

## On the constraints and consistency in implicit constitutive modelling using ANNs and indirect training

LOURENÇO Rúben<sup>1,2,a</sup>, CUETO Elías<sup>3,b</sup>, GEORGIEVA Pétia<sup>4,c</sup>  
and ANDRADE-CAMPOS António<sup>1,2,d\*</sup>

<sup>1</sup> Dept. of Mechanical Engineering, Centre for Mechanical Technology and Automation, University of Aveiro, Portugal

<sup>2</sup> LASI, Intelligent Systems Associate Laboratory, Guimarães, Portugal

<sup>3</sup> Aragon Institute of Engineering Research (I3A), University of Zaragoza, Spain

<sup>4</sup> Institute of Electronics and Informatics Engineering of Aveiro (IEETA); Dept. of Electronics, Telecommunications and Informatics, University of Aveiro, Portugal

<sup>a</sup>rubenl@ua.pt, <sup>b</sup>ecueto@unizar.es, <sup>c</sup>petia@ua.pt, <sup>d</sup>gilac@ua.pt

**Keywords:** Constitutive Model, Elastoplasticity, Neural Networks, Indirect Training, Constrained Optimization

**Abstract.** The training of an Artificial Neural Network (ANN) for implicit constitutive modelling mostly relies on labelled data pairs, however, some variables cannot be physically measured in real experiments. As such, the training should preferably be carried out indirectly, making use of experimentally measurable variables. The unconstrained training of an ANN's parameters often leads to spurious responses that do not comply with the physics of the problem. Applying constraints during training ensures not only the physical meaning of the ANN predictions but also potentially increases the convergence to a global minimum, while improving the model's performance. An ANN material model is trained using a novel indirect approach, where the local and global equilibrium conditions are ensured employing the Virtual Fields Method (VFM). An example of physical constraint is analyzed and applied during the training process.

### Introduction

ANNs are powerful function approximators that can be used to implicitly learn constitutive relations directly from data, without having to postulate a mathematical formulation [1–3]. Several successful applications of ANNs for implicit modelling of material behavior have been reported in the literature (e.g., [4–6] among others). Most of the approaches rely on training the ANNs with paired data, usually stress-strain, from numerically generated datasets. Nevertheless, in a real experiment, certain variables, such as stresses, are not measurable and, therefore, the training should preferably be carried out indirectly making use of experimentally measurable variables only.

Although a standard ANN could be able to learn the constitutive behavior of a material, given enough data, it usually works as a black-box model in which its structure is not easily interpretable and there is no guarantee that its predictions are usable, as they can violate fundamental laws of mechanics and thermodynamics [3–5]. Thus, it is necessary to enforce physics-based constraints when using ANNs for implicit constitutive modelling. The incorporation of this knowledge into the network allows it (i) to learn the structure of the underlying constitutive relations, (ii) reduce its sensitivity to noise and (iii) increase its performance regarding inputs outside the training domain [7]. Physics-based constraints act as a regularization agent for ANNs, reducing the space of admissible solutions and allowing the network to learn with smaller datasets, as it already does not have to learn those relationships from data [6,7]. These constraints can be enforced using

custom ANN architectures [3], model constraints (e.g., weight constraints) [7] or penalty/regularization terms [5,8]. Some examples of constraints are shown in Table 1.

In the present work two single layer perceptron models are used to model the linear elastic response of a virtual material. A novel indirect training methodology employing the sensitivity-based Virtual Field Method (VFM) [17] is used to train both models and study the application of constraints during training. Here, the applied constraint was the positive definiteness of the tangent stiffness, which in the absence of plastic deformation, degenerates into the elastic stiffness matrix **D**.

Table 1. Examples of material model constraints to enforce during ANN training.

	Formulation	Description
1 <sup>st</sup> law of thermodynamics	$\gamma_{loc} = \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} + \theta \dot{\eta} - \dot{\mathbf{E}}$	The work done by stress must either be stored as recoverable internal energy in the solid or dissipated as heat [8,9]
2 <sup>nd</sup> law of thermodynamics	$\gamma_{loc} \geq 0$	For a sample of material subjected to a cycle of deformation, starting and ending with identical strain and internal energy, the total work must be positive or zero [8,9]
Drucker's postulate	$\Delta\sigma_{ij}\Delta\epsilon_{ij} \geq 0$	The work done by the tractions through the displacements is positive or zero [12]
Symmetric positive definiteness of tangent stiffness	$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} = \mathbf{H} > 0$	The tangent stiffness matrix is symmetric positive definite, ensuring that the strain energy is weakly convex [13]
Time consistency	$\lim_{\Delta\boldsymbol{\epsilon} \rightarrow 0} \Delta\boldsymbol{\sigma} = 0$	A consistent material law maps a state of zero strain onto a state of zero stress [13]

The Drucker's stability criterion bears no physical meaning. Not all materials are stable in this sense, however, issues will arise for materials that do not respect it, when used to solve boundary problems [9,17].

