technical transactions

No. 2023/007

civil engineering

sciendo

The modelling of layered rocks using a numerical homogenisation technique and an artificial neural network

Aleksander Urbański
aurbansk123@gmail.com | https://orcid.org/0000-0002-5544-9134
Cracow University of Technology, Idealogic Ltd.

Szymon Ligęza
sligeza@yandex.com | https://orcid.org/0000-0001-6702-9260
Doctoral Studies, AGH University of Science and Technology, Idealogic Ltd.

Piotr Przecherski
piotr.przecherski@pk.edu.pl | https://orcid.org/0000-0001-5871-5496
Cracow University of Technology, Idealogic Ltd.

Scientific Editor: Andrzej Winnicki, Cracow University of Technology
Technical Editor: Aleksandra Urzędowska, Cracow University of Technology Press
Language Verification: Timothy Churcher, Merlin Language Services
Typesetting: Małgorzata Murat-Drożyńska, Cracow University of Technology Press

Received: January 26, 2023
Accepted: May 8, 2023

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Competing interests: The authors have declared that no competing interests exist.

Funding: The work was supported by Polish research grant NCBiR no. POIR 01.01.01-00-0276/16-00, entitled "Modern application of artificial neural networks in software applying the finite- element method, dedicated to solving complex engineering problems in construction", which is gratefully acknowledged.

Citation: Urbański, A., Ligęza, S., Przecherski, P. (2023). The influence of carbide reinforcement on the properties of sintered aluminium alloy matrix composites. *Technical Transactions, e2023007*. https://doi.org/10.37705/TechTrans/e2023007

Abstract

A method of creating a constitutive model of layered rocks based on an artificial neural network (ANN) is reported in this work. The ANN gives an implicit constitutive function $\boldsymbol{\Sigma}^{n+1} = \mathrm{F}(\boldsymbol{\Sigma}^{n}, \Delta\mathbf{E})$, relating the new state of homogenized stresses $\boldsymbol{\Sigma}^{n+1}$ with the old state $\boldsymbol{\Sigma}^{n}$ and with the increment of homogenized strains $\Delta\mathbf{E}$. The first step is to repeatedly run a strain- controlled homogenisation on an uni-dimensional finite element model of a periodic cell with elastic-plastic models (Drucker-Prager) of the components. Paths are created in $(\boldsymbol{\Sigma},\mathbf{E})$ space, from which, a set of patterns is formed to train the ANN. A description of how to prepare this data and a discussion on ANN training issues are presented. Finally, the procedure based on trained ANN is put into a finite-element code (ZSoil.PC) as a user-delivered constitutive function. The approach is verified by comparing the results of the developed model basing on ANN with a direct (single-scale) analysis, which showed acceptable accuracy.

Keywords: layered rocks, finite element method, homogenisation, artificial neural network

1. Introduction

Layered rocks are an example of a medium for which different strategies of numerical analysis related to the problems of geomechanics can be employed. Generally, layered rocks possess a periodically inhomogeneous micro-structure which consists of two (or more) layers with different material properties. While using the finite element approach, the main division may be done between direct (or single-scale) and multi-scale modelling.

In the first type of analysis, FE meshing should correspond to the layers set up, see Fig. 1, and at least one finite element must be placed in the depth of each layer. This causes a very rapid growth of the FE mesh size, with up to multi-million DOFs, particularly for 3D problems. Furthermore, mesh generation is difficult and demanding in such cases. Nevertheless, single-scale modelling can be used in small problems and provides results which constitute a reference for the other approaches, as shown in Section 6.

Multi-scale modelling requires the introduction of homogenised media in terms of averaged stresses $\boldsymbol{\Sigma}$ and strains $\mathbf{E}$. A brief description of the homogenisation procedure of the layered media, such that each layer is described by a material model belonging to wider class of elastic-plastic models, is given in Section 2.

One of the ways of proceeding is known as FE^2 (see Giuntoli 2019, Feyel 1999), in which at each integration point of macro-level finite elements, a micro-level BVP set on the domain of periodic cell is solved simultaneously with a macro-level analysis. In the case of layered media, it seems to be particularly promising, mainly due to the simplicity of the micro-level problem. Nevertheless, in the present work the other approach is applied. First, a large set of separate problems of micro-level analysis is performed, creating data for of a constitutive model identification. These are paths created in $(\boldsymbol{\Sigma}, \mathbf{E})$ space, from which, a set of patterns is formed to train the artificial neural network (ANN). A detailed description of this way of proceeding given in Section 3.

Obviously, the described procedure is not the only possibility. In the other, elastic-plastic models based on yield surfaces with a prescribed shape are used. Instead, in the ANN based procedure, a constitutive function is sought, taking an implicit form that is consistent with an incremental solution algorithm for any problem with a strain-driven nonlinear relation, see Fig. 1. The complete approach to numerically analysing layered media using an artificial neural network as the constitutive driver is shown in Fig. 2. To verify the approach, the obtained results are compared with a direct (single-scale) model.

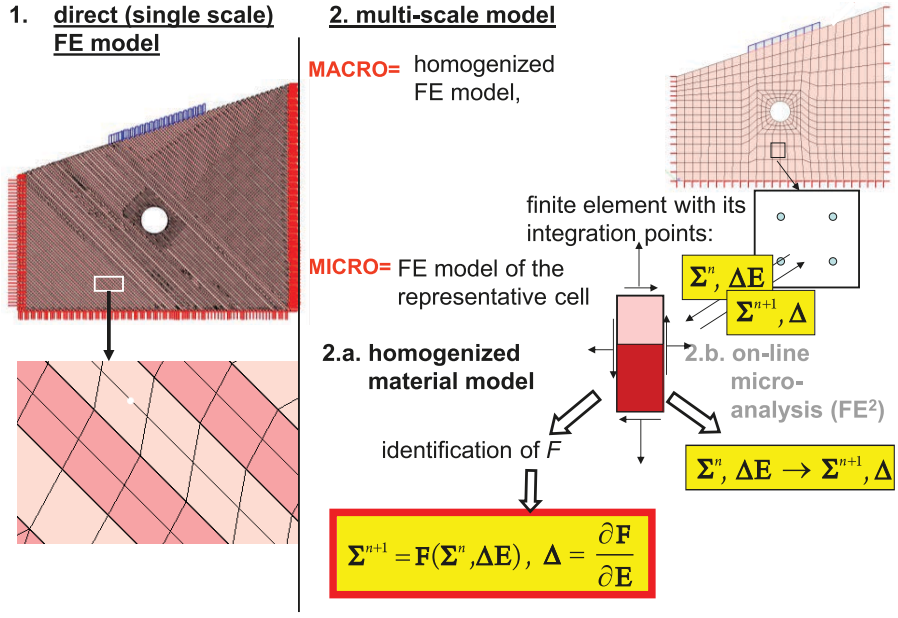

Fig. 1. Strategies of numerical analysis applicable to layered rocks (own elaboration)

https://doi.org/10.37705/TechTrans/e2023007

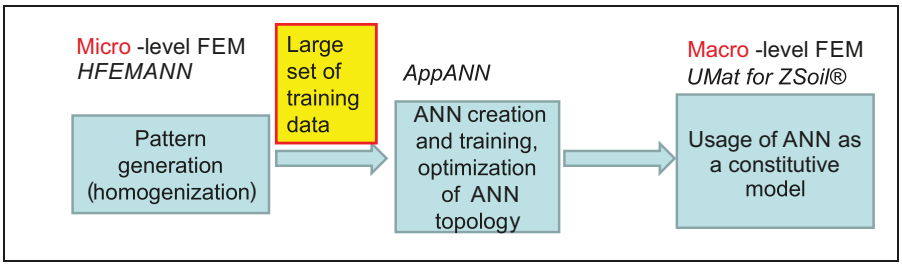

Fig. 2. Application schema and software used in the FE analysis of the layered media (own elaboration)

