Neural networks meet physics-based material models

Accelerating concurrent multiscale simulations of path-dependent composite materials

Alves Maia, M.; Rocha, I.B.C.M.; van der Meer, F.P.

Publication date
2022

Document Version
Final published version

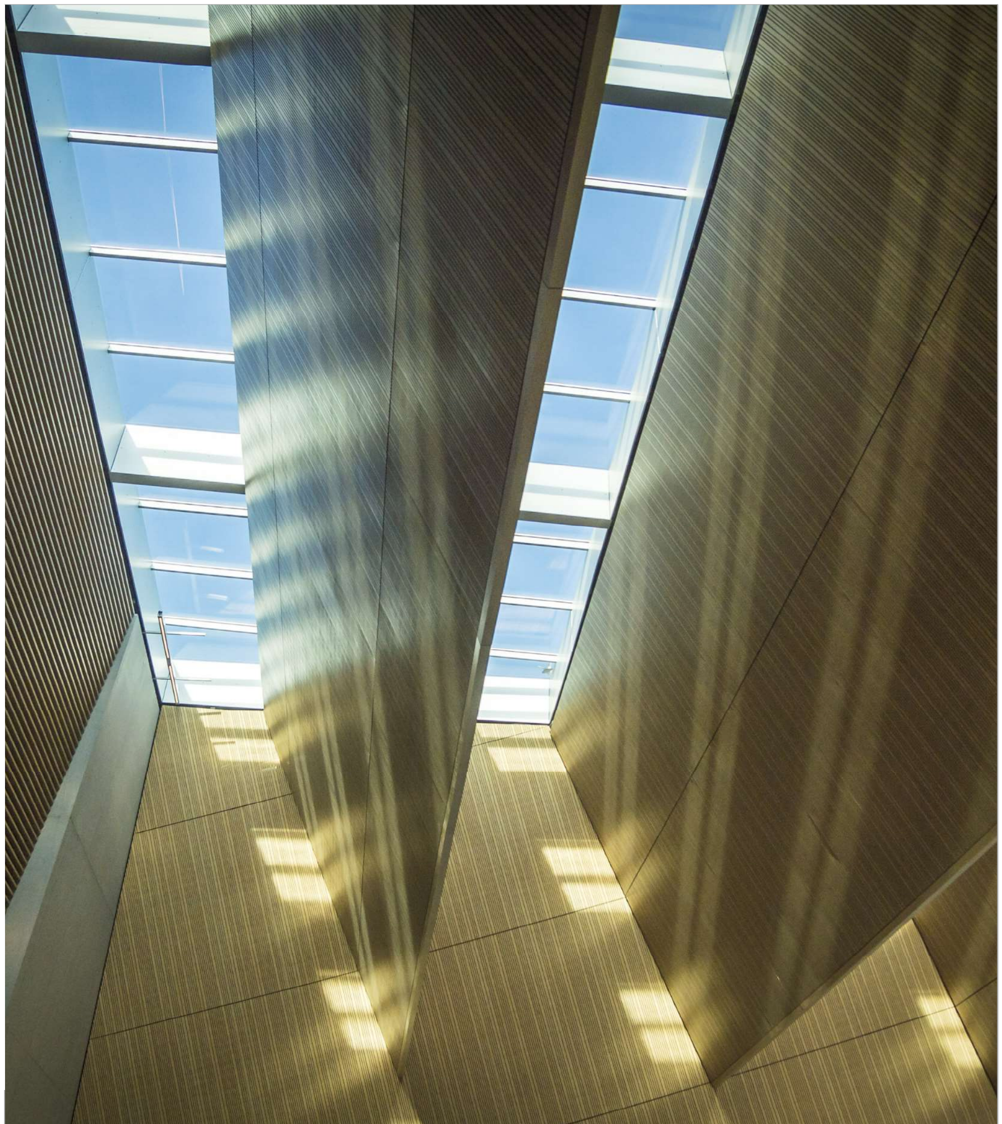
Published in
Proceedings of the 20th European Conference on Composite Materials: Composites Meet Sustainability

Citation (APA)
Alves Maia, M., Rocha, I. B. C. M., & van der Meer, F. P. (2022). Neural networks meet physics-based material models: Accelerating concurrent multiscale simulations of path-dependent composite materials. In A. P. Vassilopoulos , & V. Michaud (Eds.), *Proceedings of the 20th European Conference on Composite Materials: Composites Meet Sustainability: Vol 4 – Modeling and Prediction* (pp. 891-898). EPFL Lausanne, Composite Construction Laboratory.

