15th World Congress on Computational Mechanics (WCCM-XV)
8th Asian Pacific Congress on Computational Mechanics (APCOM-VIII)
Virtual Congress: 31 July – 5 August 2022
S. Koshizuka (Ed.)

ON THE PERFORMANCE OF DIFFERENT ARCHITECTURES IN MODELLING ELASTO-PLASTICITY WITH NEURAL NETWORK

REEM S. ALHAYKI[1*], EUGENIO J. MUTTIO[2*], WULF G. DETTMER[3*] and DJORDJE PERIĆ[4*]

[*] Swansea University, Bay Campus, Fabian Way, Swansea SA1-8EN, Wales, UK

[1] r.s.h.j.alhayki@swansea.ac.uk, [2] e.j.muttiozavala@swansea.ac.uk, [3] w.g.dettmer@swansea.ac.uk, [4] d.peric@swansea.ac.uk

Key words: Neural Network, Constitutive Modelling, Elasto-Plasticity

Abstract. *Constitutive models have been utilized to study the mechanical behaviors of solid material. The formulation of constitutive relations is difficult and could be associated with limiting hypothesis. This work proposes neural network-based approaches to reproduce the complex nonlinear constitutive relations of solid materials including elastic behavior and plastic deformation.*
It is shown that the proposed history-based and internal variable-based strategies can represent exactly the von Mises elasto-plastic material model in uni-axial stress state. Furthermore, close investigation suggests that the internal variable-based approach is most suitable.

1 INTRODUCTION

New and innovative materials with wide engineering and industrial applications continue to be one of the main topics of interest in Materials Science research. Historically, constitutive models have been utilized to describe the mechanical behavior of materials. The formulation of these constitutive relations is difficult and is associated with limiting hypothesis. The source of complexity is attributed to the composition and geometrical structure of the materials which causes isotropic, kinematic and distortional hardening. Artificial Neural Networks (ANNs) have shown promising capabilities that can be used as an alternative solution for modelling complex nonlinear constitutive relations [1]. Research shows that neural networks can be trained using experimental data to learn the relationships between stresses and strains without explicit mathematical formulation. Moreover, improvements in computer hardware processing power and architecture like GPUs, as well as software frameworks like Tensorflow [2] and PyTorch [3] allows building and deploying machine learning models easily.

The use of neural networks in modelling material behavior such as plastic deformation was pioneered by Ghaboussi et al. [4]. The learning was afterwards utilized in various research to model complex nonlinear mechanical behaviors including continuum damage [5], hyper-elasticity [6], visco-plasticity [7] and cyclic plasticity [8].

This paper contributes to this active research area by proposing and comparing two neural network-based approaches for constitutive modelling of solid material. The first model uses a history-based strategy while the second model uses an internal variable-based strategy.

2 ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) mimic the activity of the human brain, allowing computer models to discover patterns and solve common challenges in many disciplines such as life sciences, engineering and economics. ANNs are made up of nodes, also referred to as neurons, which are arranged into layers. An ANN includes an input layer, one or more hidden layers, and an output layer. When the network includes many hidden layers it is called a deep neural network. In feed-forward neural networks the data flows from the input layer to the output layer passing through the layers sequentially. In Recurrent Neural Networks (RNNs) the output of the network becomes the input in the next network evaluation. RNNs are used to process sequential data [9].