Nowadays, ANN based computations are increasingly used in geotechnical engineering, including in problems of: slope stability analysis, see Ferentinou (2007), Goh (2003); estimating TBM method performance, see Benardos (2004); bearing capacity of a pile submitted to lateral load, see Das (2006). The constitutive behaviour of homogenous soil can also be simulated using ANN, for example, in the work of Najjar (2007) and Lucon (2007). A thorough review of the ANN application in the field of geotechnics can be found in the work of Shahin (2008) and more recently, Moayedi (2020) and Baghbani (2022). Our goal is to show how an ANN-based approach can be used to create a constitutive model of layered rocks as an example of composite material with nonlinear materials of components, in a fully automated way, without any assumption of its form. This follows along the general path of proceeding shown in the works of (Waszczyszyn, 1999; Lefik, 2002; Ghaboussi, 1998; Hashash, 2004).

2. Homogenisation technique of layered media

According to the general formulation of the homogenisation problem, the total strain $\boldsymbol{\varepsilon}$ is the sum of macro-level averaged strains $\mathbf{E}(\mathbf{X})$ and the fluctuating, periodic strains $\boldsymbol{\varepsilon}^p(\mathbf{x})$. The same split relates to displacements.

$$\begin{aligned} \boldsymbol{\varepsilon}(\mathbf{X},\mathbf{x}) &= \mathbf{E}(\mathbf{X}) + \boldsymbol{\varepsilon}^p(\mathbf{x}), \\ \mathbf{u}(\mathbf{X},\mathbf{x}) &= \mathbf{U}(\mathbf{X}) + \mathbf{E}(\mathbf{X}) : \mathbf{x} + \mathbf{u}^p(\mathbf{x}) \end{aligned} \tag{1}$$

In the case of a layered (uni-periodic) medium, the microstructure geometry resulting from parallel material layers is homogeneous in two directions, namely (y,z), while it is periodic in the third direction (x), normal to the layers direction. A periodic cell has a finite length in the direction x, named p_X while it degenerates in the two others. This means that the dimensions of the periodicity cell in y and z tend to 0, $p_Y \to 0$, $p_Z \to 0$. The periodic cell is a 1D line segment. Periodicity conditions imposed on the all displacement vector components at two pairs of parallel surfaces result with the conditions:

$$\left.\begin{aligned} \mathbf{u}^p(y) = \mathbf{u}^p(y+p_Y),\ p_Y = \Delta y \to 0 \Rightarrow \frac{\partial \mathbf{u}^p}{\partial y} = 0 \\ \mathbf{u}^p(z) = \mathbf{u}^p(z+p_Z),\ p_Z = \Delta z \to 0 \Rightarrow \frac{\partial \mathbf{u}^p}{\partial z} = 0 \end{aligned}\right\} \Rightarrow \mathbf{u}^p(\mathbf{x}) = \{u(x), v(x), w(x)\}^T; \tag{2}$$

The displacement vector of the periodic perturbation $\mathbf{u}^p$ then has all three nonzero components, but they are functions of only one spatial coordinate x, it is in a direction perpendicular to the stratification. The periodicity condition has to be imposed at the opposite ends of the segment on the displacement field in the form: $\mathbf{u}^p(x^+) = \mathbf{u}^p(x^-)$. Figure 3 shows the setup of a 1D model of the periodic cell for analysis of the layered medium.

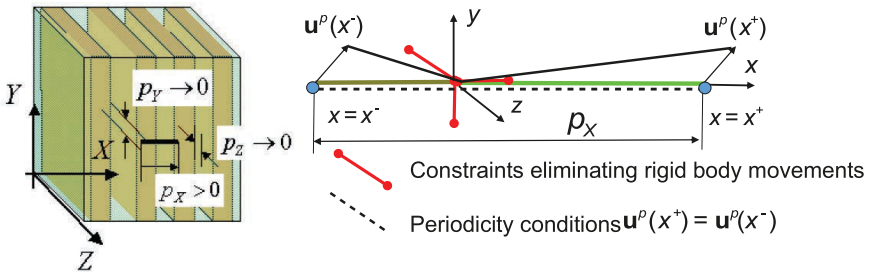

Fig. 3. Setup of the 1D model of the periodic cell of the layered medium (own elaboration)

In the strain field of the periodic perturbation, the components $\varepsilon^p_{zz} = \varepsilon^p_{yy} = \varepsilon^p_{yz} = 0$, thus the total strains in these directions are directly equal to the strains of the macro-model. $\varepsilon_{yy} = E_{YY}, \varepsilon_{zz} = E_{ZZ}, \varepsilon_{yz} = E_{YZ}$. The relationship between the strains and displacements that define the form of the operator B^{1D} takes the form:

$$\begin{Bmatrix} \varepsilon^p_{xx} \\ \varepsilon^p_{yy} \\ \varepsilon^p_{zz} \\ \varepsilon^p_{xy} \\ \varepsilon^p_{xz} \\ \varepsilon^p_{yz} \end{Bmatrix} = \begin{bmatrix} \partial_x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2}\partial_x & 0 \\ 0 & 0 & \frac{1}{2}\partial_x \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u(x) \\ v(x) \\ w(x) \end{Bmatrix} \Leftrightarrow \boldsymbol{\varepsilon}^p = \mathrm{B}^{1D}\mathbf{u}^p \tag{3}$$

Equilibrium conditions in a weak form leads to equation (4):

$$\left(\mathrm{B}^{1D}\right)^T \boldsymbol{\sigma}\left(\mathbf{E} + \mathrm{B}^{1D}\mathbf{u}^p\right) = \mathbf{0} \tag{4}$$

The solution of Eq. (4) is established formally by the finite elements method, see (Urbański, 2005), which in this case leads to a very simple model. The number of 1D finite elements with linear shape functions should be equal to the number of layers in the periodic cell, leading, in the case of two layers, to a very small, 3-node FE model of the cell, with only three degrees of freedom. This is the case firstly because of the periodicity condition and secondly because in order to avoid singularity of the finite element equations system, a set of three kinematical constraints eliminating rigid body movements have to be introduced to the model, fixing translations at the internal node. After solving it, obviously with an iterative procedure which must be run when any layer has a nonlinear constitutive model, micro-level strains and in consequence also stresses which are a piecewise constant within each layer are set. The average stresses $\boldsymbol{\Sigma}$ are evaluated as integrals along the line divided by the cell volume which is equivalent to its length.

$$\boldsymbol{\Sigma} = \frac{1}{p_X}\int_{x^-}^{x^+} \boldsymbol{\sigma}\, dx \tag{5}$$