ECCM 20

26–30 JUNE 2022

LAUSANNE SWITZERLAND



Proceedings of the 20th European Conference on Composite Materials

COMPOSITES MEET SUSTAINABILITY

Vol 4 – Modeling and Prediction

Editors : Anastasios P. Vassilopoulos, Véronique Michaud

Organized by :

EPFL

Under the patronage of :

CCLAB Composite Construction Laboratory

LPAC Laboratory for Processing of Advanced Composites

ESCM EUROPEAN SOCIETY FOR COMPOSITE MATERIALS

Proceedings of the 20th
European Conference on Composite Materials
ECCM20
26-30 June 2022,
EPFL Lausanne Switzerland

Edited By :

Prof. Anastasios P. Vassilopoulos, CCLab/EPFL
Prof. Véronique Michaud, LPAC/EPFL

Oragnized by:

Composite Construction Laboratory (CCLab)
Laboratory for Processing of Advanced Composites (LPAC)
Ecole Polytechnique Fédérale de Lausanne (EPFL)

ISBN: 978-2-9701614-0-0
DOI: http://dx.doi.org/10.5075/epfl-298799_978-2-9701614-0-0

Published by :

Composite Construction Laboratory (CCLab)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
BP 2225 (Bâtiment BP), Station 16
1015, Lausanne, Switzerland

https://cclab.epfl.ch

Laboratory for Processing of Advanced Composites (LPAC)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
MXG 139 (Bâtiment MXG), Station 12
1015, Lausanne, Switzerland

https://lpac.epfl.ch

Cover:

Swiss Tech Convention Center
© Edouard Venceslau - CompuWeb SA

Cover Design:

Composite Construction Laboratory (CCLab)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Disclaimer:

NEURAL NETWORKS MEET PHYSICS-BASED MATERIAL MODELS: ACCELERATING CONCURRENT MULTISCALE SIMULATIONS OF PATH-DEPENDENT COMPOSITE MATERIALS

Marina, A. Maia[a], Iuri, B. C. M. Rocha[a], Pierre, Kerfriden[b,c], Frans P., van der Meer[a],

a: Delft University of Technology, P.O. Box 5048. 2600GA Delft, The Netherlands
b: Mines ParisTech (PSL University), Centre des Matériaux, 63-65 Rue Henri-Auguste Desbrueres BP87, F-91003, Évry, France
c: Cardiff University, School of Engineering, Queen's Buildings, The Parade, Cardiff, CF24 3AA, United Kingdom
e-mail: m.alvesmaia@tudelft.nl

Abstract: *In a concurrent multiscale (FE^2) modeling approach the complex microstructure of composite materials is explicitly modeled on a finer scale and nested to each integration point of the macroscale. However, such generality is often associated with exceedingly high computational costs in real-scale applications. In this work, a novel Neural Network (NN) is used as the constitutive model for the microscale to tackle that issue. Unlike conventional NNs, the proposed network employs the actual material models used in the full-order micromodel as the activation function of one of the layers. The NN's capabilities are assessed (i) for a single micromodel level, where its performance is compared to that of a Recurrent Neural Network (RNN), and (ii) for an FE^2 example. A highlight of the proposed network is the ability to predict unloading/reloading behavior without ever seeing it during training, a stark contrast with highly popular but data-hungry models such as RNNs.*

Keywords: Neural Networks (NNs); Multiscale; Path-dependency

1. Introduction

Machine learning techniques are an increasingly popular alternative to time-consuming simulations in various fields. Recognizing this potential, the development and application of such methods gained traction in recent years in the field of concurrent multiscale analysis. However, a critical limitation of data-driven models is that they do not perform as well in extrapolation as they do within their training space. This can be especially critical when they are used as constitutive models in FE^2 where the lack of basic physics-related constraints can cause numerical instabilities and convergence issues.