Each neuron in this system processes data with a linear operation in the form $Wx+b$. In this case, x represents the input vector, which can be either the initial input data in the first layer or the processed data in the downstream layers. During model training, the neural network learns the best values for biases b and weights W by passing the entire training data set through the network and comparing the generated output with the desired output then adjusting their values. This process is done many times and the biases and weights are adjusted using an optimizer with every pass of the data. In general, the model prediction accuracy improves with longer training. However, when the model is trained excessively on the given inputs then a problem of data generalisation and over-fitting could emerge. The model training process can be summarised in the following steps:

1) Input data is sent to the neuron which uses the biases and weights to compute $Wx+b$.
2) The result is then routed through an activation function, such as the Rectified Linear Unit (ReLU), the logistic (Sigmoid), or the hyperbolic tangent (Tanh). The activation function selection depends on the type of problem [10].
3) The output of the activation function is either passed to the next layer of neurons or returned as the model output.

a) $a_i = max(0, n_i)$ b) $a_i = \tanh(n_i)$ c) $a_i = \frac{1}{1+e^{-n_i}}$

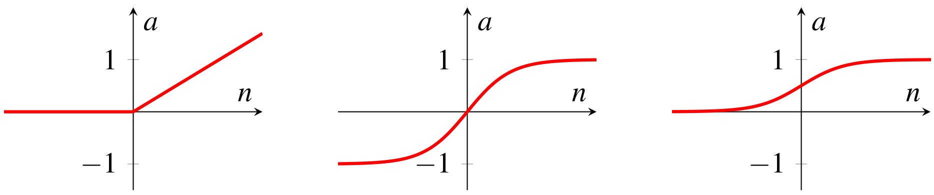

Figure 1: Commonly used activation functions.

3 EXACT REPRESENTATION OF 1D ELASTO-PLASTICITY

This section provides a brief overview of the von Mises algorithm for 3D elasto-plasticity. In small strains, deformation of elasto-plastic solids is determined by adding elastic strain $\boldsymbol{\varepsilon}^e$ and plastic strain $\boldsymbol{\varepsilon}^p$ components. The stress $\boldsymbol{\sigma}$ can be expressed by the elastic constitutive equation as:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \tag{1}$$

$$\boldsymbol{\sigma} = \mathbf{C}[\boldsymbol{\varepsilon}^e] = \mathbf{C}[\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p] \tag{2}$$

The yield criterion with no hardening can be expressed by:

$$f(\boldsymbol{\sigma}) = \mathbf{s} - \sqrt{\frac{2}{3}}(\sigma_y) \leq 0 \tag{3}$$

where $\mathbf{s}$ is the deviatoric stress tensor and $\sigma_y > 0$ is the yield stress. The flow rule equation reads:

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda}\mathbf{n} \tag{4}$$

where $\mathbf{n}= \frac{\mathbf{s}}{||\mathbf{s}||}$ is the unit normal on the yield surface and $\dot{\lambda}$ is the amount of plastic strain rate. The standard Kuhn-Tucker loading/unloading conditions are given as follows:

$$\dot{\lambda} \geq 0, \qquad f(\boldsymbol{\sigma}) \leq 0, \qquad \dot{\lambda} f(\boldsymbol{\sigma}) = 0 \tag{5}$$

The standard elastic predictor - plastic corrector backward Euler method is used for the numerical integration of elasto-plastic model due to its accurate approximation for small plasticity deformation.

3.1 History-based strategy

The behavior of materials depends on the historical values of stresses and strains. Because of the time-series nature of stress and strain, RNN architectures can be utilised to predict material behavior. However, gradient-based training of RNNs typically suffers from vanishing and exploding gradients [11]. In order to overcome the gradient problems a simplified architecture similar to the Nonlinear Autoregressive Network with Exogenous Inputs (NARX), which can be trained as a simple static feed-forward network, can be adopted. The NARX open-loop form is used for training and known as series-parallel architecture as it enables estimating the time-series for the next time-step. It includes a defined sequence of past values of inputs $u(t)$ and past measured values of outputs $y(t)$, and can be defined as:

$$\hat{y}(t+1) = f(y(t), y(t-1), ..., y(t-n_y), u(t+1), u(t), u(t-1)..., u(t-n_u)) \tag{6}$$