The method outlined above provides an exact solution for the homogenisation problem for any layered media. When strain-controlled homogenisation is used as a tool to create a macro-constitutive model then the increment of macro-strains $\Delta\mathbf{E}$ is imposed, and the response of the medium expressed in macro-stresses $\boldsymbol{\Sigma}$ create the required path in the $(\mathbf{E},\boldsymbol{\Sigma})$ space. Obviously, in the case of material nonlinearity, for each path, the homogenisation procedure must be run in incremental way in *n* steps until a given strain range is achieved:

https://doi.org/10.37705/TechTrans/e2023007

$$i=0,\boldsymbol{\Sigma}^i=\mathbf{0}$$

$$\begin{cases}\mathbf{X}^i\equiv(\boldsymbol{\Sigma}^i,\Delta\mathbf{E})\rightarrow\boldsymbol{\Sigma}^{i+1}\equiv\mathbf{Y}^i\\ \text{if } i<n: i=i+1 \text{ else: } stop\end{cases} \tag{6}$$

3. Data preparation for model identification

The fundamental problem when dealing with the problem of preparing data for ANN-based identification of the model is to organise a large number of separate runs of the homogenisation procedure, covering the whole expected states of stresses and strain increments. This creates a domain in 12-dimensional space. In the case of layered media, this task is relatively simple because homogenisation is performed on a very small FE model, and the problem possesses a level of symmetry which allows the reduction of this effort. This was accomplished in the following way:

1. The increment ΔE_{XX} was chosen covering the whole range of expected strains for compression or tension in a direction normal to the layers.
2. For each of the above cases, a reduced space of strain increments is considered. They are set in subsidiary 3D space as shown in Fig. 4. The vector $\Delta\mathbf{E}'=\{\Delta E_{XX},\Delta E_{Y'Y'},\Delta E_{Y'Y'},\Delta E_{XY'},0,\Delta E_{Y'Z'}\}^T$ is used as an increment of control data of the homogenisation procedure which is run across the full range of the expected strains. This enables the user to keep control of the data preparation procedure in an easier way.
3. For obtained points in stresses and strain increments, space transformation due to rotation around the X axis is performed (in the example $\Delta\alpha$=15°, thus it is done twenty-three times). In this way, axial symmetry with respect to the X axis and the tensorial character of data is preserved and full 3D tensors consisting of input $\boldsymbol{X}$ and output $\boldsymbol{Y}$ vectors are used later to train the ANN.

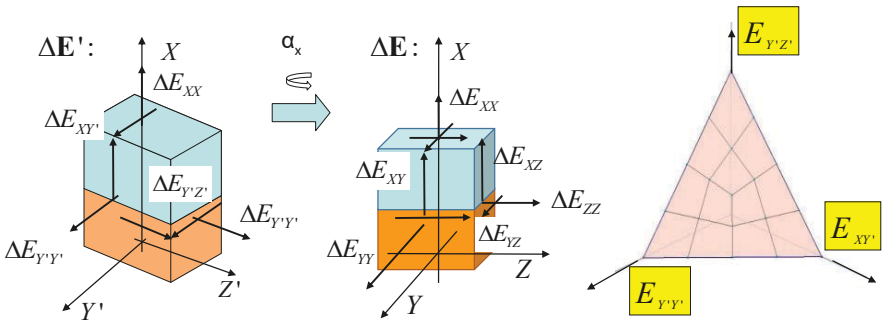

Fig. 4. The idea of pattern generation to train the ANN (own elaboration)

In the procedure described above, the number of $\boldsymbol{X},\boldsymbol{Y}$ patterns obtained from one path is equal to the number N of stress-strain points in it, $\{\mathbf{E}(\lambda_i),\boldsymbol{\Sigma}(\lambda_i)\},i=0,N$, with $\mathbf{E}(\lambda_i)=\widehat{\mathbf{E}}\cdot f(\lambda_i)$ where f is the control function of load factor λ, leading to:

$$\mathbf{X}=\{\boldsymbol{\Sigma}^{i-1},\Delta\mathbf{E}^i=\mathbf{E}(\lambda_i)-\mathbf{E}(\lambda_{i-1})\},\mathbf{Y}=\{\boldsymbol{\Sigma}^i\},i=1,N \tag{7}$$

The number of patterns might substantially increase, up to $Nmax=N(N\text{-}1)/2$, when for any given stress point, a larger set of strain increments and the corresponding stresses are included:

$$\mathbf{X}=\{\boldsymbol{\Sigma}^{i-1},\Delta\mathbf{E}_j^i=\mathbf{E}(\lambda_j)-\mathbf{E}(\lambda_{i-1})\},\mathbf{Y}=\{\boldsymbol{\Sigma}^j\},i=1,N,\ j=i+1,N. \tag{8}$$

This procedure, called “diluting”, is shown in Fig. 5. In addition to increasing the number of patterns, it widens the range of strain increments. It may only

be performed for smooth paths and with controlled intensity. This is essential when using a trained ANN as the constitutive function in a boundary value problem; however, it does not require any additional computational effort. In the presented example, the procedure of diluting is performed.

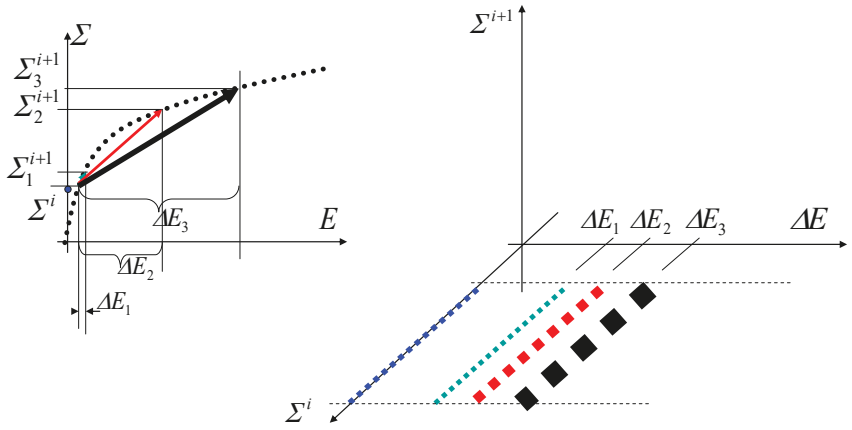

Fig. 5. Increasing the number of patterns and widening the range of strain increments by the diluting procedure (own elaboration)

Another method of enhancement was to randomise the input values of E_{mn} by introducing a randomised control function, instead of steadily increasing one, see Fig. 6 for details.

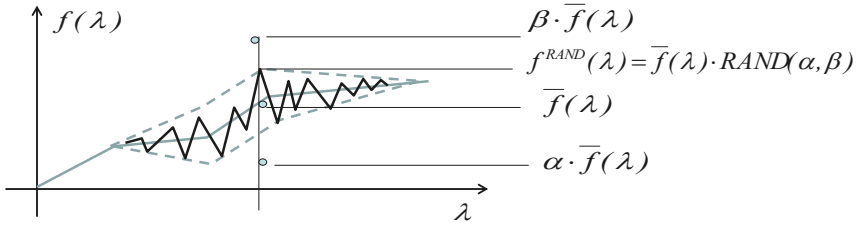

Fig. 6. Randomised control function (own elaboration)

