation strategies are based on specific information of the weight values, such as the temporal behavior of the partial derivative of the weights. The local approach is better related to the natural neural networks concept of distributed processing, where the computations are performed independently. Moreover, it appears that for many applications local strategies achieve faster and more reliable predictions than global techniques (Riedmiller,1994)

ANN model with the data model 162 PLAXIS trained. Model input data for ANN (E_1, E_2, φ_1, φ_2, L_1, L_2), output guards are removable wall. ANN to predict the movement used is composed of three layers:

a: input layer with six node (E_1, E_2, φ_1, φ_2, L_1, L_2). b: the hidden layer, and c: the output layer with one node (displacement) After an initial investigation with respect to the number of the hidden layer's nodes, the ANN configuration resulted in a [6-9-1] architecture (Figure 1). In addition, it was observed that increasing the number of hidden layers did not alter significantly the performance of the ANN, thus the runs were performed using one intermediate layer.

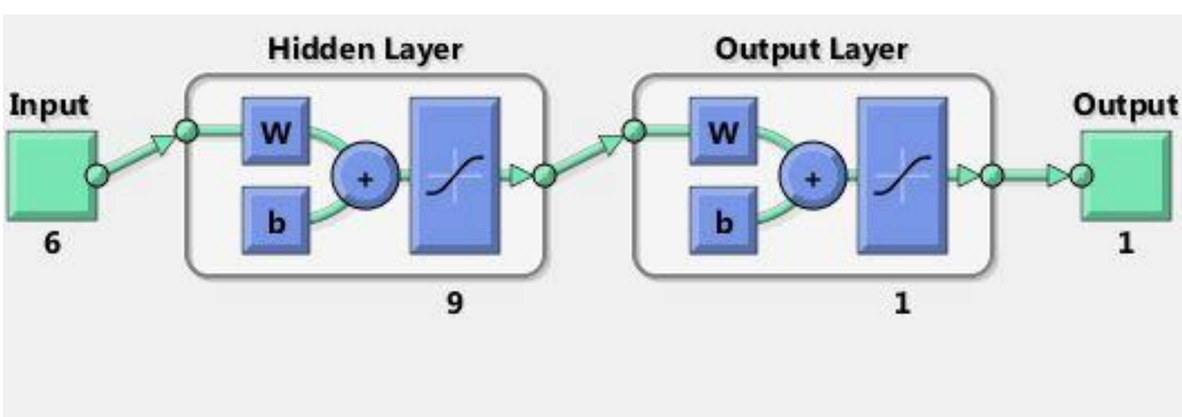

Fig 1: Hidden Layers

Previous estimates of the training set and a full investigation to obtain more information and a better approximation of the time the forecasts were reduced. input data and architecture of the ANN used in the training set is used in Table 1 and 2 have shown and the results of the performance of ANN for four model inputs varied showed that From these results it can be seen that the accuracy of the ANNs is not deteriorated significantly by the reduction of the training data .

Table1: Input Parameters

Model	The input parameters
1	$E_1,E_2,\varphi_1,\varphi_2,L_1,L_2$
2	$E_1,E_2,\varphi_1,\varphi_2,L_1$
3	$E_1,E_2,\varphi_1,\varphi_2$
4	E_1,E_2,φ_1

Table2: Architecture of ANN

ANN	Architecture
1	[6-9-1]
2	[5-9-1]
3	[4-9-1]
4	[3-9-1]

Network training error with respect to changes in input data, the number of hidden layers and number of neurons in the hidden layer is calculated. The observations of how the error decreases and the network test Found wrong in the past for a model that has the best answer, is as follows (Table 3):

Table3: The Observations of how the error decreases and the network test Found wrong in the past for a model that has the best answer

Model 1	Parameters
6	The number of input
1	The number of hidden layers
9	The number of neurons in the hidden layer
inf	Time
0.05	Learning rate

1000	Repeat

Network is trained to evaluate the data series of 18 shrimp were not used in the training phase. The following Table 4 Comparison between the output of network and application data is done.

Table4: Comparison Between the Output of Network and Application Data

Model	$E_{\text{first layer}}$ (KN/m^2)	$E_{\text{second layer}}$ (KN/m^2)	$\varphi_{\text{first layer}}$	$\varphi_{\text{second layer}}$	**L_{UNB} of first anchor** (m)	**L_{UNB} of second anchor** (m)	**Plaxis results** (m)	**Ann results** (m)	**Error** (%)
1	23450	10000	29	29	2	1.5	0.04129	0.04296	2.69
2	23450	10000	32	29	2	1.5	0.04052	0.04039	0.31
3	23450	10000	35	29	8	7.5	0.02279	0.02268	0.48
4	23450	15000	39	32	2	1.5	0.0308	0.03087	0.24
5	23450	15000	35	32	2	1.5	0.02739	0.02703	1.29
6	23450	20000	29	35	2	1.5	0.02055	0.02052	0.11
7	23450	20000	32	35	8	7.5	0.00762	0.00743	2.38
8	35000	10000	32	29	8	7.5	0.02462	0.02644	0.12
9	35000	15000	29	32	8	7.5	0.01591	0.01593	0.14
10	35000	15000	32	32	2	1.5	0.02843	0.02838	0.16
11	35000	15000	35	29	5	4.5	0.02335	0.02307	0.36
12	35000	20000	32	29	5	4.5	0.02063	0.02031	1.5
13	35000	20000	35	32	8	7.5	0.01158	0.01148	0.82
14	35000	20000	29	29	2	1.5	0.03238	0.03250	1.76
15	23450	20000	29	29	2	1.5	0.03318	0.0336	1.19
16	23450	20000	32	32	2	1.5	0.02692	0.0266	1.56
17	35000	10000	35	32	2	1.5	0.0311	0.03108	0.1
18	23450	20000	35	35	8	7.5	0.00699	0.00715	2.3

