Chapter 13

Advanced Methods in Neural Networks-Based Sensitivity Analysis with their Applications in Civil Engineering

Maosen Cao , Nizar F. Alkayem , Lixia Pan and
Drahomír Novák

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/64026

Abstract

Artificial neural networks (ANNs) are powerful tools that are used in various engineering fields. Their characteristics enable them to solve prediction, regression, and classification problems. Nevertheless, the ANN is usually thought of as a black box, in which it is difficult to determine the effect of each explicative variable (input) on the dependent variables (outputs) in any problem. To investigate such effects, sensitivity analysis is usually applied on the optimal pre-trained ANN. Existing sensitivity analysis techniques suffer from drawbacks. Their basis on a single optimal pre-trained ANN model produces instability in parameter sensitivity analysis because of the uncertainty in neural network modeling. To overcome this deficiency, two successful sensitivity analysis paradigms, the neural network committee (NNC)-based sensitivity analysis and the neural network ensemble (NNE)-based parameter sensitivity analysis, are illustrated in this chapter. An NNC is applied in a case study of geotechnical engineering involving strata movement. An NNE is implemented for sensitivity analysis of two classic problems in civil engineering: (i) the fracture failure of notched concrete beams and (ii) the lateral deformation of deep-foundation pits. Results demonstrate good ability to analyze the sensitivity of the most influential parameters, illustrating the underlying mechanisms of such engineering systems.

Keywords: civil engineering, neural networks, sensitivity analysis, NNC-based sensitivity analysis, NNE-based sensitivity analysis

1. Introduction

In solving complex civil engineering problems, conventional analytical and empirical methodologies suffer from many difficulties. This is mainly because of the limitations of such methods in handling large, complex structures that may require time-consuming and exhausting tasks. In such situations, soft-computing techniques come into the picture. They are effective estimation tools that reduce the cost and time of design and analysis. Neural networks are useful soft-computing tools that can be used for classification and prediction in complex civil engineering problems [1–3].

Sensitivity analysis is a necessary approach for understanding the relationship and the influence of each input parameter on the outputs of a problem. The key point behind sensitivity analysis is that by slightly varying each explicative input parameter and registering the response in the output, the explicative parameters with the highest sensitivity values gain the greatest importance. Sensitivity analysis of the most significant parameters can be very useful for analyzing complex engineering problems.

Neural network-based parameter sensitivity analysis in civil engineering systems is gaining more importance due to the remarkable ability of neural networks to explain the nonlinear relationships between the explicative and response variables of a problem [1, 4]. Commonly, a specific training technique is used to develop one optimal neural network to be a system model, and this model is then used for sensitivity analysis [5–10]. Yet, it is relatively difficult to determine the most optimal neural network model, for reasons such as random initialization of the underlying connection weights in the neural network model, different features of various learning techniques used to train the neural network, the absence of a reliable technique for defining the optimal structure in neural network modeling, etc. To overcome these difficulties, two potential techniques, namely neural network committee (NNC)-based sensitivity analysis [1] and neural network ensemble (NNE)-based sensitivity analysis [11], are illustrated. These two paradigms utilize a group of pre-trained optimal neural networks to handle the neural network modeling, thereafter implementing parameter sensitivity analysis individually and lastly defining the sensitivity of parameters. This chapter is organized as follows. A complete explanation is given of some traditional neural network-based sensitivity analysis. Thereafter, the NNC-based parameter sensitivity analysis method is presented, followed by a geotechnical engineering case study of strata movement and two case studies related to classical civil engineering. Then, the NNE-based sensitivity analysis paradigm is described, followed by two illustrative case studies. Finally, a complete summary of the chapter is presented.

2. Typical neural networks-based sensitivity analysis algorithms

Many studies have been concerned with improving existing neural network-based sensitivity analysis methods [9]. Among the different techniques, the partial derivative algorithm [5] and the input perturbation algorithm [10] have superior performance compared to other techniques based on the magnitude of weights [6, 7]. Therefore, these two algorithms are explored in detail in this chapter, along with some other techniques.

2.1. Partial derivative algorithm

The partial derivative algorithm is a famous neural network-based sensitivity analysis technique [5, 11]. Its characteristics enable it to deal with neural networks that apply first-derivative activation functions, such as back-propagation neural networks (BPNNs) and radial basis function neural networks (RBFNNs) [1, 8]. By implementing the partial derivative algorithm, it is possible to identify the variations of output parameters of neural networks with respect to small changes in each input parameter, thereby defining the contribution of each such input on the output parameters. This can be done by deriving the output parameters of the neural network with respect to input parameters, in other words, by calculating the Jacobian matrix that contains the partial derivatives of outputs with respect to inputs [5, 11].

For a successful BPNN model having input x_i with n_i as the total number of inputs and output y_k with n_k as the total number of outputs, the Jacobian matrix $\frac{\partial y_k}{\partial x_i}$ can be defined by using the chain rule as [1]

$$\frac{\partial y_k}{\partial x_i} = \frac{\partial y_k}{\partial O_k}\frac{\partial O_k}{\partial y_{h_n}}\cdots\cdots\frac{\partial y_{h_1}}{\partial O_{h_1}}\frac{\partial O_{h_i}}{\partial x_i} = \sum\nolimits_{h_i}\sum\nolimits_{h_{i-1}}\sum\nolimits_{h_1}\begin{bmatrix} W_{h_n k} f'(O_k) W_{h_{n-1}h_n} \\ f'(O_{h_n})\cdots\cdots W_{ih_1} f'(O_{h_1}) \end{bmatrix} \tag{1}$$