On top of that, devising a sampling plan to train Neural Networks (NN) to capture path-dependent behavior is itself a convoluted task. This is because stresses depend on the strain history of the material. Thus, independent pairs of strains and stresses cannot fully describe how the material should evolve. A common approach to handle that is to augment the feature space with an extra variable that carries information about the history of stress and/or strain:

$$\hat{\boldsymbol{\sigma}}^t = f_{NN}(\mathbf{W}, \mathbf{b}, \boldsymbol{\varepsilon}^t, \boldsymbol{\alpha}) \tag{1}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the model parameters of a NN, $\boldsymbol{\varepsilon}^t$ and $\hat{\boldsymbol{\sigma}}^t$ are the macroscopic strain and the approximated stress tensors at time step t, respectively, and $\boldsymbol{\alpha}$ is the history variable vector.

Typically, the previous strain state $\boldsymbol{\varepsilon}^{t-1}$ or the accumulated absolute strain are chosen for that [1].

Another highly popular alternative is to use Recurrent Neural Networks (RNNs). These networks can account to some extent for the typical loading/unloading by incorporating information from previous inputs as in the following parametric regression:

$$\widehat{\boldsymbol{\sigma}}^t = f_{RNN}(\mathbf{W}, \mathbf{b}, \mathbf{W}_h, \mathbf{b}_h, \boldsymbol{\varepsilon}^t) \tag{2}$$

where $\mathbf{W}_h$ and $\mathbf{b}_h$ represent additional model parameters shared across time to keep track of history-dependent materials in an implicit way. These parameters describe the evolution of the so-called hidden state and can capture information from previous iterations without the need to include previous strain states in the input vector as shown in Eq. (1). This way, the network ca learn how to process and predict a sequence of strains. As such, RNNs rapidly became a popular choice to model composite materials with path-dependency [2-4].

Despite their popularity, RNNs are still severely limited by the curse of dimensionality associated with sampling arbitrarily long strain paths. In this work, a physics-infused network is proposed to overcome this issue and accelerate concurrent multiscale simulations. In Section 2, the computational bottleneck in FE^2 is briefly discussed, while in Section 3 the main features of the novel NN are described. Finally, results are presented in Section 4 and conclusions are shown in Section 5.

2. Multiscale analysis

Let M define the macroscopic domain being modeled subjected to a set of Neumann and Dirichlet boundary conditions acting on the body surface. To find the internal stresses and displacement field of such body, a boundary value problem that satisfies the following equilibrium equation is defined:

$$\mathrm{div}\left(\boldsymbol{\sigma}^{\mathrm{M}}\right) = 0 \tag{3}$$

where $\mathrm{div}(\cdot)$ is the divergence operator. To relate strains and stresses, a constitutive model $\mathcal{D}$ is required:

$$\boldsymbol{\sigma}^{\mathrm{M}} = \mathcal{D}\,(\boldsymbol{\varepsilon}^{\mathrm{M}}, \boldsymbol{\alpha}^{\mathrm{M}}) \quad \text{with} \quad \boldsymbol{\varepsilon}^{\mathrm{M}} = \frac{1}{2}\left(\nabla\mathbf{u}^{\mathrm{M}} + (\nabla\mathbf{u}^{\mathrm{M}})^{\mathrm{T}}\right) \tag{4}$$

where $\boldsymbol{\alpha}^{\mathrm{M}}$ is a history term that accounts for path-dependency and $\mathbf{u}^{\mathrm{M}}$ is the macroscopic displacement field. However, in the concurrent multiscale approach, $\mathcal{D}$ is not directly formulated but is instead obtained by nesting a lower scale model to each integration point, as illustrated in Fig. 1. In that scale, complex materials can be explicitly modeled using simpler constitutive models and a geometrical representation of the microstructure.

The computational bottleneck arises from the fact that to obtain the internal forces and the tangent stiffness matrix of a single integration point of the macroscale, an entire FE model is run instead of a single evaluation of a homogenous material model. To alleviate that, the alternative explored in this work consists in replacing the solution of a Representative Volume Element (RVE) subjected to a periodic boundary value problem with a physics-based neural network.