It has seen from the table that has a high success rate in the network and 2.69% error rate predicted to move the wall guards. In order to more closely define the output PLAXIS and ANN , the above table can be found in the following chart (Fig 2):

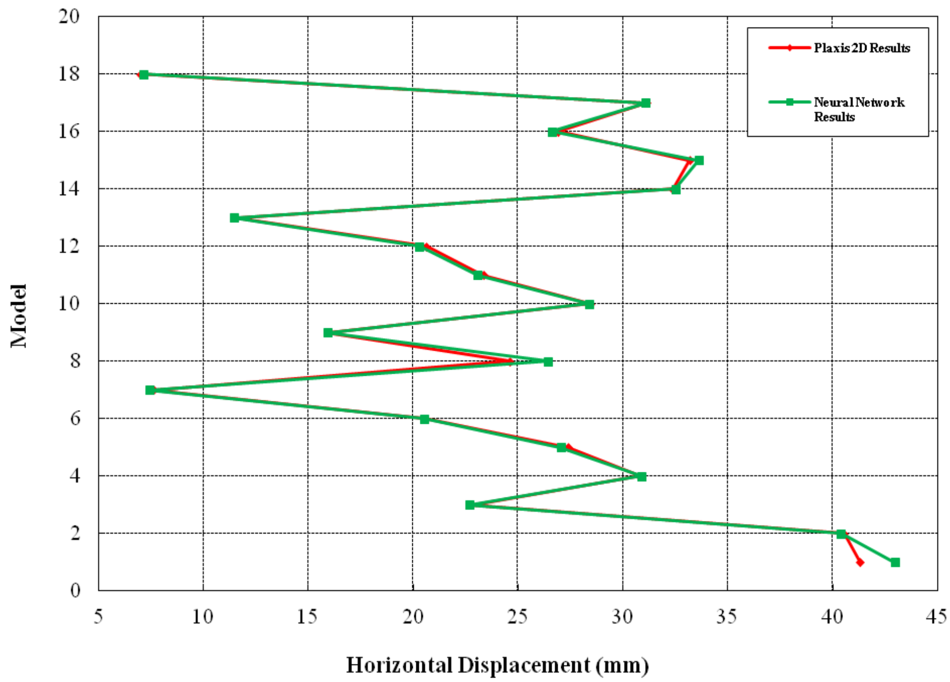

Fig 2: Comparison between PLAXIS output and Neural Network

5. Conclusion

Purpose of The current study was to estimate the parameters of the soil and anchor by means of artificial neural network to determine its success in moving the walls of the guard. Our finding suggest that With increasing soil mechanical parameters such as modulus of elasticity of soil and soil internal friction angle of the wall decreases the maximum amount of transformation and With increasing amounts of inhibitory non tangly and transformation up to the wall decreases. Reducing the inhibitory non tangly and transformation values of maximum wall increases. During the transformation non tangly on the maximum wall length is more tangly. Stress reduction in force, with a maximum inhibitory amounts of deformation increases. using the artificial neural network in this article is cited can be made saving time and cust.

Further work needs to be done to establish whether other software , such results are confirmed and also to apply these results in real cases from the field compared with the reports mentioned in this article.

Refrences:

1. Lagaros ND, Papadrakakis M. (2004) "*Improving the condition of the Jacobin in neural network training*", Adv Eng Softy, 9–25.

2. Lagaros ND, Tsompanakis Y, editors. (2006) “*Intelligent computational paradigms in earthquake engineering*”. Idea Publishers,103-106.

3. MacKay DJC.(1992) "*A practical Bayesian framework for back prop networks*", Neural Compute, 72–448.

4. Papadrakakis M, Lagaros ND, Tsompanakis Y. (1998) "*Structural optimization using evolution strategies and neural networks*", Compute Methods Apple Mach Eng, 33–309.
5. Riedmiller M. (1994) "*Advanced supervised learning in multi-layer perceptions – from back propagation to adaptive learning algorithms*", International Journal of Computer Standards and Interfaces – Special Issue on Neural Networks, 78–265.
6. Riedmiller M, Braun H.(1993) "*A direct adaptive method for faster back-propagation learning: the RPROP algorithm*", In: Proceedings of the IEEE international conference on neural networks (ICNN). San Francisco, 91–586.
7. Wu CL, Chau KW. (2006) "*A flood forecasting neural network model with genetic algorithm*", Int J Environ Pollute, 73–261.