where x_i is the ith input variable $h_n, h_{n-1}, \ldots$, and h_i are the hidden neurons from the nth to the first hidden layer, respectively; y_k, y_{h_n}, and y_{h_1} are the output values for output neuron k, hidden neurons h_n, and n_1 in the respective nth and the first hidden layer; $W_{h_n k}$ is the connection weight between the kth output neuron and the hidden neuron h_n; $W_{h_{n-1}h_n}$ is the connection weight between the hidden neurons h_{n-1} and h_n and W_{ih_1} is the connection weight between the ith input neuron and the hidden neuron h_1; O_k, O_{h_n}, and O_{h_1} are the weighted sums of kth output neuron, the hidden neuron h_n, and h_1, respectively; f^1 denotes the derivative of the activation function f. y_k, y_{h_n}, and y_{h_1} can be given as

$$\begin{cases} y_k = f(O_k), O_k = \sum_{h_n} y_{h_n} W_{h_n,k} + b_k \\ y_{h_n} = f(O_{h_n}), O_{h_n} = \sum_{h_n} y_{h_{n-1}} W_{h_{n-1},k} + b_{h_n} \\ \vdots \\ y_{h_1} = f(O_{h_1}), O_{h_1} = \sum_{h_n} x_i W_{i,h_1} + b_{h_1} \end{cases} \tag{2}$$

where b_k, b_{h_n}, and b_{h_1} are the biases of the kth output neuron, the hidden neuron h_n, and, h_1, respectively.

For p training samples of each input x_i on the output y_k of the neural network, c_{ik} can be calculated as

$$c_{ik} = \sum_p \left| \left(\frac{\partial y_k}{\partial x_i} \right)_p \right| \tag{3}$$

For each input parameter, the value of c can be used as a factor for classification of the influence of total inputs on the outputs of the neural network model. The most important or crucial input parameter may have the highest c value [1].

2.2. Input perturbation algorithm

The input perturbation algorithm is another common method for neural-network-based sensitivity analysis [6, 9]. It implements a small perturbation on each input of the neural network model and measures the corresponding change in the outputs. This perturbation is applied on one input individually at a time while all other inputs are fixed, and the response for perturbation of each output is registered. Sensitivity analysis is performed by giving a rank for each response of the output generated by the same perturbation in every input parameter. The input that has the highest effect on the outputs after perturbation is considered the most influential or important [1].

In essence, when a larger amount of perturbation is added to the selected input parameter, the mean square error (MSE) of the neural network increases. The variance of the input parameter can be represented as $x_i = x_i + \Delta x_i$, where x_i is the current selected input variable and Δx_i is the perturbation. The perturbation can be varied from 0 to 50% by steps of 5% of the input value. Depending on the increasing value of the MSE corresponding to each perturbed input, outputs can be ranked and thus sensitivity analyses are performed [1, 8].

2.3. Weights method

This method was proposed by Garson [12] and Goh [13]. In this method, for each hidden neuron, the connection weights are divided into components related to each input neuron. This method was simplified by Gevrey et al. [8] to give the same results as the initial method. For the purpose of illustration, a multilayer neural network with a single hidden layer is considered; thereafter, for each hidden neuron the following calculations are used:

For $i = 1$ to n_i

  For $j = 1$ to n_j

http://dx.doi.org/10.5772/64026

$$D_{ij} = \frac{|W_{ij}|}{\sum_{i=1}^{n_i} |W_{ij}|} \tag{4}$$

End

End

where n_i and n_j are the number of input and hidden neurons, respectively; W_{ij} is the weight corresponding to input neuron *i* and hidden neuron *j*. The percentage relative contribution of all inputs RC_i is then calculated as

For $i = 1$ to n_i

$$RC_i = \frac{\sum_{j=1}^{n_j} D_{ij}}{\sum_{j=1}^{n_j} \sum_{i=1}^{n_i} D_{ij}} \tag{5}$$

End

2.4. Profile method

This method was proposed by Lek et al. [14–16], and further explained by Gevrey et al. [8]. The key point behind this method is to analyze one particular input at a time while fixing the values of all other inputs. The procedure starts by dividing the value of each input parameter into equal subintervals, whereas all other inputs are set prior to minimum, quarter, half, three quarters of the maximum and maximum, respectively. At the end of this task, patterns of five values corresponding to different input parameters result and the median value for each pattern is calculated. The median values are plotted with respect to the subintervals to form a profile that explains the contribution of the input parameter. Finally, for all inputs, a set of curves explaining the relative importance for all input parameters is obtained [8].

2.5. Stepwise method

In this method, one input parameter is blocked and the responses of the outputs are recorded. This process is performed step by step for all input parameters and the responses of the outputs are recorded by means of the MSE. Depending on the MSE, the relative importance of each input variable is ranked correspondingly. There are two main strategies for the stepwise method. The first is to construct a number of neural network models by evolving the input parameters one by one. This strategy is called forward stepwise, while the backward stepwise strategy can be implemented in the reverse way, that is, constructing neural network models by first using all input parameters and then blocking each input parameter [8, 17].

This method can be improved to reduce the difficulty of producing many neural network models by using a single model. In this model, one input parameter is blocked and the MSE is calculated. The parameter with the maximum MSE value is ranked as the most important and can then be either removed from the model or fixed at its mean value so that the contribution of other parameters can be found, and so on.

3. Neural network committee-based sensitivity analysis

Consider a neural network model with a sensitivity analysis-ranking vector $R = [r_1, r_2, ..., r_n]$ and the actual sensitivity analysis-ranking vector $R_0 = [a_1, a_2, ..., a_n]$, where r_i and a_i are the calculated and actual ranks of *i*th input parameter, respectively, and *n* is the number of input parameters. To reduce the difference between *R* and R_0 to minimum, it is not efficient to use single neural network model to perform sensitivity analysis. The reason is the absence of persistence in sensitivity analysis of one neural network model even when a major sensitivity analysis strategy is implemented. In recognition of this fact, it is more effective to utilize a set of good pre-trained neural network models instead of using a single optimal model for sensitivity analysis. This procedure is well used in neural network committee (NNC)-based sensitivity analysis [1].

