Thermodynamics-based Artificial Neural Networks for constitutive modeling

Filippo Masi, Ioannis Stefanou, Paolo Vannucci, Victor Maffi-Berthier

Thermodynamics-based Artificial Neural Networks for constitutive modeling

Filippo Masi[a,b], Ioannis Stefanou[a,*], Paolo Vannucci[c], Victor Maffi-Berthier[b]

[a]Institut de Recherche en Génie Civil et Mécanique,
UMR 6183, CNRS, Ecole Centrale de Nantes, Université de Nantes,
1 rue de la Nöe, F-44300, Nantes, France.

[b]Ingérop Conseil et Ingénierie,
18 rue des Deux Gares, F-92500, Rueil-Malmaison, France.

[c]LMV, UMR 8100, Université de Versailles et Saint-Quentin,
55 avenue de Paris, F-78035, Versailles, France.

Abstract

Machine Learning methods and, in particular, Artificial Neural Networks (ANNs) have demonstrated promising capabilities in material constitutive modeling. One of the main drawbacks of such approaches is the lack of a rigorous frame based on the laws of physics. This may render physically inconsistent the predictions of a trained network, which can be even dangerous for real applications.

Here we propose a new class of data-driven, physics-based, neural networks for constitutive modeling of strain rate independent processes at the material point level, which we define as Thermodynamics-based Artificial Neural Networks (TANNs). The two basic principles of thermodynamics are encoded in the network's architecture by taking advantage of automatic differentiation to compute the numerical derivatives of a network with respect to its inputs. In this way, derivatives of the free-energy, the dissipation rate and their relation with the stress and internal state variables are hardwired in the architecture of TANNs. Consequently, our approach does not have to identify the underlying pattern of thermodynamic laws during training, reducing the need of large datasets. Moreover the training is more efficient and robust, and the predictions more accurate. Finally and more important, the predictions remain thermodynamically consistent, even for unseen data. Based on these features, TANNs are a starting point for data-driven, physics-based constitutive modeling with neural networks.

We demonstrate the wide applicability of TANNs for modeling elasto-plastic materials, using both hyper- and hypo-plasticity models. Strain hardening and softening are also considered for the hyper-plastic scenario. Detailed comparisons show that the predictions of TANNs outperform those of standard ANNs.

Finally, we demonstrate that the implementation of the laws of thermodynamics confers to TANNs high degrees of robustness to the presence of noise in the training data, compared to standard approaches.

TANNs ' architecture is general, enabling applications to materials with different or more complex behavior, without any modification.

Keywords: Data-driven modeling; Machine learning; Artificial neural network; Thermodynamics; Constitutive model.

Masi, et al. (2020) preprint, Journal of the Mechanics and Physics of Solids. doi: 10.1016/j.jmps.2020.104277

1. Introduction

A large spectrum of constitutive models have been proposed in the literature, based on observations and experimental testing. Existing constitutive laws can account for phenomena taking place at various length scales. This is achieved either through heuristic approaches and assumptions or through asymptotic approximations and averaging (e.g. Lloberas Valls et al., 2019; Nitka et al., 2011; Feyel, 2003; Bakhvalov and Panasenko, 1989). The history and the state of a material is commonly taken into account through ad hoc enrichment of simpler constitutive laws and extensive calibration. For this purpose, the laws of thermodynamics offer a useful framework for deriving more sophisticated laws, by intrinsically respecting the energy balance and the entropy production requirements (see e.g. Houlsby and Puzrin, 2000; Einav et al., 2007; Houlsby and Puzrin, 2007; Einav, 2012, among others).

An important limitation in constitutive modeling is the availability of data at different time- and length-scales. However, with the increase of computational power, it is nowadays possible to foresee micromechanical simulations that can account for realistic physics and explore stress paths and non-linear phenomena, which are experimentally inaccessible with the current methods. Of course, some constitutive assumptions will be always necessary, but these might be at a smaller scale, where the material properties are measurable and easier to identify. This scale is for instance the scale of the microstructure of a material (e.g. the scale of sand grains, crystals, alloys' grains, composites' fibers, masonry bricks' etc. including their topological configuration).

However, it is likely that the existing constitutive models might not be sufficient for describing complex material behaviors emerging from the microstructure. Therefore, calibration (parameter fitting) of known constitutive descriptors might be insufficient for representing the full space of material response, provided by sophisticated micromechanical simulations. Moreover, micromechanical simulations have currently a tremendous calculation cost, which is impossible to afford in large-scale, non-linear, incremental simulations (e.g. Finite Elements) that are usually needed in applications (cf. Masi et al., 2020, 2018; Rattez et al., 2018a,b; Collins-Craft et al., 2020; Lloberas Valls et al., 2019; Nitka et al., 2011; Eijnden et al., 2016; Feyel, 2003).

A promising solution to this issue seems to be Machine Learning. According to Geron (2015), "Machine Learning is the science (and art) of programming computers so they can learn from data". In the context of computer programming, learning is defined by Mitchell et al. (1997) as follows: "A computer program is said to learn from experience $\boldsymbol{E}$ with respect to some task $\boldsymbol{T}$ and some performance measure $\boldsymbol{P}$, if its performance on $\boldsymbol{T}$, as measured by $\boldsymbol{P}$, improves with experience". In the frame of constitutive modeling, a Machine Learning program can learn the stress-strain behavior of a material, given examples of stress-strain increments, which are either determined experimentally or through detailed micromechanical simulations. The data that the system uses to learn are called the training data-set and

*Corresponding author.

Email addresses: filippo.masi@ec-nantes.fr (Filippo Masi), ioannis.stefanou@ec-nantes.fr (Ioannis Stefanou), paolo.vannucci@uvsq.fr (Paolo Vannucci), victor.maffi-berthier@ingerop.com (Victor Maffi-Berthier)

each training example is called a training instance (or sample). In our case, the task $\boldsymbol{T}$, for instance, can be the prediction of the stress for a given strain increment and internal state of the material. The experience $\boldsymbol{E}$ is the training data-set and the performance measure $\boldsymbol{P}$ can be the prediction error. Machine Learning is a general term to describe a large spectrum of numerical methods. Some of them offer very rich interpolation spaces, which, in theory, could be used for approximating complicated functions belonging to uncommon spaces. Here we focus on the method of Artificial Neural Networks (ANNs), which is considered to be a sub-class of Machine Learning methods. According to Cybenko (1989) and Chen and Chen (1995), ANNs have proved to be universal approximators, due to their rich interpolation space. Therefore, they seem to be a useful and promising tool for approximating constitutive laws of many materials (e.g. sand, masonry, alloys, ceramics, composites etc.).

Recognizing this potential, there is an increasing amount of new literature employing ANNs successfully in constitutive modeling of non-linear materials from model identification based on experiments and detailed numerical simulations. Starting form the seminal work of Ghaboussi et al. (1991) and without being exhaustive, we refer to Ghaboussi and Sidarta (1998); Lefik and Schrefler (2003); Jung and Ghaboussi (2006); Settgast et al. (2019); Liu and Wu (2019); Lu et al. (2019); Xu et al. (2020); Huang et al. (2020); Liu and Wu (2019); Gajek et al. (2020); Gorji et al. (2020) and references therein. The main idea in these works is to appropriately train ANNs, feeding them with material data, and predict the material response at the material point level. In this sense ANNs can be seen as rich interpolation spaces, able to represent complex material behavior. For instance, we record the works of Heider et al. (2020); Ghavamian and Simone (2019); Mozaffar et al. (2019); Frankel et al. (2019); González et al. (2019); Gorji et al. (2020), who demonstrated that Recurrent Neural Networks (RNNs), an extension of neural networks, can be particularly useful for modeling path-dependent plasticity models. RNNs, differently from ANNs, process time sequences. As suggested by Gorji et al. (2020), the history-dependent variables of RNNs can potentially mimic the role of physical quantities.

The Boundary Value Problem (BVP), set to determine the behavior of a solid under mechanical and/or multiphysics couplings, is then solved by replacing the standard constitutive equations or algorithms by the trained ANN. This replacement is straightforward and non-intrusive in Finite Element (FE) codes. We record, without being exhaustive, the successful embedding of ANNs as material description subroutines in FE codes by Lefik and Schrefler (2003); Jung and Ghaboussi (2006); Lefik et al. (2009); Settgast et al. (2019). Ghavamian and Simone (2019) further implemented ANNs in a FE^2 scheme for accelerating multiscale FE simulations for materials displaying strain softening, with Perzyna viscoplaticity model.

It is worth emphasizing that the aforementioned data-driven approaches are different from another promising data-driven method (i.e., data driven computing Kirchdoerfer and Ortiz, 2016) in which the BVP is solved directly from experimental material data (measurements), bypassing the empirical material modeling step, involving the calibration of constitutive parameters (Kirchdoerfer and Ortiz, 2016; Ibañez et al., 2017; Kirchdoerfer and Ortiz, 2018,?; Ibanez et al., 2018; Eggersmann et al., 2019). While data-driven computing can be extremely powerful in many applications (Eggersmann et al., 2019), the first class of methods above-mentioned (based on the constitutive behavior at the material point level)

can be advantageous when modeling complex and abstract constitutive behaviors, which are not a priori known. Moreover, they can be used even if the BVP does not have a unique solution due to important non-linearities and bifurcation phenomena (e.g. loss of uniqueness, strain localization at the length of interest, multiphysics, runway instabilities etc.).

Nevertheless, until now ANNs for constitutive modeling are mainly used as a 'black-box' mathematical operator, which once trained on available data-sets, does not embody the basic laws of thermodynamics. As a result, vast amount of high quality data (e.g. with reduced noise and free of outliers) are needed to enable ANNs to identify and learn the underlying thermodynamic laws. Moreover, nothing guarantees that the predictions of trained ANNs will be thermodynamically consistent, especially for unseen data.

In this paper, we encode the two basic laws of thermodynamics in the architecture of neural networks. This assures thermodynamically consistent predictions, even for unseen data (which can exceed the range of training data-sets). Therefore, we assure thermodynamically consistent network's predictions, both for seen and unseen data (which can exceed the range of the training data-sets). Moreover, our network does not have to identify/learn the underlying pattern of thermodynamical laws. Consequently, smaller data-sets are needed in principle, the training is more efficient and the accuracy of the predictions higher. The price to pay, in comparison with existing approaches, is the need of two additional scalar functions (outputs) in the training data-set. These are the free-energy and the dissipation rate. However, these quantities are easily accessible in micromechanical simulations (e.g. Nitka et al., 2011; Eijnden et al., 2016; Feyel, 2003) and can also be obtained experimentally in some cases. Then, based on classical derivations in thermodynamics (e.g. Houlsby and Puzrin, 2007; Einav, 2012) specific interconnections are programmed inside our ANN architecture to impose the necessary thermodynamic restrictions. These thermodynamic restrictions concern the stresses and internal state variables and their relation with the free-energy and the dissipation rate. Our approach is inspired by the so-called Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019), in which reverse-mode autodiff (Baydin et al., 2017) is used, allowing the numerical calculation of the derivatives of an ANN with respect to its inputs.

The calculation of these derivatives, imposes some numerical requirements regarding the mathematical class of the activation functions to be used. More specifically, the internal ANN restrictions, derived from the first law of thermodynamics, require activation functions whose second gradient does not vanish. Otherwise, the problem of second-order vanishing gradients, as it is called here (cf. classical vanishing gradients problem in ANNs, e.g. Geron, 2015), can inhibit back-propagation and make training to fail. This new problem and its remedy is extensively explored and discussed herein.

For the sake of simplicity and for distinguishing our approach from existing ones, we call the proposed ANN architecture Thermodynamics-based Artificial Neural Networks (TANNs). In our opinion TANNs should be the starting point for data-driven and physics-based constitutive modeling at the material point level.

The paper is structured as follows. Section 2 presents a brief summary of the theoretical background of thermodynamics. In Section 3 an overview of the methodology proposed and architecture of TANNs is given. The main differences with classical, standard ANNs for

material constitutive modeling are also discussed. Particular attention is given to the choice of activation functions and the issue of second-order vanishing gradient is investigated in detail. Generation of material data-sets, with which the training of ANNs is performed, is presented in Section 4. In a first phase, we apply TANNs for the constitutive modeling of three-dimensional elasto-plastic material models, Section 5. In particular, we consider both hyper-plasticity models and smoother hypo-plasticity ones. Extensive comparisons with standard ANNs, which are not based on thermodynamics, are also presented. In a second phase, we investigate the performance and robustness of TANNs with the presence of noise in the training data, Section 6. This is achieved by generating a set of pseudo-experimental data, adding several levels of artificial noise. Supplementary figures and data are available in Supplementary data file. For the implementation of Artificial Neural Networks and Thermodynamics-based Artificial Neural Networks, we leverage Tensorflow v2.0. All code accompanying this manuscript is available upon request.

2. Thermodynamics principles: energy conservation and dissipation inequality

2.1. Energy conservation

A convenient way to express the (local) energy conservation is

$$\rho \dot{\mathrm{e}} = \boldsymbol{\sigma} \cdot \nabla^{\mathrm{Sym}} \mathrm{v} - \mathrm{div} \boldsymbol{q} + \rho h, \tag{1}$$

with ρ being the material density; e the specific internal energy (per unit mass); $\boldsymbol{\sigma}$ the Cauchy stress tensor; $\nabla \mathrm{v}$ the spatial velocity gradient tensor; $\boldsymbol{q}$ the rate of heat flux per unit area; h the specific energy source (supply) per unit mass, and "$\cdot$" denotes contraction of adjacent indices.

2.2. Second principle

The second law of thermodynamics can be formulated in terms of the local Clausius-Duhem inequality

$$\rho \dot{\mathrm{s}} \geq \frac{\rho h}{\theta} - \mathrm{div} \left(\frac{\boldsymbol{q} \cdot \boldsymbol{n}}{\theta} \right), \tag{2}$$

with s being the specific (per unit mass) entropy; h/θ and $-(\boldsymbol{q} \cdot \boldsymbol{n})/\theta$ the rate of entropy supply and flux, respectively. By removing the heat supply h between the energy equation (1) and the entropy inequality (2) leads to

$$\rho \left(\theta \dot{\mathrm{s}} - \dot{\mathrm{e}} \right) + \boldsymbol{\sigma} \cdot \nabla^{\mathrm{Sym}} \mathrm{v} - \frac{\boldsymbol{q} \cdot \nabla \theta}{\theta} \geq 0, \tag{3}$$

where the first two terms represent the rate of mechanical dissipation $\mathrm{D} = \rho \left(\theta \dot{\mathrm{s}} - \dot{\mathrm{e}} \right) + \boldsymbol{\sigma} \cdot \nabla^{\mathrm{Sym}} \mathrm{v}$ and the latter the thermal dissipation rate, i.e., $\mathrm{D}^{th} = -\frac{\boldsymbol{q} \cdot \nabla \theta}{\theta}$. The thermal dissipation is non-negative because heat only flows from regions of higher temperature to lower temperature–that is, the heat flux $\boldsymbol{q}$ is always in the direction of the negative thermal gradient. As it follows we argue that the mechanical dissipation rate must itself be non-negative (point-wise), i.e., $\mathrm{D} \geq 0$.

2.3. Dissipation function

The definition of the (mechanical) dissipation rate D leads to

$$\rho\dot{\mathrm{e}} = \rho\theta\dot{\mathrm{s}} + \boldsymbol{\sigma}\cdot\nabla^{\mathrm{Sym}}\mathrm{v} - \mathrm{D}. \tag{4}$$

Let us define the specific (per unit volume) internal energy $\mathrm{E} = \rho\mathrm{e}$ and entropy $\mathrm{S} = \rho\mathrm{s}$ and further assume constant material density, i.e., $\frac{\mathrm{d}}{\mathrm{d}t}\rho = 0$–that is, $\dot{\mathrm{E}} = \rho\dot{\mathrm{e}}$ and $\dot{\mathrm{S}} = \rho\dot{\mathrm{s}}$. We shall assume a small strain regime, i.e., $\nabla\boldsymbol{u} \ll 1$, with $\boldsymbol{\varepsilon} := \nabla^{\mathrm{Sym}}\boldsymbol{u}$ the small strain tensor, where $\boldsymbol{u}$ is the displacement vector field, and $\dot{\boldsymbol{\varepsilon}} := \nabla^{\mathrm{Sym}}\mathrm{v}$ its rate of change. Equation (4) hence becomes

$$\dot{\mathrm{E}} = \theta\dot{\mathrm{S}} + \boldsymbol{\sigma}\cdot\dot{\boldsymbol{\varepsilon}} - \mathrm{D}. \tag{5}$$

Let assume a strain-rate independent material such that the energy potential is

$$\mathrm{E} := \widetilde{\mathrm{E}}\left(\mathrm{S}, \boldsymbol{\varepsilon}, \mathcal{Z}\right), \tag{6}$$

and the dissipation rate, being a first-order homogeneous function of $\dot{\mathcal{Z}}$, is

$$\mathrm{D} := \widetilde{\mathrm{D}}\left(\mathrm{S}, \boldsymbol{\varepsilon}, \mathcal{Z}, \dot{\mathcal{Z}}\right), \tag{7}$$

where $\mathcal{Z} = (\zeta_i, \ldots, \zeta_N)$ denotes a set of N (additional) internal state variables, ζ_i, $i = 1, \ldots, N$. We define here (thermodynamic) state variables those macroscopic quantities characterizing the state of a system, see e.g. Maugin and Muschik (1994). The physical representation of ζ_i is not a priori prescribed. For instance, in the case of isotropic damage, ζ is a scalar; in anistotropic damage, a tensor; in the case of elasto-plasticity, a second order tensor, etc. The generalization to a finite-strain formulation can be achieved by considering the deformation gradient, $\boldsymbol{F}$, and the first Piola-Kirchhoff tensor, $\boldsymbol{P}$, as strain and stress measures, respectively (see e.g. Mariano and Galano, 2015; Anand et al., 2012). Nevertheless, as it would presented in Section 3, an incremental formulation of the material response is herein adopted. Therefore, the hypothesis of a small strain regime is usually realistic, at least for a large class of materials and an updated Lagrangian scheme.
Time differentiation of the internal energy gives

$$\dot{\mathrm{E}} = \frac{\partial \mathrm{E}}{\partial \mathrm{S}}\cdot\dot{\mathrm{S}} + \frac{\partial \mathrm{E}}{\partial \boldsymbol{\varepsilon}}\cdot\dot{\boldsymbol{\varepsilon}} + \sum_{i=1}^{N}\frac{\partial \mathrm{E}}{\partial \zeta_i}\cdot\dot{\zeta}_i, \tag{8}$$

which is equal to (5) and, grouping terms, it leads to

$$\left(\frac{\partial \mathrm{E}}{\partial \mathrm{S}} - \theta\right)\dot{\mathrm{S}} + \left(\frac{\partial \mathrm{E}}{\partial \boldsymbol{\varepsilon}} - \boldsymbol{\sigma}\right)\cdot\dot{\boldsymbol{\varepsilon}} - \left(\sum_{i=1}^{N}\frac{\partial \mathrm{E}}{\partial \zeta_i}\cdot\dot{\zeta}_i + \mathrm{D}\right) = 0. \tag{9}$$