Once the training process is completed, the NARX-based neural model utilizes the close-loop form known as parallel architecture. In close loop the previous values of the output $(y(t), y(t-1), ..., y(t-n_y))$ are redirected back to the input layer. The values of an independent (exogenous) input $u(t)$ are also introduced into the network as a history input vector that corresponds with each output $y(t)$. The lengths n_y and n_u need to be defined, which depend on the data and the problem restrictions. It is recommended to use less historical values to minimize the network size and consequently reduce computational cost. When higher accuracy is needed additional historical values could be considered. The sequence of the history inputs in this case can be defined as:

$$\hat{y}(t+1) = f(\hat{y}(t), \hat{y}(t-1), ..., \hat{y}(t-n_y), u(t+1), u(t), u(t-1)..., u(t-n_u)) \tag{7}$$

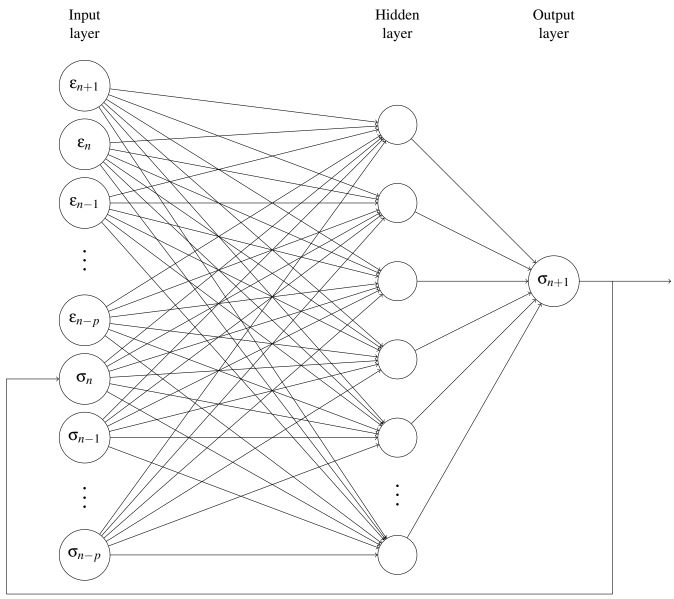

Figure 2: History-based strategy in terms of stresses and strains.

The modelling performance of a neural network is influenced by the neural architecture. Therefore, a case study on neural architectures was undertaken and the best architecture in terms of modeling accuracy, while requiring minimum number of hidden layers and nodes per layer was determined. Moreover, this study demonstrates how the complex elastic-plastic behavior could be reproduced using a model that involves only linear algebra operations and evaluation of a neural network model.

1D Ideal elasto-plasticity without hardening: For the history-based strategy, with no hardening, the proposed neural network suggest that a history of 1 is required to reproduce the elasto-plastic constitutive model output. The optimal network architecture achieved with an identical constitutive model representation is presented in Figure 3 and characterized with the following:

- An input layer that consists of 3 inputs of ε_{n+1}, ε_n and σ_n.
- Following that, the node connectivity is adjusted by assigning the right combination of weight W_{ij} and bias b_i parameters. The weighted sum is then delivered to the proceeding layer namely the first hidden layer which includes 4 neurons. The Rectified Linear Unit is selected as an activation function for the hidden layer.
- Finally, a linear activation function is used for the output layer and the defined weights for the output layer are used to modify and compute the next step σ_{n+1}

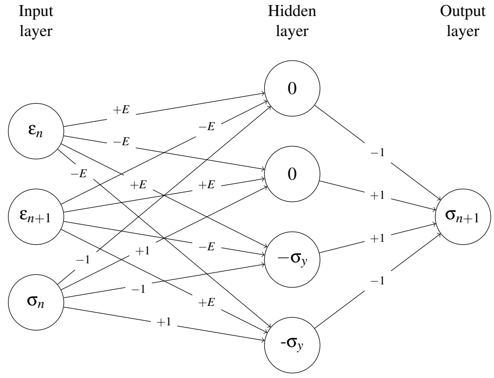

Figure 3: 1D ideal linear elasto-plasticity model in a network form.