The mathematical foundation of NNC-based sensitivity analysis starts from the weak law of large numbers in probability. Having $x_1, x_2, ...$ infinite set of random variables with no correlation between any two of them, each having the exact value of μ and variance σ^2, the sample average convergence in probability can be written as [18]

$$\bar{x}_n = \frac{(x_1 + x_2 + \ldots + x)}{n} \tag{6}$$

or, in other words, for a small number ε, the following can be expressed:

$$\lim_{n\to\infty} P\left(|\bar{x}_n| - \mu < \varepsilon\right) = 1 \tag{7}$$

By considering single neural network sensitivity analysis-ranking vector *R*, the elements $r_1, r_2, ...$ can be defined as random variables; in other words, *R* is composed of *n* random variables. In the case of neural network ensemble-based sensitivity analysis, a set of random variables $r_i^1, r_i^2, ..., r_i^m$ related to r_i are obtained. Depending on the weak law of large numbers, for a large number *m*, the mean of $r_i^1, r_i^2, ..., r_i^m$ can converge to the actual ranking values a_i in R_0. Therefore, in NNC-based sensitivity analysis, it is possible to find a ranking vector *R* that is close to the actual ranking vector R_0.

As the number of input variables is specified and the input variables are not completely random, due to the many specifications that appear during neural network model training, the condition of the weak law of large numbers that is applied on an infinite number of random variables is not satisfied. For this reason, optimization strategy can be an efficient tool to select a number of good pre-trained neural network models and skip those with weak performance. By electing the best neural network elements and eliminating the bad ones, optimization can generate good predictions of sensitivity analysis-ranking vectors [1].

Depending on the above principles, we can summarize NNC-based sensitivity analysis in three basic procedures. First, groups (seeds) of successful neural network models are prepared using neural network-training techniques such as back propagation (BP) or radial basis functions, etc. Then, a set of best-performance models are chosen to compose the optimal NNC that is used in performing ensemble neural network sensitivity analysis by individual applications of sensitivity analysis, giving large numbers of *R*. Finally, the mean of *R* is calculated to find the accurate approximation of R_0. A schematic diagram of NNC-based sensitivity analysis strategy is given in **Figure 1**.

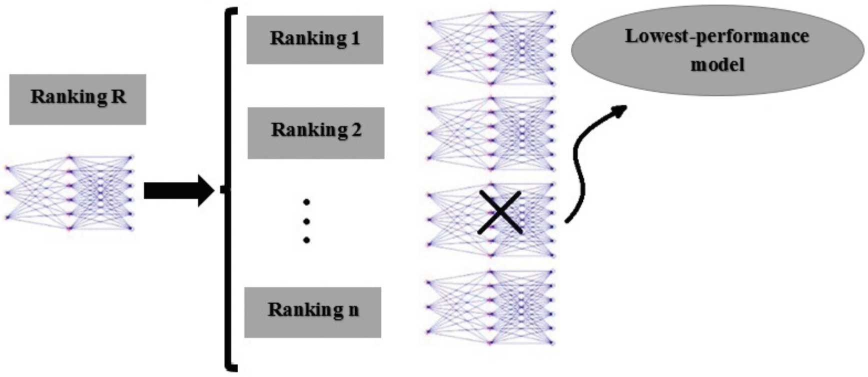

Figure 1. A schematic diagram of NNC-based sensitivity analysis strategy.

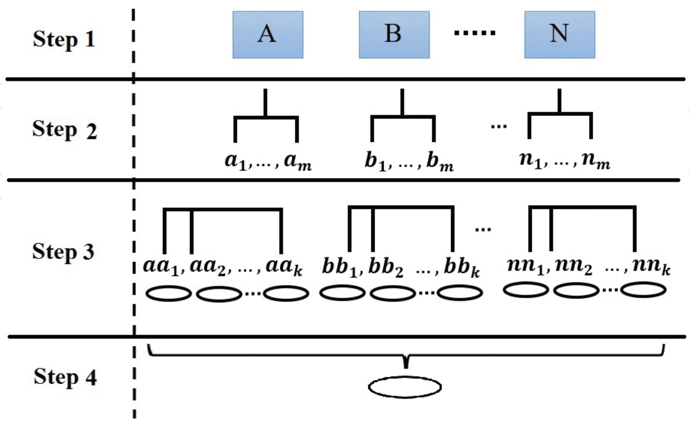

Figure 2. NNC-based sensitivity analysis strategy stepwise procedure: A–N, the neural network model seeds; $a_1 - a_{m'}$ $b_1 - b_{m,}$ and $n_1 - n_{m'}$ the candidate groups of neural network models; $a_1 - aa_{k'}$ $bb_1 - bb_{k'}$ and $nn_1 - nn_{k'}$ the superior neural network models; ellipse refers to a sensitivity analysis ranking of an input parameter.

The basic steps for the NNC-based sensitivity analysis algorithm are shown in **Figure 2** and can be explained as follows:

1. Select the best available types of neural network model empirically. These are called "seeds" for NNC-based sensitivity analysis.
2. Each seed involves a set of neural network models. These models are varied by means of a number of hidden neurons or hidden layers to produce a candidate group of neural network models.
3. Depending on the MSE, a subset k of superior neural network models is picked up, where $k = \frac{3}{10}m$ has been experimentally specified and m is the number of neural network models in the candidate group. Thereafter, sensitivity analysis is employed on each model to generate a group of sensitivity analysis-ranking vectors R.
4. For each input parameter, the mean of the related ranking number in the sensitivity analysis-ranking vector R is calculated to form a predicted ranking vector close to the actual ranking vector R_0, which is calculated as

$$a_i \approx \hat{a}_i = \frac{1}{NK}\sum_{s=1}^{N}\sum_{t=1}^{K} r_i^{st}, i = 1, 2, \ldots, l \tag{8}$$

where $\hat{a}_i$ is the predicted value of a_i in R_0 for variable x_i in R, K is the number of elements in the candidate group of neural network models (committee), N is the number of neural network seeds, and l is the number of input parameters.