The arbitrariness of $\dot{\mathrm{S}}$, $\dot{\boldsymbol{\varepsilon}}$, and $\dot{\zeta}$ leads to the following relations

$$\theta = \frac{\partial \mathrm{E}}{\partial \mathrm{S}}, \tag{10a}$$

$$\boldsymbol{\sigma} = \frac{\partial \mathrm{E}}{\partial \boldsymbol{\varepsilon}}, \tag{10b}$$

$$\sum_{i=1}^{N}\frac{\partial \mathrm{E}}{\partial \zeta_i}\cdot\dot{\zeta}_i + \mathrm{D} = 0. \tag{10c}$$

Further introducing the thermodynamic stress, conjugate to ζ_i, $\mathcal{X} = (\chi_1, \dots, \chi_N)$, with

$$\chi_i := -\frac{\partial \mathrm{E}}{\partial \zeta_i} \qquad \forall\, i \in [1, N]\,, \tag{11}$$

we obtain the following, alternative definition of the dissipation

$$\mathrm{D} = \sum_{i=1}^{N} \chi_i \cdot \dot{\zeta}_i \tag{12}$$

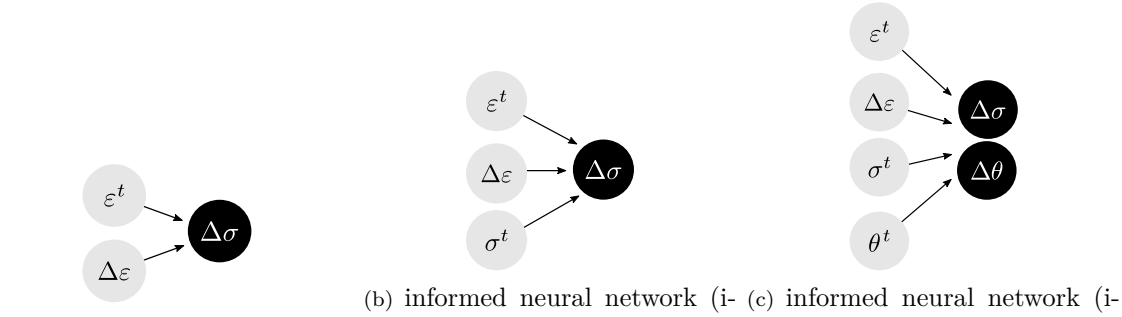
2.4. Isothermal processes

In the case of isothermal process, the (specific) Helmholtz free-energy, $\mathrm{F} := \mathrm{E} - \mathrm{S}\theta = \widetilde{\mathrm{F}}(\theta, \boldsymbol{\varepsilon}, \mathcal{Z})$, which is the Legendre transform conjugate of e, is preferable. In this case, the dissipation rate is such that $\mathrm{D} := \widetilde{\mathrm{D}}(\theta, \boldsymbol{\varepsilon}, \mathcal{Z}, \dot{\mathcal{Z}})$. The equations presented above (9-30) still hold (by replacing E with F)

$$\mathrm{S} = -\frac{\partial \mathrm{F}}{\partial \theta}, \quad \boldsymbol{\sigma} = \frac{\partial \mathrm{F}}{\partial \boldsymbol{\varepsilon}}, \quad \mathrm{D} = -\sum_i \frac{\partial \mathrm{F}}{\partial \zeta_i} \cdot \dot{\zeta}_i = \sum_i \chi_i \cdot \dot{\zeta}_i. \tag{13}$$

3. Thermodynamics-based Artificial Neural Networks

Within the framework of ANN or RNN material models, we can distinguish two main classes. The first consists of direct, so-called "black-box", approaches, where the information flow passes through the machine learning tool which operates as a mere regression operator, see e.g. Ghaboussi et al. (1991); Lefik and Schrefler (2003). The second class coincides with ANN and/or RNN models incorporating some knowledge in an informed, guided graph with intermediate history-dependent variables or detecting history-dependent features, see Heider et al. (2020); Mozaffar et al. (2019); Gorji et al. (2020), among others. Whilst the latter case has demonstrated to be extremely successful for path-dependent plasticity models, both classes are affected by the lack of physics, being the predictions not always compatible with thermodynamic principles (at least). Figure 1a depicts the direct approach



(a) black-box (BB) network.

(b) informed neural network (i-NN1).

(c) informed neural network (i-NN2).

Figure 1: Examples of direct, black-box (BB) (a) and informed (b, c) neural networks for material laws modeling. Inputs are highlighted in gray (●), outputs in black (●).

(BB), in which ANNs, usually Feed-Forward Neural Networks (FFNNs), are used to predict the stress increment, (output, $\mathcal{O}$) $\mathcal{O} = \Delta\sigma = \sigma^{t+\Delta t} - \sigma^t$, from the input $\mathcal{I} = (\varepsilon^t, \Delta\varepsilon)$, being ε^t the precedent strain and $\Delta\varepsilon$ its increment. In concise form, we write $\mathcal{O} = \text{BB@}\,\mathcal{I}$. In this scheme, ε^t and $\Delta\varepsilon$ can be regarded as the state variables, namely the ANN state variables (not necessarily coinciding with those introduced in Sect. 2), on which the updated material stress depends on. Two examples of guided, informed ANNs, either FFNNs or RNNs, are illustrated in Figures 1b and 1c. In both cases, the neural network intrinsically accounts for path-dependency, see e.g. Heider et al. (2020), making sequence of predictions of the main output. The network i-NN1 makes use of the last predicted output, i.e., σ^t, to make predictions of the next output, $\mathcal{O} = \Delta\sigma$. The inputs are hence $\mathcal{I} = (\varepsilon^t, \Delta\varepsilon, \sigma^t)$. We shall notice that, differently from BB, the stress at the precedent state, σ^t, is also considered to be an ANN state variable. Other alternatives exist in the selection of the ANN variables of state. One may chose, as we shall see in Section 5, (thermodynamic) state variables to be ANN state variables.
In the case of temperature-dependent material response, the second case (i-NN2) allows to make predictions that depend on the precedent temperature state, θ^t, namely $\mathcal{O} = \text{i-NN2@}\,\mathcal{I}$, with $\mathcal{I} = (\varepsilon^t, \Delta\varepsilon, \sigma^t, \theta^t)$ and $\mathcal{O} = (\Delta\sigma, \Delta\theta)$.

The main aim of this work is to change the classical paradigm of data-driven ANN material modeling into physics-based ANN material modeling. We propose a new class of ANNs based on thermodynamics, which are Thermodynamics-based Artificial Neural Networks (TANNs). By exploiting the theoretical background presented in Section 2, we propose neural networks which, by definition, respect the thermodynamic principles, holding true for any class of material. In this framework, TANNs posses the special feature that the entire constitutive response of a material can be derived from definition of only two (pseudo-) potential functions: an energy potential and a dissipation pseudo-potential (Houlsby and Puzrin, 2007). TANNs are fed with thermodynamics "information" by relying on the automatic differentiation technique (Baydin et al., 2017) to differentiate neural networks outputs with respect to their inputs. This strategy allows to construct a general framework of neural networks material models which, in principle, can be exploited to predict the behavior of any material and assure that the predictions of TANNs will be thermodynamically consistent even for inputs that exceed the training range of data. In this paper, we only focus on strain-rate independent processes. Moreover, our approach can be extended, following the developments in Houlsby and Puzrin (2000), to materials showing viscosity and strain-rate dependency.
The model relies on an incremental formulation and can be used in existing Finite Element formulations (among others), see e.g. Lefik and Schrefler (2003). Figure 2 illustrates the scheme of TANNs. The model inputs are the strain increment, the previous material state at time t, which is identified herein through the material stress, σ^t, temperature, θ^t, and the internal state variables, ζ_i^t, as well as the time increment Δt, namely $\mathcal{I} = (\varepsilon^t, \Delta\varepsilon, \sigma^t, \theta^t, \zeta_i^t, \Delta t)$. The primary outputs, $\mathcal{O}_1$, are internal variables increment, $\Delta\zeta_i$, the temperature increment, $\Delta\theta$, and the energy potential at time $t+\Delta t$, $\mathrm{F}^{t+\Delta t}$, i.e. $\mathcal{O}_1 = (\Delta\zeta_i, \Delta\theta, \mathrm{F}^{t+\Delta t})$. Secondary outputs, $\mathcal{O}_2$–that is, outputs computed by differentiation of the neural network with respect to the

inputs−are the stress increment, $\Delta\sigma$, and the dissipation rate, $\mathrm{D}^{t+\Delta t}$, which we denote as $\mathcal{O}_2 = \nabla_{\mathcal{I}}\mathcal{O}_1 = (\Delta\sigma, \mathrm{D}^{t+\Delta t})$.
The class of neural networks we propose differs from the previous ones by the fact that the quantity of main interest, i.e., the stress increment, is obtained as a derived one, which intrinsically satisfies the first principle of thermodynamics (and, as we shall see, the second principle, as well). In the following, we briefly recall the basic concepts of artificial neural networks (paragraph 3.1), we then focus on the issue of the second-order vanishing gradients that may afflict the training and the performance of an ANN model (paragraph 3.2). In particular, it is shown that, in the framework of Thermodynamics-based Artificial Neural Networks, particular attention has to be paid to the selection of activation functions. Finally, we present in detail the architecture of our model (paragraph 3.3).

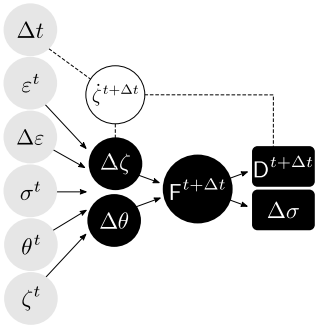


Figure 2: Schematic architecture of TANN. Inputs are highlighted in gray (●); outputs in black, (●) and (■); and intermediate quantities in white (○). Dashed lines represent definitions, while arrows are used to denote neural network links.

3.1. Artificial neural networks overview

We give herein a brief overview of the basic concepts of ANNs and in particular FFNNs. For more details, we refer to Hu and Hwang (2002) and Géron (2019). ANNs can be regarded as non-linear operators, composed of an assembly of mutually connected processing units−nodes−, which take an input signal $\mathcal{I}$ and return the output $\mathcal{O}$, namely

$$\mathcal{O} = \mathrm{ANN}@\mathcal{I}. \tag{14}$$

ANNs consist of at least three types of layers: input, output and hidden layers, with equal or different number of nodes. Figure 3 depicts a network composed of one hidden layer, with 3 nodes, an input layer with 2 inputs, and an output layer with 1 node. When an ANN has two or more hidden layers, it is called a deep neural network (Géron, 2019). Denoting the input array with $\mathcal{I} = (i_t)$, with $t = 1, 2 \ldots, n_{\mathcal{I}}$ ($n_{\mathcal{I}}$ is the number of inputs), and the outputs with $\mathcal{O} = \left(o_j\right)$, with $j = 1, 2 \ldots, n_{\mathcal{O}}$ ($n_{\mathcal{O}}$ is the number of outputs), the signal flows from layer $(l-1)$ to layer (l) according to

$$p_k^{(l)} = \mathcal{A}^{(l)}\left(z_k^{(l)}\right), \quad \text{with} \quad z_k^{(l)} = \sum_s^{n_N^{(l-1)}} \left(w_{ks}^{(l)} p_s^{(l-1)}\right) + b_k^{(l)}, \tag{15}$$

where $p_k^{(l)}$ are the outputs of node k, at layer (l); $\mathcal{A}^{(l)}$ is the activation function of layer (l); $n_{\mathcal{N}}^{(l-1)}$ is the number of neurons in layer $(l-1)$; $w_{ks}^{(l)}$ are the weights between the s-th node in layer $(l-1)$ and the k-th node in layer (l); and $b_k^{(l)}$ are the biases of layer (l). With reference to Figure 3, the output is given by

$$\begin{aligned} O = &\quad \mathcal{A}^{(o)}\left(z^{(o)}\right) \quad \text{with} \quad z^{(o)} = \textstyle\sum_r w_r^{(2)} p_r^{(1)} + b^{(2)} \\ p_r^{(1)} = &\quad \mathcal{A}^{(1)}\left(z_r^{(1)}\right) \quad \text{with} \quad z_r^{(1)} = \textstyle\sum_t w_{rt}^{(1)} i_t + b_r^{(1)}, \end{aligned}$$

where the activation function of the output layer, $\mathcal{A}^{(\text{out})}$, is a linear function, as in most of the cases for regression problems. The weights and biases of interconnections are adjusted, in an iterative procedure (gradient descent algorithm Géron, 2019), to minimize the error between the benchmark, $\bar{O}$, and prediction, O, that is measured by a loss function, $\mathcal{L}$. In the following, the Mean (over a set of N samples) Absolute Error (MAE) is used as loss function, i.e.,

$$\mathcal{L} = \frac{\sum_{i=1}^{N} |\bar{O}_i - O_i|}{N}, \tag{16}$$

where $i = 1, 2, \ldots N$. The errors related to each node of the output layer are hence back-propagated to the nodes in the hidden layers and used to calculate the gradients of the loss function, namely

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_{ks}^{(l-m)}} &= \frac{\partial z_k^{(l-m+1)}}{\partial w_{ks}^{(l-m)}} \frac{\partial p_k^{(l-m+1)}}{\partial z_k^{(l-m+1)}} \frac{\partial \mathcal{L}}{\partial p_k^{(l-m+1)}} \\ \frac{\partial \mathcal{L}}{\partial p_k^{(l-m)}} &= \sum_{j=1}^{n_{\mathcal{N}}^{(l-m+1)}} \frac{\partial z_j^{(l-m+1)}}{\partial p_k^{(l-m)}} \frac{\partial p_j^{(l-m+1)}}{\partial z_j^{(l-m+1)}} \frac{\partial \mathcal{L}}{\partial p_j^{(l-m+1)}} \end{aligned} \tag{17}$$

which are then used to update weights and biases, and force the minimization of the loss function values, i.e.

$$w_{ks}^{(l)-\text{new}} := w_{ks}^{(l)} - \epsilon \frac{\partial \mathcal{L}}{\partial w_{ks}^{(l)}}, \tag{18}$$

where ϵ is the so-called learning rate. The weights and biases updating, the so-called training process, is performed on a subset of the input-output data-set, defined as training set, known from experimental tests or numerical simulations of the phenomenon investigated. The ANN is trained. The training process is stopped as the loss function is below a specific tolerance. Then a test set, a subset of the input-output data-set different to the training set, is used to check the error of the network predictions. Once the ANN is trained, it is used in recall mode to obtain the output of the problem at hand.

Due to their rich interpolation space, ANNs have proved to be universal approximators, see e.g. Cybenko (1989); Chen and Chen (1995), although the choice of hyper-parameters, such as the number of neurons, the network topology, the weights, etc. are problem-dependent. The same stands for the activation functions, which may be chosen to have some desirable properties of non-linearity, differentiation, monotonicity, etc. Most of these properties stem

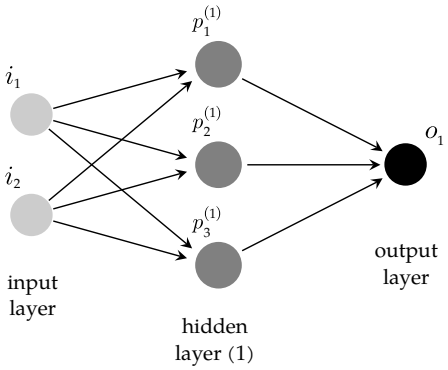


Figure 3: Graph illustration of an ANN structure with two inputs, one output, and one hidden layer with three nodes.

from issues related to the gradient descent algorithm and the so-called (first-order) vanishing gradient problem. As it follows, we briefly present this well-known issue and we further give insights in a variation of it: the second-order vanishing gradient.

