Computer Assisted Mechanics and Engineering Sciences, **18**: 303–311, 2011.

Application of artificial neural network in soil parameter identification for deep excavation numerical model

Marek Wojciechowski
Chair of Geotechnics and Engineering Structures, Technical University of Łódź
Al. Politechniki 6, 90-924 Łódź, Poland
e-mail: mwojc@p.lodz.pl

In this paper, an artificial neural network (ANN) is used to approximate response of deep excavation numerical model on input parameters. The approximated model is then used in minimization procedure of the inverse problem, i.e. minimization of the differences between the response of the model (now, neural network) and the field measurements. ANN based objective function is continuous and differentiable thus gradient based optimization algorithm can be efficiently used in this problem. It is showed that initial approximation of the numerical model by means of ANN increase efficiency of the identification process without loss of accuracy.

Keywords: artificial neural network, parameter identification, deep excavation.

1. Introduction

Results provided by numerical models of deep excavations (and other geotechnical structures) often differ from field measurements and case histories. This is due to many uncertainties regarding the true soil properties at the stage of structure design. On the other hand the field measurements performed at the construction stages state a potential source of knowledge about the soil material response. These data could be used to improve the numerical models via inverse analysis techniques. This should, in turn, result in better prediction of the behavior of the structure at further construction stages.

When applying inverse analysis techniques to study the behavior of an actual supported excavation, concerns exist about the proper representation of the real system, as well as the efficiency of the inverse analysis technique and its ability to find a unique set of parameters for a particular problem.

Artificial neural networks (ANN) can be used in a number of ways in this context. The most common approach applied to the geotechnical problems (and especially deep excavations) is the so called self-learning finite element framework introduced by Ghaboussi [2]. In this method a neural network learns its own constitutive law form field measurements. Process of training ANN is then a process of creating an equivalent material model which tries to fit available field data. Weights of the network are then considered as material parameters. Another approach is to train an ANN in such a way, that it reflects an *inverse relationship* between numerical model response and its parameters (e.g. [6]). Application of the field measurements at the input of such a trained network should result in correct parameters of the numerical model at its output. The third way is to approximate, by means of ANN, a *direct relationship* between the numerical model answer and its parameters. After that, the trained ANN is used in the minimization procedure of the inverse problem, i.e. minimization of the error between the response of the model (now, artificial neural network) and the field measurements [8].

The last method is used in this paper for identification of the soil parameters for deep excavation systems. Feed-forward artificial neural network with continuous and differentiable transfer functions is used. Thus the neural network based objective function of the inverse problem is also continuous and differentiable. Once we train a neural network, the process of finding material parameters is very efficient, because of the use of the truncated Newton minimization algorithm [7].

A similar identification problem, but without exploiting artificial neural networks, can be found in [9]. In this paper the minimized objective is not differentiable and the numerical model of the excavation have to be evaluated directly when solving inverse problem. Thus the finite difference method must be used to calculate gradients or a non-gradient minimization method have to be applied (like genetic algorithm). Our approach is superior to the methods proposed in [9] because of higher efficiency (we need smaller number of model evaluations and we are able to identify higher number of parameters) and satisfactory accuracy.

In the following we define the identification procedure and the feed-forward neural network (along with its derivatives) used in this work. Then we present numerical model of deep excavation system and finally we present the results of identification with discussion and conclusions.