4. NNC-based sensitivity analysis of strata movement

Strata movement is a critical problem in geotechnical engineering because of the complex highly nonlinear properties involved. It is necessary to define the most significant factors involved in strata movement. Therefore, NNC-based sensitivity analysis strategy is used. The dataset of strata movement is composed of 168 samples taken from multiple typical observation stations of earth surface movement above underground metal mines. The dataset has six input parameters and three output parameters as shown in **Table 1**. These parameters characterize the working operation of strata movement of underground metal mines.

In NNC-based sensitivity analysis, four scenarios are chosen, depending on the output variables (**Table 1**): scenario (1) all output parameters, (2) only *MAU*, (3) only *MAL*, and (4) only *AA*. At the beginning, radial basis function and BP neural networks are selected as seeds, because of their proven ability to handle nonlinear features. Then, 50 neural network models are generated by each seed to construct two candidate sets of neural network models. Thereafter, 15 superior neural network models are chosen from each set to form a committee containing the best-performed neural network models. After that, sensitivity analysis is applied to each model by utilizing both a perturbation algorithm and a partial derivative algorithm to produce a group of ranking vectors R. Next, the sum of corresponding ranking numbers that is considered as a score for input parameters is calculated. The score is a reflection of the near actual ranking R_0. The sum is used instead of the mean to prevent the repetition of the identical values for different parameters, in order to have fewer neural network models from which to decide the final ranking. The best-performed neural networks and the input parameter ranking for scenario (1) are illustrated in **Table 2**.

Parameter	Characteristics	Parameter type
MCU	Mean consistency of upper wall rock	Input
LCL	Mean consistency of lower wall rock	Input
SAO	Slope angle of ore body	Input
TO	Thickness of ore body	Input
LO	Length of ore body	Input
DE	Depth of excavation	Input
MAU	Movement angle of upper wall rock	Output
MAL	Movement angle of lower wall rock	Output
AA	Avalanche angle	Output

Table 1. Measured parameters of strata movement [1].

Best-performed neural network model	MCU	LCL	SAO	TO	LO	DE
RBF1	5	6	1	4	3	2
RBF2	2	1	6	5	4	3
RBF3	3	6	5	2	4	1
RBF4	5	6	3	1	4	2
RBF5	4	6	3	2	5	1
RBF6	5	4	2	3	6	1
RBF7	4	6	5	2	3	1
RBF8	2	6	3	5	4	1
RBF9	4	6	5	2	3	1
RBF10	2	5	6	3	4	1
RBF11	3	4	1	6	5	2
RBF12	3	4	2	6	5	1
RBF13	2	4	3	6	5	1
RBF14	2	4	3	6	5	1
RBF15	4	2	3	5	6	1
BP1	1	4	3	6	5	2

Best-performed neural network model	MCU	LCL	SAO	TO	LO	DE
BP2	2	4	3	6	5	1
BP3	4	2	3	5	6	1
BP4	4	3	2	6	5	1
BP5	2	3	4	6	5	1
BP6	4	2	1	5	6	3
BP7	5	3	2	6	4	1
BP8	4	3	1	5	6	2
BP9	4	3	2	6	5	1
BP10	4	3	1	5	6	2
BP11	3	4	2	5	6	1
BP12	3	5	1	4	6	2
BP13	4	3	2	5	6	1
BP14	4	3	1	6	5	2
BP15	3	5	1	4	6	2

Table 2. Sensitivity analysis rankings produced by best-performed neural network model groups [1].

Scenarios	Score and ranking	MCU	LCL	SAO	TO	LO	DE
(1)	Score	101	120	80	138	148	43
	Ranking	3	4	2	5	6	1
(2)	Score	96	114	64	162	145	49
	Ranking	3	4	2	6	5	1
(3)	Score	96	81	87	75	155	136
	Ranking	4	2	3	1	6	5
(4)	Score	109	121	113	103	86	98
	Ranking	4	6	5	3	1	2

Table 3. NNC-based sensitivity analysis results for strata movement.

The outcome sensitivity analysis for the four scenarios is illustrated in **Table 3**. It is clear from the table that for scenario (1), *DE* has the highest importance, followed by *SAO, MCU, LCL, TO,* and *LO*, respectively. In scenario (2), the degree of importance is the same as in scenario (1), but *LO* is more significant than *TO*. Nevertheless, in scenario (3), *TO* has the highest significance, above that of *LCL, SAO,* and *MCU*, which have approximately similar significance, and then *DE* and *LO* have the least significance. Finally, in scenario (4) *LO* has the highest contribution followed by *DE, TO, MCU, SAO*, and *LCL*, respectively. However, the contributions of *DE* and *TO* are very close to those of *MCU, SAO*, and *LCL*.