3.2. First- and second-order vanishing gradients

During the training process, if the gradient of the loss function with respect to a certain weight tends to zero−that is, see Eq. (18), when $\mathcal{A}'^{(l)} = \partial p_j^{(l)}/\partial z_j^{(l)} \approx 0$ (with $\mathcal{A}'$ the first-derivative of the activation function with respect to its arguments)−the update operation can fail, and weights and biases are not updated. In this case, we have the so-called first-order vanishing gradient (Géron, 2019). Figure 4 displays some of the most common activation

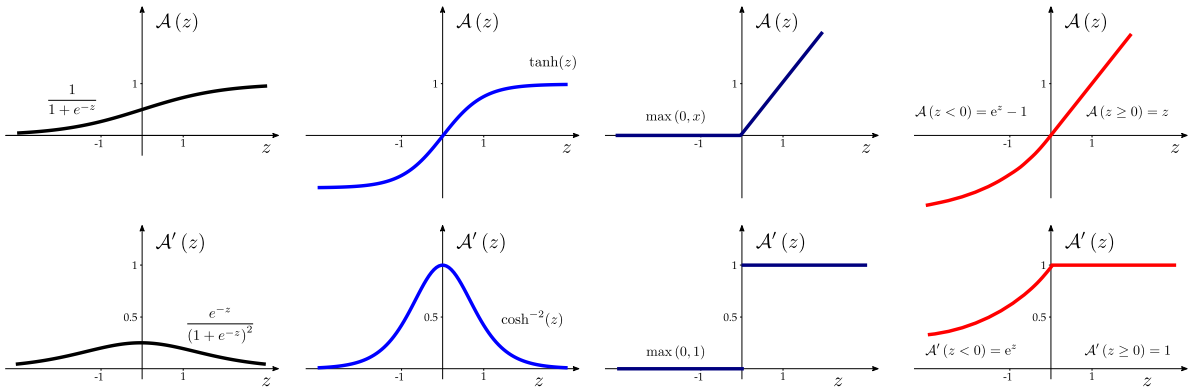


Figure 4: Some of the most common activation functions and their first-order gradient. From left to right: the logistic (sigmoid) function, the hyperbolic tangent, the Rectified Linear Unit (ReLU), and the Exponential Linear Unit (ELU).

functions and their derivatives−that is, the logistic (sigmoid) function, the hyperbolic tangent, the Rectified Linear Unit (ReLU), and the Exponential Linear Unit (ELU). The sigmoid function is S-shaped, continuous, differentiable, its output values range from 0 to 1, and its first-order gradient (derivative) assumes values much smaller than 1. When inputs become large (negative or positive), the function saturates at 0 or 1, with a derivative extremely close

to 0. Thus when backpropagation kicks in, it has virtually no gradient to propagate back through the network, which is problematic for training. The hyperbolic tangent activation function is very similar to the sigmoid, but it is centered at zero allowing to maintain the output values within a normalized range (between -1 and 1). Nevertheless, it suffers from saturated gradients (at $z = 0$, for $z << -1$ and $z >> 1$). ReLU is continuous but not differentiable at $z = 0$. Nevertheless it is an unsaturated activation function for positive values of z (its gradient has no maximum) and, therefore, it allows to avoid vanishing gradient issues for $z > 0$. Nevertheless, it suffers from a problem known as the dying ReLUs: during training, some neurons are effectively deactivated, meaning they stop outputting anything other than 0 (for $z < 0$). To this purpose many variants exist. The ELU activation, for instance, takes on negative values when $z < 0$, which allows the unit to have an average output closer to 0. This helps alleviate the vanishing gradient problem, as discussed earlier. Second, it has a nonzero gradient for $z < 0$, which avoids the dying units issue. Finally, the function is smooth everywhere, including z = 0, which helps speed up gradient descent.

When dealing with TANNs, second-order vanishing gradients can appear. This is a new concept and, in order to illustrate it, we will use a simple example. Assume an ANN which takes as input some $\mathcal{I} = x$ and returns (a) $O_1 = x^2$ and (b) its derivative with respect to the input, i.e., $O_2 = \nabla_{\mathcal{I}} O_1 = 2x$ (see Figure 5). Let us consider one hidden layer, with activation function $\mathcal{A}$ and N_n nodes. The activation function of the single output layer, which returns x^2, is assumed to be linear. In this case, the output (a) is given by

$$\begin{aligned} O_1 &= p^{(o)} = \mathcal{A}^{(o)}\left(z_k^{(o)}\right) \\ O_1 &= \sum_j w_j^{(o)} p_j^{(1)} + b^{(o)} \\ O_1 &= \sum_j w_j^{(o)} \mathcal{A}\left(w_j^{(1)} i + b_j^{(1)}\right) + b^{(o)}. \end{aligned} \tag{19}$$

The derivatives of the outputs with respect to the inputs can be easily computed, in this simple example, by taking advantage of the automatic (numerical) differentiation (Baydin et al., 2017). Output (b) is hence computed by the ANN as

$$\begin{aligned} O_2 = \nabla_{\mathcal{I}} O_1 = \frac{\partial O_1}{\partial \mathcal{I}} &= \sum_j \frac{\partial p^{(o)}}{\partial z^{(o)}} \frac{\partial z^{(o)}}{\partial p_j^{(1)}} \frac{\partial p_j^{(1)}}{\partial z_j^{(1)}} \frac{\partial z_j^{(1)}}{\partial \mathcal{I}} \\ \frac{\partial O_1}{\partial \mathcal{I}} &= \sum_j w_j^{(o)} w_j^{(1)} \mathcal{A}'\left(z_j^{(1)}\right). \end{aligned} \tag{20}$$

Consider the following loss function

$$\mathcal{L} = \mathrm{w}_o \mathcal{L}_o + \mathrm{w}_{\nabla_{\mathcal{I}} O} \mathcal{L}_{\nabla_{\mathcal{I}} O},$$

where $\mathcal{L}_o$ and $\mathcal{L}_{\nabla_{\mathcal{I}} O}$ are the loss functions corresponding to output O_1 and $O_2 = \nabla_{\mathcal{I}} O_1$, respectively. Regularized weights, w_o and $\mathrm{w}_{\nabla_{\mathcal{I}}}$, can be used to obtain comparable order

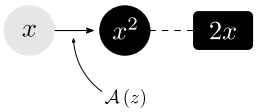


Figure 5: ANN which takes as input x and returns (a) $O_1 = x^2$ and (b) its derivative with respect to the input, i.e., $\nabla_I O_1 = 2x$, with one hidden layer whose activation function is $\mathcal{A}$.

of magnitude of the two loss functions. During training, weights and biases are updated according to Eq. (18) where the computed gradients are

$$\frac{\partial \mathcal{L}}{\partial w_j^{(o)}} = \mathcal{A}\mathcal{L}'_o + w_j^{(1)}\mathcal{A}'\mathcal{L}'_{\nabla_I O} \tag{21a}$$

$$\frac{\partial \mathcal{L}}{\partial w_j^{(1)}} = i w_j^{(o)}\mathcal{A}'\mathcal{L}'_o + \left(w_j^{(o)}\mathcal{A}' + i w_j^{(1)}\mathcal{A}''\right)\mathcal{L}'_{\nabla_I O} \tag{21b}$$

$$\frac{\partial \mathcal{L}}{\partial b^{(o)}} = \mathcal{L}'_o \tag{21c}$$

$$\frac{\partial \mathcal{L}}{\partial b_j^{(1)}} = w_j^{(o)}\mathcal{A}'\mathcal{L}'_o + w_j^{(o)} w_j^{(1)}\mathcal{A}''\mathcal{L}'_{\nabla_I O}. \tag{21d}$$

It follows, from relations (21b) and (21d), that the gradient descent algorithm needs the computation of both first- and second-order gradients of the activation function $\mathcal{A}$. This particular result is a direct consequence of the minimization of the error between the gradient of the outputs with respect to the inputs, i.e. $\boldsymbol{O}_2 = \nabla_I \boldsymbol{O}_1$, and the corresponding benchmark values, $2x$. This is what we call second-order vanishing gradient problem. It is tantamount to the first-order variant, but it involves the second derivatives (and not only the first) of the activation functions in an ANN. With reference to Figure 4, none of the depicted, classical activation functions is suitable for such class of problems. Consequently, care must be taken in selecting activation functions that do not have second-order vanishing gradients. To this purpose, Appendix A presents an example illustrating the issue of second-order vanishing gradients and proper solutions are given to this problem.

3.3. Architecture of Thermodynamics-based Artificial Neural Networks

Herein we detail the architecture and the internal steps/definitions TANNs are relying on. The architecture is detailed in Figure 6. The input vector is $\boldsymbol{\mathcal{I}} = (\boldsymbol{\varepsilon}^t, \Delta\boldsymbol{\varepsilon}, \boldsymbol{\sigma}^t, \theta^t, \boldsymbol{\zeta}_i^t, \Delta t)$, the primary and secondary outputs are $\boldsymbol{O} = (\Delta\boldsymbol{\zeta}_i, \Delta\theta, \mathrm{F}^{t+\Delta t})$ and $\nabla_{\boldsymbol{\mathcal{I}}}\boldsymbol{O} = (\Delta\boldsymbol{\sigma}, \mathrm{D}^{t+\Delta t})$, respectively. TANN involves the following steps:

1. computation of the updated strain (definition): $\boldsymbol{\varepsilon}^{t+\Delta t} := \boldsymbol{\varepsilon}^t + \Delta\boldsymbol{\varepsilon}$
2. prediction of the kinematic variables and temperature increments with two sub-ANNs:

$$\Delta\boldsymbol{\zeta} = \mathrm{sNN}_{\boldsymbol{\zeta}} @ \left(\boldsymbol{\varepsilon}^{t+\Delta t}, \Delta\boldsymbol{\varepsilon}^t, \boldsymbol{\sigma}^t, \theta^t, \boldsymbol{\zeta}^t\right)$$

and

$$\Delta\theta = \mathrm{sNN}_{\theta} @ \left(\boldsymbol{\varepsilon}^{t+\Delta t}, \Delta\boldsymbol{\varepsilon}, \boldsymbol{\sigma}^t, \theta^t, \boldsymbol{\zeta}^t\right)$$

3. computation of

   (*a*) the updated kinematic variables rates (backward finite difference approximation): $\dot{\zeta}^{t+1} \approx \frac{\Delta\zeta}{\Delta t}$

   (*b*) the updated kinematic variables (definition): $\zeta^{t+1} := \zeta^t + \Delta\zeta^t$

   (*c*) the updated temperature (definition): $\theta^{t+1} := \theta^t + \Delta\theta$

4. prediction of the updated energy potential:

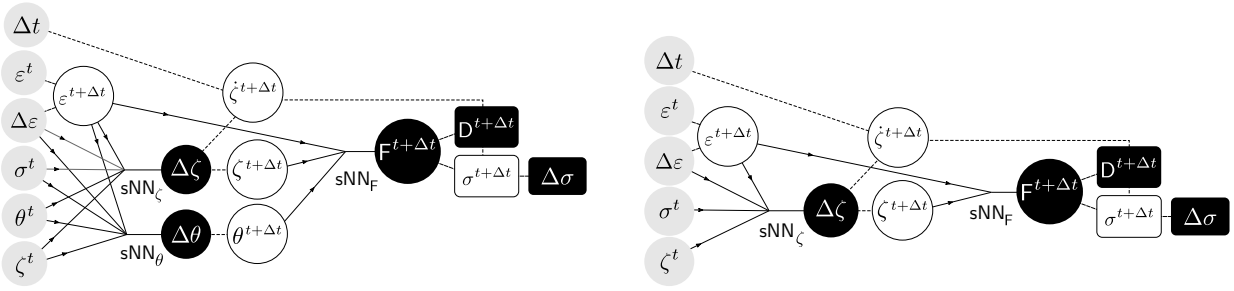
$$\mathrm{F}^{t+\Delta t} = \mathrm{sNN}_\mathrm{F}@\{\varepsilon^{t+\Delta t} \quad \zeta^{t+\Delta t} \quad \theta^{t+\Delta t}\}$$

5. computation of the updated dissipation rate (definition, Eq. (13)): $\mathrm{D}^{t+\Delta t} := -\frac{\partial \mathrm{F}^{t+\Delta t}}{\partial \zeta^{t+\Delta t}} \cdot \dot{\zeta}^{t+\Delta t}$

6. computation of

   (*a*) the updated stress (definition, Eq. (13)): $\sigma^{t+\Delta t} := \frac{\partial \mathrm{F}^{t+\Delta t}}{\partial \varepsilon^{t+\Delta t}}$

   (*b*) the stress increment (definition): $\Delta\sigma := \sigma^{t+\Delta t} - \sigma^t$



(a) non-isothermal processes.

(b) isothermal processes.

Figure 6: Architecture of TANNs: general case (a) and for isothermal processes (b). Inputs are highlighted in gray (●); outputs in black, (●) for direct ANN predictions and (■) for derived outputs; and intermediate quantities (definitions) are in white (○) and (▭). Relationships obtained from definitions are represented with dashed lines, while arrows denote ANNs.

TANNs are thus composed of three sub-ANNs; sNN_ζ predicts the internal variables increment, sNN_θ predicts the temperature increment (note that in case of the isothermal conditions, this component can be removed from the architecture, see Fig. 6b), and sNN_F predicts the Helmholtz free-energy. The main output, the increment in stress, is computed according to expression (13), which stems from thermodynamic requirements. By virtue of the fact that the entire constitutive response of a material can be derived from definition of only two (pseudo-)potential functions, the model is able to predict the stress increment from the knowledge of the energy potential (and the internal variables ζ_i). It is worth noticing that, differently from common approaches (cf. Sect. 3), the sub-network sNN_F is required to learn a scalar quantity–that is, the Helmholtz free-energy potential. This offer compelling advantages. When dealing with ANNs, the curse of dimensions (increasing effort in training

and large amount of training data required) is an important issue when the studied problem passes to higher dimensions, see e.g. Bessa et al. (2017). Passing from 1D to 3D, for instance, increases the number of variables the ANNs need to learn. For stresses, from one single scalar value, in 1D, we pass to a vector with six-components, in 3D. The computational effort is thus not trivial. Nevertheless, TANNs are, in principle, less affected by these issues as the two (pseudo-)potentials, on which the entire set of predictions relies on, are scalar functions. The computation of dissipation, from expression (13), plays a double role. First, it assures thermodynamic consistency of the predictions of TANNs (first law). Second, it brings the information to distinguish between reversible and irreversible processes, e.g. elasticity from plasticity/damage, etc., and it is trained to be positive or zero (second law).
It is worth noticing that further improvements of the performance of TANNs may be obtained, as suggested in the work of Karpatne et al. (2017), by adding a physical inconsistency term to the loss functions (e.g., with respect to dissipation).

4. Generation of data

We present the procedures used to generate material data TANNs are trained with in the following applications (see Sect. 5). We distinguish two different strategies. The first one, based on the numerical integration of an incremental form of the constitutive relations, is used to generate data for an hyper-plastic von Mises constitutive model with kinematic hardening (Houlsby and Puzrin, 2000, 2007). A different procedure is instead used to generate data for von Mises hypo-plasticity (Einav, 2012).

In the case of hyper-plasticity models, we assume the Ziegler's orthogonality condition (see paragraph 4.1 and Ziegler, 2012; Houlsby and Puzrin, 2000, 2007), which, in general, it is not a strict requirement. Nevertheless, it is worth noticing that this restriction applies only on the generated data, and not on the ANN class here proposed. More precisely, TANN architecture still holds even for materials for which the Ziegler's normality condition does not apply. We shall recall that the aim is to demonstrate the advantages of thermodynamics-based neural networks with respect to classical approaches. Hence the restrictions, imposed by the orthogonality hypothesis for the generation of data, are expected not to affect the comparisons presented in Section 5.

Hypo-plasticity is here used to show that the framework of thermodynamics encoded in TANNs is general and does not depend on restrictive assumptions such as the Ziegler's orthogonality condition afflicting hyper-plasticity. Furthermore, we consider the hypo-plastic material case to test TANNs against materials with a smooth response, which is more representative of realistic materials.

4.1. Incremental formulation

4.1.1. Hyper-plasticity

Following the hyper-plasticity framework proposed in Einav et al. (2007), the thermo-mechanical, non-linear, incremental constitutive relation for strain-rate independent materials,

undergoing infinitesimal strains, is here derived in the framework of isothermal processes (θ = cost). By differentiating the energy expressions (13) and rearranging the terms, we obtain the following non-linear incremental relations

$$\dot{\sigma} = \partial_{\varepsilon\varepsilon}\mathrm{F} \cdot \dot{\varepsilon} + \sum_k \partial_{\varepsilon\zeta_k}\mathrm{F} \cdot \dot{\zeta}_k, \tag{22a}$$

$$-\dot{\chi}_i = \partial_{\zeta_i\varepsilon}\mathrm{F} \cdot \dot{\varepsilon} + \sum_k \partial_{\zeta_i\zeta_k}\mathrm{F} \cdot \dot{\zeta}_k. \tag{22b}$$

where the following notation is adopted

$$\partial_{\varepsilon\varepsilon}\mathrm{F} = \frac{\partial^2 \mathrm{F}}{\partial\varepsilon_{ij}\partial\varepsilon_{kl}}, \quad \partial_{\varepsilon\zeta_k}\mathrm{F} = \frac{\partial^2 \mathrm{F}}{\partial\varepsilon_{ij}\partial\zeta_k}, \quad \partial_{\zeta_i\zeta_k}\mathrm{F} = \frac{\partial^2 \mathrm{F}}{\partial\zeta_i\partial\zeta_k}.$$

Further, introducing the thermodynamic dissipative stresses $\mathcal{X}^\dagger = (X_1, \ldots, X_N)$ and assuming the Ziegler's orthogonality condition (Ziegler, 2012), the following non-linear, incremental constitutive relation can be found

$$\dot{\Xi} = \begin{cases} \mathcal{M}|_{y=0} \cdot \dot{\varepsilon} & \text{if } y = 0 \\ \mathcal{M}|_{y<0} \cdot \dot{\varepsilon} & \text{else} \end{cases} \tag{23}$$

with

$$\dot{\Xi} = \begin{bmatrix} \dot{\sigma} \\ -\dot{X}_i \\ \dot{\zeta}_i \\ \lambda \end{bmatrix}, \quad \mathcal{M}|_{y=0} = \begin{bmatrix} \partial_{\varepsilon\varepsilon}\mathrm{F} - \sum_k \partial_{\varepsilon\zeta_k}\mathrm{F} \cdot \left(\frac{C_\varepsilon}{B} \cdot \frac{\partial y}{\partial X_k}\right) \\ \partial_{\zeta_i\varepsilon}\mathrm{F} - \sum_k \partial_{\zeta_i\zeta_k}\mathrm{F} \cdot \left(\frac{C_\varepsilon}{B} \cdot \frac{\partial y}{\partial X_k}\right) \\ -\frac{C_\varepsilon}{B} \cdot \frac{\partial y}{\partial X_i} \\ -\frac{C_\varepsilon}{B} \end{bmatrix}, \quad \text{and} \quad \mathcal{M}|_{y<0} = \begin{bmatrix} \partial_{\varepsilon\varepsilon}\mathrm{F} \\ \partial_{\zeta_i\varepsilon}\mathrm{F} \\ \emptyset \\ \emptyset \end{bmatrix}, \tag{24}$$

and $\cdot$ denotes the contraction of adjacent indices. In the above relations (23-24), whose derivation is presented in Appendix B, $y = \tilde{y}(\varepsilon, \mathcal{Z}, \mathcal{X}^\dagger)$ is the yield function, $\emptyset$ denotes a quantity (scalar or tensorial, depending on the dimensionality of the internal variable set) equal to zero, and

$$C_\varepsilon = \frac{\partial y}{\partial \varepsilon} - \sum_{i=1}^{N} \frac{\partial y}{\partial X_i} \cdot \partial_{\zeta_i\varepsilon}\mathrm{F},$$

$$B = \sum_{i=1}^{N} \frac{\partial y}{\partial \zeta_i} \cdot \frac{\partial y}{\partial X_i} - \sum_{i=1}^{N} \frac{\partial y}{\partial X_i} \left(\sum_{k=1}^{N} \partial_{\zeta_k\varepsilon}\mathrm{F} \cdot \frac{\partial y}{\partial X_k} \right).$$

4.1.2. Hypo-plasticity

The theoretical framework used here to generate the hypo-plastic data can be found in Einav (2012). Einav (2012) proposed a new theoretical model, called h^2plastic, unifying hypo- and hyper-plasticity models. In particular, compared to standard hypo-plasticity, the incremental material formulation can be derived from (pseudo-) potentials. The h^2plastic

model allows ease integration of the incremental constitutive equations, i.e., Eq. (5.15a) in Einav (2012).