2. Identification procedure

Parameter identification of the numerical model $\mathcal{M}(\mathbf{p})$ consist in finding a set of parameters $\mathbf{p}$ in such a way, that the answer of the model at the chosen points of the solution domain, i.e.: $\mathbf{u}_{\mathcal{M}} = \mathcal{M}(\mathbf{p})$ fits best the field measurements $\mathbf{u}_{\mathcal{R}}$. This can be formulated as the minimization of the following objective function:

$$\Omega(\mathbf{p}) = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{u}_{\mathcal{M},i} - \mathbf{u}_{\mathcal{R},i})^2, \tag{1}$$

where N is the number of the monitored points. In case of using ANN the answer of model $\mathcal{M}(\mathbf{p})$ is replaced by the answer of appropriately trained ANN model $\mathcal{N}$, i.e. $\mathbf{u}_{\mathcal{M}} \approx \mathbf{u}_{\mathcal{N}} = \mathcal{N}(\mathbf{p})$. The objective function to be minimized can be then written as:

$$\Omega^{\star}(\mathbf{p}) = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{u}_{\mathcal{N},i} - \mathbf{u}_{\mathcal{R},i})^2. \tag{2}$$

Model $\mathcal{N}$ can be realized by a single network with N outputs or by N networks with a single output. The second possibility is used in this paper.

The identification procedure taking advantage of ANN approximation of the numerical model consist in three stages:

1. Generation of training data for neural network – this is done by a number of runs of model $\mathcal{M}$ with different sets of parameters $\mathbf{p}$.
2. Training the ANN model $\mathcal{N}(\mathbf{p})$.
3. Minimization of the function $\Omega^{*}(\mathbf{p})$.

The second and the third stage do not need additional evaluations of numerical model $\mathcal{M}$ (approximated model $\mathcal{N}$, ie. trained ANN, is used instead).

3. Artificial neural network with derivatives

3.1. Formulation

Two layer artificial neural network (Fig. 1) with nonlinear activation functions in hidden nodes, identity inputs and single identity output can be represented by the following closed formula:

$$\mathcal{N}(p_1,\ldots,p_n) = \sum_{j=1}^{m} v_j g_j \left(\sum_{i=1}^{n} w_{ji} p_i + b_j\right), \tag{3}$$

where n is a number of network inputs, m is a number of hidden neurons, v_j, w_{ij}, b_j are parameters of the network (weights and biases adjusted during training) and g_j is an activation function at hidden node j. Such a network is proved to be able to approximate any continuous multivariate function and, simultaneously, its partial derivatives, if only g_j is continuous, bounded and differentiable [3]. These criteria are fulfilled by sigmoid function used in this work:

$$g(x) = \frac{1}{1+e^{-x}}. \tag{4}$$

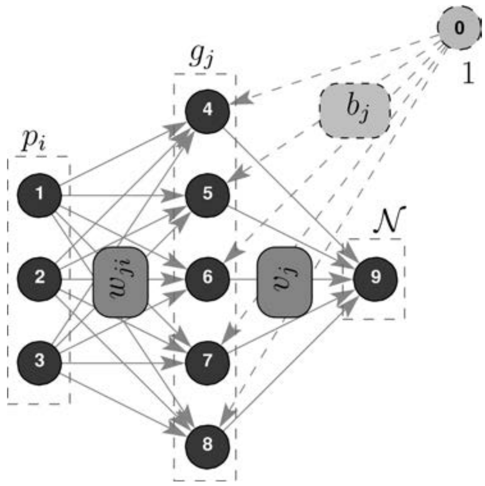

Fig. 1. Two layer artificial neural network.

It is straightforward to show (e.g. [5, 11, 12]) that k-th partial derivative of the function realized by the network $\mathcal{N}$ can be computed by means of the following formula:

$$\frac{\partial^{k_1}}{\partial p_1^{k_1}} \frac{\partial^{k_2}}{\partial p_2^{k_2}} \cdots \frac{\partial^{k_n}}{\partial p_n^{k_n}} \mathcal{N} = \sum_{j=1}^{m} \left(v_j g_j^{(k)} P_j \right), \tag{5}$$

where $g_j^{(k)}$ is k-th derivative of g_j and:

$$k = \sum_{i=1}^{n} k_i, \qquad P_j = \prod_{l=1}^{n} w_{jl}^{k_l}. \tag{6}$$

Once the ANN is trained, it should approximate a relationship contained in training data and also the derivatives of this relationship. Equation (6) shows that we have access to these derivatives, thus minimization of $\Omega^*(\mathbf{p})$ with gradient methods is possible.