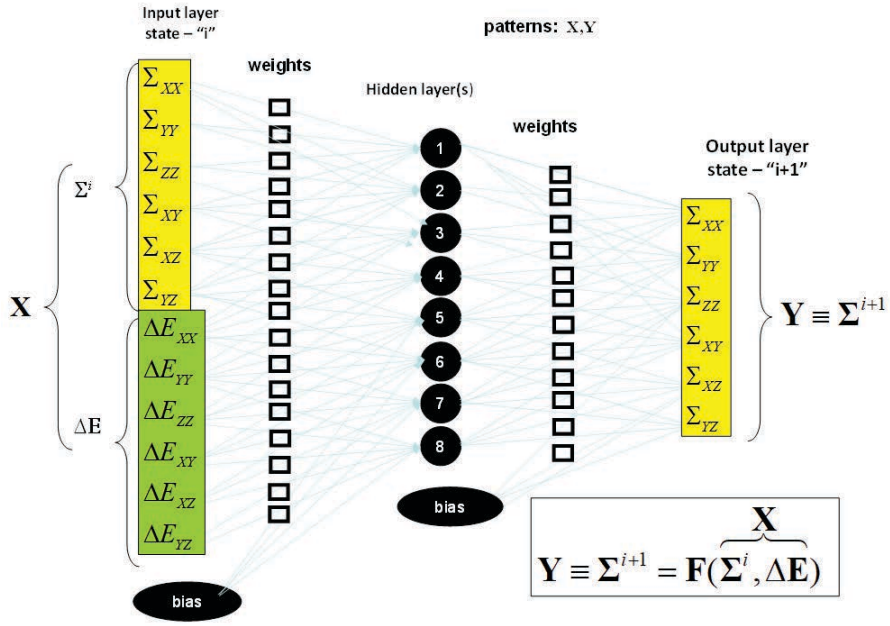

Fig. 7. Multilayer perceptron-type ANN as a constitutive function (own elaboration)

https://doi.org/10.37705/TechTrans/e2023007

A single pattern record for training ANN and its schema is shown in Fig. 7. In practice, a large number of such records is needed, for example, the number of records finally used to train the ANN from the example described in Section 6 was about 200,000.

4. ANN-based creation of the constitutive model

The multi-layered, unidirectional perceptron neural net (shown schematically in Fig. 7) is a nonlinear function $\mathbf{\Phi}$: $\mathbf{X} \in R^N \rightarrow \mathbf{Y} \in R^M$, defined recursively as:

$$\begin{aligned}
&\mathbf{\Phi}: R^N \ni \mathbf{X} = \{x_i\} = \{y_i^{(0)}\}, \quad i = 1, n^{(0)}, \quad n^{(0)} = N \\
&\ldots. \\
&y_i^{(k)} = \alpha^{(k)} \left(\sum_{j=1, n^{(k-1)}} w_j^{(k-1)} y_j^{(k-1)} + b^{(k-1)} \right), \quad k = 1, L \\
&\ldots.. \\
&\{y_i^L\} = \mathbf{Y} \in R^M, \quad i = 1, n^L, \quad n^L = M.
\end{aligned} \tag{9}$$

In Eq. (9), $L > 1$ is the number of layers, and $n^{(k)}$ is the number of neurons in the k-th layer. These parameters define the topology of the artificial neural net. $w_j^{(k)}$ is the coefficients of the linear combination named as the weights, and $b^{(k)}$ is the bias, which is a term that shifts values of linear combinations. $\alpha^{(k)}$ is the activation function, which is generally nonlinear. The activation functions available in our implementation are presented in Table 1.

Table 1. Activation functions available in AppANN

Name:	LINEAR	SIGMOID	TANH	SOFTPLUS	SOFTSIGN	RELU
$\alpha(x)=$	x	$\frac{1}{1+e^{-x}}$	$\tanh(x)$	$\ln\left(1+e^{x}\right)$	$\frac{x}{1+\lvert x\rvert}$	$\begin{matrix}0, x<0\\ x, x\geq 0\end{matrix}$

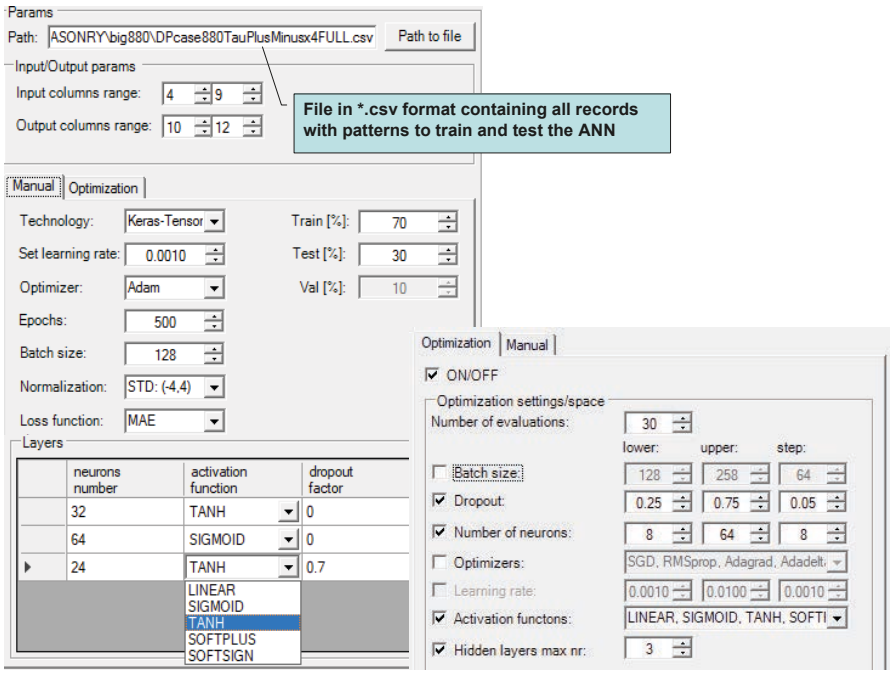

Fig. 8. Dialogues enabling the user to set hyper-parameters in manual and optimisation mode (source: own elaboration)

For an assumed set of topological parameters L and $n^{(k)}$, the values of $w_j^{(k)}$ and $b^{(k)}$ are the results of using an objective loss function to minimise error. The theoretical background for this is provided by the mathematical theory of artificial neural networks, with the universal approximation theorem (Hertz 1991, Bishop 2006). It has been proven that a multi-layered, unidirectional perceptron neural network is capable of approximating an arbitrary multi-dimensional nonlinear function e.g. $\mathbf{\Phi}: \mathbf{X} \in R^n \rightarrow \mathbf{Y} \in R^m$.

Using the original software AppANN of IdeaLogic Ltd., the building of multi-layered perceptron architecture could be easily achieved. In the subsection "Layers", the user can determine the number of layers and the corresponding parameters for each layer, such as the number of neurons, the activation function and the dropout factor. The user interface of AppANN is shown in Fig. 8.

Learning rate is a parameter that controls the adjustment of the weights of a neural network with respect to the loss gradient. During the training process, the loss is calculated for inputs and the gradient of that loss is calculated with respect to the model's weights. The learning rate is calculated after obtaining values of the gradients – the gradients get multiplied by the learning rate. The learning rate affects how huge the update is of the step to move towards the minimum point in the loss function. The lower the value, the slower we travel along the downward slope. If the step is too big, the minimum point on the loss function could be missed. If the step is too small, it might take too long to converge to the minimum point. More information can be found in the work of Smith (2017).

Batch size is the number of samples that are passed to the network at once. This is one of the most important hyperparameters to tune in the training of neural networks. The larger batch size allows computational speedups from the parallelism. Too large a batch size will lead to poor generalisation, see the work of Keskar (2017).

Standardisation is an important step in data pre-factoring. In practical applications, the data set contains a variety of features, often with different distributions and intervals, with different levels (dimension), which easily affect our model training. Data standardisation removes the effects of scale, feature, and distribution differences in the model.