Here we use the following incremental equations (Eq.s 7.14 and 7.15 in Einav, 2012) for the relaxation strain rate ($\dot{z}$) and stress increment ($\dot{\sigma}$), according to von Mises model:

$$\dot{z} = \frac{|\sigma' \cdot \dot{e}|}{2k^2} \left(\frac{1}{k} \sqrt{\frac{\sigma' \cdot \sigma'}{2}} \right)^{s-2} \cdot \sigma' \tag{25a}$$

$$\dot{\sigma} = K\dot{\varepsilon}_p + 2G \left(\dot{e} - \frac{1}{k} \sqrt{\frac{\sigma' \cdot \sigma'}{2}} \right)^{s-2} \cdot \sigma' \tag{25b}$$

where k represents the elastic limit in simple shear; s is a material parameter ($s > 0$); K and G are, respectively, the bulk and shear moduli; $\dot{\varepsilon}_p$ is the mean strain rate; $\dot{e}$ and $\dot{z}$ are, respectively, the deviatoric total and relaxation strain rate tensors; and σ' is the deviatoric stress.

4.2. Data generation

Data are generated in a Python environment, where SymPy and SciPy libraries are used for symbolic calculations and numerical integration. The accuracy of the generation process is 10^{-6} for strains and 10^{-4} MPa for stresses.

For the case of hyer-plasticity, data are generated by identifying an initial state for the material at time t,

$$\text{state at time } t: \qquad \Xi^t = \begin{bmatrix} \sigma^t \\ -X_i^t \\ \zeta_i^t \\ 0 \end{bmatrix} \quad \text{and} \quad \varepsilon^t,$$

and a given strain increment $\dot{\varepsilon}$, assuming constant and unitary time increment $\Delta t = 1$ ($\dot{\varepsilon}^t = \Delta\varepsilon^t$). Numerical integration of the ordinary differential equations (23) is performed with an explicit solver (Bogacki and Shampine, 1989) to obtain the state at the new time $t + \Delta t$, i.e.,

$$\text{state at time } t + \Delta t: \qquad \Xi^{t+\Delta t} = \begin{bmatrix} \sigma^{t+\Delta t} \\ -X_i^{t+\Delta t} \\ \zeta_i^{t+\Delta t} \\ \lambda^{t+\Delta t} \end{bmatrix}$$

For hypo-plasticity, data are generated similarly but only internal variables ζ_i, deformation ε, and stress σ are used to represent the material state at time t and $t+\Delta t$, through numerical resolution of Eq.s (25a) and (25b).

The training data play a crucial role for both the accuracy of the predictions and the generalization with respect to the ANN state variables, e.g., strain increments. The generalization capability of a network is here defined as the ability to make predictions for loading paths different from those used in the training operation. Nevertheless, a significant

dependency on the ANN state variables is usually observed. This may result in a poor network generalization. In Lefik and Schrefler (2003), an improvement of the generalization capability of ANNs is proposed. Artificial sub-sets of data, with zero strain increments, are added in the set of training data to force the network in learning that to zero input increments correspond zero output increments.
In the available literature, strain-stress loading paths are commonly used in training. If recursive neural networks are used, feeding them with history variables (loading paths) is the only possible solution (see e.g. Mozaffar et al., 2019). Nevertheless, ANNs do not necessary need the data-sets to be (historical) paths.

Herein, we generate data randomly. Conversely, this allows us to (1) improve the representativeness of the material data and (2) improve the generalization of the network on the strain increments. For the hyper-plastic material model, the initial state, Ξ^t and $\boldsymbol{\varepsilon}^t$, and the strain increment, $\Delta\boldsymbol{\varepsilon}$, are randomly generated from standard distributions with mean value equal to zero and standard deviation equal to $\Xi_{\max}$, $\boldsymbol{\varepsilon}^t_{\max}$, and $\Delta\boldsymbol{\varepsilon}_{\max}$, respectively. The Cauchy and thermodynamic stresses, $\boldsymbol{\sigma}^t$ and X_i^t, as well as the internal variables ζ_i^t are then calculated to satisfy the constraint $y^t \leq 0$. This incremental procedure is repeated for N_{samples}, resulting in a set of N_{samples} ordered pairs $\{\Xi^t, \boldsymbol{\varepsilon}^t, \Delta\boldsymbol{\varepsilon}; \Xi^{t+\Delta t}\}$, from which the corresponding energy potential and dissipation rate at time $t + \Delta t$ are evaluated. For the case of hypo-plasticity, data are generated by random loading paths as the procedure aforementioned for hyper-plasticity is not applicable to the theoretical framework in Einav (2012), as no definition of yield surface is needed for the derivation of the incremental material constitutive law. Figure 7 depicts the sampling for one of the investigated applications (see paragraph 5.1).

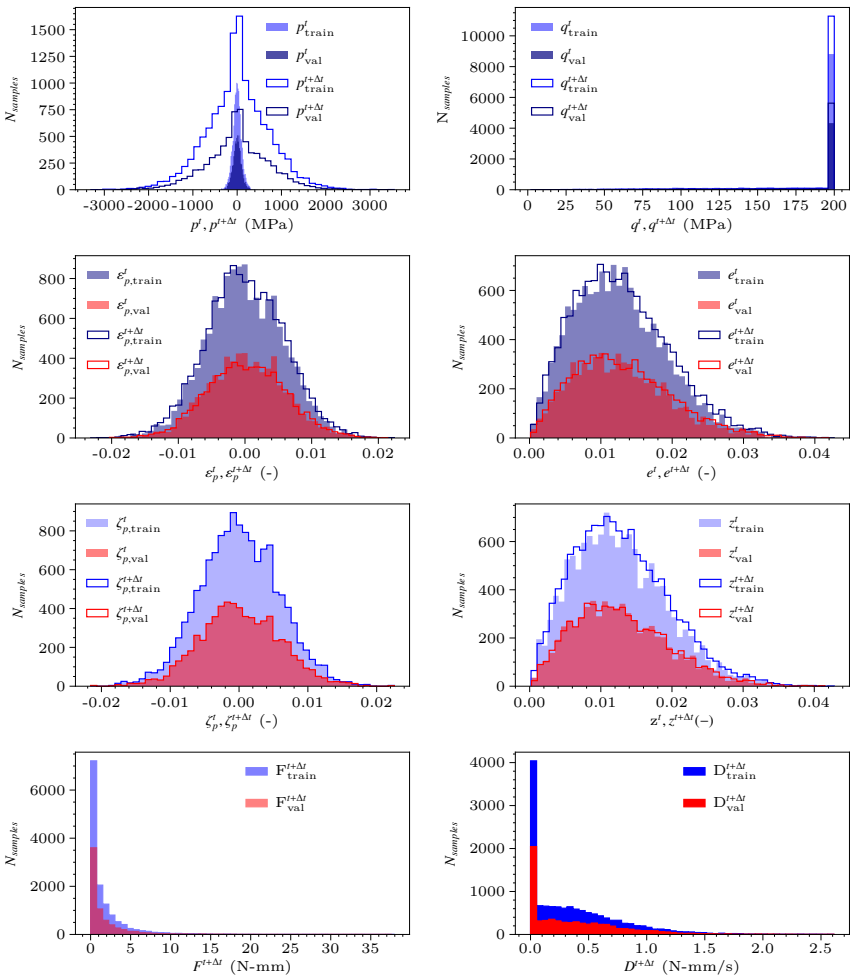


Figure 7: Sampling for material case H-1 (cf. Table 1). From top to bottom: mean and deviatoric stress (p and q); mean and deviatoric total deformation (ε_p and e); mean and deviatoric plastic deformation (ζ_p and z); energy (F) and dissipation rate (D). Training and validation data-sets are also distinguished.

5. Applications

Herein we use TANNs to the modeling of multi-dimensional elasto-plastic materials and demonstrate their wide applicability and effectiveness. It is worth noticing that, even though the applications here investigated consist of elasto-plastic materials, the proposed class of neural networks can be successfully applied (without any modification) to materials with different or more complex behavior, accounting e.g. for damage and/or other non-linearities (in the framework of strain-rate independent processes). In paragraph 5.1, von Mises hyper-plasticity is accounted for, considering perfect-plasticity, hardening and softening behaviors. In paragraph 5.1, we further investigate hypo-plastic material models.
In the examples presented herein, reference dependent variables, such as the total plastic strain, were considered. However, the internal state variables set $\mathcal{Z}$, see Eq. (7), and, consequently, our approach are not limited to this kind of state variables.

As it follows, the hyper-parameters (i.e., number of hidden layers, neurons, activation functions, etc.) of the networks are selected to give the best predictions, while requiring minimum number of hidden layers and nodes per layer. This is accomplished by comparing the learning error on the set of test patterns, per each trial choice of the hyper-parameters. In each training process, we use early-stopping. In other words, training is stopped as the error of a validation set starts to increase while the learning error still decreases (Géron, 2019). The validation set is used to avoid over-fitting of the training data.
Throughout this Section relatively simple deep feed-forward neural networks architectures are used (with, at maximum, two hidden layers) and no additional regularization techniques are employed (e.g., L1/L2 penalties, dropout, etc.). Each numerical example is accompanied with a detailed discussion about the network architecture.

5.1. von Mises hyper-plasticity

In order to illustrate the performance of TANNs, we use the simple von Mises elasto-plastic model with kinematic hardening and softening. The model can be derived from the following expressions of the energy potential and dissipation rate

$$\begin{aligned} \mathrm{F} &= \frac{9K}{2}\left(\varepsilon_p - \zeta_p\right)\cdot\left(\varepsilon_p - \zeta_p\right) + \\ &\quad + G\,(e - z)\cdot(e - z) + \frac{H}{2} z\cdot z, \\ \mathrm{D} &= k\sqrt{2}\,\sqrt{\dot{z}\cdot\dot{z}}, \end{aligned}$$

where k represents the elastic limit in simple shear; K and G are the bulk and shear moduli; H the hardening (softening) parameter; ε_p and ζ_p are, respectively, the mean total and plastic deformation; and e and z are, respectively, the total and plastic deviatoric strain tensors. The yield surface can be derived as shown in Appendix B (Houlsby and Puzrin, 2007) and is defined as

$$y = D - X'\cdot z = \sqrt{X'\cdot X'} - \sqrt{2}k \le 0, \tag{26}$$

with $X'_{ij} = 2G\left(e_{ij} - z_{ij}\right) + Hz_{ij}$.

Table 1: Material parameters for 3D elasto-plastic von Mises material.

case	K (GPa)	G (GPa)	k (MPa)	H (GPa)
H-1	167	77	140	0
H-2	167	77	140	-10
H-3	167	77	140	10

5.1.1. Training

Data are generated as detailed in Section 4. A total of 6000 data with random increments of deformation are generated. In order to improve the performance of the network in recall mode, additional sampling with random uni-axial and bi-axial loading paths are also used. The samples are split into training (50%), validation (25%), and test (25%) sets. The sampling in terms of the mean and deviatoric stresses, p and q, and deformations, ε_p and e, is presented in Figure 7 for material case H-1 (perfect plasticity). We distinguish between training and validation sets. For the sake of simplicity, stress and deformation are converted in the principal axes frame of reference. Table 2 shows the mean, standard deviation, and maximum values of the training data-sets. Adam optimizer with Nesterov's acceleration gradient (Dozat, 2016) is selected and a batch size of 10 samples is used. Data are normalized between -1 and 1.

We use the Mean Absolute Error (MAE), and not the Mean Square Error (MSE), as loss function for each output in order to assure the same precision between data of low and high numerical values. Regularized weights are used to have consistent order of magnitude of different quantities involved in the loss functions.

The network architecture is adapted to the size of the inputs and outputs, with respect to the mono-dimensional case. In particular, the sub-network sNN_ζ consists of two hidden layers, with 48 neurons (leaky ReLU activation function), and three output layers, one per each (principal) component of (increment of) ζ. The sub-network s-NN_F has one hidden layer with 36 neurons (activation ELU_{z^2}). The output layers for both sub-networks have linear activation functions and biases set to zero. The resulting number of hyper-parameters is ≈ 3000 (cf. Ghaboussi and Sidarta, 1998; ?; Lefik et al., 2009; Mozaffar et al., 2019).. Figure 8 displays the loss functions of each output as the training is performed, for material case H-1 (perfect plasticity). The early stopping rule assures convergence, after approximately 1000 epochs, with MAEs of the same order of magnitude for the 4 outputs, $\Delta\zeta$, $\text{F}^{t+\Delta t}$, $\Delta\sigma$, and $\text{D}^{t+\Delta t}$. The adimensional MAE is approximately equal to 1×10^{-4} for all outputs at the end of the training. Similar behaviors are also recovered for cases H-2 (softening), H-3 (hardening).

5.1.2. Predictions in recall mode

Once the network has been trained, it is used, in recall mode, to make predictions. We briefly present the performance of TANNs in predicting the material response for a random loading path. Figure 9 depicts the comparison with the target material model for material case H-1. The network displays extremely good performance and the ability to predict

Table 2: Mean (μ), standard deviation (st), and maximum values of the training data-sets.

data		μ	st	max
ε_i^t	(-)	8×10^{-5}	0.010	0.041
$\Delta\varepsilon_i$	(-)	2×10^{-5}	0.003	0.014
ζ_i^t	(-)	8×10^{-5}	0.010	0.041
$\Delta\zeta_i$	(-)	0	1×10^{-4}	0.0011
σ_i^t	(MPa)	-1.4	143	544
$\Delta\sigma_i$	(MPa)	123	7577	36744
$F^{t+\Delta t}$	(N-mm)	1.82	2.72	37.9
$D^{t+\Delta t}$	(N-mm/s)	0.41	0.38	2.61

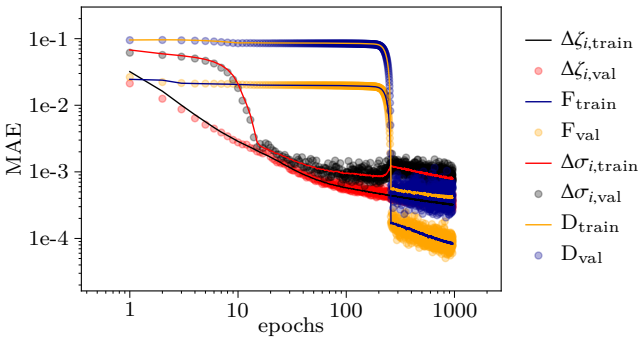


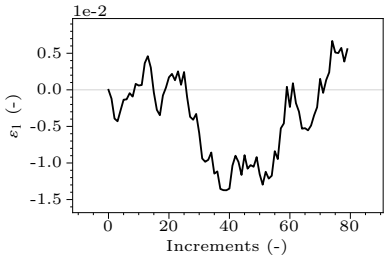
Figure 8: Errors in terms of the adimensional Mean Absolute Error (MAE) of the predictions of TANN (loss functions), as the training is being performed, evaluated with respect to the training (train) and validation (val) sets. Weights and biases update are computed only on the training set.

random loading path.

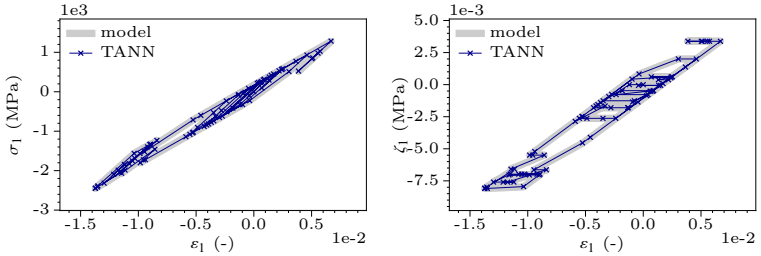
5.1.3. TANN vs standard ANN. Generalization of the network

Herein we investigate the performance of TANNs with respect to the classical approach of ANNs (Ghaboussi et al., 1991; Lefik and Schrefler, 2003), as well as the sensitivity with respect to the input variables range. Figure 10 displays the architecture of the network, ANN, with inputs $\mathcal{I} = (\varepsilon_i^t, \Delta\varepsilon_i, \sigma_i^t, \zeta_i^t)$ and output $\mathcal{O} = (\Delta\zeta_i, \Delta\sigma_i)$, with $i = 1, 2, 3$ denoting the principal components. The architecture is selected to give the best performance, preserving the same number of hyper-parameters between TANN and standard ANN. The network, ANN, consists of the two sub-networks, aNN_ζ and aNN_σ, with two hidden layers, each one, leaky ReLU activation functions, and number of neurons per layer equal to 48. As for sNN_ζ and sNN_σ, in aNN_ζ and aNN_σ three output layers (1 neuron each) are used, with linear activation functions and zero biases.
In Figure 11 we present the comparison of the MAE of the network predictions with respect to the target values (training and validation data-sets).