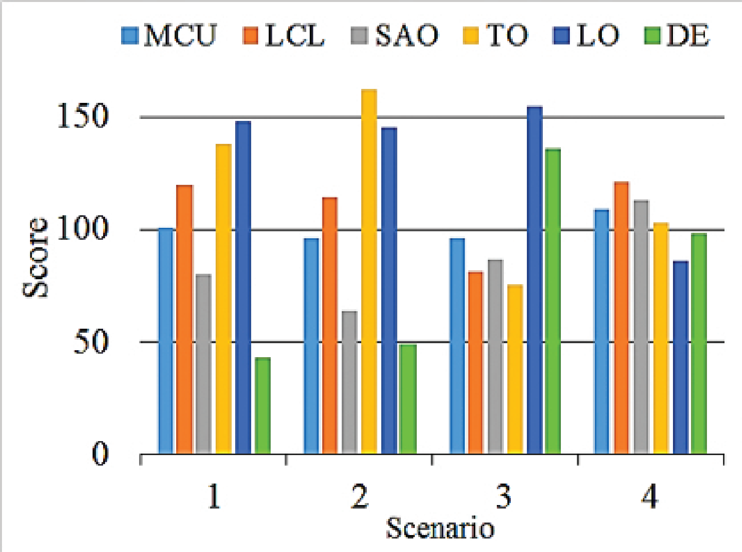

Figure 3. Activity analysis of dependent variables for strata movement based on NNC-based sensitivity analysis results [1].

The working condition of strata movement is defined by the predictability of response parameter (output parameters). For this reason, the scores of the input variables after sensitivity analysis for three scenarios (*MAU, MAL*, and *AA*) that are related to the response variables are plotted in **Figure 3**. The response variable with the highest sensitivity against explicative variables has the highest predictability, and this can be calculated by finding the variance of the score vector of the explicative variables. The result of that procedure is 1965.6, 1068.4, and 150, corresponding to the response variables *MAU, MAL*, and *AA*, respectively. It is obvious that *MAU* has the highest predictability, followed by *MAL* and *AA*. Therefore, we can consider the angles of the upper wall rocks as the most significant feature, ahead of the lower wall rocks and the avalanche angle that are less important.

5. NNE-based parameter sensitivity analysis

The NNE-based parameter sensitivity analysis technique is a modified version of the NNC-based sensitivity analysis. It reduces the time-consuming procedure of using different neural network types as seeds by using just one preferred neural network type as the seed [4]. NNE-based parameter sensitivity analysis incorporates the following steps: (1) one preferred type of neural network is considered as the seed, (2) a set of *k*-neural network models that are varied with regard to the number of hidden neurons and hidden layers is defined, (3) from *k*-neural network models, a group of *n* best-performed models ($n < k$) is picked up and the other poorly performed models are eliminated to form an NNE model, and (4) a sophisticated sensitivity analysis algorithm is performed on the NNE model to obtain a sensitivity ranking of all input variables of the engineering problem under consideration. A schematic diagram of NNE-based parameter sensitivity analysis is shown in **Figure 4**.

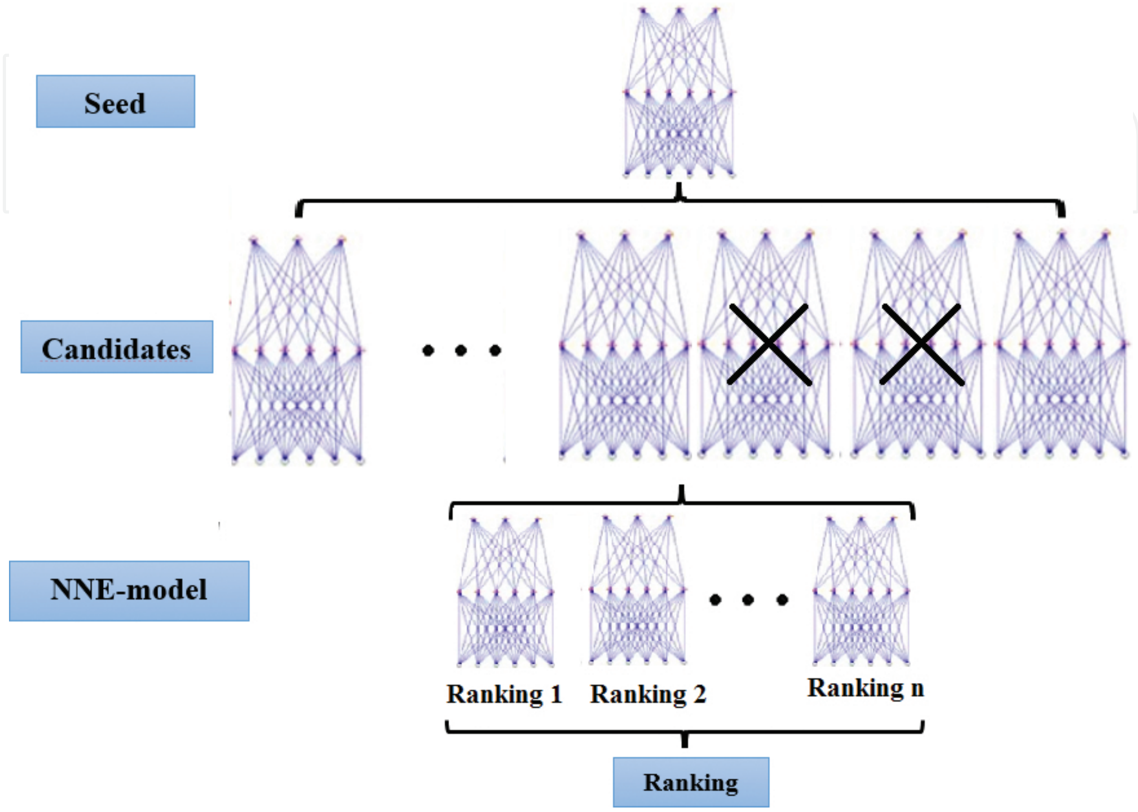

Figure 4. A schematic representation of NNE-based parameter sensitivity analysis.

6. Illustrative case studies

To highlight the application of NNE-based parameter sensitivity analysis technique, two civil engineering case studies are explained. The first is the determination of the importance of material properties in the fracture failure of a notched concrete beam and the second is the specification of significant parameters in the lateral deformation of a deep-foundation pit [4].