3.2. Training

For ANN training we need a database of input – output pairs which should represent sufficiently a problem domain. For databases with small number of samples (in this paper we deal with such a situation, because training samples are obtained by numerically expensive evaluations of deep excavation model) special techniques are used to increase the quality of training results. The main goal is to avoid over-fitting while keeping good approximation of the model. One of these techniques is an addition of Tikhonov regularization term to the ANN cost function used in training. Cost function exploited in this paper is given by the formula:

$$E^*(\mathbf{p},\mathbf{w}) = \sum_{i=1}^{s} \left(\mathcal{N}(p_i,\mathbf{w}) - \mathcal{M}(p_i)\right)^2 + \lambda \sum_{j=1}^{n}\sum_{i=1}^{s} \left(\frac{\partial^2 \mathcal{N}(p_i,\mathbf{w})}{\partial p_j^2}\right)^2, \tag{7}$$

where s is a number of training samples and $\mathbf{w}$ is vector of all parameters (v_j, b_j, w_{ji}) of the network $\mathcal{N}$. The second term of E^* is a sum of second derivatives of ANN approximation, which can be calculated with (5). Shape of E^* means, that curvature of the ANN mapping is minimized. This smoothing behavior is controlled by scaling coefficient λ (which should be kept small).

In order to exploit Newton methods in ANN training gradient of the mapping $\mathcal{N}$ and its (second) derivatives with respect to weights and biases is needed. Let's represent the formula for network derivatives (5) in the following way:

$$\partial\mathcal{N}(v_j, b_j, w_{ji}) = \sum_{j=1}^{m} \left(v_j g_j^{(k)} P_j\right). \tag{8}$$

Then, gradient of $\partial\mathcal{N}$ with respect to output weights v_j, biases b_j and input weights w_{ij} is given by formulas:

$$\frac{\partial(\partial\mathcal{N})}{\partial v_j} = g_j^{(k)} P_j, \tag{9}$$

$$\frac{\partial(\partial\mathcal{N})}{\partial b_j} = v_j g_j^{(k+1)} P_j, \tag{10}$$

$$\frac{\partial(\partial\mathcal{N})}{\partial w_{ji}} = x_i v_j g_j^{(k+1)} P_j + v_j k_i w_{ji}^{k_i-1} g_j^{(k)} Q_j, \tag{11}$$

where

$$Q_j = \prod_{l=1, l\neq i}^{n} w_{jl}^{k_l}. \tag{12}$$

If differentiation order $k = 0$ then we have $\partial\mathcal{N} \equiv \mathcal{N}$ and the above equations represent the usual gradient of the network output with respect to network parameters. Thus the presented formulas are considered as a generalization of the standard two layer feed-forward neural network definition. The above formulas were used to calculate gradient of the introduced cost function, i.e. $\dfrac{\partial E^*(\mathbf{p},\mathbf{w})}{\partial \mathbf{w}}$, needed by truncated Newton optimization procedure [7] implemented in `ffnet` [13] software used for training.

4. Numerical model of deep excavation

Deep excavations support systems and its modeling by means of finite element method are wildely discussed in the literature (see. e.g. [4, 9, 10]). It have to be noted that the geometry of the excavation, its overall behavior, the material descriptions used in modeling can be quite complex. However, in this paper, geometry of the excavation model was kept relatively simple as the main goal is presentation of the usage of ANN in the identification task. Deep excavation system was modeled in Plaxis [1] geotechnical software using elastic – perfectly plastic material model with Mohr-Coulomb failure criterion for soils and linear elastic materials for diaphragm wall and struts (Fig. 2). Plain strain finite element formulation was used with standard boundary conditions.