### Mechanical Test and Dataset Creation

A heterogeneous test was used to generate synthetic data to train the ANN models. The geometry consists of  $3 \times 3 \text{ mm}^2$  plate with thickness  $t = 0.1 \text{ mm}$ . The initial mesh, geometry and boundary conditions are depicted in Fig. 1. Symmetry boundary conditions are applied to the boundaries at  $x = 0$  and  $y = 0$  and a surface traction is applied to the boundary at  $x = 3 \text{ mm}$ . The traction follows a non-uniform distribution with a single component along the  $x$ -direction, varying linearly in the  $y$ -direction, according to:  $f_x(y) = my + b$ , where  $m$  and  $b$ , respectively, control the slope and intercept of the distribution.

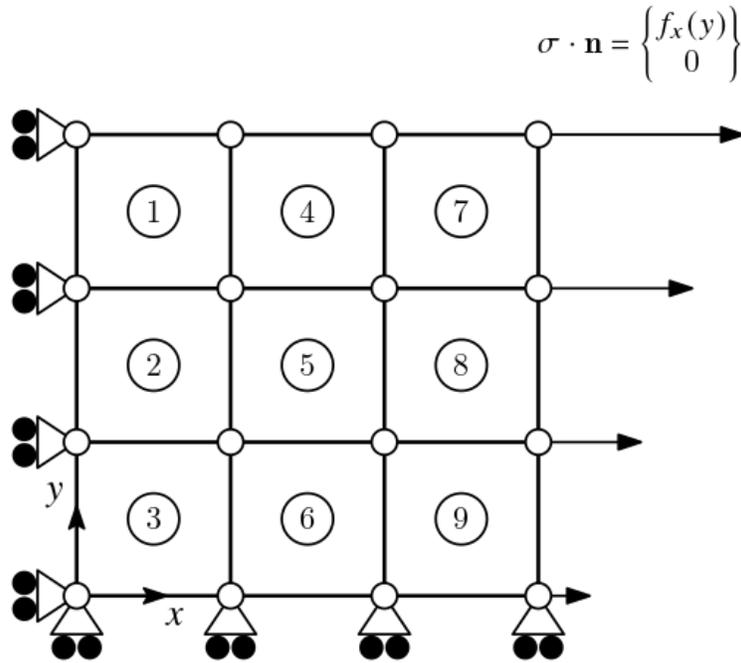


Fig. 1. Heterogeneous test: initial geometry, mesh and boundary conditions, adapted from [14].

The numerical simulations were conducted using the commercial finite element code Abaqus. The model was built with CPS4R elements (bilinear reduced integration plane stress). The elastic parameters were defined as  $E = 210$  GPa and  $\nu = 0.3$ . To generate a training dataset, various simulations were performed with the time period set to 1 and a fixed time increment  $\Delta t = 0.001$ . For each time step, the strain tensor at the centroid was extracted for all the elements and the resultant force computed from the equilibrium of the internal forces, such that:

$$Fl = \sum_{i=1}^n \sigma_i A_i t \tag{1}$$

where  $F$  is the global force,  $l$  the length of the solid,  $A$  is the element's area and  $t$  is the thickness. The training and validation datasets were generated for different load distribution parameters, depicted in Table 2. Prior to training and for each mechanical test, the dataset was organized into batches of 9 elements per time increment and shuffled before being split into training (67%) and test data (33%). The input features were scaled in order to have zero mean and unit variance.

Table 2. Loading parameters for the generation of the training and validation datasets.

Training set	$m = \{60,80,100,120\} [N/mm]$	$b = \{60,80,100\} [N]$
	$m$	$b$
Validation set	50	50
	70	90
	90	65

### Neural Network Model

A single layer perceptron with linear activation was chosen to predict the elastic response. The inputs were the components of the strain tensor at a given time  $t$  and the outputs were the components of the corresponding stress tensor, as depicted in Fig. 2.

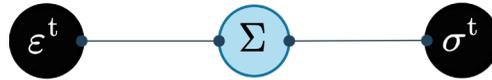


Fig. 2. Single layer perceptron with linear activation.

The model outputs the stress directly according to the following relationship during the forward pass:

$$\boldsymbol{\sigma} = \mathbf{W}\boldsymbol{\epsilon} + \mathbf{b} \Leftrightarrow \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{Bmatrix} + \begin{Bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{Bmatrix} \quad (1)$$

where  $w_{ij}$  are the layer weights and  $b_{ij}$  the biases. An analogy can be established between the terms of the weight matrix  $\mathbf{W}$  and the elasticity matrix  $\mathbf{D}$  in the Hooke's law, for an isotropic material under plane stress:

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon} \Leftrightarrow \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{Bmatrix} \quad (2)$$

with  $\mathbf{D}$  defined in terms of the elastic constants  $E$ , the Young's modulus and  $\nu$ , the Poisson's ratio. For the material chosen for this work, the elasticity matrix  $\mathbf{D}$  is the following:

$$\mathbf{D} = \begin{bmatrix} 230769.231 & 69230.769 & 0 \\ 69230.769 & 230769.231 & 0 \\ 0 & 0 & 80769.231 \end{bmatrix} \quad (3)$$

The architecture presented above was used to train two models in order to learn the elastic response of the material. One of the models was trained following an unconstrained optimization approach, while the other model was trained using a constrained optimization, in order to study how adding constraints would influence the resulting stress predictions.