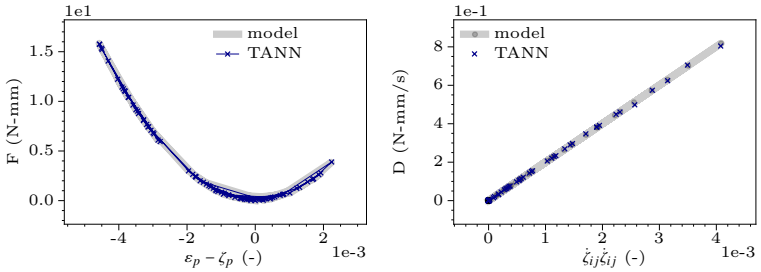
It is worth emphasizing that both ANNs and TANNs are dependent on the choice of the user, concerning, for instance, the hyper-parameters. Moreover, the actual configurations of both networks may benefit of alternatives/extensions, such as RNNs. Nevertheless, the



(a) random loading path.



(b) stress and internal variable.



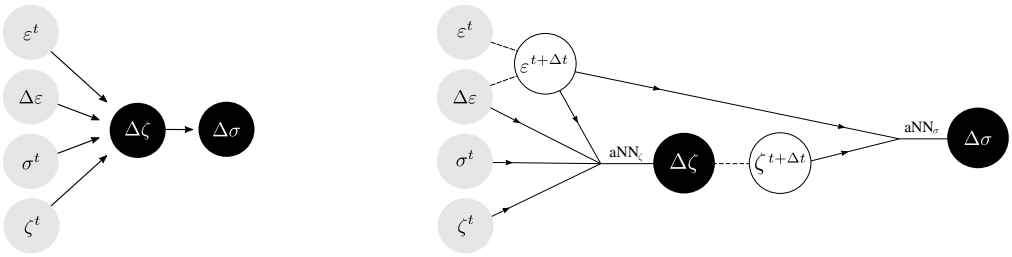
(c) energy and dissipation rate.

Figure 9: Predictions of TANN for a uni-axial random loading path, compared with the target constitutive model, case H-1, perfect plasticity: (a) loading path; (b) principal stress, σ_1, and internal variable, ζ_1, predictions; (c) energy and dissipation rate predictions.

following comparisons show the added value of our approach compared to standard ones that do not explicitly contain physics, as TANNs.

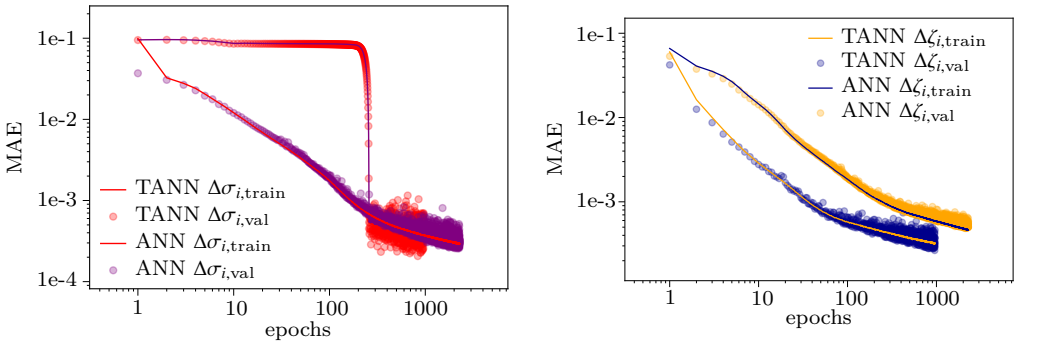
We first compare the performance of both networks, TANNs and standard ANNs, in predicting the material response for cyclic isotropic loading paths (material case H-1, cf. Table 1). A linear elastic material response is expected and retrieved. Figure 12 displays the stress predictions of TANNs and ANNs, compared with the target values, for different strain increments. It is worth mentioning that the standard approach of ANNs does not succeed in accurately predicting the elastic deformation range. Moreover, contrary to TANNs, the stress predictions of standard ANNs, depend strongly on the cyclic loading. As the network is used recursively, in recall mode, the stress predictions rapidly become less and less precise, due to error accumulation.

The performance of both networks is further compared for the following tri-axial loading



(a) ANN scheme. (b) ANN architecture.

Figure 10: Schematic (a) and full architecture (b) of the network, not based on thermodynamics, standard ANNs. Inputs are highlighted in gray (●), outputs in black (●).



(a) mean absolute error of $\Delta\zeta$ prediction (b) mean absolute error of $\Delta\sigma_i$ prediction

Figure 11: Training of ANNs compared with TANNs evaluated with respect to the training (train) and validation (val) sets.

path

$$\Delta\varepsilon_1^n = \Delta\varepsilon\ \text{sgn}\left(\cos\frac{n\pi}{2N}\right), \quad \Delta\varepsilon_2^n = \Delta\varepsilon_3^n = \Delta\varepsilon\ \text{sgn}\left(\sin\frac{n\pi}{2N}\right), \tag{27}$$

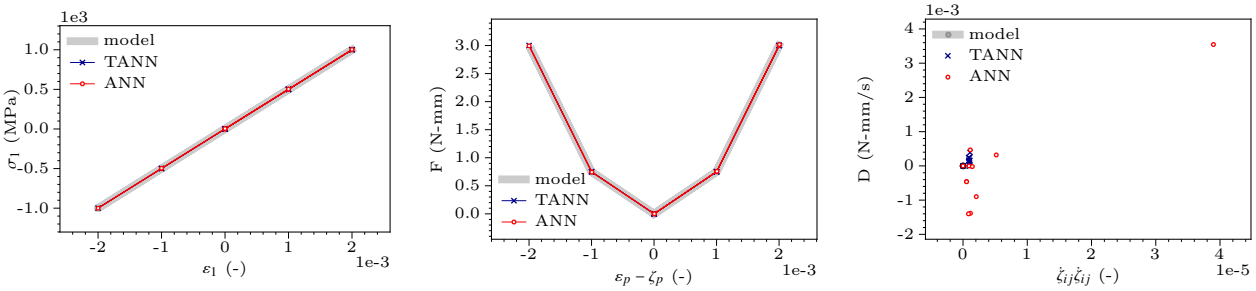
with $N = \varepsilon_{\max}/\Delta\varepsilon$, $\varepsilon_{\max} = 2\times10^{-3} \div 1$, and $\Delta\varepsilon = 1\times10^{-5} \div 1\times10^{-1}$.
Figures 13 and 14 display the material response in terms of the principal stresses, σ_1 and σ_3, and inelastic strains, ζ_1 and ζ_3, respectively. We show in Figure 15 the energy and dissipation rate predicted by TANNs with those computed, with standard ANNs, directly using the corresponding definitions for the free-energy and dissipation rate, Eq. (5.1). The predictions of TANNs are in good agreement with the constitutive model, independently from the strain increment, which exceeds considerably the training range. Nevertheless, the performance of ANNs is found to be strongly affected by the values of $\Delta\varepsilon$. For strain increments well inside the training range, i.e., $\Delta\varepsilon = 1\times10^{-3}$, standard ANNs are well predict the material response. In particular, computing, through Eq. (5.1, the dissipation rate and energy from the ANNs' predictions reveals that ANNs can successfully predict output respecting the requirements of the thermodynamics. The first and second principles of thermodynamics are indirectly learned during training (on thermodynamic consistent data). However, standard ANNs perform poorly for strain increments smaller and larger than the ones at which it was trained ($\Delta\varepsilon = 1\times10^3$, cf. Table 2). And in these cases,

standard ANNs predict thermodynamically inconsistent outputs.
The predictions of TANNs are, instead, always thermodynamically consistent. Moreover, the quantities of primary interest, such as the stress, the internal state variable, and the energy are in extremely good agreement with the reference model. The same stands also for the dissipation rate. We notice, once more, that its values are always positive, even when the network is used for predictions beyond the training range.
Figure S3 displays the predictions for very small strain increments, i.e. $\Delta\varepsilon = 1\times10^{-5}$. TANNs successfully still predict the response in this limiting case, while ANNs do not. Indeed, the training data were generated guaranteeing an accuracy of the order of 10^{-6} in terms of strains and such small strain increments are at the margin of the computing precision.

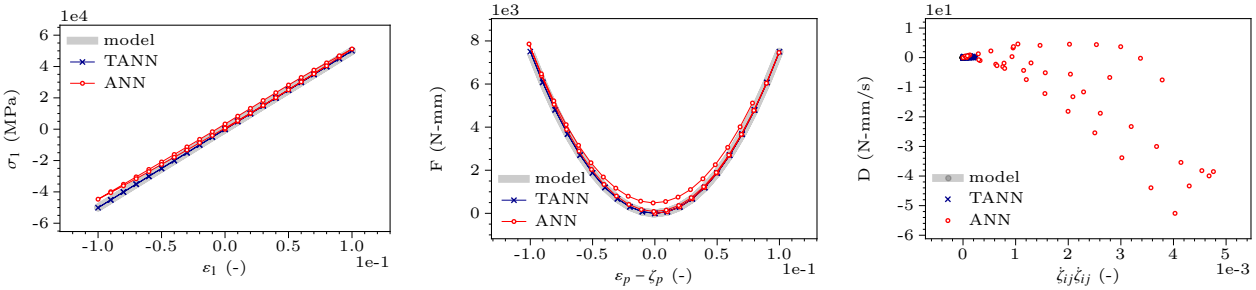
In the Supplementary Material, we present the results of a uni-axial loading scenario, in Figures S1 and S2, for material case H-1 (perfect plasticity). Kinematic hardening and softening material cases and the predictions of TANNs and ANNs are shown in Figures S4-S9.

It is worth noticing that in all the cases, even for very large strain increments–for which the predictions of the network in terms of dissipation rate differ from the target values–, TANNs successfully predict the Jacobian, i.e., $\frac{\partial\sigma_i}{\partial\epsilon_j}$ $(i, j = 1, 2, 3)$, in very good agreement with the reference model. This is of particular importance for numerical simulations with implicit algorithms. Therefore, TANNs can successfully replace complicated constitutive models or multiscale approaches, but considerably and safely decreasing the calculation cost, even when the requested increments are outside the training range.

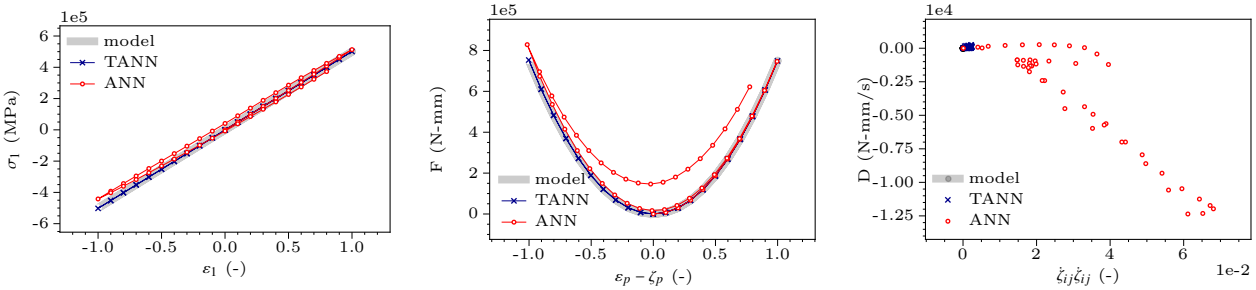
We emphasize that the performance of TANNs and standard ANNs can be improved by increasing the dimension of the training data-sets, the number of the hyper-parameters (e.g. numbers of hidden layers, etc.). Nevertheless, the fundamental gap between the two approaches in assuring thermodynamically consistent quantities still persist.



(a) strain increment $\Delta\varepsilon = 1 \times 10^{-3}$.

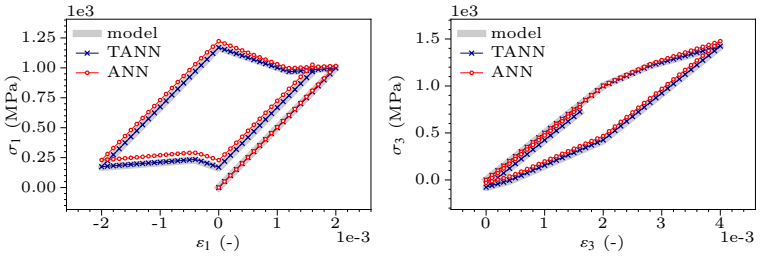


(b) strain increment $\Delta\varepsilon = 1 \times 10^{-2}$.

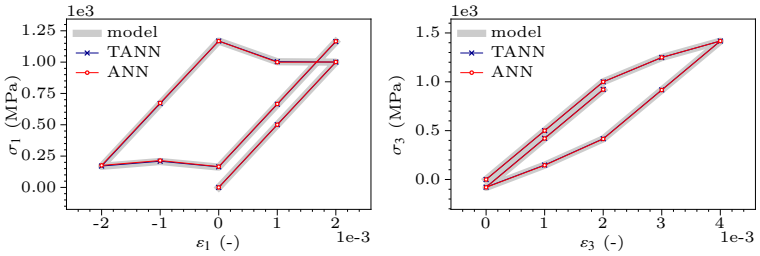


(c) strain increment $\Delta\varepsilon = 1 \times 10^{-1}$.

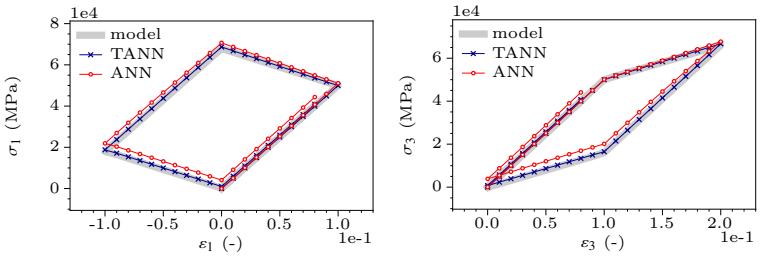
Figure 12: Comparison of the stress, energy, and dissipation predictions of TANNs and standard ANNs. Energy and dissipation for ANNs are computed according to Eq. (5.1), for the cyclic, isotropic loading path $\Delta\varepsilon_1^n = \Delta\varepsilon_2^n = \Delta\varepsilon_3^n = \Delta\varepsilon \operatorname{sgn}\left(\cos\frac{n\pi}{2N}\right)$ –with $N = \varepsilon_{\max}/\Delta\varepsilon$, $\varepsilon_{\max} = 2 \times 10^{-3}$ (a), $\varepsilon_{\max} = \times 10^{-1}$ (b), and $\varepsilon_{\max} = 1$ (c), for material case H-1 (perfect plasticity). Each row represents the prediction at different $\Delta\varepsilon$ increments.



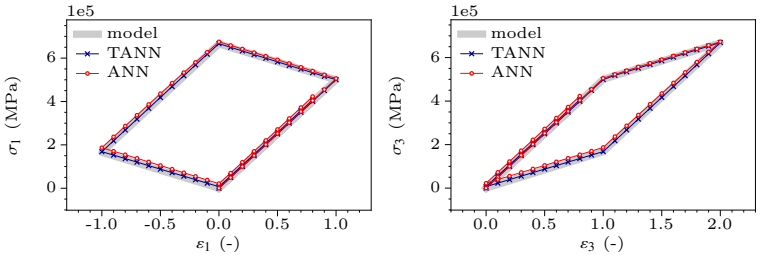
(a) strain increment $\Delta\boldsymbol{\varepsilon} = 1 \times 10^{-4}$.



(b) strain increment $\Delta\boldsymbol{\varepsilon} = 1 \times 10^{-3}$.

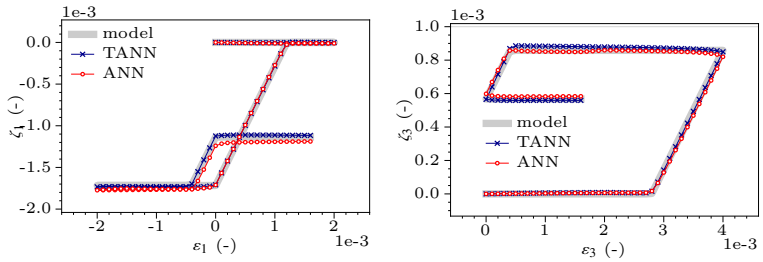


(c) strain increment $\Delta\boldsymbol{\varepsilon} = 1 \times 10^{-2}$.

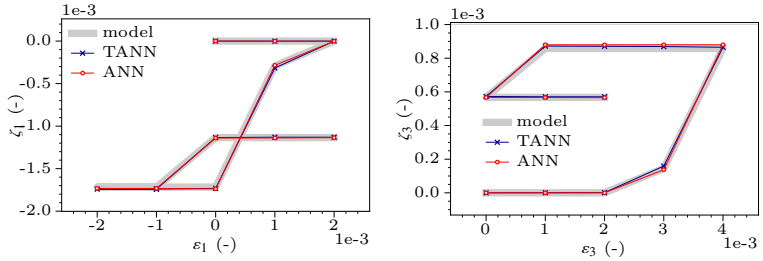


(d) strain increment $\Delta\boldsymbol{\varepsilon} = 1 \times 10^{-1}$.