In plain strain all quantities are calculated for 1 meter of the excavation length, thus stiffness of struts was divided by their spacing which was assumed 2.5 m. Two stages of construction are considered: partial excavation up to 15 m, where training data for ANN is gathered, and full excavation depth 20 m, where identified parameters are verified. The most important soil parameters used in simulations, i.e. unit weight of saturated and unsaturated soil (γ_{sat}, γ_{unsat}), Poisson's ratio (ν), Young's modulus (E_1 or E_2), friction angle (ϕ_1 or ϕ_2), cohesion (c), dilatatancy angle (ψ) are shown in Table 1. These parameters are considered to be typical values for soft unconsolidated clay and medium dense sand. From this parameter set four quantities were chosen for identification, i.e.: Young moduli and friction angles for both, clay and sand soil layers. Cohesion of soft clay was skipped because of its relatively small value and also relatively small impact on the diaphragm wall behavior. Thus the numerical model $\mathcal{M}$ of excavation has been attributed with the parameter list $\mathbf{p} = [E_1, \phi_1, E_2, \phi_2]$. Other parameters were kept constant during performed simulations.

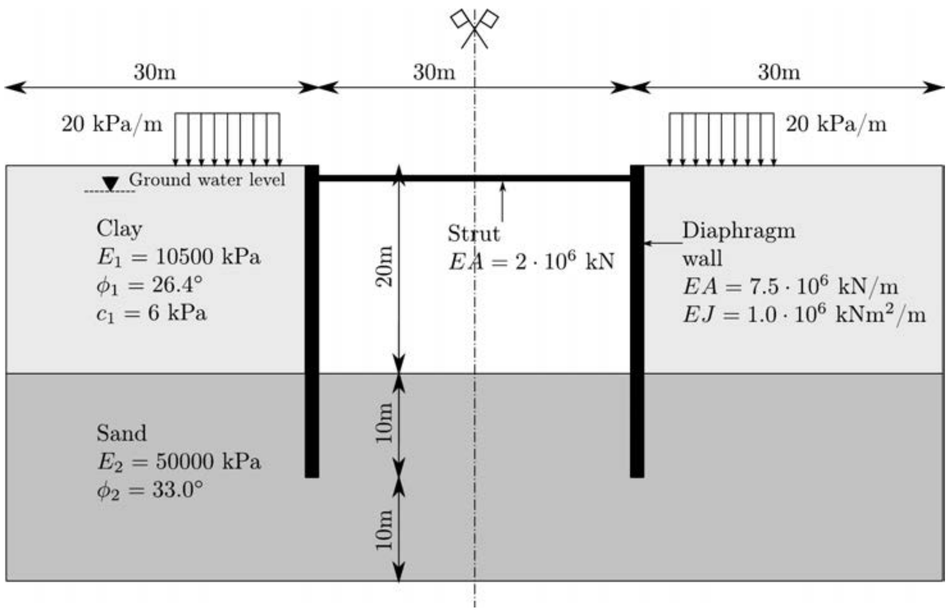

Fig. 2. Deep excavation model.

Table 1. Parameters of soils.

	γ_{unsat} [kN/m^3]	γ_{sat} [kN/m^3]	ν [–]	E [kPa]	c [kPa]	ϕ [°]	ψ [°]
clay (1)	15.9	19.6	0.33	10500	6.0	26.4	0.0
sand (2)	17.5	20.0	0.3	50000	0.0	33.0	3.0

5. Database preparation and training ANN

In order to prepare data for training ANNs, 36 different parameter sets $\mathbf{p}$ were generated and numerical model was evaluated 36 times. At this point partial excavation up to the depth 15 m is assumed as shown on Fig. 3. Parameter sets were generated carefully to be evenly distributed in the chosen parameter space (see Table 2). Generally, it was assumed that Young moduli and friction angles vary in the range $\pm 20\%$ starting from the values presented in Table 1. For each model evaluation the lateral displacements at all nodes of the finite elements constituting diaphragm wall were picked up, then the wall's shape was smoothed by means of the 3rd order splines. Values of