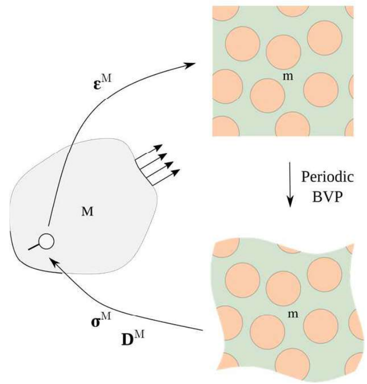


Figure 1. FE² scheme

3. Neural networks

Consider the parametric regression model in Eq. (2). When training a neural network for $\boldsymbol{\sigma}^{\mathrm{M}}$, strains are fed its first layer (input) and the values are propagated until the final layer (output) to give the predicted stresses $\widehat{\boldsymbol{\sigma}}^{\mathrm{M}}$, which are in turn compared to the ground truth value using the loss function:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left\|\boldsymbol{\sigma}^{\mathrm{M}}(\boldsymbol{\varepsilon}_i^{\mathrm{M}}) - \widehat{\boldsymbol{\sigma}}^{\mathrm{M}}(\boldsymbol{\varepsilon}_i^{\mathrm{M}})\right\|^2 \tag{5}$$

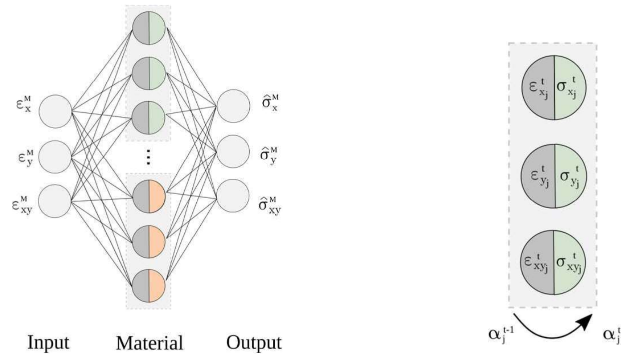
where N is the number of pairs of $\boldsymbol{\varepsilon}^{\mathrm{M}} - \boldsymbol{\sigma}^{\mathrm{M}}$ obtained from microscopic simulations. Eq. (5) is then minimized by updating the model parameters according to an optimization algorithm.

3.1 Bayesian Recurrent Neural Network

In practice, RNNs struggle with vanishing gradient problems and are not suitable for long-term dependent problems. Among other architectures, the Gated Recurrent Unit (GRU) has become a widely used alternative to circumvent that issue. The GRU contains more operations and parameters than a regular RNN and can control more precisely the flow of information, being able to retain or forget information in a long sequence. In this work, a GRU with Variational dropout (i.e., Gaussian dropout where the rates are learned implicitly by the network) [5], also referred as Bayesian Recurrent Neural Network (BNN), is used for comparison purposes.

3.2 Adding physics-based material models

The proposed regression model consists of a neural network composed of one fully-connected material layer followed by a Dense layer, as illustrated Fig. 2. The material layer is responsible for explicitly incorporating into the network the same physics-based material model used in the homogenization of the RVE.



(a) Proposed neural network (b) Fictitious integration point j

Figure 2. Physics-infused neural network

Two different material models are used to describe the fibers and the matrix of the composite microscopic models in this work: the first consists of a linear elastic model and the second is an elastoplastic model $\mathcal{M}$. Since the latter is more complex, it is used to illustrate how its features are incorporated into the network. Model $\mathcal{M}$ takes as input the current strain $\boldsymbol{\varepsilon}^t \in \mathbb{R}^{n_\varepsilon}$ and the internal variables from previous time step $\boldsymbol{\alpha}^{t-1} \in \mathbb{R}^{n_{IntVar}}$, where n_ε and n_{IntVar} are the number of strain components and number of internal variables of the material model, respectively.

First, neurons are grouped in sets of the size of the input layer (light grey boxes in Fig. 2a) and only then activated as a subgroup, or *fictitious material point*. To store the internal variables used as input/output of the material model, an auxiliary vector $\mathbf{h}_j \in \mathbb{R}^{n_{IntVar}}$ is defined. For the first time step, $\mathbf{h}_j$ is initialized as zero for all subgroups.