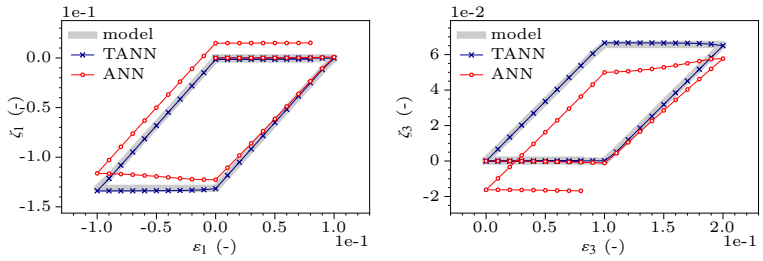
Figure 13: Comparison of the stress predictions of TANNs and standard ANNs with respect to the target values, for the tri-axial cyclic loading path, Eq. (27), for material case H-1 (perfect plasticity). Each row represents the prediction at different $\Delta\boldsymbol{\varepsilon}$ increments.



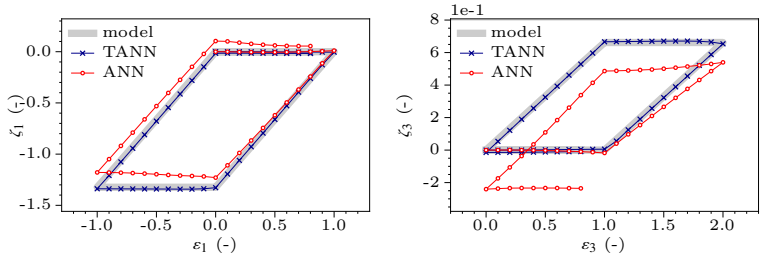
(a) strain increment $\Delta\varepsilon = 1 \times 10^{-4}$.



(b) strain increment $\Delta\varepsilon = 1 \times 10^{-3}$.

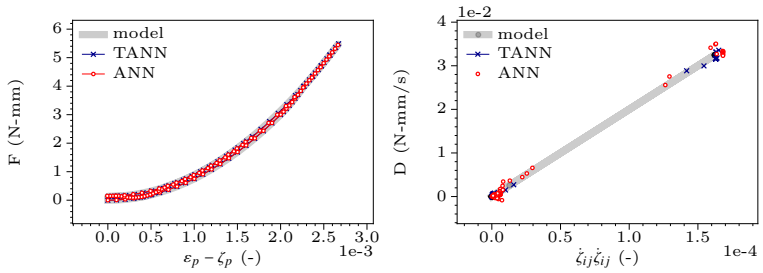


(c) strain increment $\Delta\varepsilon = 1 \times 10^{-2}$.

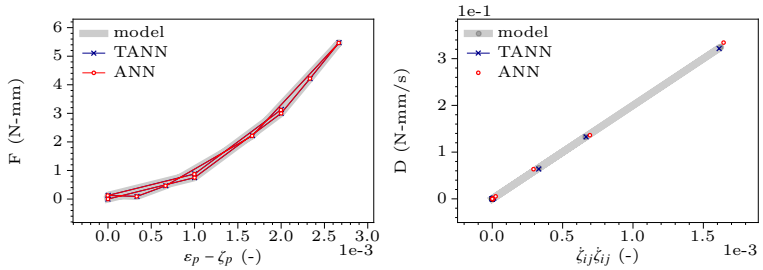


(d) strain increment $\Delta\varepsilon = 1 \times 10^{-1}$.

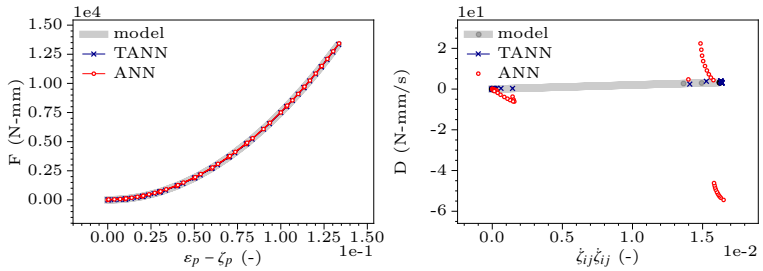
Figure 14: Comparison of the internal variable predictions of TANNs and standard ANNs with respect to the target values, for the tri-axial cyclic loading path, Eq. (27), for material case H-1 (perfect plasticity). Each row represents the prediction at different $\Delta\varepsilon$ increments.



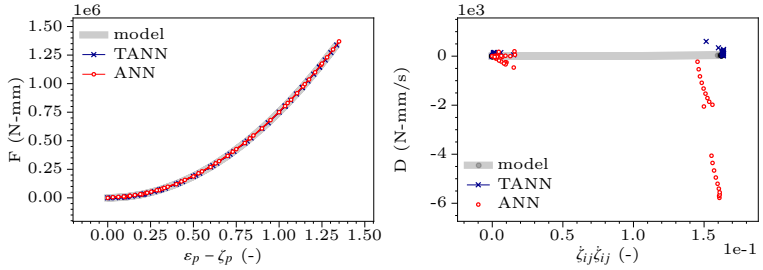
(a) strain increment $\Delta\varepsilon = 1 \times 10^{-4}$.



(b) strain increment $\Delta\varepsilon = 1 \times 10^{-3}$.



(c) strain increment $\Delta\varepsilon = 1 \times 10^{-2}$.



(d) strain increment $\Delta\varepsilon = 1 \times 10^{-1}$.

Figure 15: Comparison of the energy and dissipation rate predictions of TANNs and computation according to Eq. (5.1) for standard ANNs with respect to the target values, for the tri-axial cyclic loading path, Eq. (27), for material case H-1 (perfect plasticity). Each row represents the prediction at different $\Delta\varepsilon$ increments.

5.2. von Mises hypo-plasticity

We illustrate the performance of TANNs in predicting smooth material behaviors as well, modeled here using the hypo-plasticity model, presented in Einav (2012) and paragraph 4.1.2. The energy potential and dissipation rate are given by

$$\begin{aligned} \mathrm{F} &= \frac{9K}{2}\left(\varepsilon_p - \zeta_p\right)\cdot\left(\varepsilon_p - \zeta_p\right) + \\ &\quad + G\left(e - z\right)\cdot\left(e - z\right), \\ \mathrm{D} &= \sigma' \cdot \dot{z}, \end{aligned}$$

with $\dot{z}$ being defined in Eq. (25a). We consider $K = 167$ GPa, $G = 77$ GPa, and $s = 1$, see Eq.s (25a) and (25b).

Data are generated as detailed in Section 4. 8000 are data generated through random loading paths. As in the case of hyper-plasticity, additional sampling with random uni-axial and bi-axial random loading paths are also used. The samples are split into training (50%), validation (25%), and test (25%) sets. The sampling in terms of the mean and deviatoric stresses, p and q, and deformations, ε_p and e, is presented in Figure S10.

The architecture and hyper-parameters of TANNs are maintained equal to the hyper-plastic case (see paragraph 5.1). The internal variables ζ_i are selected to coincide with the inelastic strain. We emphasize that this particular choice does not affect the results of TANNs. As extensively discussed in Einav (2012), an alternative choice to the selection of the inelastic strain as internal variable can be the material porosity.

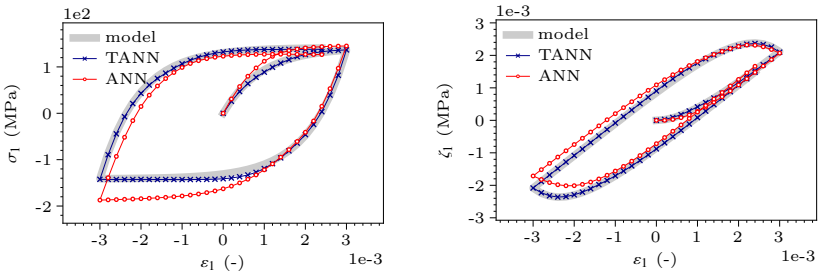
The early stopping rule assures convergence, after approximately 1000 epochs, with MAEs of the same order of magnitude for the 4 outputs, $\Delta\zeta$, $\mathrm{F}^{t+\Delta t}$, $\Delta\sigma$, and $\mathrm{D}^{t+\Delta t}$. The (adimensional) MAE is approximately equal to 1×10^{-4} for all outputs at the end of the training.

5.2.1. TANN vs standard ANN. Generalization of the network

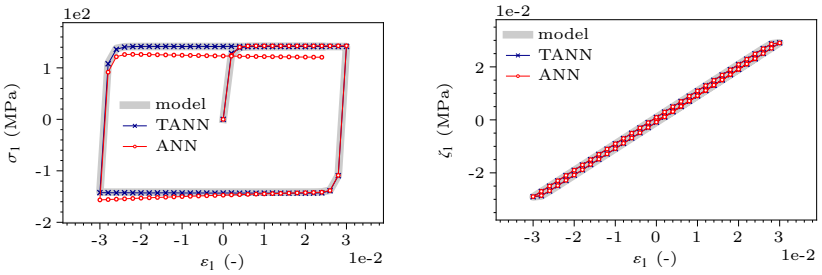
As for the hyper-plastic cases, we investigate the performance of TANNs with respect to standard ANNs through illustrative examples. The architecture and hyper-parameters of ANNs are maintained equal to the hyper-plastic case (see paragraph 5.1.3).
Figure 16 shows the predictions of both networks for the following bi-axial loading path

$$\Delta\varepsilon_1^n = -\Delta\varepsilon_2^n = \Delta\varepsilon\ \mathrm{sgn}\left(\cos\frac{n\pi}{2N}\right), \quad \Delta\varepsilon_3^n = 0,$$

with $N = \varepsilon_{\max}/\Delta\varepsilon$, $\varepsilon_{\max} = 2 \times 10^{-3} \div 1$, and $\Delta\varepsilon = 2 \times 10^{-4}, 2 \times 10^{-3}$. TANNs' predictions are in excellent agreement with the target model. The smoother material response, with respect to the hyper-plastic scenario, is well captured by the networks. Standard ANNs clearly underperform.



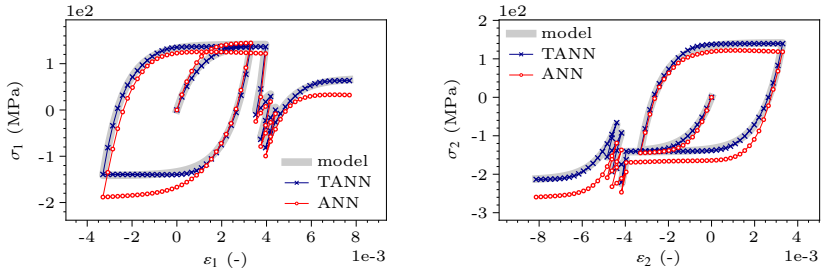
(a) strain increment $\Delta\varepsilon = 2 \times 10^{-4}$.



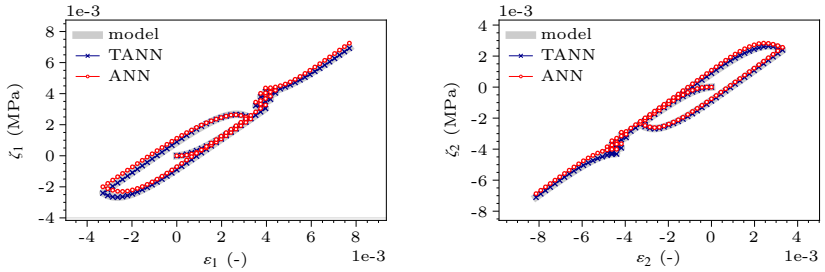
(b) strain increment $\Delta\varepsilon = 2 \times 10^{-3}$.

Figure 16: Comparison of the stress and internal variable predictions of TANNs and standard ANNs with respect to the target values, for a bi-axial cyclic loading path, $\Delta\varepsilon_2 = -\Delta\varepsilon_1$, with $\Delta\varepsilon_1$ as in Eq. (5.2.1), for a perfect hypo-plastic material. Each row represents the predictions at different $\Delta\varepsilon$ increments.

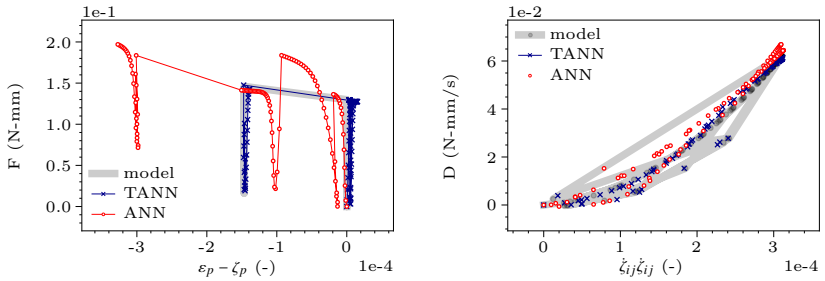
Additional demonstration of the performance of TANNs is given in Figure 17, for a bi-axial loading path with strain-controlled ratcheting. Ratcheting is a well known phenomenon shown by many materials during cyclic loading, which has been modeled here with the h^2plasticity framework (Einav, 2012). In particular, we show that TANNs, contrary to ANNs, successfully predict principal stresses, inelastic strains, energy potential, and dissipation rate.



(a) principal stresses.



(b) principal internal variables.



(c) energy and dissipation rate.

Figure 17: Comparison of the predictions of TANNs and standard ANNs with respect to the target values, for the bi-axial loading path with strain-controlled ratcheting, for a perfect hypo-plastic material.

6. Noise in training data and robustness of predictions

After having demonstrated the performance of TANNs and their superiority to standard approaches in modeling path-dependent material behaviors, we investigate the effect of noise in the measurements of the data used to train artificial neural networks. This is achieved by training TANNs (and ANNs) using the previously generated data and adding, in the training and validation sets, artificial noise. For sake of clarity, we consider a perfectly plastic material (case H-1, cf. Tab. 1). The additive noise, ns, is based on a normal distribution with standard deviation (sd) equal to 10% of the mean value of the clean data. In particular, we consider the following scenarios, independently:

(1) noise in $\zeta_i^{t+\Delta t}$, i.e., ns_ζ, with sd = 10% of the mean value of $\zeta_i^{t+\Delta t}$;

(2) noise in $\sigma_i^{t+\Delta t}$, i.e., ns_σ, with sd = 10% of the mean value of $\sigma_i^{t+\Delta t}$;

(3) noise in $\mathsf{F}^{t+\Delta t}$, i.e., ns_F, with sd = 10% of the mean value of $\mathsf{F}^{t+\Delta t}$; and

(4) noise in $\mathsf{D}^{t+\Delta t}$, i.e., ns_F, with sd = 10% of the mean value of $\mathsf{D}^{t+\Delta t}$.

We emphasize that the aforementioned noise levels were chosen to demonstrate the performance of TANNs and generally lower levels of noise are expected in practical applications. However, we examine each scenario independently in order to explore better the effect of noise on training and on the accuracy of the predictions. In cases (1) and (2), once the noised quantities are computed (denoted with $\bar{\sigma}_i^{t+\Delta t}$ and $\bar{\zeta}_i^{t+\Delta t}$), the increments, i.e., $\Delta\bar{\sigma}_i$ and $\Delta\bar{\zeta}_i$, are re-evaluated as $\Delta\bar{\sigma}_i = \bar{\sigma}_i^{t+\Delta t} - \sigma_i^t$ and $\Delta\bar{\zeta}_i = \bar{\zeta}_i^{t+\Delta t} - \zeta_i^t$, respectively.
The architecture and hyper-parameters of the neural networks, both TANNs and ANNs, designated in this study are the same as those used in Section 5. It should be noticed that, for each case (1-4), the data used to train the networks are not respecting the thermodynamics requirements due to the added noise, i.e., Eq. (13).

The addition of noise can have an impact on the training of the networks and their predictions. We first focus on the former. Figure 18 displays the loss functions of each output as the training of TANNs is performed, for noise added in stresses, case (2). The MAE is evaluated between the TANNs' predictions and the (noised) training and validation data-sets. Table 3 shows the MAEs of the predictions of TANNs with respect to the validation data-sets, for each level of noise, at the end of the training. Although the earlystopping technique is used, training is accomplished, in all cases of noise, after approximately 1000 epochs.

By comparing the training using the original, un-noised data (Fig. 8) and that using the noised ones (Fig. 18), we can observe that TANNs are unable to learn the noised signal, i.e., $\Delta\bar{\sigma}_i$. This is a direct consequence of the fact that the network evaluates the stress increments from the knowledge of the stress state at time t and the energy potential predictions. When noise is added, the first law of thermodynamics is violated and the training operation with noised data is unsuccessful, with respect to the noised training and validation data-sets. However this is not a drawback of our approach. On the contrary, it is an indication of the

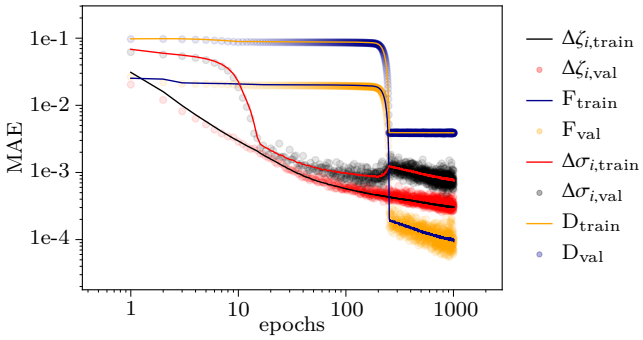


Figure 18: Errors of the predictions of TANN, as the training is being performed, evaluated with respect to the training (train) and validation (val) sets. Noise is added in stress to both training and validation data-sets, case (2).

Table 3: MAEs of the predictions of TANNs with respect to the validation data-sets, for the original, un-noised data and each level of noise, at the end of the training, Early-stopping is used and the training is always completed at approximately 1000 epochs.

	Mean Absolute Error (1e-4)			
	$\Delta\zeta_i$	$\Delta\sigma_i$	$\mathsf{F}^{t+\Delta t}$	$\Delta\mathsf{D}^{t+\Delta t}$
un-noised	3.2	4.1	0.8	7.5
ns_ζ	324	2.5	0.9	5.9
ns_σ	3.3	39	1.0	5.4
ns_F	3.3	3.1	19	7.3
ns_D	3.4	4.9	0.9	57

quality of the data, which in this case they don't respect the laws of thermodynamics due to measurement noise. Notice that the values of the training error is consistent with Eq. (13), the expression given in Section 3.4 and the magnitude of the noise. The implementation of the laws of thermodynamics in the network's architecture shields the learning process and prohibits learning of inconsistent data.