### Indirect Training for Linear Model

Implicit constitutive modelling using ANNs relies on paired data, usually strain and stress tensors, in order to learn the material behavior. However, variables such as stress cannot be obtained from experiments [15]. Therefore, the training must be carried out indirectly, using only measurable data. The VFM, first introduced by Grédiac [16], is a state-of-the-art method employed in the identification of constitutive parameters, known by its computational efficiency and does not resort to FEM for any forward calculations [17]. The key elements behind the VFM are the Principle of Virtual Work (PVW) and the choice of virtual fields. According to the PVW, the internal virtual work must be equal to the external virtual work performed by the external forces and is written by [18]:

$$-\int_V \boldsymbol{\sigma} : \boldsymbol{\varepsilon}^* dV + \int_{\partial V} \mathbf{T} \cdot \mathbf{u}^* dS = 0 \tag{4}$$

where  $\boldsymbol{\varepsilon}^*$  is the virtual strain,  $\mathbf{u}^*$  is the virtual displacement,  $V$  is the volume of the solid and  $\mathbf{T}$  is the traction vector. The virtual entities work can be defined independently of the measured displacements/strains. Any number of virtual fields can be used; however, the following conditions should be honored [5,6]: the functions defining the virtual fields should be piece-wise differentiable and kinematically admissible, in order to satisfy the displacement boundary conditions. The virtual fields can be manually defined, though the choice is tied to the user's own experience and intuition. Moreover, manually defined virtual fields do not show a temporal evolution [17]. Nonetheless, systematic procedures to automatically define these virtual entities exist, namely: the stiffness-based and the sensitivity-based virtual fields [4,7].

In the present work, sensitivity-based virtual fields were employed to indirectly train a single-layer perceptron for implicit constitutive modelling, following the workflow presented in Fig. 3.

The key concept of the sensitivity-based virtual fields is to apply a perturbation to each of the model's parameters in order to obtain a stress sensitivity and its temporal evolution [17]. The stress field is the only quantity that depends directly on the constitutive parameters in the VFM, so the stress sensitivity maps highlight areas that strongly depend on a given parameter [20]. In the context of this work, the single-layer perceptron from Fig. 2 is taken as a representation of the constitutive model. The spatial sensitivity of stress to each model parameter is computed as:

$$\delta \boldsymbol{\sigma}^{(i)}(\boldsymbol{\varepsilon}, \mathbf{W}, t) = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}, \mathbf{W}, t) - \boldsymbol{\sigma}(\boldsymbol{\varepsilon}, \mathbf{W} + \delta w_i, t), \delta w_i = -0.1 w_i \tag{5}$$

with  $i$  being the  $i$ -th term of the perceptron's weight matrix  $\mathbf{W}$  and  $t$  the time step. The virtual displacements  $\mathbf{u}^*$  are found starting from the following system of equations applied to a virtual mesh:

$$\delta \boldsymbol{\sigma}^{(i)} = \mathbf{B} \mathbf{u}^{*(i)} \tag{6}$$

where  $\mathbf{B}$  is the global strain-displacement matrix, used to map the virtual displacements at the nodes to every virtual strain. If the displacement is prescribed at the boundaries, the traction is unknown and, as such, the displacements are set to zero. If, on the other hand, the traction distribution is unknown and only the resultant force is known, a constant virtual displacement is set at the boundaries. By applying these constraints, a modified global strain-displacement matrix  $\bar{\mathbf{B}}$  is obtained and the virtual displacements  $\mathbf{u}^*$  can finally be computed as follows:

$$\mathbf{u}^{*(i)} = \text{pinv}(\bar{\mathbf{B}}) \delta \boldsymbol{\sigma}^{(i)} \tag{7}$$

with  $\text{pinv}(\bar{\mathbf{B}})$  being the pseudo-inverse of the modified strain-displacement matrix. The virtual strains  $\boldsymbol{\varepsilon}^*$  are then computed as:

$$\boldsymbol{\varepsilon}^{*(i)} = \mathbf{B} \mathbf{u}^{*(i)} \tag{8}$$

The training process is carried out, with the virtual fields  $\mathbf{u}^{*(i)}$  and  $\boldsymbol{\varepsilon}^{*(i)}$  being updated multiple times in order to evaluate the loss resulting from the application of the PVW and update the weight matrix  $\mathbf{W}$  accordingly. The loss to be minimized is the following:

$$\mathcal{L}(\mathbf{W}, \mathbf{b}, \boldsymbol{\varepsilon}) = \sum_{i=1}^{n_{VFS}} \left[ \frac{1}{(\alpha^{(i)})^2} \sum_{t=1}^{n_t} \left( \sum_{j=1}^{n_{pts}} \boldsymbol{\sigma}^j(\mathbf{W}, \mathbf{b}, \boldsymbol{\varepsilon}, t) \cdot \boldsymbol{\varepsilon}^{*j(i)}(t) \cdot S^j - W_{\text{ext}}^*(t) \right)^2 \right] \tag{9}$$

with  $S^j$  being the surface area of the  $j$ -th measurement point and  $\alpha^{(i)}$  the scaling factor of the  $i$ -th virtual field, employed to guarantee similar magnitudes between different virtual fields. In the present work, the scaling factor was defined based on the mean of the 30% highest internal virtual work values.