The material properties of steel A36 were used in all the examples. The Young's modulus is chosen as E= 200 GPa, isotropic hardening modulus H= 10 GPa and yield stress σ_y= 250MPa.

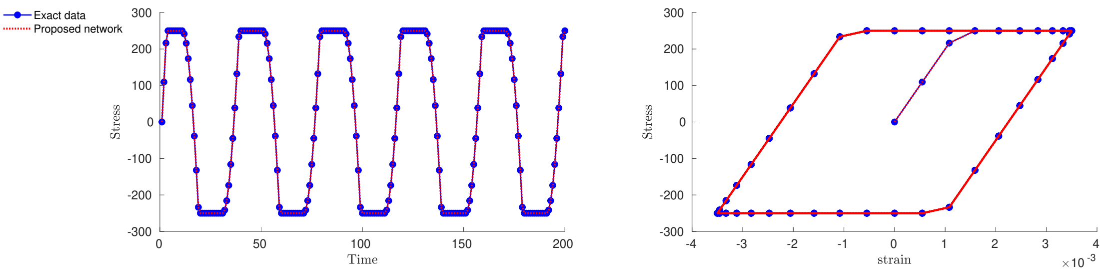

Figure 4: The performance of proposed history based strategy without hardening presenting stress over time on the left and the stress and strain relationship on the right.

1D elasto-plasticity with hardening: The complexity level increases when isotropic hardening is introduced. This is because of the internal hardening variable α which should be defined in terms of historical strains and the corresponding stresses. The study indicated that history of 2 is needed to get acceptable estimation of the elasto-plastic behavior when considering hardening. The derivation procedure to determine the equivalent term of α is briefly outlined:

- First, the total stresses during plasticity can be calculated by:

$$\begin{aligned} \sigma_{n+1} &= \sigma_{n+1}^{trial} - \frac{E}{E+H}\operatorname{sign}(\sigma_{n+1}^{trial})\Big(\sigma_{n+1}^{trial} - (\sigma_y + H\alpha_n)\Big) \\ \sigma_n &= \sigma_n^{trial} - \frac{E}{E+H}\operatorname{sign}(\sigma_n^{trial})\Big(\sigma_n^{trial} - (\sigma_y + H\alpha_{n-1})\Big) \end{aligned} \tag{8}$$

- Then, solving for α_n yields:

$$\alpha_n = \frac{\sigma_n^{trial} - \sigma_y + \operatorname{sign}(\sigma_n^{trial})(\sigma_n - \sigma_n^{trial})}{H} \tag{9}$$

- Finally, σ_{n+1} can be expressed in terms of historical strains and it corresponding stresses as:

$$\sigma_{n+1} = \sigma_{n+1}^{trial} - \frac{E}{E+H}\operatorname{sign}(\sigma_{n+1}^{trial})\left(\sigma_{n+1}^{trial} - \left(\sigma_y + H\left(\frac{\sigma_n^{trial} - \sigma_y + \operatorname{sign}(\sigma_n^{trial})(\sigma_n - \sigma_n^{trial})}{H}\right)\right)\right) \tag{10}$$

The proposed neural network for 1D elasto-plasticity when hardening introduced in Figure 5 could be summarised as:

1) The input layer which consists of 5 inputs including 2 historical stress and strain values of:

$$\boldsymbol{x} = \begin{bmatrix}\varepsilon_{n+1} & \varepsilon_n & \varepsilon_{n-1} & \sigma_n & \sigma_{n-1}\end{bmatrix}^T \tag{11}$$