For instance, with reference to Table 3, we can see that for case (2), ns_σ, the MAEs in the predictions of the inelastic strains, energy and dissipation rate approximately coincide with those obtained with the un-noised data.

The aforementioned behavior is not observed in standard ANNs. As an example, we show in Table 4 the MAEs of the predictions of standard ANNs with respect to the validation data-sets, for noised stresses. In this case, we can see that the network, unaware of the requirements of the thermodynamics, learns successfully the noised outputs. This means that, once standard ANNs are asked to make predictions, in recall mode, the outputs will be affected by the noisy training in an unpredicted way. For the levels of noise cases (1), (3), and (4), similar results are obtained.

Table 4: MAEs of the predictions of standard ANNs with respect to the validation data-sets, for the original, un-noised data and noise on stresses, at the end of the training (approximately 1000 epochs).

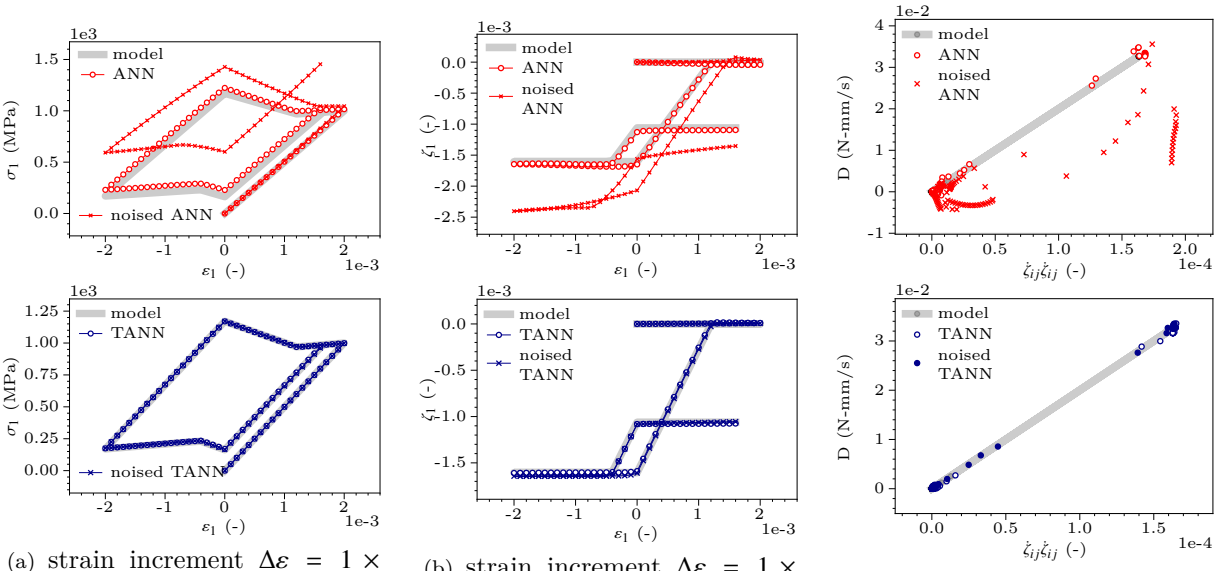
	Mean Absolute Error (1e-4)	
	$\Delta\zeta_i$	$\Delta\sigma_i$
un-noised	5.8	4.2
ns_σ	5.9	4.2

In Figure 19 compared the predictions in recall mode of both networks based on noise data. The predictions of the training with clean (un-noised) data is also presented for helping the comparison. We notice that TANNs, whilst trained on data with relatively large levels of noise, successfully predict the material response and perform more or less as when trained on data free of noise. On the contrary, standard ANNs are strongly affected by the large levels of noise of the data used to train the network. Similar results are found in presence of noise in the training and validation data of the internal variable, ζ_i, see Figure S11, in the Supplementary Material. It should be noticed that, in this case, ANNs do not manage to successfully minimize the loss function of $\Delta\zeta_i$, with the selected number of hyper-parameters. This is the consequence of the ANN architecture which have been chosen to achieve the best performance with thermodynamic consistent (clean of noise) data. However, we emphasize that, if the number of hyper-parameters of the ANN model were increased to achieve convergence with respect to the noised data, then ANNs would learn the noised material response, resulting to be highly affected by noise measurements.
Consequently, we can state that TANNs show high degree of robustness to noise, when compared to ANNs.

7. Concluding remarks

A new class of artificial neural networks models to replace constitutive laws and predict the material response at the material point level was proposed. The two basic laws of thermodynamics were directly encoded in the architecture of the model, which we refer to as Thermodynamics-based Neural Network (TANN). Our approach was inspired by the so-called Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019), where the automatic differentiation was used to perform the numerical calculation of the derivative of a neural network with respect to its inputs. Feed-Forward Neural Networks were used herein, but the approach is general and can be applied to Recurrent Neural Networks (RNNs) or other types of ANNs as well.

The numerical requirements regarding the mathematical class of appropriate activation functions to be used together with automatic differentiation were investigated. More specifically, the internal restrictions, derived from the first law of thermodynamics, require activation functions whose second gradient does not vanish. This new problem and its remedy was extensively explored and discussed in the manuscript.



(a) strain increment $\Delta\varepsilon = 1 \times 10^{-4}$.

(b) strain increment $\Delta\varepsilon = 1 \times 10^{-3}$.

(c) strain increment $\Delta\varepsilon = 1 \times 10^{-2}$.

Figure 19: Influence of noise in the stress, σ_i, for the predictions of the stress, internal variable, and dissipation rate of TANNs and of standard ANNs with respect to the target values, for the tri-axial cyclic loading path for material case H-1 (perfect plasticity). Noise strongly affect the predictions of standard ANNs, see Fig.s 13-15.

TANN, relying on an incremental formulation and on the theoretical developments in Houlsby and Puzrin (2007), posses the special feature that the entire constitutive response of a material can be derived from definition of only two scalar functions: the free-energy and the dissipation rate. This assures thermodynamically consistent predictions both for seen and unseen data. Differently from the standard ANN approaches, TANN does not have to identify, through learning, the underlying thermodynamic laws. Indeed, predictions of standard ANNs may be thermodynamically inconsistent, even though the training of the network has been performed on consistent material data. Being aware of physics, TANNs are found to be a robust approach with the presence of noise measurements in the training data, contrary to the standard ANN approach.

For the cases here investigated, we showed that TANNs are characterized by high accuracy of the predictions, higher than those of standard approaches. The integration of thermodynamic principles inside the network renders TANN's ability of generalization (i.e., make predictions for loading paths different from those used in the training operation) remarkably good. Consequently, TANN is an excellent candidate for replacing constitutive calculations at Finite Element incremental formulations. Moreover, thanks to the implementation of the free-energy in the network predictions and its thermodynamical relation with the stresses, the Jacobian $\frac{\partial\sigma}{\Delta\varepsilon}$ at the material point level is better predicted even for increments far beyond the training data-set range. As a result quadratic convergence in implicit formulations can be preserved, reducing the calculation cost.

Finally, we investigated the presence of noise in data and the effect on the training process

and predictions in recall mode. The thermodynamic framework of TANNs shields the training operation and prohibits learning of inconsistent data. As a result, TANNs posses high degrees of robustness to noise, compared to standard ANNs.

Further extensions of TANN in a wide range of applications, for complex materials, are straightforwards, as the thermodynamics principles hold true for any known class of material, at any length (micro- and macro-scale).

Acknowledgments

The authors would like to acknowledge the anonymous reviewers whose feed-backs helped improve this work.
The author I.S. would like to acknowledge the support of the European Research Council (ERC) under the European Union Horizon 2020 research and innovation program (Grant agreement ID 757848 CoQuake).

7.1. Appendix A. Understanding second-order vanishing gradient

In the following, we investigate the performance and influence of different activation functions on the computational time to train an ANN with input $\mathcal{I}$, primary output $\mathcal{O}_1$, and secondary output $\mathcal{O}_2 = \nabla_{\mathcal{I}}\mathcal{O}_1$. Consider the above discussed example with $\mathcal{I} = x$, $\mathcal{O}_1 = x^2$, and $\mathcal{O}_2 = 2x$. The ANN has one hidden layer, with $N_n = 6$ nodes, and activation functions as reported in Table 5. The output layer has linear activation and null bias. The absolute error is selected as loss function for both $\mathcal{O}_1$ and $\mathcal{O}_2$. Training is performed on 1000 samples, normalized between -1 and 1. A very small value for the learning rate is selected, i.e., $\epsilon = 10^{-5}$ in order to facilitate the gradient descent algorithm in reaching small values of the loss function. We use early-stopping. In other words, training is stopped as the error of a validation set (500 samples) starts to increase while the learning error still decreases (Géron, 2019). The validation set is used to avoid over-fitting of the training data.

Table 5: Set of activation functions considered to investigate the performance of the network with outputs $\mathcal{O} = x^2$ and $\nabla_{\mathcal{I}}\mathcal{O} = 2x$, with $\mathcal{I} = x$, in the framework of first- and second-order vanishing gradients.

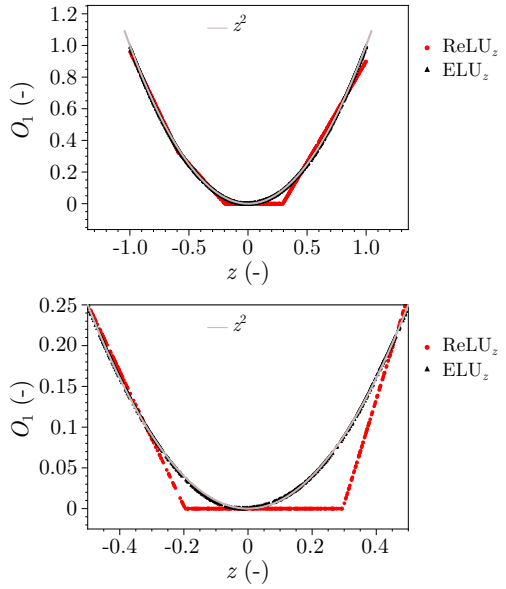
Function	z range	$\mathcal{A}(z)$	$\mathcal{A}'(z)$	$\mathcal{A}''(z)$
ReLU_z	$z < 0$	0	0	0
	$z \geq 0$	z	1	0
$\mathrm{ReLU}_{0.5z^2+z}$	$z < 0$	0	0	0
	$z \geq 0$	$0.5z^2 + z$	$z + 1$	1
ReLU_{z^2}	$z < 0$	0	0	0
	$z \geq 0$	z^2	$2z$	2
$\mathrm{ELU}_{\mathrm{e}}$	$\forall z$	$\mathrm{e}^z - 1$	e^z	e^z
ELU_z	$z < 0$	$\mathrm{e}^z - 1$	e^z	e^z
	$z \geq 0$	z	1	0
$\mathrm{ELU}_{0.5z^2+z}$	$z < 0$	$\mathrm{e}^z - 1$	e^z	e^z
	$z \geq 0$	$0.5z^2 + z$	$z + 1$	1
ELU_{z^2}	$z < 0$	$\mathrm{e}^z - 1$	e^z	e^z
	$z \geq 0$	z^2	$2z$	2
ELU_{z^4}	$z < 0$	$\mathrm{e}^z - 1$	e^z	e^z
	$z \geq 0$	z^4	$4z^3$	$12z^2$
$\mathrm{ELU}_{z^4+0.5z^2+z}$	$z < 0$	$\mathrm{e}^z - 1$	e^z	e^z
	$z \geq 0$	$z^4 + 0.5z^2 + z$	$4z^3 + z + 1$	$12z^2 + 1$

For each tested activation function, Table 6 shows the adimensional Mean Absolute Error (MAE) calculated using a set of new, unseen data (500 samples) of input-output predictions for x^2 and $2x$. The advancement of training is quantified herein as the number of epochs, i.e., the number with which the training algorithm works with the training data-set Géron (2019). Activation functions with quadratic terms, or of higher degree, perform very well, compared to their linear equivalents. RELU_{z^2}, ELU_{z^2} outperform as their shape is very similar to the input-output regression they are trained to learn. Nevertheless, it is worth

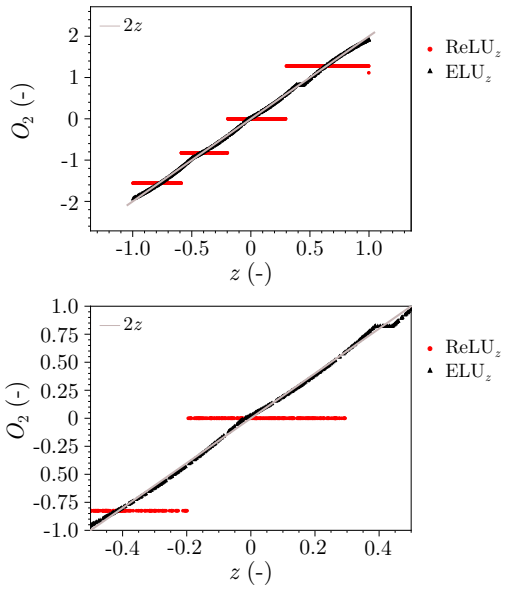
noticing that training fails when activation functions with vanishing second gradient are used (e.g. RELU_z and ELU_z). Figure 20 compares the ANN predictions for a selection of activation functions with the analytical (exact) results. Whilst RELU_z is clearly inadequate, ELU_z predictions overall agree with the analytical values. This is due to the fact that the ANN takes advantage of the exponential term, for negative z and thus successfully manage to satisfy both $\boldsymbol{O}$ and $\nabla_{\boldsymbol{I}}\boldsymbol{O}$. Additional hidden layers may improve the performance of the network. It can be further noticed that activation function of high degree, e.g. ELU_e, ELU_{z^4}, and $\text{ELU}_{z^4+0.5z^2+z}$, even if successful, require a large number of epochs.

Table 6: Activation functions and performance with unseen data.

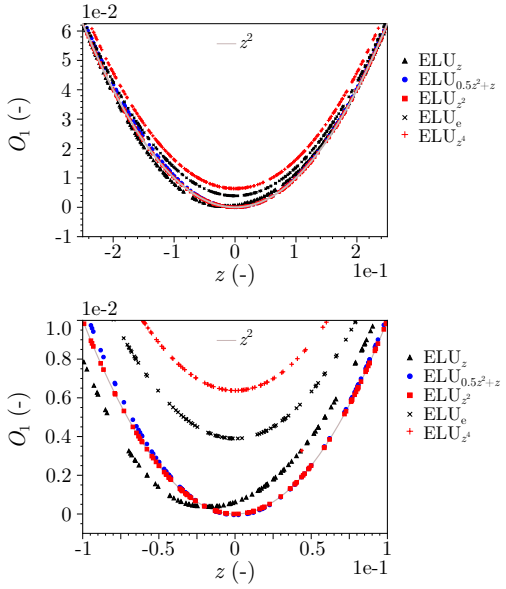
Activation function $\mathcal{A}$	$\mathcal{L}$ (10^{-4})	$\mathcal{L}_{\boldsymbol{O}}$ (10^{-4})	$\mathcal{L}_{\nabla_{\boldsymbol{I}}\boldsymbol{O}}$ (10^{-4})	no. epochs (-)
ReLU_z	1521.2	205.98	1315.18	920
$\text{ReLU}_{0.5z^2+z}$	762.4	93.58	668.85	8054
ReLU_{z^2}	0.061	0.0241	0.0371	148
ELU_e	127.2	26.83	100.38	19477
ELU_z	108.56	12.12	96.44	17280
$\text{ELU}_{0.5z^2+z}$	65.5	10.91	54.63	12178
ELU_{z^2}	0.13	0.067	0.067	88
ELU_{z^4}	65.36	33.75	31.61	20051
$\text{ELU}_{z^4+0.5z^2+z}$	12.94	1.81	11.13	9683



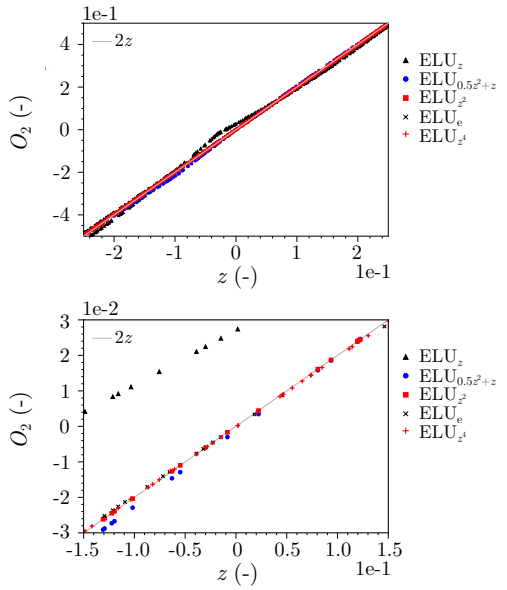
(a) x^2 predictions, O_1, using ReLU and ELU$_z$.



(b) $2x$ predictions, O_2, using ReLU and ELU$_z$.



(c) x^2 predictions, O_1, using ELU$_z$, ELU$_{0.5z^2+z}$, ELU$_{z^2}$, ELU$_e$, and ELU$_{z^4}$.



(d) $2x$ predictions, O_2, using ELU$_z$, ELU$_{0.5z^2+z}$, ELU$_{z^2}$, ELU$_e$, and ELU$_{z^4}$.

Figure 20: Comparison of different activation functions for the prediction of the primary output, x^2 (a), and secondary output, $2x$ (b). From top to bottom the range of z decreases from larger to smaller values, to observe the behavior at $z \approx 0$.