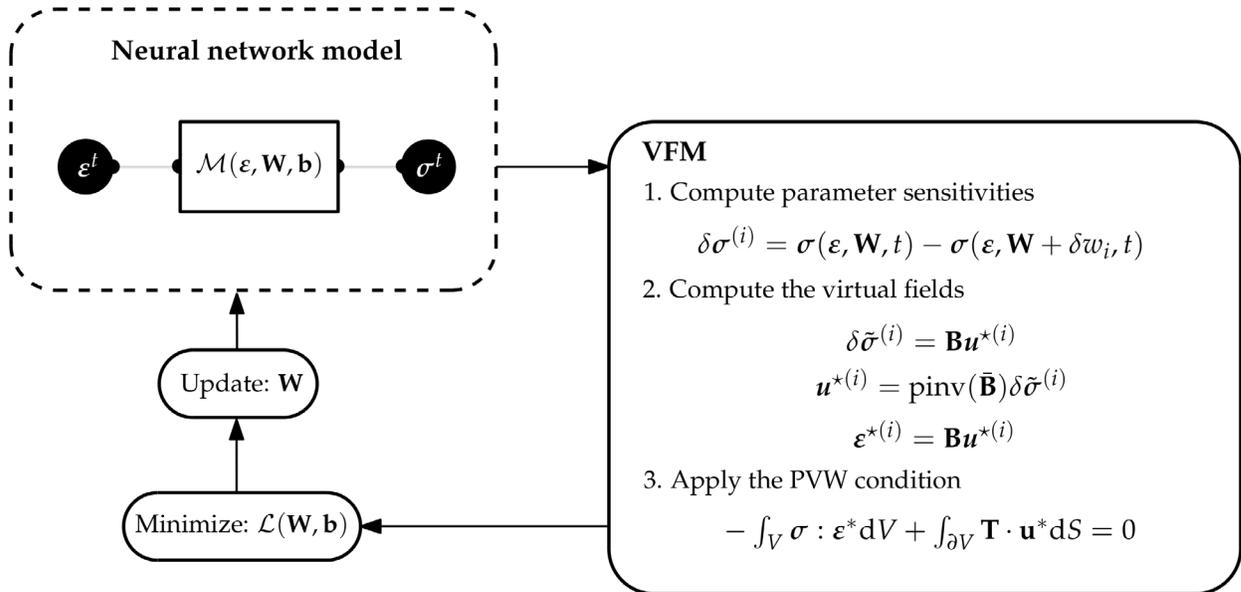


Fig. 3. Coupled ANN-VFM method for implicit constitutive modelling

### Unconstrained Training Analysis

The Adam algorithm was used to optimize the network weights, with an initial learning rate set to 0.1, scheduled to be reduced using a multiplier of 0.2 if no improvement in the training loss was registered after 3 epochs. For the unconstrained training model, the network was set to train during a maximum of 10000 epochs. However, an early-stopping criterion was triggered when no further improvement was observed in the test loss, after 2886 epochs. The learning curves are plotted in Fig. 4, showing a sharp decline during training with a low value in the order of  $10^{-3}$  achieved at the end and almost no gap being observed between the train and test curves, indicating the model did not overfit the data.

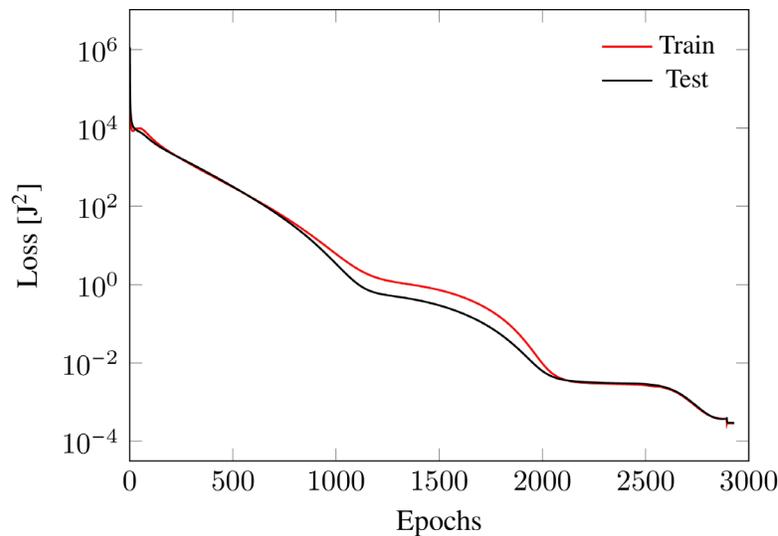


Fig. 4. Learning curves for the unconstrained training model.