6.1. Fracture failure of notched concrete beam

Fracture failure is the most common problem facing engineers in the analysis and usage of concrete structures [19,20]. Good knowledge of appropriate material properties is necessary during modeling of the fracture behavior of concrete structures. Such material properties are defined by a three-point bending of a notched concrete specimen. Therefore, the NNE-based parameter sensitivity analysis strategy is used to find the most crucial material properties in the fracture failure of a notched concrete beam. The geometry of the notched concrete beam is shown in **Figure 5**, with experimentally determined mean values of material properties [21]: modulus of elasticity $E_c = 35$ GPa, tensile strength $f_t = 3$ MPa, compressive strength $f_c = 65$ MPa, fracture energy $G_f = 100$ N/m, and compressive strain at compressive strength in the uniaxial compressive test $e_c = 0.003$. A group of 20 notched concrete beam samples is prepared depending on a stratified Monte Carlo-type simulation called Latin hypercube sampling (LHS) [22], using FReET software [23] with a correlation control procedure [24].

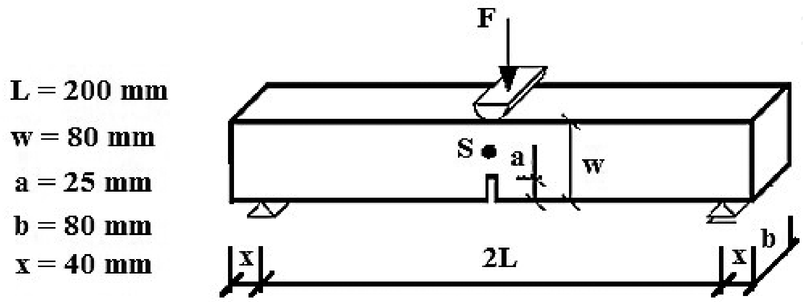

Figure 5. Notched concrete beam under three-point bending [4].

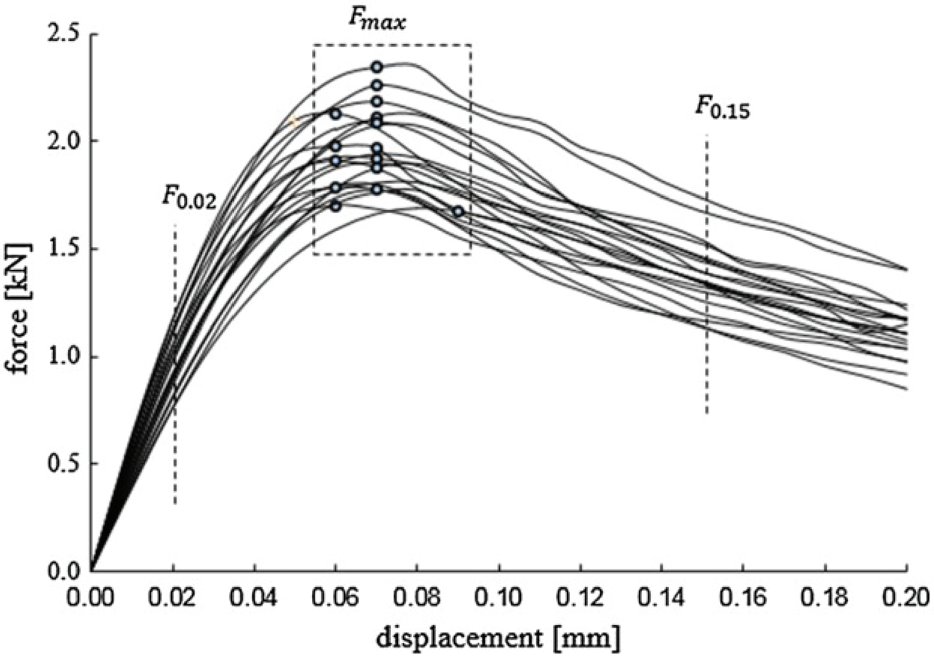

Figure 6. Force-displacement curves at the notch tip *S* from 20 simulated realizations of notched concrete beams [4].

The 20 notched concrete beam samples are determined by employing the following steps: (1) material properties are considered as random variables and mean values are obtained by experiments; (2) for each property, the LHS stochastic simulation is utilized to produce 20 random realizations of $\{E_c, f_t, f_c, G_f, e_c\}$ that feature variation of 0.15 and that obey a rectangular probability distribution, to impose variability for the creation of the training set. Each random realization determines a numerical nonlinear fracture mechanic calculation of a notched concrete beam; and (3) the finite element method (FEM) software ATENA [25] is applied to each realization to simulate the tensile fracture of the corresponding notched concrete beam. The fracture failure is described by a force-displacement curve at the notch tip *S* (**Figure 5**). A set of 20 force-displacement curves is illustrated in **Figure 6**. This set can be used as input data for the NNE-based sensitivity analysis. These curves describe the correlation between material fracture-mechanical properties and the nonlinear response of the beam. The sensitivity of the material properties to tensile fracture is studied depending on three forces: $F_{0.02}$, the force corresponding to 0.02-mm displacement; $F_{\max}$ the maximum force; and $F_{0.15}$ the force corresponding to 0.15-mm displacement. For each force, NNE-based parameter sensitivity analysis is applied to determine the significance of the material properties.

Parameter	Characteristics	Parameter type
E_c	Modulus of elasticity	Input
f_t	Tensile strength	Input
f_c	Compressive strength	Input
G_f	Fracture energy	Input
ε_c	Compressive strain	Input
$F_{0.02}$	Force at 0.02-mm displacement	Output
F_{max}	Maximum force	Output
$F_{0.15}$	Force at 0.15-mm displacement	Output

Table 4. Material properties in fracture failure of notched concrete beam [4].