As information reaches the material layer and the material model is evaluated (or updated), three features are obtained: the stresses $\boldsymbol{\sigma}_j^t$, the updated internal variables $\boldsymbol{\alpha}_j^t$,and the tangent stiffness matrix $\mathbf{D}_j^t \in \mathbb{R}^{n_\varepsilon \times n_\varepsilon}$. Then, stresses are propagated forward, and the updated internal variables $\boldsymbol{\alpha}^t$ are stored in $\mathbf{h}_j^t$ so that when new strains $\boldsymbol{\varepsilon}_j^{t+1}$ are fed to the fictitious material point j in the next time step, the material model is aware of its own history so far, as illustrated in Fig. 2b. Finally, to obtain the stiffness matrix, a full backwards differentiation pass is required.

3.3 Decoders

The decoder converts the outputs from the Material layer and combines them into the predicted macroscopic stress $\hat{\boldsymbol{\sigma}}^{\mathrm{M}}$. In that sense, the decoder acts as the averaging operator in the multiscale approach. Therefore, weights of the output layer can be seen as the relative contribution of each fictitious material point to the average macroscopic stress as if they were obtained from a Gaussian quadrature, which are always positive.

Based on that, four different approaches are investigated: (i) first, no constraints are applied and the weights can be positive and negative, then weight positivity is enforced either with (ii) a penalty approach or by applying the (iii) *relu* function or the (iv) *softplus* function on the weights. For the decoder with the penalty (ii), an extra term is added to the loss function in Eq. (5):

$$P\,(w_k) = \begin{cases} 0.0, & w_k \geq 0 \\ \theta\,|w_k|, & w_k < 0 \end{cases} \quad (6)$$

where θ is a penalty parameter term introduced to penalize negative weights w_k in the output layer. Finally, for the approaches (iii) and (iv), the transformation functions are applied element-wise on the weight matrix.

It is worth mentioning that the first two decoders are shown as reference results since only (iii) and (iv) guarantee that, after the transformation, weights will be positive. This is an important outcome for the FE^2 framework because by constraining the decoder to be positive, the spectral properties of the Jacobian of the material model $\mathcal{M}$ are inherited by the network when calculating the tangent stiffness matrix.

4. Results

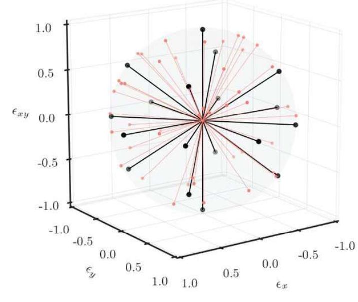
In this section, the performance of the proposed physics-infused network, or Material Neural Network (MNN), is compared to a Bayesian Recurrent Neural Network (BNN). The MNN was implemented in an in-house Finite Element code using the open-source Jem/Jive C++ numerical analysis library, while PyTorch was used to construct the BNN. The goal is to demonstrate the capabilities of the proposed network to capture path-dependent behavior in comparison to a popular method using exclusively monotonic data for training. The maximum number of epochs for both networks is 60000. For the BNN, an early stopping criterion of 5000 epochs is used.

The microscopic model consists of an RVE with 9 elastic fibers with properties $E = 74000$ MPa and $\nu = 0.2$ embedded in an elastoplastic matrix with isotropic hardening. The latter is modeled using the von Mises yield criterion with properties $E = 3130$ MPa, $\nu = 0.3$ and yield stresses given by:

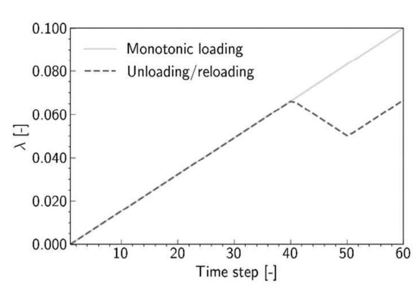
$$\sigma_t = \sigma_c = 64.8 - 33.6\, e^{-\varepsilon_{\mathrm{eq}}^{\mathrm{p}}/0.003407} \quad (7)$$

where $\varepsilon_{\mathrm{eq}}^{\mathrm{p}}$ is the equivalent plastic strain. Plane strain conditions are assumed.