2) As mentioned previously the complexity level increases when introducing isotropic hardening to the problem due to the hardening parameter α non-linearity. In this case, the defined total combination of weights is 40, while the total number of assigned biases is 8. The proposed neural network performance is not exact but with relatively acceptable level of accuracy to the 1D elasto-plasticity material behavior. In addition, the ReLU activation function is chosen for the hidden layer as explained below:

$$\mathrm{H} = \mathrm{ReLU}\begin{bmatrix} E & -E & 0 & 1 & 0 \\ -E & E & 0 & -1 & 0 \\ \frac{-E^2}{E+H} & \frac{E^2}{E+H} & 0 & \frac{-E}{E+H} & 0 \\ \frac{E^2}{E+H} & \frac{-E^2}{E+H} & 0 & \frac{E}{E+H} & 0 \\ 0 & \frac{-E^2}{E+H} & \frac{E^2}{E+H} & 0 & \frac{-E}{E+H} \\ 0 & \frac{E^2}{E+H} & \frac{-E^2}{E+H} & 0 & \frac{E}{E+H} \\ 0 & \frac{-E^2}{E+H} & \frac{E^2}{E+H} & \frac{E}{E+H} & \frac{-E}{E+H} \\ 0 & \frac{E^2}{E+H} & \frac{-E^2}{E+H} & \frac{-E}{E+H} & \frac{E}{E+H} \end{bmatrix}\begin{bmatrix}\varepsilon_{n+1} \\ \varepsilon_n \\ \varepsilon_{n-1} \\ \sigma_n \\ \sigma_{n-1}\end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{-E\sigma_y}{E+H} \\ \frac{-E\sigma_y}{E+H} \\ \frac{-E\sigma_y}{E+H} \\ \frac{-E\sigma_y}{E+H} \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

3) Finally, the output of the neural network is computed with the following mentioned weights $W_{ij}^{(O)}$ and linear activation function is given for the output layer

$$W_{ij}^{(O)} = \begin{bmatrix}1 & -1 & 1 & -1 & -1 & 1 & 1 & -1\end{bmatrix}^T \tag{13}$$

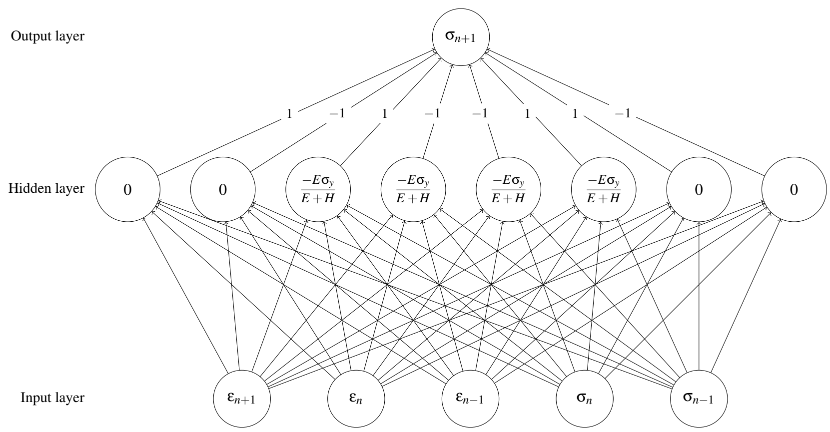

Figure 5: 1D proposed elasto-plasticity with hardening model in a network form.

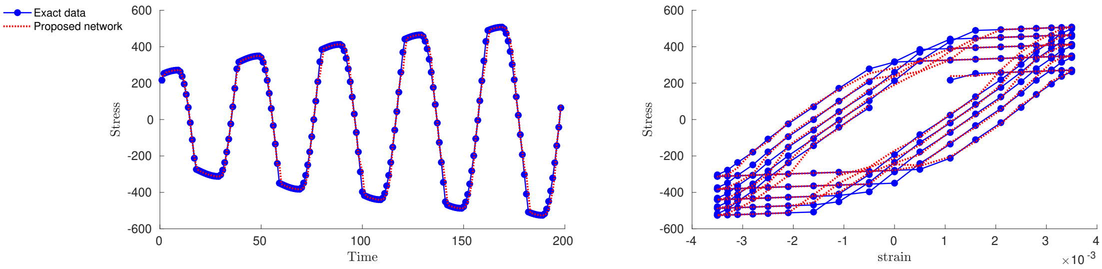

Figure 6: The response of the suggested architecture for history based strategy with hardening presented in stress over time (left) and stress and strain relationship (right).