Standardisation means subtracting the mean value of all features from each feature and dividing the result by the standard deviation σ according to the standardisation equation:

$$x' = \frac{x - \sigma}{\sigma} \tag{10}$$

Normalisation is the process of reducing data values to specific ranges in order to be able to compare them. Most often, they are normalised to ranges: <0:1> or <-1:1> with the minmax technique:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{11}$$

Optimiser – the user can choose between the following optimisers: stochastic gradient descent (SGD), adaptive moment estimation (Adam), root mean square propagation (RMSprop), adaptive gradient (Adagrad), and adaptive delta (Adadelta). For their descriptions, see (Smith 2017).

Number of epoch specifies complete passes of the entire training dataset passing through the training or learning process and it decides how many times the change in the weights of the network will happen.

Loss function. The available functions are means squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and coefficient of determination (R2). Exact formulae can be found in numerous references, e.g. Bishop (2006), Raschka (2017) and Szeliga (2017).

https://doi.org/10.37705/TechTrans/e2023007

Dropout factor is one of most efficient regularisation techniques to prevent overfitting. It is the percentage of randomly deactivated neurons during training in each training iteration. Some neurons tend to be too strong and when they are deactivated, the neural network is forced to find a different decision path and activate neurons which have previously not been used so much.

Train/test defines the proportion of data used for training and for testing.

The neural network training uses the Keras package with TensorFlow in the back-end. For the purposes of tuning the neural network hyperparameters, the python library Hyperopt has been implemented in software. It uses the tree of Parzen estimators (TPE) and adaptive TPE techniques to find the optimal set of parameters. The user can define the space of the optimised parameters by setting the lower range limit, the upper range limit, the step (for numerical parameters), and by the specification of other items. The algorithms of the above are given in the literature (Bergstra 2011, Bergstra 2013).

5. Implementation issues concerning ANN used as a constitutive model

The computations were run within the FE system ZSoil.PC®, which contains the option of a user material model (UMat). A piece of UMat code shown in Fig. 9 utilises the universal routine RunModel (of IdeaLogic) for reading ANN in a form $\mathbf{X} \rightarrow \mathbf{Y}$ written in C++ .

```
ANN_ToZS( iModel,…,sOld,dE,sNew) //  C++ called from Fortran code
……..
double X[12]; //  temporary vector for stresses and increment of strains
for (int i = 0; i < 6; i++)
        {
        X[i] = sOld[i]; //setting old stress
        X[i + 6] = dE[i]; //setting strain increment
        }
ANNModels[iModel]->RunModel (12,6,X, sNew); //evaluate new stress
```

Fig. 9. Routine used to read ANN within UMat code (source own elaboration)

Other required parts of code (transformation local-global of stresses, strains and stiffness, Gauss point storage handling and updating) are also contained in the ZSoil® UMat. The evaluation of material stiffness Δ for monotropic material as an inverse of compliance matrix is given in Eq. (12). Elastic constants $E_{vv}, E_{hh}, G_v, G_h, \nu_{vh}, \nu_{hh}$ are based on (Vlasov 1990), where they are given in a closed form for a two-layer composite material.

$$\Delta = \Xi^{-1}, \quad \Xi = \begin{bmatrix} \frac{1}{E_{vv}} & -\frac{\nu_{vh}}{E_v} & -\frac{\nu_{vh}}{E_v} & 0 & 0 & 0 \\ & \frac{1}{E_{hh}} & -\frac{\nu_{hh}}{E_v} & 0 & 0 & 0 \\ & & \frac{1}{E_{hh}} & 0 & 0 & 0 \\ & & & \frac{1}{G_v} & 0 & 0 \\ & symm & & & \frac{1}{G_v} & 0 \\ & & & & & \frac{1}{G_h} \end{bmatrix} \tag{12}$$

Although the described method of proceeding with the constitutive matrix of the model has a limited choice of nonlinear problem-solution algorithms to the initial stiffness algorithm only, it is more convenient than the alternative. Note, however, that all material model nonlinearities are reproduced by the ANN function.

6. Comparison between direct and ANN-based homogenised model

As an example of layered media following a two-layer composite was considered. In both layers of equal thickness, material behaviour is assumed to be perfectly elastic-plastic with a Drucker-Prager yield surface. The Drucker-Prager plasticity model was chosen among the others available mainly because of its simplicity. Note that the main goal of this paper was to investigate the effectiveness of the ANN approach with regard to the constitutive modelling of any layered media with the nonlinear behaviour of each component but not an exact reproduction of a given layered material. Model data are given in Table 2.

Table 2. Drucker-Prager elastic-plastic model data for the components

Material layer	Young modulus E_i [GPa]	Poisson ratio ν_i [-]	Friction angle φ_i [°]	Dilatancy angle ψ_i [°]	Cohesion c_i [MPa]
1. Slate	4.0	0.17	32	16	2
2. Sandstone	11.9	0.20	39	39	16

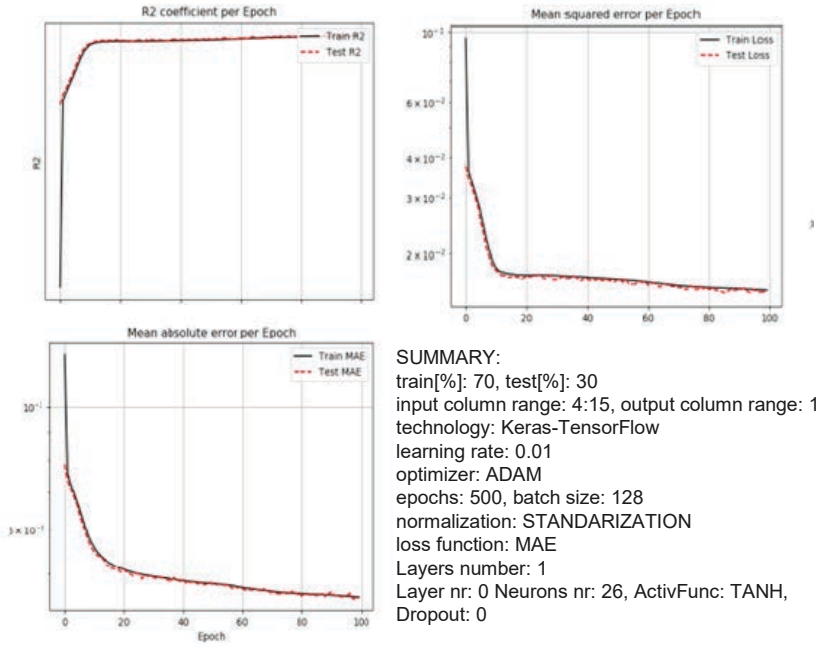

Fig. 10. Error measures obtained during the training procedure and its summary (source own elaboration)