For training the MNN, 18 curves with *a priori* known directions (black lines in Fig. 3a) are generated. Each curve consists of 60 pairs of $\boldsymbol{\varepsilon}^{\mathrm{M}} - \boldsymbol{\sigma}^{\mathrm{M}}$ with monotonic loading (solid line in Fig. 3b). These directions comprehend typical loading cases used for calibrating mesomodels and comprise pure uniaxial, shear, biaxial and biaxial with shear cases. The validation set consists of another 54 monotonic curves in random directions (red lines in Fig. 3a).



(a) Directions as unit load vectors



(b) Monotonic and non-monotonic loading

Figure 3. Loading directions in (a) and loading function in (b)

Based on that, a preliminary study with 10 different initializations was carried out for each decoder and size of the material layer as shown in Fig. 4. It is clear from the results that using only 3 fictitious integration points (in contrast, the original FE micromodel comprises 7088 integration points) is enough to accurately represent the homogenized material behavior. We, therefore, adopt a layer size of 3 points (9 units) with softplus-activated weights for the sake of parsimony. A similar procedure is followed for the BNN, and a network with a single GRU layer with 128 units is selected.

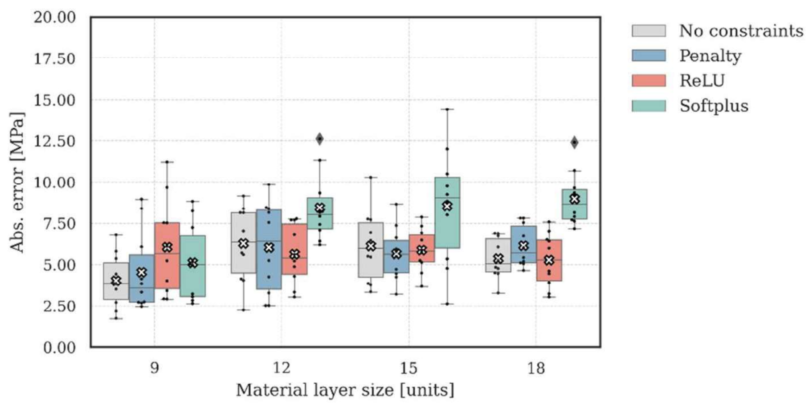


Figure 4. Abs. error for validation set with different combinations of decoder and layer size

4.1 Single scale

In this section, the 18 curves with known directions are kept as a fixed part of the dataset of both networks, while the BNN is trained with additional random monotonic curves for different training dataset sizes. The test set consists of 100 random curves with unloading/reloading as shown in Fig. 3b. Again, 10 different initializations were considered for each case, but only the best performance is depicted in Fig. 5.

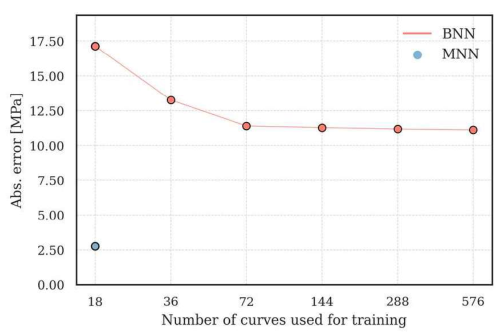
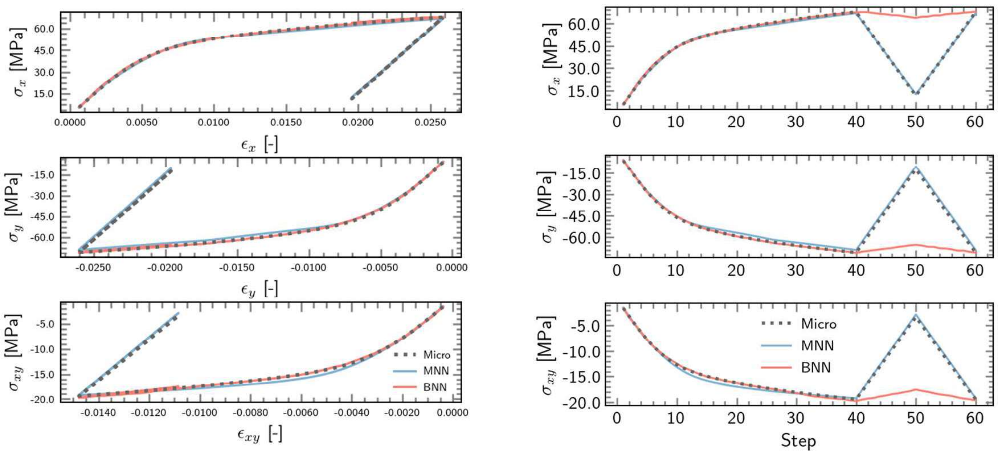