During training, the layer weights should evolve in such a manner that, at the end of the process, the unscaled matrix  $\mathbf{W}$  approximates  $\mathbf{D}$  as much as possible, with the bias vector  $\mathbf{b}$  being equal to or almost zero. Due to the scaling of the inputs, the parameters hold no significance, so the inverse scaling is needed here to compare both entities. An overview of both scaled and unscaled model parameters obtained after training is shown in Table 3.

Table 3. Scaled and unscaled model parameters for the unconstrained model after training.

	Layer weights $\mathbf{W}$	Biases $\mathbf{b}$
<b>Scaled</b>	$\begin{bmatrix} 76.6645 & 0.01328 & 0.0159 \\ 23.1111 & 27.4231 & 0.0667 \\ 0.6532 & 0.3366 & 5.1392 \end{bmatrix}$	$\{111 \quad 0.0997 \quad 5.80\}^T$
<b>Unscaled</b>	$\begin{bmatrix} 209891.4619 & 101.9638 & 234.9259 \\ 63273.3868 & 210502.5075 & 981.5443 \\ 1788.5317 & 2584.3104 & 75627.5458 \end{bmatrix}$	$\{0.0305 \quad -0.0369 \quad -0.286\}^T$
<b>Rel. error [%]</b>	$\begin{bmatrix} 9.047 & 99.853 & \text{N/A} \\ 8.605 & 8.782 & \text{N/A} \\ \text{N/A} & \text{N/A} & 6.366 \end{bmatrix}$	N/A

Examining the unscaled parameters, one can observe that although the single layer perceptron was able to naturally learn nonnegative terms, it was not able to replicate the symmetry of the elasticity matrix, thus failing to guarantee the isotropy of the material. Furthermore, there are significant errors between the unscaled parameters and the terms of the elasticity matrix  $\mathbf{D}$ . Nevertheless, the indices of the most dominant terms ( $w_{11}$ ,  $w_{22}$ ,  $w_{33}$ ) correlate well with those from  $\mathbf{D}$ . The unscaled bias vector holds very small terms in comparison, however not close to zero. From here, we conclude that the model is not able to fully predict the linear elastic response. This is confirmed by observing the plots comparing the real and predicted elastic curves corresponding to two example elements, shown on Fig. 5. The differences stated above, caused the single layer perceptron predictions to achieve correlations ranging from 0.85 to 0.999 and mean absolute errors (MAE) from 0.266 to 5.173.

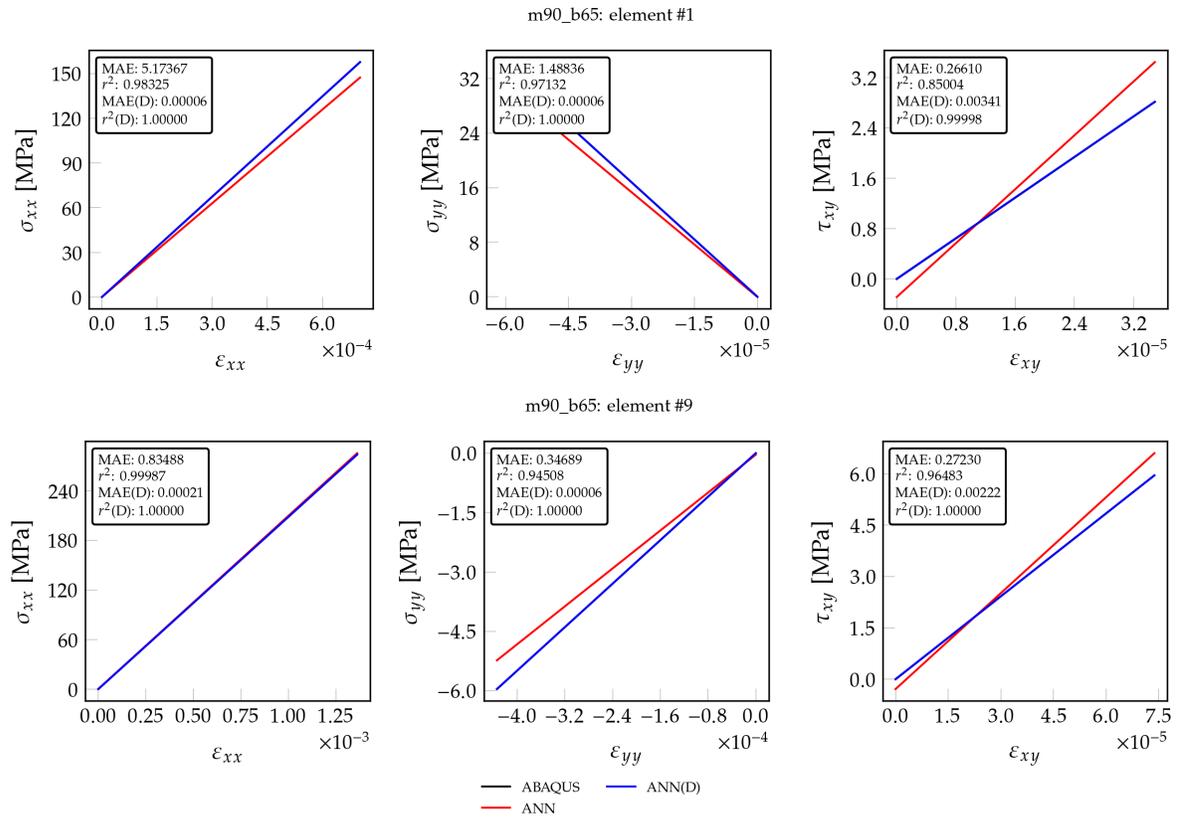


Fig. 5. Validation results for the unconstrained training model for elements 1 and 9.