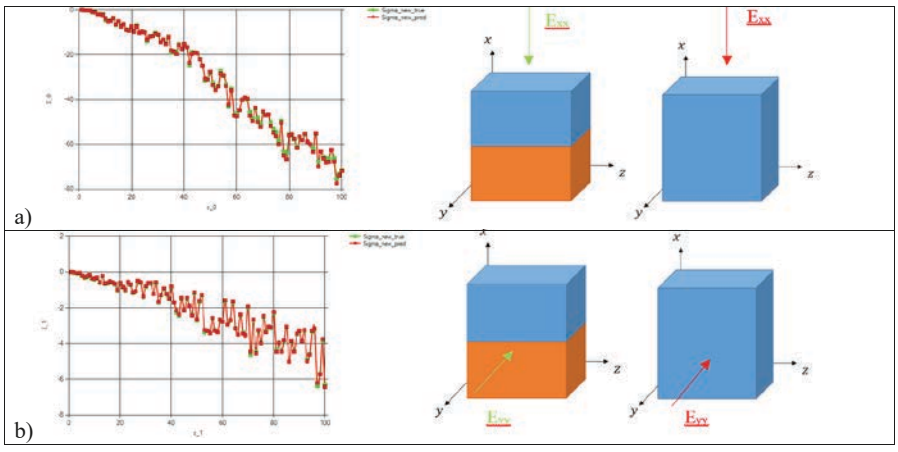

Fig. 11. Comparison of stress-strain path (red-ANN, green-homogenisation) for: a) compression in a direction normal to the layers, b) compression in the direction of the layers (own elaboration)

https://doi.org/10.37705/TechTrans/e2023007

The range of strains used for the creation of patterns was (0-0.002) for tension, (-0.01-0) for compression in both the XX and YY directions and (0-0.005) for shear. The number of increments reached N=100, giving 181,600 records. Then, the procedure of diluting, described in Section 3, and randomisation was additionally used. On such a pattern set, training of ANN was performed. The split of the data between the test subset (70%) and the validation subset (30%) was accomplished. Searching for the optimal configuration of ANN was performed, yielding: number of hidden layers (=1), number of neurons at each layer (=26), the type of activation functions (tanh). Different error measure graphs are shown in Fig. 10.

Comparison of stress-strain path in the composite (XX and YY components) in full a range of compression ($Ninc$=100) for randomly induced strains, for both the homogenisation procedure and the created ANN model is shown in Fig. 11. In the authors opinion, this results show a satisfactory accordance.

The simplest way to verify the results of the ANN-based approach in the analysis of the layered media is to compare it with the results of the single-scale, direct analysis of the same object. This reference object is shown in the Fig. 12. This is a model of a slope with tunnel openings and its lining modelled with shell elements. This is in fact a plane-strain model but modelled with one layer of 3D brick elements in the Z direction. In the XY plane, in this example, FE meshing corresponds to the spreading of material layers of equal thickness.

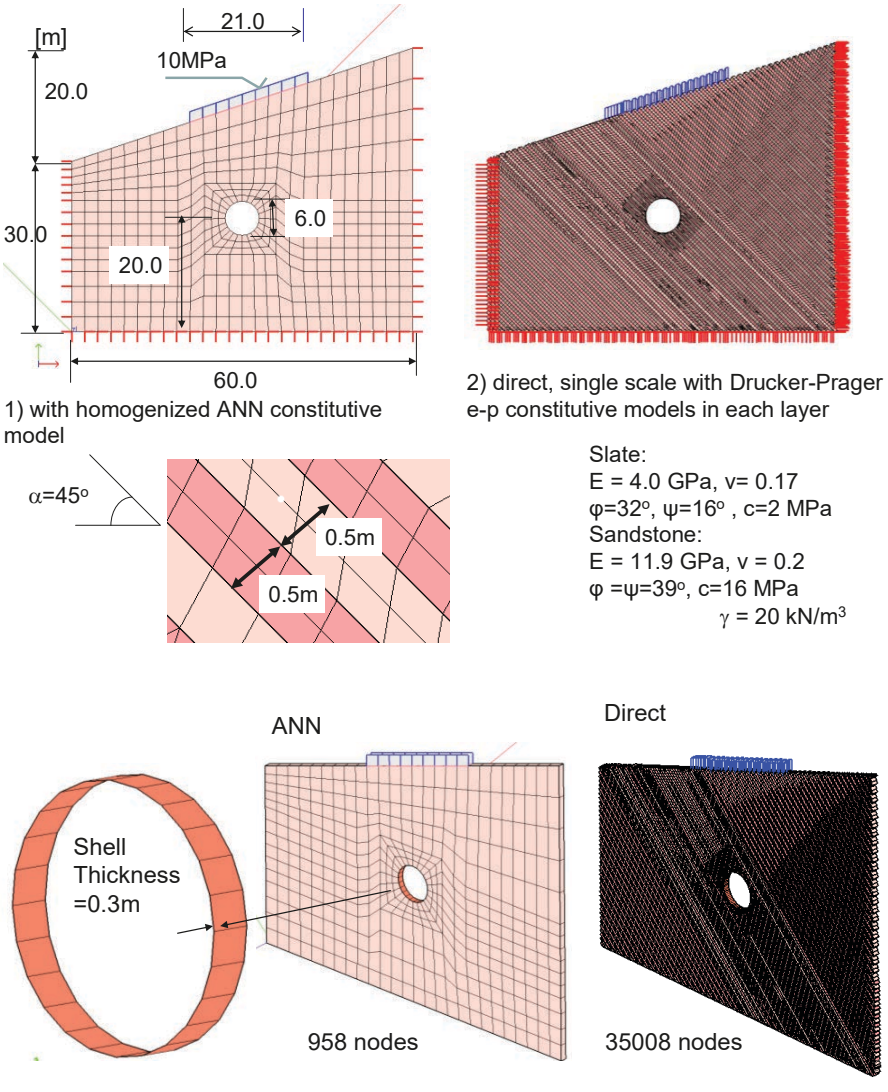

Fig. 12. ANN based model versus referential direct (single scale). Outlook and details (own elaboration)

For both components, the gravity is the same, $\gamma = 20$ kN/m^3. Obviously, identical plasticity model data and layer thicknesses were used for the analysis of the periodic cell.

The referential direct model possesses 38,000 nodes while the ANN-based model has only 980 nodes. The comparison of the results for the first deformation and the plastic zones is presented in Fig. 13. This is for the direct model only, as the ANN-based model cannot produce such a result. In Figs. 14, 15 and 16 stresses and strains are shown, both in frame rotated (*X* axis-perpendicular to the layers). Additionally, the internal force graphs in shell tunnel lining are given in Fig.17.

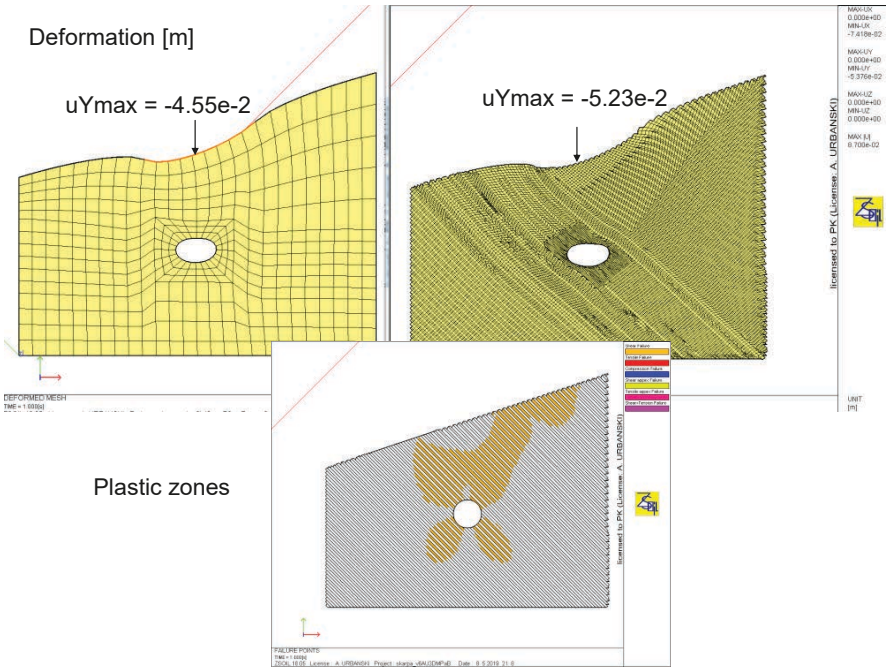

Fig. 13. Deformation for ANN-based and referential (direct) object. Plasticity zones (own elaboration)