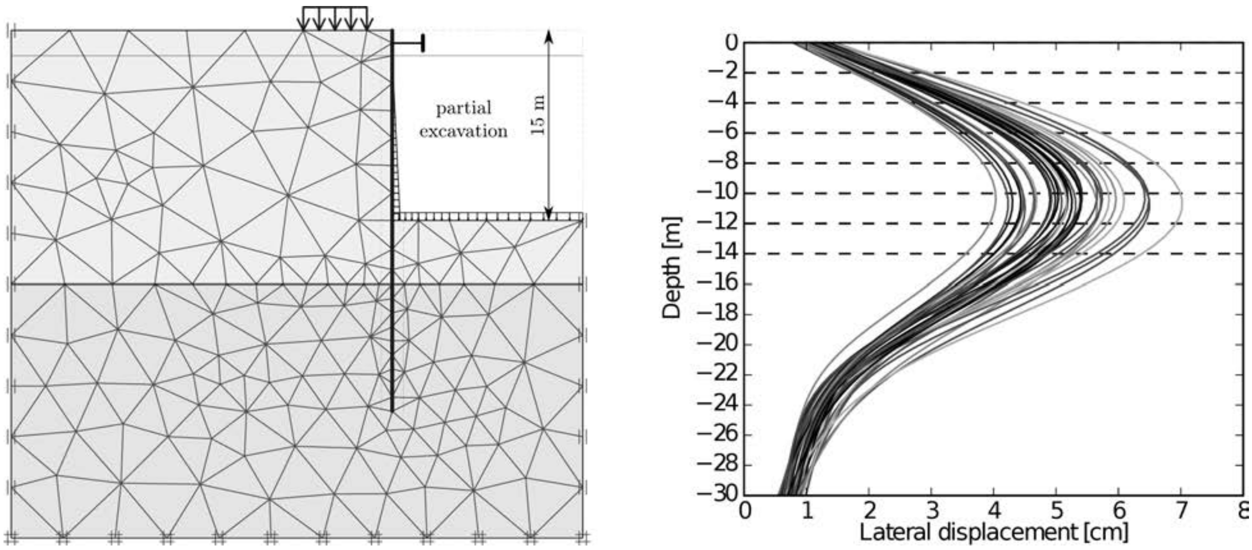

Fig. 3. Partial excavation model (left) and the lateral displacements of diaphragm wall for different sets of soil parameters (right). Dashed lines indicate monitored levels.

Table 2. Parameters of soils.

No.	E_1 [kPa]	ϕ_1 [°]	E_2 [kPa]	ϕ_2 [°]	No.	E_1 [kPa]	ϕ_1 [°]	E_2 [kPa]	ϕ_2 [°]
1	11200	24.6	46667	30.8	19	12600	31.7	40000	26.4
2	9800	28.2	46667	30.8	20	12600	21.1	60000	26.4
3	9800	24.6	53333	30.8	21	8400	31.7	60000	26.4
4	11200	28.2	53333	30.8	22	12600	31.7	60000	26.4
5	9800	24.6	46667	35.2	23	8400	21.1	40000	39.6
6	11200	28.2	46667	35.2	24	12600	31.7	40000	39.6
7	11200	24.6	53333	35.2	25	12600	31.7	60000	39.6
8	9800	28.2	53333	35.2	26	8400	21.1	60000	30.8
9	11900	22.9	43333	28.6	27	8400	21.1	46667	26.4
10	9100	29.9	43333	28.6	28	8400	31.7	46667	39.6
11	9100	22.9	56667	28.6	29	8400	24.6	60000	39.6
12	11900	29.9	56667	28.6	30	12600	21.1	46667	39.6
13	9100	22.9	43333	37.4	31	11200	21.1	60000	39.6
14	11900	29.9	43333	37.4	32	12600	26.4	50000	33.0
15	11900	22.9	56667	37.4	33	10500	31.7	50000	33.0
16	9100	29.9	56667	37.4	34	10500	26.4	40000	33.0
17	12600	21.1	40000	26.4	35	10500	26.4	50000	26.4
18	8400	31.7	40000	26.4	36	10500	26.4	50000	33.0