### Constrained Training Analysis

For the constrained training model, the same training parameters were used, with the only difference being that the following set of constraints were applied to the entries of the parameter matrix  $\mathbf{W}$ :

$$w_{ij} > 0 \wedge w_{13} = w_{31} = 0 \wedge w_{23} = w_{32} = 0 \quad (10)$$

meaning all the terms of  $\mathbf{W}$  are forced to be nonnegative and some terms set to be zero, matching the indices of the terms with zero values in  $\mathbf{D}$ . The training progressed with the early-stopping criterion being triggered when no further improvement was observed in the test loss, after 5381 epochs. The resulting learning curves are plotted in Fig. 6, showing a sharp decline during training with a lower value of loss being achieved at the end (order of  $10^{-9}$ ) when compared to the unconstrained model. Similarly, almost no gap is observed between the train and test curves, indicating the model did not overfit the data.

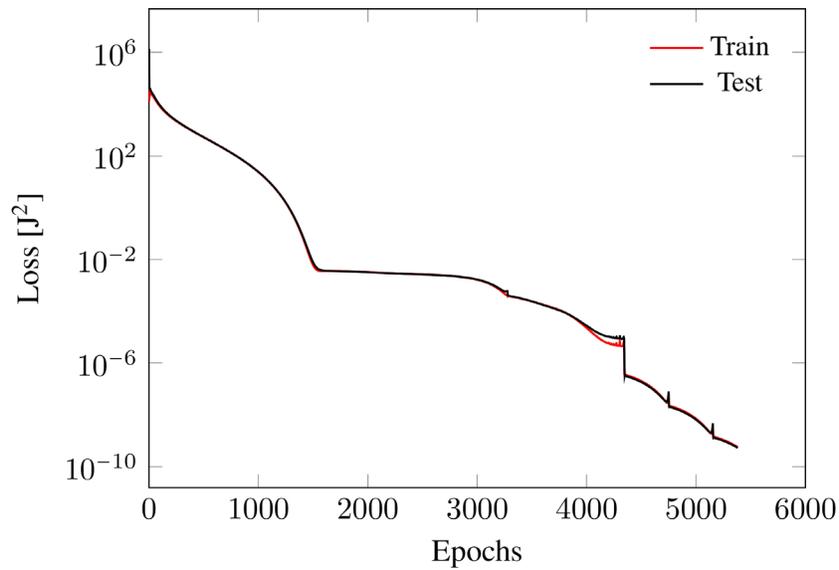


Fig. 6. Learning curves for the constrained training model.

An overview of both scaled and unscaled model parameters obtained after training is shown in Table 4. Examining the unscaled parameters, one can observe that enforcing the constraints defined in (10) was enough for the single layer perceptron to naturally replicate the symmetry of the elasticity matrix, thus guaranteeing the isotropy of the material. Although no constraints were applied to the biases, the unscaled bias vector holds values very close to zero, as it was expected. The model perfectly predicts the elastic response, as it is confirmed by the plots comparing the real and predicted elastic curves corresponding to two example elements, shown on Fig. 7. There are some differences between the unscaled parameters and the terms of the elasticity matrix  $\mathbf{D}$ , however these are not significantly high to cause the model to output badly predicted curves. The constraints allowed the model to achieve lower mean absolute errors and the highest possible correlations  $r^2$ .

Table 4. Scaled and unscaled model parameters for the constrained model after training.

	Layer weights $\mathbf{W}$	Biases $\mathbf{b}$
<b>Scaled</b>	$\begin{bmatrix} 78.4168 & 8.2999 & 0 \\ 23.5103 & 27.8005 & 0 \\ 0 & 0 & 4.9949 \end{bmatrix}$	$\{104.3700 \quad 0 \quad 5.4179\}^T$
<b>Unscaled</b>	$\begin{bmatrix} 230643.294 & 68815.410 & 0 \\ 69149.634 & 230497.092 & 0 \\ 0 & 0 & 80669.215 \end{bmatrix}$	$\{-0.0043 \quad -0.0001 \quad -0.0007\}^T$
<b>Rel. error [%]</b>	$\begin{bmatrix} 0.054 & 0.600 & \text{N/A} \\ 0.117 & 0.118 & \text{N/A} \\ \text{N/A} & \text{N/A} & 0.124 \end{bmatrix}$	N/A