Referring to Figure 6 the material behavior during both elastic and plastic phases can be reproduced with a good level of accuracy. However, the predicted behavior for the transition from elastic to plastic phase is relatively poor. In order to enhance the network response performance additional historical stresses and strains would be essential.

3.2 Internal variable-based strategy

Two types of state variables can be identified in any physical system which are:

- *Observable (measurable):* can be determined directly by physical measurements such as strains and temperature.
- *Internal non-observable (not directly measurable):* an integration process that works with other observable magnitudes and are dependant on the system specific structure. For example, internal stresses, plastic strains and damage depend on the system historical values, and record the internal changes in the micro-structure.

The significance of the proposed methodology is that only observable values are needed to train the network, and the internal state equations may be derived as a result of the training procedures. For this case, a two network formulation is used to model the elastic predictor and plastic corrector return mapping algorithm. The first network takes in the input strain and current state hidden variables. The number of hidden variables depends on the problem and degree of accuracy. Network 1 estimates the increment in the hidden variable which is used to update the current state internal variables before passing that information to Network 2. Finally, stress response can be obtained by performing a linear operation.

1D Ideal elasto-plasticity without hardening: The return mapping algorithm is used to develop the proposed neural network constitutive model. The study concluded that only one internal variable is needed to reproduce 1D elastoplasticity without hardening. The following is an explanation of the steps used by the proposed two network formulation model presented in Figure 7:

1) The first neural network computes the plastic corrector in the return mapping procedure. The network consists of an input layer with 2 neurons then 1 hidden layer with 2 neurons and finally an output layer with 1 neuron. The neurons in the hidden layers have ReLU activation functions while the neuron in the output layer has a linear activation function. The total strain ε_{n+1} and plastic strain ε_n^p are fed into the input layer and the network computes an incremental plastic strain $\Delta\varepsilon^p$.

2) Then the next step plastic strain ε_{n+1}^p is updated by adding the plastic strain ε_n^p and output of incremental plastic strain $\Delta\varepsilon^p$ from the first network.

3) Following that, the updated internal variable of ε^p_{n+1} is given to the second neural network to perform the elastic predictor step in the return mapping algorithm. The second neural network consist of two inputs of ε_{n+1} and ε^p_{n+1} connected with a linear activation function to an output layer holding the desired updated stress σ_{n+1}.

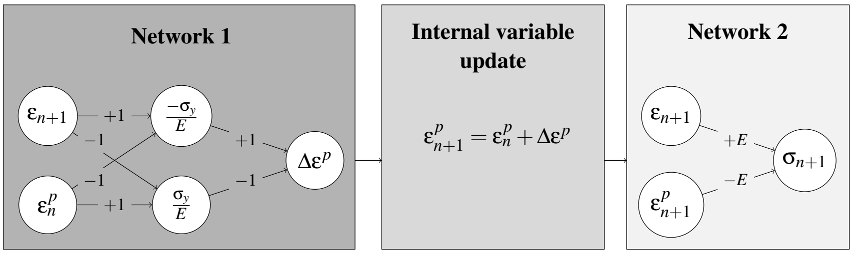

Figure 7: Two neural network formulation of 1D ideal linear elasto-plasticity model.