such processed displacements at 8 monitored depth levels (from 0 to 15 m) stated the answer of the model $\mathbf{u}_{\mathcal{M}}$ used for training artificial neural network.

8 networks of the architecture 4-4-1 were then trained – one ANN for each monitored displacement. All 36 samples were used for training and Tikhonov regularization with coefficient $\lambda = 0.005$ (as described in Subsec. 3.2) was applied. Very good training results were obtained: the root mean square error for all networks did not exceed value of 5e-3, and the correlation was about 0.998.

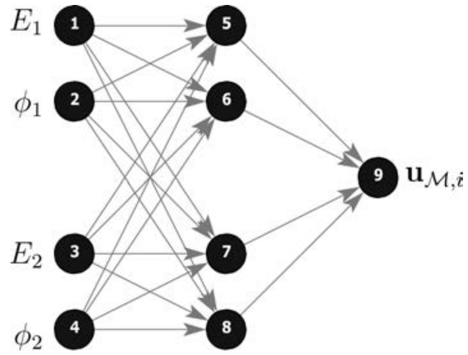

Fig. 4. Neural network architecture used for model approximation. 8 networks were trained – one ANN for each monitored displacement.

6. Identification results

6.1. Partial excavation

Field measurements for excavation problem were generated numerically in this paper. For this purpose a parameter set:

$$\mathbf{p}_{\mathcal{R}} = [8925,\ 22.4,\ 42500,\ 28.1] \tag{13}$$

was chosen and the finite element numerical model was used to get field data $\mathbf{u}_{\mathcal{R}}$. Equipped with the trained neural networks we performed minimization of the objective function $\Omega^*(\mathbf{p})$. A truncated Newton optimization method was used [7] (the same as for ANN training). The convergence is very fast, i.e. 30–50 function calls and gradient computations were needed. The obtained minimum considered as the identified parameters:

$$\mathbf{p}_{\mathcal{I}} = [9031,\ 22.3,\ 43311,\ 27.8] \tag{14}$$

is very close to the expected values. This can be observed in Fig. 5, where $\Omega^*(\mathbf{p})$ is visualized at chosen sections of the 4-dimensional parameter space.