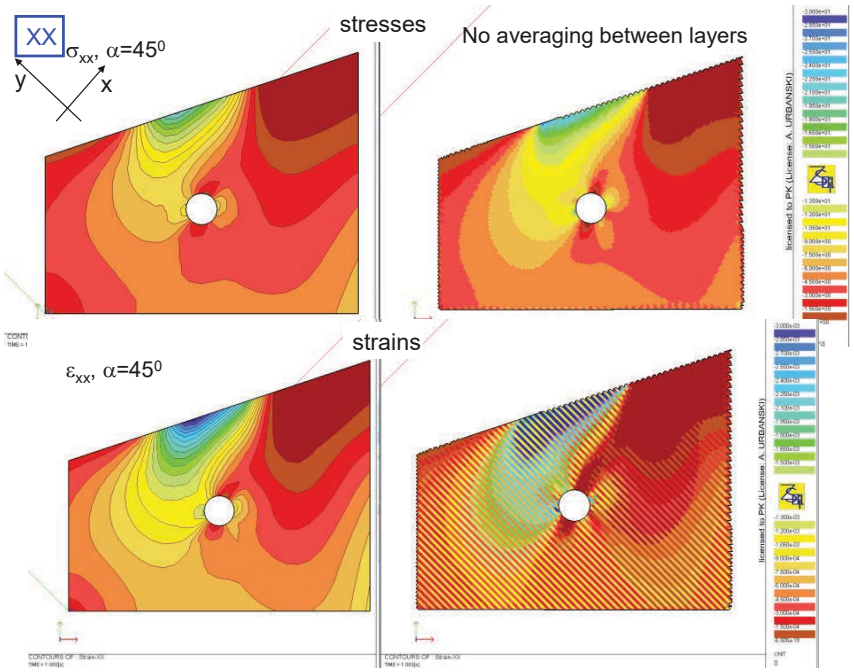

Fig. 14. Stresses and strains for ANN-based and referential (direct) object component XX (normal to the layers) (own elaboration)

https://doi.org/10.37705/TechTrans/e2023007

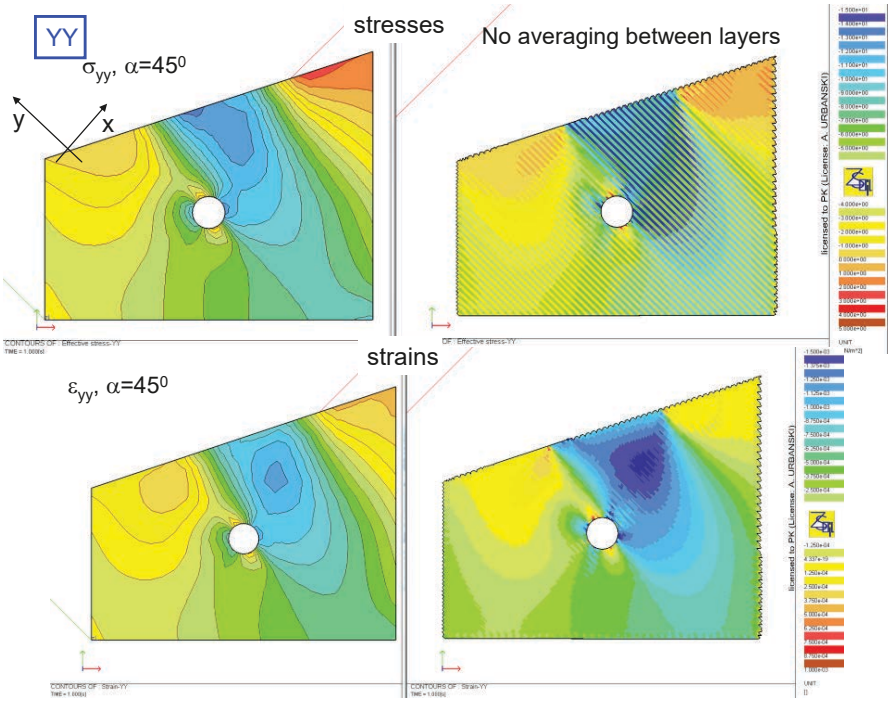

Fig. 15. Stresses and strains for ANN-based and referential (direct) object component YY (parallel to the layers) (own elaboration)

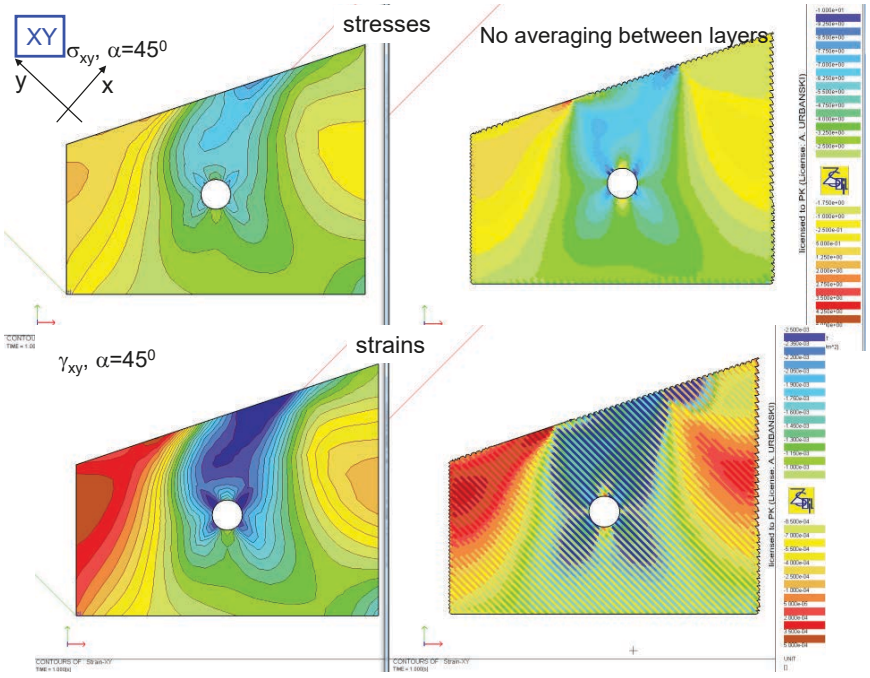

Fig. 16. Stresses and strains for ANN-based and referential (direct) object component XY (shear) (own elaboration)