Figure 5. Networks trained on monotonic data and tested for curves with unloading/reloading

Note that as more curves are added, the BNN's error decreases, but around 144 curves, the addition of more monotonic data is no longer useful to the network in this scenario. The initial error decrease is actually associated with the points before the unloading. Once that part of the curve is accurate enough, the error in the unloading will remain unchanged (and high) while the MNN can capture accurately the entire strain path, as illustrated in Fig. 6. Although this is a simplified scenario, it helps in elucidating the ability of the proposed network to predict non-monotonic data without the need to extend the training dataset with curves with arbitrary unloading/reloading cycles, as typically done for RNNs.



(a) Strain-stress view (b) Time step-stress view

Figure 6. BNN trained on 144 monotonic curves cannot capture unloading/reloading behavior

4.2 Multi-scale

Here, the MNN trained in the previous section is tested as the constitutive model in a multiscale application. The structure consists of a composite tapered specimen with a length of 128 mm and a height of 8 mm loaded in transverse tension. The boundary and loading conditions are shown in Fig. 7, where the load-displacement curve using the full-order solution is plotted along with the network's response. Good agreement is observed between the curves.

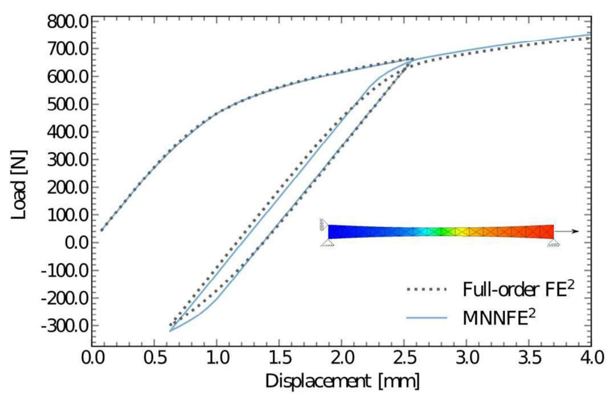


Figure 7. Load-displacement curves using the full-order solution and the MNN

5 Conclusions

A network with embedded physics-based constitutive models was presented. The network captures unloading without ever seeing it during training, which is not observed in the BNN regardless of the number of monotonic curves considered. In the multiscale example, the proposed approach showed good accuracy in an structure subjected to different strain states and reduced the CPU time from 9077 s to 3 s (excluding training and data generation times). Further details on the training of the networks and results over a broader range of test cases will be presented in a future publication.

Acknowledgements

The authors acknowledge the TU Delft AI Initiative for their support through the SLIMM AI Lab. FM acknowledges financial support from the Dutch Research Council (NWO) under Vidi grant 16464.

6 References

1. Huang D, Fuhg JN, Weißenfels C, Wriggers P. A machine learning based plasticity model using proper orthogonal decomposition. Computer Methods in Applied Mechanics and Engineering 2020; 365:113008.
2. Ghavamian F, Simone A. Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. Computer Methods in Applied Mechanics and Engineering 2019; 357:112594.
3. Wu L, Nguyen VD, Kilingar NG, Noels L. A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths. Computer Methods in Applied Mechanics and Engineering 2020; 369:113234.
4. Mozaffar M, Bostanabad R, Chen W, Ehmann K, Cao J, Bessa MA. Deep learning predicts path-dependent plasticity. Proceedings of the National Academy of Sciences 2019; 116:26414–26420.
5. Kingma DP, Salimans T, Welling M. Variational Dropout and the Local Reparameterization Trick. arXiv 2015; arXiv:1506.02557.