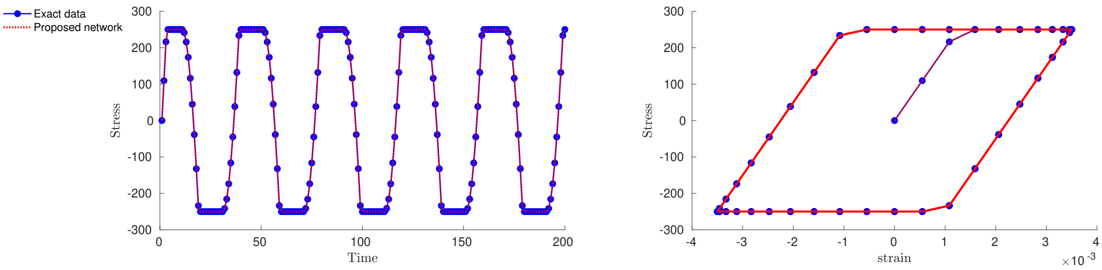

Figure 8: The two network formulation performance for 1D linear elasto-plasticity presented in stress over time on the left and stress and strain relationship on the right.

1D elastoplasticity with isotropic hardening: When introducing isotropic hardening to the problem the network would need two internal variables to reproduce the constitutive relationship. Similar procedure is followed to formulate the proposed neural network in Figure 9 and could be summarised as:
1) The first neural network takes three inputs namely the total strain ε_{n+1}, plastic strain ε^p_n and hardening parameter α_n. The input layer is linked to one hidden layer that include two neurons with ReLU activation function. The computed results are then passed to the output layer which includes two output neurons of $\Delta\varepsilon^p$ and $\Delta\alpha_n$.
2) Following that, the two internal variables of plastic strain ε^p_n and α_n are then updated by adding them to the incremental output computed in the first network as following:

$$\varepsilon^P_{n+1} = \varepsilon^P_n + \Delta\varepsilon^p \tag{14}$$
$$\alpha_{n+1} = \alpha_n + \Delta\alpha_n \tag{15}$$

3) Next, the updated internal variables are directed to the second neural network which perform a linear operation to represent the elastic predictor in the return mapping algorithm. The network consists of 3 inputs of ε_{n+1}, ε^p_{n+1} and α_{n+1} linked linearly with the updated stress σ_{n+1} in the output layer.

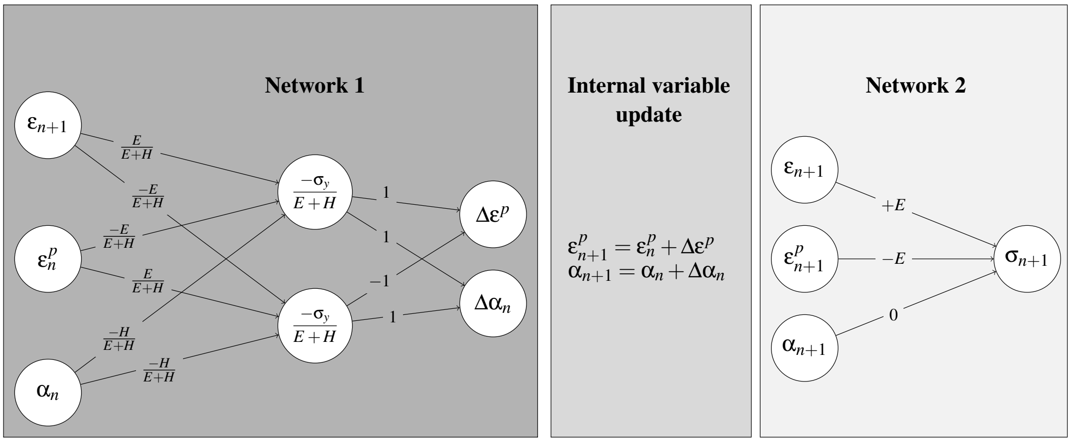

Figure 9: 1D elasto-plasticity with hardening model in a network form.

With the given architecture and defined combination of weights and biases in Figure 9 an identical stress response could be achieved as illustrated in Figure 10 below.