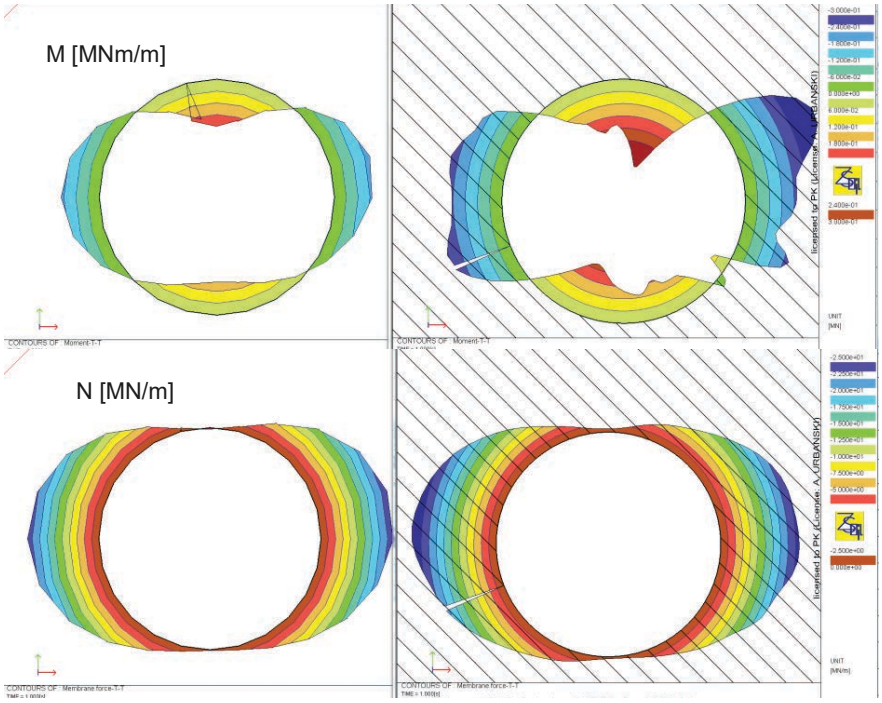

Fig. 17. Internal forces for shell lining for ANN-based and referential (direct) object (own elaboration)

7. Conclusions

During the research on the applicability of an artificial neural network based on constitutive modelling within a finite element analysis, particularly in the context of layered rocks, a few issues should be considered. Firstly, large pattern sets must be created on which the artificial neural networks will be trained. The range of these sets should cover the range of stress points and strain increments expected to be encountered during the intended finite element analysis. When this postulate is not fulfilled, the correct results can hardly be expected.

The "diluting" procedure, described in Section 3, may be seen as a remedy, at least when monotonic loads are considered as was the case in the assessed examples. However, applying a randomised path may be helpful to make the ANN model usable in cases when some loading-unloading action takes place.

Because of the developed AppANN application described in Section 4, the user does not need sophisticated knowledge about artificial neural network algorithms and their optimisation. Our final conclusion is that the artificial neural network-based constitutive models can be used in FE modelling; however, more research is needed to make them a "black box tool" for general use during finite element analysis.

References

Baghbani, A., Choudhury, T., Costa, S., Reiner, J. (2022). Application of artificial intelligence in geotechnical engineering: A state-of-the-art review. *Earth-Science Reviews*, Vol. 228, https://doi.org/10.1016/j.earscirev.2022.103991

Benardos, A., Kaliampakos, D. (2004). Modelling TBM performance with artificial neural networks. *Tunnelling and Underground Space Technology 19*(6): 597–605, http://doi.org/10.1016/j.tust.2004.02.128

Bergstra, J., Bardenet, R., Bengio, Y., Kegl, B. (2011). Algorithms for Hyper-Parameter Optimization. *Advances in Neural Information Processing Systems* 24: 2546–2554.

Bergstra, J., Yamins, D., Cox, D.D. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *ICML'13: Proceedings of the 30th International Conference on International Conference on Machine Learning*, Vol. 28: I-115–I-123.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning.* New York: Springer.

Das, S.K., Basudhar, P. (2006). Undrained lateral load capacity of piles in clay using artificial neural network. *Computers and Geotechnics 33*(8): 454–459, http://doi.org/10.1016/j.compgeo.2006.08.006

Ferentinou, M., Sakellariou, M. (2007). Computational Intelligence tools for the prediction of slope performance. *Computers and Geotechnics 34*(5): 362–384, http://doi.org/10.1016/j.compgeo.2007.06.004, 2007

Feyel, F. (1999). Multiscale FE2 elastoviscoplastic analysis of composite structures. *Computational Materials Science*, Vol. 16, Issues 1–4: 344–354.

Ghaboussi, J., Pecknold, D.A., Zhang, M., HajAli, R.M. (1998). Autoprogressive training of neural network constitutive models. *Int. J. Numer. Meth. Engng.*, 42: 105–126.

Giuntoli, G., Aguilar, J., Vázquez, M., Oller, S., Houzeaux, G. (2019). A FE 2 multi-scale implementation for modelling composite materials on distributed architectures. *Coupled Systems Mechanics*, Vol. 8: 99–109.

Goh, A., Kulhawy, F.H. (2003). Neural network approach to model the limit state surface for reliability analysis. *Canadian Geotechnical Journal 40*(6): 1235–1244, http://doi.org/10.1139/t03-056

Hashash, Y.M.A., Jung, S., Ghaboussi, J. (2004). Numerical implementation of a neural network based material model in finite element analysis. *International Journal for Numerical Methods in Engineering 59*(7): 989–1005.

Hertz, J., Krogh, A., Palmer, R.G. (1991). *Introduction to the theory of neural computation.* Redwood City, Calif: Addison-Wesley Pub.

Keskar, N.S., Nocedal, J., Tang, P.T.P., Mudigere, D., Smelyanskiy, M. (2017). *On large-batch training for deep learning: Generalization gap and sharp minima*. 5th International Conference on Learning Representations, ICLR 2017, Toulon, France.

Lefik, M. (2002). *Artificial neural network for modelling an effective behavior of composite materials.* Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V), Vienna, Austria, July 7–12. Austria: Vienna University of Technology.

Lucon, P.A., Donovan, R.P. (2007). An artificial neural network approach to multiphase continua constitutive modelling. *Composites*, Vol. 38.

Moayedi, H., Mosallanezhad, M., Rashid, A.S.A. (2020). A systematic review and meta-analysis of artificial neural network application in geotechnical engineering: theory and applications. *Neural Comput. & Applic. 32*: 495–518, https://doi.org/10.1007/s00521-019-04109-9

Najjar, Y.M., Huang, C. (2007). Simulating the stress–strain behavior of Georgia kaolin via recurrent neuronet approach. *Computers and Geotechnics 34*(5): 346–361, http://doi.org/10.1016/j.compgeo.2007.06.006

Raschka, S. (2017). *Python Machine Learning* (in Polish: *Python. Uczenie maszynowe*). Gliwice: Helion.

Shahin, M., Jaksa, M.B., Maier, H.R. (2008). State of the Art of Artificial Neural Networks in Geotechnical Engineering. *Electronic Journal of Geotechnical Engineering*, *8*: 1–26.

Smith, L.N. (2017). Cyclical Learning Rates for Training Neural Networks, *IEEE Winter Conference on Applications of Computer Vision (WACV)*: 464–472, http://doi.org/10.1109/WACV.2017.58

Szeliga, M. (2017). *Data Science and Machine Learning.* (in Polish: *Data Science i uczenie maszynowe*). Gliwice: Helion.

Urbański, A. (2005). *The unified, finite element formulation of homogenization of structural members with a periodic microstructure.* Monograph no. 320. Kraków: Cracow University of Technology.

Vlasov, A.N., Merzlakov, W.P., Uzow, B.S. (1990). *Effective characteristics of deformational properties of layered media* (in Russian: *Effektivnyje harakteristiki deformacionnyh svojstv sloistyh porod*, Osnowanija, fundamenty i mehanika gruntov).
Waszczyszyn, Z. (1999). *Neural Networks in the Analysis and Design of Structures.* CISM Courses and Lectures No. 404. Wien–New York: Springer.
ZSoil.PC. User Manual. https://zsoil.com/zsoil/tutorials

https://doi.org/10.37705/TechTrans/e2023007