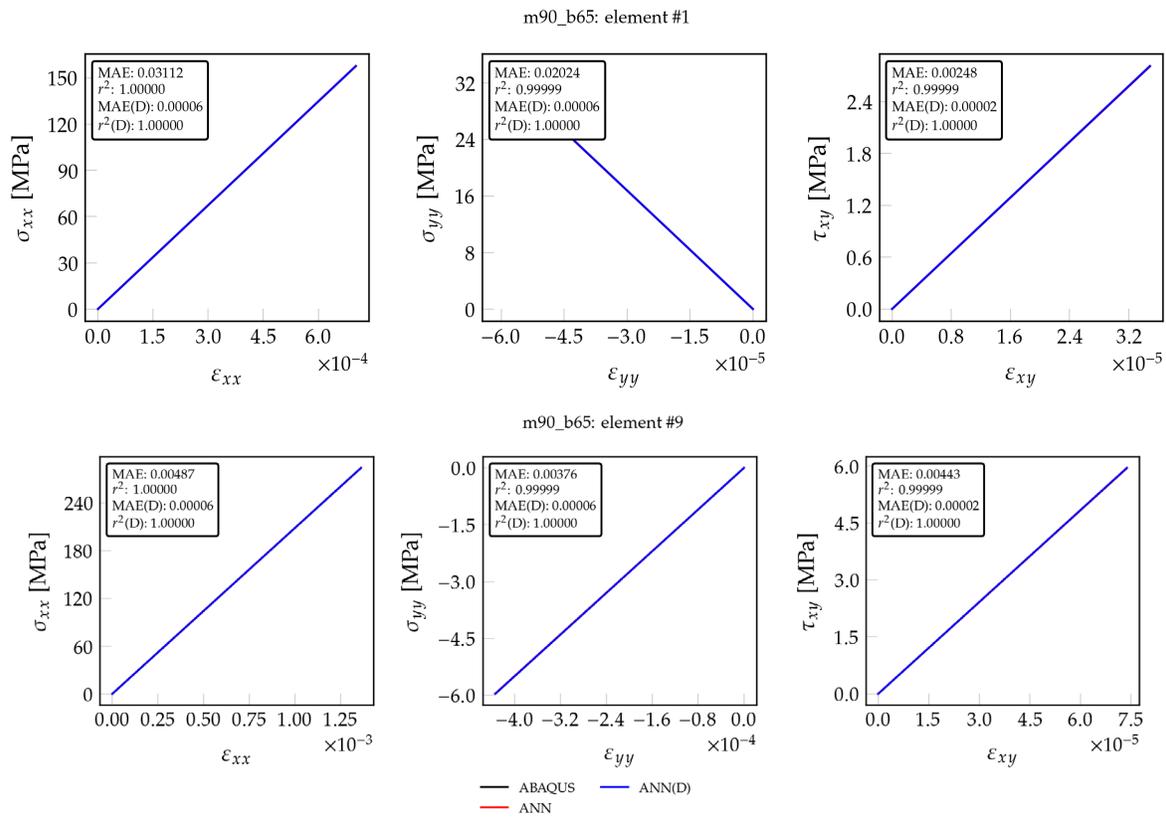


Fig. 7. Validation results for the constrained training model for elements 1 and 9.

### Summary

A single layer perceptron model was trained without stress labels employing a new indirect training methodology based on the Virtual Fields Method, which makes use of the Principle of Virtual Work and sensitivity-based virtual fields to guarantee the equilibrium between the external and virtual work. The sensitivity-based virtual fields are a more robust approach than the manually defined virtual fields, providing a systematic way of generating virtual fields that are not static and do not dependent on the user's intuition.

Two different perceptron models with the same architecture were trained in order to learn the linear elastic response of a virtual material and to study the influence of the application of constraints during training. Considering a single layer perceptron with 3 inputs and 3 outputs, an analogy could be established between the mathematical description of the forward pass and the Hooke's law under plane stress, with the perceptron's weight matrix  $\mathbf{W}$  resembling the elasticity matrix  $\mathbf{D}$ . There are several possible ways to constrain the space of admissible solutions of a neural network model. In the present work, by adapting the perceptron's model to the real material model, its parameters gain a real meaning, thus making it easier to control their evolution during training and ensure the predictions are physically admissible. As such, the constraints were applied directly to the perceptron's parameters in order to guarantee nonnegative terms in  $\mathbf{W}$  and force some of its terms to be zero, matching those in the elasticity matrix  $\mathbf{D}$ . It was shown that by applying these constraints, the single layer perceptron naturally learned the symmetry of the elasticity matrix  $\mathbf{D}$ , respecting the isotropy of the material, and replicated its terms without significant errors, being able to predict a linear elastic response that matches the real one.

## Disclaimer

The results reflect only the authors' view, and the European Commission is not responsible for any use that may be made of the information it contains.

## Acknowledgements

Rúben Lourenço acknowledges the Portuguese Foundation for Science and Technology (FCT) for the financial support provided through the grant 2020.05279.BD, co-financed by the European Social Fund, through the Regional Operational Programme CENTRO 2020. The authors also gratefully acknowledge the financial support of FCT under the projects UID/EMS/00481/2013-FCT under CENTRO-01-0145-FEDER-022083. This project also supported by the Research Fund for Coal and Steel under grant agreement No 888153.