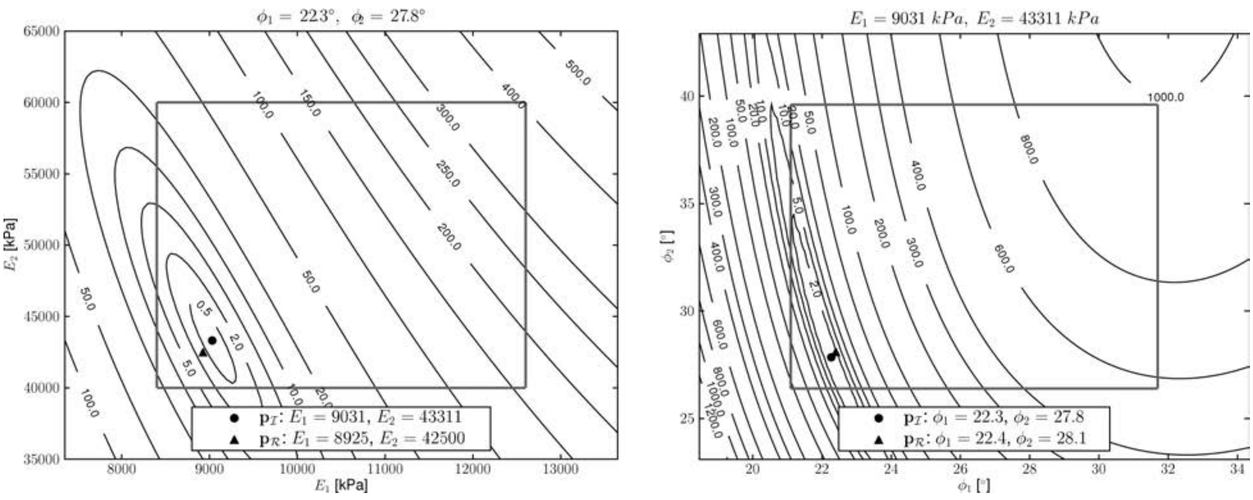

Fig. 5. $\Omega^*(\mathbf{p})$ in function of Young moduli with constant friction angles (left) and the opposite situation (right). Internal rectangle indicates the area of ANN training domain. Close position of the minimum $\mathbf{p}_{\mathcal{R}}$ and the identified parameters $\mathbf{p}_{\mathcal{I}}$ is observed.

6.2. Full excavation

Identification results, i.e. parameters $\mathbf{p}_{\mathcal{I}}$, were verified additionally by performing a simulation of full depth excavation (20 m) and comparing the results with a solution obtained for parameters $\mathbf{p}_{\mathcal{R}}$, i.e. with field measurements. Again, verification shows that the response of the numerical model is very similar for both sets of parameters (Fig. 6). This is true although the identification was performed (and ANNs were trained) at earlier construction stage, for different behavior of the structure and qualitatively different shapes of the curves in Fig. 3 and in Fig. 6.

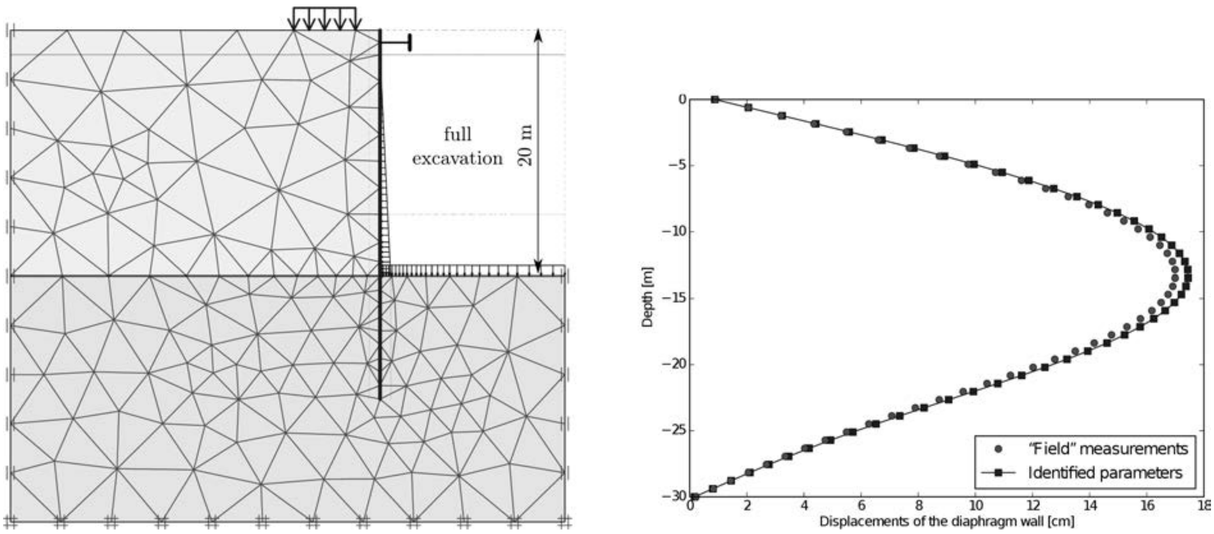

Fig. 6. Full excavation model (left) and the lateral displacements of diaphragm wall calculated with identified parameters compared to numerically generated field measurements (right).