Force	Ranking				
	E_c	f_t	f_c	G_f	ε_c
$F_{0.02}$	1	2	3	4	5
F_{max}	2	1	4	3	5
$F_{0.15}$	4	2	3	1	5

Table 5. Sensitivity analysis results of material parameters in fracture failure [4].

In NNE-based sensitivity analysis paradigm, a BP neural network with five input neurons and one output neuron (**Table 4**) is used as the seed to create a set of *k*-candidate neural network models. These models correlate the relationship between the material properties and the fracture failure. Depending on the performance of these models, the three best-performed neural network models are selected in the NNE model and the input perturbation algorithm is used for parameter sensitivity analysis. The result of the sensitivity analysis in this case is shown in **Table 5**. It is obvious from the table that f_t, followed by E_c and G_f, are the most important parameters in the fracture failure of the notched concrete beam.

6.2. Lateral deformation of deep-foundation pit

The construction of underground structures such as subway system tunnels, etc. requires deep-foundation pits. The working condition of a deep-foundation pit is usually defined by means of lateral deformation [26]. This lateral deformation usually involves a group of variables (**Table 6**), namely surface load q, deformation modulus of soil E, Poisson's ratio λ, soil cohesion C, and internal friction angle of soil φ. To analyze the working process of the deep-foundation pit, it is essential to study the sensitivity of these variables in order. Therefore, NNE-based parameter sensitivity analysis is applied to determine the importance of parameters in the lateral deformation of deep-foundation pits. For such analysis, a deep polygon-shaped foundation pit, as in [27], is utilized, having an excavation depth of 9.71 m, a width of earth-retaining wall of 8.7 m, and a length of reinforcement piles of 19.0 m, with the insertion ratio about 1.0. For testing cases, an orthogonal design of experiments is used to generate 25 testing cases, as shown in **Table 7** [27]. The testing cases are employed within the NNE-based sensitivity analysis paradigm to finally specify the contribution of each parameter to the lateral deformation y of the deep-foundation pits.

Parameter	Characteristics	Parameter type
q	Surface load	Input
E	Deformation modulus of soil	Input
ε	Poisson's ratio	Input
C	Soil cohesion	Input
φ	Internal friction angle of soil	Input
y	Lateral deformation of deep-foundation pit	Output

Table 6. Properties in lateral deformation of deep-foundation pit [4].

No.	q (kPa)	E (kPa)	ε	C (kPa)	φ (rad)	y (cm)
1	1 (5.0)	1 (3855)	1 (0.325)	1 (5.63)	1 (0.1386)	63.7
2	1	2 (6168)	2 (0.376)	2 (7.44)	2 (0.1834)	35.3
3	1	3 (7710)	3 (0.410)	3 (8.65)	3 (0.2133)	26.7
4	1	4 (9252)	4 (0.444)	4 (9.86)	4 (0.2432)	20.9
5	1	5 (11,565)	5 (0.478)	5 (11.68)	5 (0.2731)	12.5
6	2 (8.0)	1	2	3	4	55.8
7	2	2	3	4	5	32.1

No.	q (kPa)	E (kPa)	ε	C (kPa)	φ (rad)	y (cm)
8	2	3	4	5	1	21.9
9	2	4	5	1	2	16.3
10	2	5	1	2	3	25.2
11	3 (10.0)	1	3	5	2	47.8
12	3	2	4	1	3	26.1
13	3	3	5	2	4	16.2
14	3	4	1	3	5	30.4
15	3	5	2	4	1	22.1
16	4 (12.0)	1	4	2	5	37.1
17	4	2	5	3	1	18.0
18	4	3	1	4	2	34.9
19	4	4	2	5	3	25.8
20	4	5	3	1	4	18.9
21	5 (15.0)	1	5	4	3	25.2
22	5	2	1	5	4	42.4
23	5	3	2	1	5	30.1
24	5	4	3	2	1	22.4
25	5	5	4	3	2	15.6

Table 7. Orthogonal experimental design for producing testing samples [27].

Model	Ranking				
	q	E	λ	C	φ
NNM1	5	1	2	3	4
NNM2	5	1	2	3	4
NNM3	5	1	2	3	4

Table 8. Sensitivity analysis results in lateral deformation of deep-foundation pit [4].

As in the previous case study, a BP neural network is chosen as the seed in NNE-based sensitivity analysis to generate a set of k-candidate neural network models having five inputs and one output as listed in **Table 6**. By selecting three superior neural network models, namely NNM1, NNM2, and NNM3, and implementing input perturbation algorithm for sensitivity analysis, the ranking of each input parameter corresponding to each neural network model is shown in **Table 8**. It is clear that E is the most important parameter, followed by λ, C, ϕ, and q, respectively.

http://dx.doi.org/10.5772/64026

7. Summary

A short review of traditional neural network sensitivity analysis techniques was illustrated, followed by the presentation of two advanced techniques, NNC-based sensitivity analysis and NNE-based sensitivity analysis. These two techniques utilized selective superior neural network models along with some mathematical concepts to analyze the sensitivity of significant explicative variables. The efficiency of NNC-based sensitivity analysis paradigm was verified by studying the underlying influential parameters in strata movement. The effectiveness of NNE-based sensitivity analysis paradigm was proved by two case studies in civil engineering, the fracture failure of notched concrete beams and the lateral deformation of deep-foundation pits. These paradigms are essential for understanding the neural-network-based sensitivity analysis of critical engineering problems, due to their ability to determine the most and least important parameters, thereby reducing the inputs of neural network models to generate better predictability. They are good tools for analyzing the mechanism of engineering problems that black-box neural network models cannot explain.