## References

- [1] Y. C. Lin, J. Zhang, J. Zhong, Application of neural networks to predict the elevated temperature flow behavior of a low alloy steel, *Comput. Mater. Sci.* 43 (2008) 752-758. <https://doi.org/10.1016/j.commatsci.2008.01.039>
- [2] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* 2 (1989) 359-366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- [3] K. Linka, E. Kuhl, A new family of Constitutive Artificial Neural Networks towards automated model discovery, *Arxiv* (2022). <https://doi.org/10.48550/arxiv.2210.02202>
- [4] J. Ghaboussi, D.A. Pecknold, M. Zhang, R.M. Haj-Ali, Autoprogressive training of neural network constitutive models, *Int. J. Numer. Methods Eng.* 42 (1998) 105-126. [https://doi.org/10.1002/\(SICI\)1097-0207\(19980515\)42:1<105::AID-NME356>3.0.CO;2-V](https://doi.org/10.1002/(SICI)1097-0207(19980515)42:1<105::AID-NME356>3.0.CO;2-V)
- [5] A. Zhang, D. Mohr, Using neural networks to represent von Mises plasticity with isotropic hardening, *Int. J. Plast.* 132 (2020) 102732. <https://doi.org/10.1016/j.ijplas.2020.102732>
- [6] F. Masi, I. Stefanou, P. Vannucci, V. Maffi-Berthier, Thermodynamics-based Artificial Neural Networks for constitutive modeling, *J. Mech. Phys. Solids* 147 (2021) 104277. <https://doi.org/10.1016/j.jmps.2020.104277>
- [7] F. As'ad, P. Avery, C. Farhat, A mechanics-informed artificial neural network approach in data-driven constitutive modeling, *Int. J. Numer. Methods Eng.* 123 (2022) 2738-2759. <https://doi.org/10.1002/NME.6957>
- [8] L. Borkowski, C. Sorini, A. Chattopadhyay, Recurrent neural network-based multiaxial plasticity model with regularization for physics-informed constraints, *Comput. Struct.* 258 (2022) 106678. <https://doi.org/10.1016/J.COMPSTRUC.2021.106678>
- [9] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686-707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [10] E. Cueto, F. Chinesta, *Thermodynamics of learning physical phenomena*, 2022.
- [11] A. Danoun, E. Prulière, Y. Chemisky, Thermodynamically consistent Recurrent Neural Networks to predict non linear behaviors of dissipative materials subjected to non-proportional loading paths, *Mech. Mater.* (2022) 104436. <https://doi.org/10.1016/j.mechmat.2022.104436>
- [12] A. Bower, *Appl. Mech. Solids*, CRC Press, 2009.
- [13] K. Xu, D.Z. Huang, E. Darve, Learning constitutive relations using symmetric positive definite neural networks, *J. Comput. Phys.* 428 (2021) 110072. <https://doi.org/10.1016/j.jcp.2020.110072>
- [14] J.M.P. Martins, A. Andrade-Campos, S. Thuillier, Comparison of inverse identification strategies for constitutive mechanical models using full-field measurements, *Int. J. Mech. Sci.* 145 (2018) 330–345. <https://doi.org/10.1016/j.ijmecsci.2018.07.013>

- [15] X. Liu, F. Tao, H. Du, W. Yu, K. Xu, Learning nonlinear constitutive laws using neural network models based on indirectly measurable data, *J. Appl. Mech. Trans. ASME* 87 (2020) 1-8. <https://doi.org/10.1115/1.4047036>
- [16] M. Grédiac, Principe des travaux virtuels et identification, *Comptes Rendus de l'Académie des Sciences* 309 (1989) 1–5.
- [17] A. Marek, F.M. Davis, F. Pierron, Sensitivity-based virtual fields for the non-linear virtual fields method, *Comput. Mech.* 60 (2017) 409–431. <https://doi.org/10.1007/s00466-017-1411-6>
- [18] F. Pierron, M. Grédiac, *The virtual fields method: extracting constitutive mechanical parameters from full-field deformation measurements*. Springer Science & Business Media, 2012.
- [19] M. Grediac, F. Pierront, S. Avrilt, E. Toussaint, The Virtual Fields Method for Extracting Constitutive Parameters From Full-Field Measurements: a Review, *Strain* 42 (2006) 233-253. <https://doi.org/10.1111/j.1475-1305.2006.tb01504.x>
- [20] A. Marek, F.M. Davis, M. Rossi, F. Pierron, Extension of the sensitivity-based virtual fields to large deformation anisotropic plasticity, *Int. J. Mater. Forming* 12 (2019) 457-476. <https://doi.org/10.1007/S12289-018-1428-1/FIGURES/13>
- [21] T.B. Stoughton, J.W. Yoon, Review of Drucker's postulate and the issue of plastic stability in metal forming, *Int. J. Plast.* 22 (2006) 391–433. <https://doi.org/10.1016/J.IJPLAS.2005.03.002>