7. Concluding remarks

The following conclusions can be stated:

1. Only 36 runs of numerical model were needed to create sufficient database for ANN training and to achieve satisfactory neural approximation of the numerical model parametrized with *four variables*. Preparation of training data is the main numerical cost of the identification procedure. Numerical costs of training the networks and then minimizing objective $\Omega^*(\mathbf{p})$ can be estimated as a cost of single run of the excavation model.

2. For comparison, in [9], where no ANN was used, 20–35 runs of model were needed by gradient optimization method (with finite difference for gradient calculation) in order to identify *two* parameters of deep excavation model (and 208 runs of model runs if genetic algorithm was used). This shows the superiority of the presented method, where four parameters were identified with the similar numerical costs. We also note, that ANN based identification method does not cause loss of accuracy since differences between the identified parameters $\mathbf{p}_{\mathcal{I}}$ and the expected ones $\mathbf{p}_{\mathcal{R}}$ do not exceed 2%.

3. Moreover, considered parameter identification problem seems to have unique solution in the considered 4-dimensional parameter space (single minimum of $\Omega^*(\mathbf{p})$ in Fig. 5). Thus the usage of ANN approximation and gradient optimization methods for solving considered inverse problem is additionally justified in this case.

4. We showed also, that identified parameters can be used to predict behavior of the structure at further excavation stages. However, the presented method should be tested with real field

measurements before taking advantage of it in real construction world. This is a subject of current work of the author.

References

[1] R.B.J. Brinkgreve, W. Broere, and R. Al-Khoury. *Plaxis: 2D, Version 8.* Balkema, 2002.
[2] J. Ghaboussi, D.A. Pecknold, and M. Zhang. Autoprogressive training of neural network constitutive models. *Int. J. Numer. Meth. Engng.*, **42**: 105–126, 1998.
[3] K. Hornik, M. Stinchcombe, and H. White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, **3**: 551–560, 1990.
[4] Y. Hou, J. Wang, and L. Zhang. Three-dimensional numerical modeling of a deep excavation adjacent to Shanghai Metro tunnels. *Computational Science – ICCS 2007*, pp. 1164–1171, 2007.
[5] I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *Neural Networks, IEEE Transactions on*, **9**(5): 987–1000, 1998.
[6] A. Nardin, B. Schrefler, and M. Lefik. Application of artificial neural network for identification of parameters of a constitutive law for soils. *Lecture Notes in Computer Science*, pp. 545–554, 2003.
[7] S.G. Nash. A survey of truncated-Newton methods. *Journal of Computational and Applied Mathematics*, **124**(1–2): 45–59, 2000.
[8] B. Pichler, R. Lackner, and H.A. Mang. Back analysis of model parameters in geotechnical engineering by means of soft computing. *International Journal for Numerical Methods in Engineering*, **57**(14): 1943–1978, 2003.
[9] C. Rechea, S. Levasseur, and R. Finno. Inverse analysis techniques for parameter identification in simulation of excavation support systems. *Computers and Geotechnics*, **35**(3): 331–345, 2008.
[10] A. Siemińska–Lewandowska. The displacements of anchored diaphragm walls. *Studia Geotechnica et Mechanica*, **27**(1–2): 155–167, 2005.
[11] M. Wojciechowski. *Feed-forward artificial neural networks and their derivatives – application possibilities in engineering and geotechnics.* Ph.D. Thesis, 2009.
[12] M. Wojciechowski. Feed-forward neural network as a tool for solving nonlinear differential equations. *Proceedings of CMM-2009 – Computer Methods in Mechanics*, Zielona Góra, May 2009.
[13] M. Wojciechowski. Feed-forward neural network for python, http://ffnet.sourceforge.net, 2007.