Author details

Maosen Cao[1*], Nizar F. Alkayem[1], Lixia Pan[1] and Drahomír Novák[2]

*Address all correspondence to: cmszhy@hhu.edu.cn

1 Department of Engineering Mechanics, Hohai University, Nanjing, People's Republic of China

2 Faculty of Civil Engineering, Institute of Structural Mechanics, Brno University of Technology, Brno, Czech Republic

References

[1] Cao MS, Qiao P. Neural network committee-based sensitivity analysis strategy for geotechnical engineering problems. Neural Computing & Applications. 2008;17(5): 509–519. DOI: 10.1007/s00521-007-0143-5

[2] Cao M, Qiao P, Ren Q. Improved hybrid wavelet neural network methodology for time-varying behavior prediction of engineering structures. Neural Computing and Applications. 2009;18(7):821–832. DOI: 10.1007/s00521-009-0240-8

[3] Waszczyszyn Z, Ziemiański L. Neural networks in mechanics of structures and materials—new results and prospects of applications. Computers & Structures. 2001;79(22-25):2261–2276. DOI: 10.1016/S0045-7949(01)00083-9

[4] Cao MS, Pan LX, Gao YF, Novák D, Ding ZC, Lehký D, Li XL. Neural network ensemble-based parameter sensitivity analysis in civil engineering systems. Neural Computing & Applications. Forthcoming. DOI: 10.1007/s00521-015-2132-4

[5] Dimopoulos Y, Bourret P, Lek S. Use of some sensitivity criteria for choosing networks with good generalization ability. Neural Processing Letters. 1995;2(6):1–4. DOI: 10.1007/BF02309007

[6] Gedeon TD. Data mining of inputs: analysing magnitude and functional measures. International Journal of Neural Systems. 1997;8(2):209–218. DOI: 10.1142/S0129065797000227

[7] Wang W, Jones P, Partridge D. Assessing the impact of input features in a feedforward neural network. Neural Computing & Applications. 2000;9(2):101–112. DOI: 10.1007/PL00009895

[8] Gevrey M, Dimopoulos I, Lek S. Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling. 2003;160(3):249–264. DOI: 10.1016/S0304-3800(02)00257-0

[9] Montaño JJ, Palmer A. Numeric sensitivity analysis applied to feedforward neural networks. Neural Computing & Applications. 2003;12(2):119–125. DOI: 10.1007/s00521-003-0377-9

[10] Zeng X, Yeung DS. A quantified sensitivity measure for multilayer perceptron to input perturbation. Neural Computation. 2003;15(1):183–212. DOI: 10.1162/089976603321043757

[11] Yang Y, Zhang Q. A hierarchical analysis for rock engineering using artificial neural networks. Rock Mechanics and Rock Engineering. 1997;30(4):207–222. DOI: 10.1007/BF01045717

[12] Garson GD. Interpreting neural network connection weights. AI Expert. 1991;6(4):46–51.

[13] Goh ATC. Back-propagation neural networks for modelling complex systems. Artificial Intelligence in Engineering. 1995;9(3):143–151. DOI: 10.1016/0954-1810(94)00011-S

[14] Lek S, Belaud A, Baran P, Dimopoulos I, Delacoste M. Role of some environmental variables in trout abundance models using neural networks. Aquatic Living Resources. 1996;9(1):23–29.

[15] Lek S, Delacosteb M, Baranb P, Dimopoulosa I, Laugaa J, Aulagnierc S. Application of neural networks to modelling nonlinear relationships in ecology. Ecological Modelling. 1996;90(1):39–52. DOI: 10.1016/0304-3800(95)00142-5

[16] Lek S, Belaud A, Dimopoulos I, Lauga J, Moreau J. Improved estimation, using neural networks, of the food consumption of fish populations. Marine and Freshwater Research. 1995;46(8):1229–1236 . DOI: 10.1071/MF9951229

[17] Sung AH. Ranking importance of input parameters of neural networks. Expert Systems with Applications. 1998;15(3-4): 405–411. DOI: 10.1016/S0957-4174(98)00041-4

[18] Durrett R. Probability: Theory and Example, 3rd ed. Pacific Grove, CA: Duxbury; 2004. 521 p.

[19] Carpinteri A, Ferro G. Size effects on tensile fracture properties: a unified explanation based on disorder and fractality of concrete microstructure. Materials and Structures. 1994;27(10):563–571. DOI: 10.1007/BF02473124

[20] Rott JG. Computational modelling of concrete fracture [dissertation]. Delft: Technische Hogeschool Delft; 1988.

[21] Novák D, Lehký D. ANN inverse analysis based on stochastic small-sample training set simulation. Engineering Applications of Artificial Intelligence. 2006;19(7):731–740. DOI: 10.1016/j.engappai.2006.05.003

[22] Stein M. Large sample properties of simulations using Latin hypercube sampling. Technometrics. 1987;29(2):143–151 . DOI: 10.2307/1269769

[23] Novák D, Vořechovský M, Teplý B. FReET: software for the statistical and reliability analysis of engineering problems and FReET-D: degradation module. Advances in Engineering Software. 2014;72:179–192. DOI: 10.1016/j.advengsoft.2013.06.011

[24] Vořechovský M, Novák D. Correlation control in small-sample Monte Carlo type simulations I: a simulated annealing approach. Probabilistic Engineering Mechanics. 2009;24(3):452–462. DOI: 10.1016/j.probengmech.2009.01.004

[25] Červenka V, Jendele L, Červenka J. ATENA Program Documentation, Part 1: Theory. Prague: Červenka Consulting; 2009. 94 p.

[26] Feng S, Wu Y, Li J, Li P, Zhang Z, Wang D. The analysis of spatial effect of deep foundation pit in soft soil areas. Procedia Earth and Planetary Science. 2012;5:309–313. DOI: 10.1016/j.proeps.2012.01.052

[27] XU C, YE GB. Parameter sensitivity analysis of numerical model by cross test design technique. Hydrogeology and Engineering Geology. 2004;1:95–97.