Appendix B. Derivation of the incremental material formulation

By differentiating the energy expressions (13) and rearranging the terms, we obtain the following non-linear incremental relations

$$\dot{\sigma} = \partial_{\varepsilon\varepsilon}\mathrm{F} \cdot \varepsilon + \sum_k \partial_{\varepsilon\zeta_k}\mathrm{F} \cdot \dot{\zeta}_k + \partial_{\varepsilon\theta}\mathrm{F}\,\dot{\theta} \tag{28a}$$

$$-\dot{\chi}_i = \partial_{\zeta_i\varepsilon}\mathrm{F} \cdot \varepsilon + \sum_k \partial_{\zeta_i\zeta_k}\mathrm{F} \cdot \dot{\zeta}_k + \partial_{\zeta_i\theta}\mathrm{F}\,\dot{\theta} \tag{28b}$$

$$-\dot{\mathrm{S}} = \partial_{\theta\varepsilon}\mathrm{F} \cdot \varepsilon + \sum_k \partial_{\theta\zeta_k}\mathrm{F} \cdot \dot{\zeta}_k + \partial_{\theta\theta}\mathrm{F}\,\dot{\theta}, \tag{28c}$$

where the following notation is adopted

$$\partial_{\varepsilon\varepsilon}\mathrm{F} = \frac{\partial^2\mathrm{F}}{\partial\varepsilon_{ij}\partial\varepsilon_{kl}}, \quad \partial_{\varepsilon\zeta_k}\mathrm{F} = \frac{\partial^2\mathrm{F}}{\partial\varepsilon_{ij}\partial\zeta_k},$$
$$\partial_{\varepsilon\theta}\mathrm{F} = \frac{\partial^2\mathrm{F}}{\partial\varepsilon_{ij}\partial\theta}, \quad \partial_{\theta\theta}\mathrm{F} = \frac{\partial^2\mathrm{F}}{\partial\theta^2}.$$

We introduce the thermodynamic dissipative stresses $\mathcal{X}^\dagger = (X_1, \ldots, X_N)$ with

$$X_i := \frac{\partial \mathrm{D}}{\partial \dot{\zeta}_i} \qquad \forall\, i \in [1, N]. \tag{29}$$

For a rate-independent material, the dissipation is a homogeneous first-order function in the internal variable rates $\dot{\zeta}_i$ (Houlsby and Puzrin, 2007). This homogeneity can be expressed by the Euler's relation

$$\mathrm{D} = \sum_{i=1}^{N} \frac{\partial \mathrm{D}}{\partial \dot{\zeta}_i} \cdot \dot{\zeta}_i = \sum_i X_i \cdot \dot{\zeta}_i, \tag{30}$$

which, together with (11), implies

$$\sum_{i=1}^{N} (X_i - \chi_i) \cdot \dot{\zeta}_i = 0 \tag{31}$$

Ziegler's orthogonality condition (Ziegler, 2012) is further assumed, i.e., $X_i = \chi_i\ \forall\, i \in [1, N]$. Being D homogeneous first-order function in $\dot{\zeta}_i$, the Legendre transform, conjugate to X_i, is degenerate, that is equal to zero, and represents the yield function $y = \tilde{y}(\theta, \varepsilon, \mathcal{Z}, \mathcal{X}^\dagger)$, i.e.

$$\lambda y = \sum_i X_i \cdot \dot{\zeta}_i - \mathrm{D} = 0, \tag{32}$$

where λ is a non-negative multiplier. From the properties of Legendre transform, the following flow rules must hold

$$\dot{\zeta}_i = \lambda \frac{\partial y}{\partial X_i} \qquad \forall\, i \in [1, N]. \tag{33}$$

Since $\lambda \geq 0$ and $\lambda y = 0$, $y \leq 0$. If $y = 0$, the following consistency equation is met

$$\dot{y} = \frac{\partial y}{\partial \varepsilon} \cdot \dot{\varepsilon} + \sum_{i=1}^{N} \frac{\partial y}{\partial \zeta_i} \cdot \dot{\zeta}_i + \sum_{i=1}^{N} \frac{\partial y}{\partial X_i} \cdot \dot{X}_i + \frac{\partial y}{\partial \theta}\, \dot{\theta} = 0. \tag{34}$$

By further using the flow rules (33) and Ziegler's normality condition, we obtain

$$\lambda = -\frac{C_\varepsilon}{B} \cdot \dot{\varepsilon} - \frac{C_\theta}{B} \cdot \dot{\theta}, \tag{35}$$

with

$$C_\varepsilon = \frac{\partial y}{\partial \varepsilon} - \sum_{i=1}^{N} \frac{\partial y}{\partial X_i} \cdot \partial_{\zeta_i \varepsilon} \mathrm{F},$$

$$C_\theta = \frac{\partial y}{\partial \theta} - \sum_{i=1}^{N} \frac{\partial y}{\partial X_i} \cdot \partial_{\zeta_i \theta} \mathrm{F},$$

and

$$B = \sum_{i=1}^{N} \frac{\partial y}{\partial \zeta_i} \cdot \frac{\partial y}{\partial X_i} - \sum_{i=1}^{N} \frac{\partial y}{\partial X_i} \left(\sum_{k=1}^{N} \partial_{\zeta_k \varepsilon} \mathrm{F} \cdot \frac{\partial y}{\partial X_k} \right).$$

Finally, we arrive to the following, incremental non-linear formulation, for $y = 0$,

$$\dot{\Xi} = \mathcal{M}|_{y=0}\, \dot{\xi}, \qquad \text{with} \qquad \dot{\Xi} = \begin{bmatrix} \dot{\sigma} \\ -\dot{X}_i \\ -\dot{\mathrm{S}} \\ \dot{\zeta}_i \\ \lambda \end{bmatrix}, \quad \dot{\xi} = \begin{bmatrix} \dot{\varepsilon} \\ \dot{\theta} \end{bmatrix}, \quad \mathcal{M}|_{y=0} = \begin{bmatrix} \mathrm{M}_{\varepsilon\varepsilon} & \mathrm{M}_{\varepsilon\theta} \\ \mathrm{M}_{\zeta\varepsilon} & \mathrm{M}_{\zeta\theta} \\ \mathrm{M}_{\theta\varepsilon} & \mathrm{M}_{\theta\theta} \\ -\frac{C_\varepsilon}{B} \cdot \frac{\partial y}{\partial X_i} & -\frac{C_\theta}{B} \cdot \frac{\partial y}{\partial X_i} \\ -\frac{C_\varepsilon}{B} \cdot & -\frac{C_\theta}{B} \end{bmatrix}, \tag{36}$$

and

$$\begin{aligned}
\mathrm{M}_{\varepsilon\varepsilon} &= \partial_{\varepsilon\varepsilon}\mathrm{F} - \textstyle\sum_k \partial_{\varepsilon\zeta_k}\mathrm{F} \cdot \left(\frac{C_\varepsilon}{B} \cdot \frac{\partial y}{\partial X_k}\right), \\
\mathrm{M}_{\varepsilon\theta} &= \partial_{\varepsilon\theta}\mathrm{F} - \textstyle\sum_k \partial_{\varepsilon\zeta_k}\mathrm{F} \cdot \left(\frac{C_\theta}{B} \cdot \frac{\partial y}{\partial X_k}\right), \\
\mathrm{M}_{\zeta\varepsilon} &= \partial_{\zeta_i\varepsilon}\mathrm{F} - \textstyle\sum_k \partial_{\zeta_i\zeta_k}\mathrm{F} \cdot \left(\frac{C_\varepsilon}{B} \cdot \frac{\partial y}{\partial X_k}\right), \\
\mathrm{M}_{\zeta\theta} &= \partial_{\zeta_i\theta}\mathrm{F} - \textstyle\sum_k \partial_{\zeta_i\zeta_k}\mathrm{F} \cdot \left(\frac{C_\theta}{B} \cdot \frac{\partial y}{\partial X_k}\right), \\
\mathrm{M}_{\theta\varepsilon} &= \partial_{\theta\varepsilon}\mathrm{F} - \textstyle\sum_k \partial_{\theta\zeta_k}\mathrm{F} \cdot \left(\frac{C_\varepsilon}{B} \cdot \frac{\partial y}{\partial X_k}\right), \\
\mathrm{M}_{\theta\theta} &= \partial_{\theta\theta}\mathrm{F} - \textstyle\sum_k \partial_{\theta\zeta_k}\mathrm{F} \cdot \left(\frac{C_\theta}{B} \cdot \frac{\partial y}{\partial X_k}\right).
\end{aligned}$$

In case of $y < 0$, relation (36) becomes

$$\dot{\Xi} = \mathcal{M}|_{y<0}\, \dot{\xi}, \qquad \text{with} \qquad \mathcal{M}|_{y<0} = \begin{bmatrix} \partial_{\varepsilon\varepsilon}\mathrm{F} & \partial_{\varepsilon\theta}\mathrm{F} \\ \partial_{\zeta_i\varepsilon}\mathrm{F} & \partial_{\zeta_i\theta}\mathrm{F} \\ \partial_{\theta\varepsilon}\mathrm{F} & \partial_{\theta\theta}\mathrm{F} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{37}$$

References

O. Lloberas Valls, M. Raschi Schaw, A. E. Huespe, X. Oliver Olivella, Reduced finite element square techniques (rfe2): towards industrial multiscale fe software, in: COMPLAS 2019: XV International Conference on Computational Plasticity: Fundamentals and Applications, International Centre for Numerical Methods in Engineering (CIMNE), 2019, pp. 157–169.

M. Nitka, G. Combe, C. Dascalu, J. Desrues, Two-scale modeling of granular materials: a DEM-FEM approach, Granular Matter 13 (2011) 277–281. doi:10.1007/s10035-011-0255-6.

F. Feyel, A multilevel finite element method (FE2) to describe the response of highly non-linear structures using generalized continua, Computer Methods in Applied Mechanics and Engineering 192 (2003) 3233–3244. doi:10.1016/S0045-7825(03)00348-7.

N. Bakhvalov, G. Panasenko, Homogenisation: Averaging Processes in Periodic Media: Mathematical Problems in the Mechanics of Composite Materials, 1989.

G. Houlsby, A. Puzrin, A thermomechanical framework for constitutive models for rate-independent dissipative materials, International journal of Plasticity 16 (2000) 1017–1047.

I. Einav, G. Houlsby, G. Nguyen, Coupled damage and plasticity models derived from energy and dissipation potentials, International Journal of Solids and Structures 44 (2007) 2487–2508.

G. T. Houlsby, A. M. Puzrin, Principles of hyperplasticity: an approach to plasticity theory based on thermodynamic principles, Springer Science & Business Media, 2007.

I. Einav, The unification of hypo-plastic and elasto-plastic theories, International Journal of Solids and Structures 49 (2012) 1305–1315.

F. Masi, I. Stefanou, V. Maffi-Berthier, P. Vannucci, A discrete element method based-approach for arched masonry structures under blast loads, Engineering Structures 216 (2020) 110721. doi:https://doi.org/10.1016/j.engstruct.2020.110721.

F. Masi, I. Stefanou, P. Vannucci, A study on the effects of an explosion in the Pantheon of Rome, Engineering Structures 164 (2018) 259–273. doi:10.1016/j.engstruct.2018.02.082.

H. Rattez, I. Stefanou, J. Sulem, The importance of thermo-hydro-mechanical couplings and microstructure to strain localization in 3d continua with application to seismic faults. part i: Theory and linear stability analysis, Journal of the Mechanics and Physics of Solids 115 (2018a) 54 – 76. doi:https://doi.org/10.1016/j.jmps.2018.03.004.

H. Rattez, I. Stefanou, J. Sulem, M. Veveakis, T. Poulet, The importance of thermo-hydro-mechanical couplings and microstructure to strain localization in 3d continua with application to seismic faults. part ii: Numerical implementation and post-bifurcation analysis, Journal of the Mechanics and Physics of Solids 115 (2018b) 1 – 29. doi:https://doi.org/10.1016/j.jmps.2018.03.003.

N. A. Collins-Craft, I. Stefanou, J. Sulem, I. Einav, A cosserat breakage mechanics model for brittle granular media, Journal of the Mechanics and Physics of Solids (2020) 103975. doi:https://doi.org/10.1016/j.jmps.2020.103975.

A. P. V. D. Eijnden, P. Bésuelle, F. Collin, R. Chambon, J. Desrues, Modeling the strain localization around an underground gallery with a hydro-mechanical double scale model ; effect of anisotropy, Computers and Geotechnics (2016). doi:10.1016/j.compgeo.2016.08.006.

A. Geron, Hands-on MachineLearning with Scikit-Learn & Tensorflow, volume 1, O'Reilly Media, 2015. doi:10.1017/CBO9781107415324.004. arXiv:arXiv:1011.1669v3.

T. M. Mitchell, et al., Machine learning. 1997, Burr Ridge, IL: McGraw Hill 45 (1997) 870–877.

G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of control, signals and systems 2 (1989) 303–314.

T. Chen, H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, IEEE Transactions on Neural Networks 6 (1995) 911–917.

J. Ghaboussi, J. H. Garrett, X. Wu, Knowledge‐based modeling of material behavior with neural networks, Journal of Engineering Mechanics 117 (1991) 132–153. doi:10.1061/(ASCE)0733-9399(1991)117:1(132).

J. Ghaboussi, D. Sidarta, New nested adaptive neural networks (nann) for constitutive modeling, Computers and Geotechnics 22 (1998) 29–52.
M. Lefik, B. A. Schrefler, Artificial neural network as an incremental non-linear constitutive model for a finite element code, Computer methods in applied mechanics and engineering 192 (2003) 3265–3283.
S. Jung, J. Ghaboussi, Neural network constitutive model for rate-dependent materials, Computers & Structures 84 (2006) 955–963.
C. Settgast, M. Abendroth, M. Kuna, Constitutive modeling of plastic deformation behavior of open-cell foam structures using neural networks, Mechanics of Materials 131 (2019) 1–10.
Z. Liu, C. Wu, Exploring the 3d architectures of deep material network in data-driven multiscale mechanics, Journal of the Mechanics and Physics of Solids 127 (2019) 20–46.
X. Lu, D. G. Giovanis, J. Yvonnet, V. Papadopoulos, F. Detrez, J. Bai, A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites, Computational Mechanics 64 (2019) 307–321.
K. Xu, D. Z. Huang, E. Darve, Learning constitutive relations using symmetric positive definite neural networks, arXiv preprint arXiv:2004.00265 (2020).
D. Z. Huang, K. Xu, C. Farhat, E. Darve, Learning constitutive relations from indirect observations using deep neural networks, Journal of Computational Physics (2020) 109491.
S. Gajek, M. Schneider, T. Böhlke, On the micromechanics of deep material networks, Journal of the Mechanics and Physics of Solids (2020) 103984.
M. B. Gorji, M. Mozaffar, J. N. Heidenreich, J. Cao, D. Mohr, On the potential of recurrent neural networks for modeling path dependent plasticity, Journal of the Mechanics and Physics of Solids (2020) 103972.
Y. Heider, K. Wang, W. Sun, SO(3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials, Computer Methods in Applied Mechanics and Engineering 363 (2020) 112875. doi:https://doi.org/10.1016/j.cma.2020.112875.
F. Ghavamian, A. Simone, Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network, Computer Methods in Applied Mechanics and Engineering 357 (2019) 112594.
M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M. Bessa, Deep learning predicts path-dependent plasticity, Proceedings of the National Academy of Sciences 116 (2019) 26414–26420.
A. L. Frankel, R. E. Jones, C. Alleman, J. A. Templeton, Predicting the mechanical response of oligocrystals with deep learning, Computational Materials Science 169 (2019) 109099.
D. González, F. Chinesta, E. Cueto, Learning corrections for hyperelastic models from data, Frontiers in Materials 6 (2019) 14.
M. Lefik, D. Boso, B. Schrefler, Artificial neural networks in numerical modelling of composites, Computer Methods in Applied Mechanics and Engineering 198 (2009) 1785–1804.
T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, Computer Methods in Applied Mechanics and Engineering 304 (2016) 81–101.
R. Ibañez, D. Borzacchiello, J. V. Aguado, E. Abisset-Chavanne, E. Cueto, P. Ladevèze, F. Chinesta, Data-driven non-linear elasticity: constitutive manifold construction and problem discretization, Computational Mechanics 60 (2017) 813–826.
T. Kirchdoerfer, M. Ortiz, Data-driven computing in dynamics, International Journal for Numerical Methods in Engineering 113 (2018) 1697–1710.
R. Ibanez, E. Abisset-Chavanne, J. V. Aguado, D. Gonzalez, E. Cueto, F. Chinesta, A manifold learning approach to data-driven computational elasticity and inelasticity, Archives of Computational Methods in Engineering 25 (2018) 47–57.
R. Eggersmann, T. Kirchdoerfer, S. Reese, L. Stainier, M. Ortiz, Model-free data-driven inelasticity, Computer Methods in Applied Mechanics and Engineering 350 (2019) 81–99.
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, Journal of Computational Physics 378 (2019) 686–707.
A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning:

a survey, The Journal of Machine Learning Research 18 (2017) 5595–5637.
G. A. Maugin, W. Muschik, Thermodynamics with internal variables. Part I. General concepts, 1994.
P. M. Mariano, L. Galano, Fundamentals of the Mechanics of Solids, Springer, 2015.
L. Anand, O. Aslan, S. A. Chester, A large-deformation gradient theory for elastic–plastic materials: Strain softening and regularization of shear bands, International Journal of Plasticity 30-31 (2012) 116 – 143. doi:https://doi.org/10.1016/j.ijplas.2011.10.002.
Y. H. Hu, J.-N. Hwang, Handbook of neural network signal processing, 2002.
A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, 2019.
M. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, W. K. Liu, A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality, Computer Methods in Applied Mechanics and Engineering 320 (2017) 633–667.
A. Karpatne, W. Watkins, J. Read, V. Kumar, Physics-guided neural networks (pgnn): An application in lake temperature modeling, arXiv preprint arXiv:1710.11431 (2017).
H. Ziegler, An introduction to thermomechanics, Elsevier, 2012.
P. Bogacki, L. F. Shampine, A 3 (2) pair of Runge-Kutta formulas, Applied Mathematics Letters 2 (1989) 321–325.
T. Dozat, Incorporating Nesterov momentum into Adam (2016).