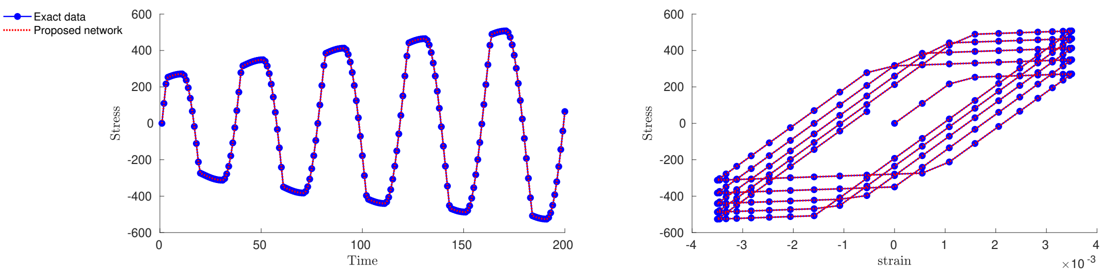

Figure 10: The proposed network response with internal variables strategy for 1D elasto-plasticity with hardening shown in terms of stress over time (left) and stress and strain relationship (right).

It is essential to mention that all the results shown in this work are not produced with neural network training but with the theoretically defined weights and biases in the proposed ideal minimum architectures for history-based and internal variable-based strategies. In addition, similar network performance could be achieved with alternative combination of weights and biases due to different scaling.

4 CONCLUSIONS

Two neural network-based strategies have been proposed and compared to reproduce the complex non-linear constitutive relations of solid materials with high computational efficiency which are history based and internal variable strategies. The data used for evaluating the proposed network performance were generated using the von Mises model, and implemented by the return mapping algorithm for elastic behavior and plastic deformation. The networks were designed to reproduce exactly the behavior of the established constitutive model. For history-based with isotropic hardening an acceptable level of accuracy is obtained but inaccuracies are observed when the direction of the strain loading is changed. Furthermore, it is clear that the history-based strategy would fail if the strain loading was kept constant over a number of time-steps. The recent history would not contain sufficient information to reconstruct the internal state of the material. This indicates that the performance of internal variable-based strategy is better suited to capture the elasto-plastic type material behavior.

REFERENCES

[1] Liu, X., Tian, S., Tao, F. and Yu, W. A review of artificial neural networks in the constitutive modeling of composite materials. *Composites Part B: Engineering* (2021) **224**:109152.

[2] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[3] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. Automatic differentiation in pytorch (2017).

[4] Ghaboussi, J., Garrett, J. and Wu, X. Knowledge-based modeling of material behavior with neural networks. *Journal of engineering mechanics.* (1991) **117**:132–153.

[5] Liu, X., Tao, F., Du, H., Yu, W. and Xu, K. Learning Nonlinear Constitutive Laws Using Neural Network Models Based on Indirectly Measurable Data. *Journal of Applied Mechanics* (2020) **87**:081003.

[6] Shen, Y., Chandrashekhara, K., Breig, W. and Oliver, L. Finite element analysis of V-ribbed belts using neural network based hyperelastic material model. *International Journal of Non-Linear Mechanics* (2005) **40**:875–890.

[7] Furukawa, T. and Yagawa, G. Implicit constitutive modelling for viscoplasticity using neural networks. *International Journal for Numerical Methods in Engineering.* (1998) **43**:195–219.

[8] Furukawa, T. and Hoffman, M. Accurate cyclic plastic analysis using a neural network material model. *Engineering Analysis with Boundary Elements.* (2004) **28**:195–204.

[9] Bengio, Y., Simard, P. and Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* (1994) **5**:157-166.

[10] Salehinejad, H., Sankar, S., Barfett, J., Colak, E. and Valaee, S. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078* (2017).

[11] Rehmer, A. and Kroll, A. On the vanishing and exploding gradient problem in Gated Recurrent Units. *IFAC-PapersOnLine* (2020) **53**:1243–1248.