Mechanics Research Communications 125 (2022) 103993

Contents lists available at ScienceDirect

Mechanics Research Communications

journal homepage: www.elsevier.com/locate/mechrescom

Polyconvex neural networks for hyperelastic constitutive models: A rectification approach

Peiyi Chen [a], Johann Guilleminot [b,*]

[a] Department of Mechanical Engineering and Materials Science, Duke University, NC, 27710, USA
[b] Department of Civil and Environmental Engineering, Duke University, NC, 27710, USA

ARTICLE INFO

Keywords:
Constitutive modeling
Hyperelasticity
Neural networks
Polyconvexity
Rectification

ABSTRACT

A simple approach to rectify unconstrained neural networks for hyperelastic constitutive models is proposed with the aim of ensuring both mathematical well-posedness (in terms of existence theorems) and physical consistency. The surrogate involves neural networks that are made admissible by selecting a proper parameterization, following standard results in continuum mechanics, and by enforcing polyconvexity through integral representations. The relevance of the formulation is demonstrated by considering digitally synthesized and experimental datasets for isotropic and anisotropic materials, including the case of soft biological tissues.

1. Introduction

Constitutive models based on neural networks (NN) have received growing attention over the past few years, owing to their ability to represent nonlinear mappings in a high dimensional setting. There is a very substantial amount of papers published on this topic, for a wide variety of material behaviors; see, *e.g.*, [1–12] and the references therein, in a non-exhaustive manner.

Beyond classical data science aspects that pertain to architecture design, training and validation strategies, and the analysis of approximation capabilities, a central concern is to make such surrogates amenable to scientific simulations where such models are typically set to parameterize systems of partial differential equations. In this context, the surrogate must satisfy both physical assumptions and mathematical properties (*e.g.*, boundedness or a certain type of convexity) to ensure the existence (and potentially, the uniqueness) of solutions. In the case of nonlinear elasticity for instance, a strain energy density function is theoretically required to satisfy frame indifference, some asymptotic behavior, and specific convexity and growth conditions. There are various ways to enforce such properties and in particular, the convexity requirement. The simplest strategy consists in using some unconstrained neural network that, if properly calibrated on a rich enough dataset, may possess desired convexity. Another way to enforce convexity is to add a penalty term in the loss function during the training stage; see, *e.g.*, [13]. This latter strategy corresponds to a weak enforcement and hence does not prevent from checking the condition *a posteriori*.

In this work, we consider enforcing convexity in the strong sense, by defining classes of surrogate models that satisfy the condition *a priori*. The issue of ensuring the convexity of a neural network is not new and was mostly tackled by constraining the network through the use of non-negative weights and convex activation functions [14]. Restriction on weights can be imposed by constraining the weights during training, or by using a mapping from $\mathbb{R}$ into $\mathbb{R}_{\geq 0}$ that acts on unconstrained weights [11,15]. Applications in computational mechanics are presented in [10,11,16] for example. A detailed analysis about the use of constrained neural networks for polyconvex anisotropic hyperelastic models can be found in [12], in particular.

Here, we aim to construct a convex neural network model without affecting expressiveness (that is, without constraining weights *a priori*) and training cost (which can be affected by transformations performed on weights at the training stage). Building upon recent works on monotonic neural networks [17] and monotone transport maps for density estimation [18], our approach relies on simple integral representations to define an operator that transforms any arbitrary function (and in particular, a *free* neural network) into a convex function. This strategy thus entails the rectification of the whole neural network model. We show, through various numerical experiments on both digitally synthesized and experimental datasets, that the proposed rectified models enable proper fitting. They are also seen to converge much faster than constrained models (in terms of number of iterations)—at the expense of an increased computational cost per iteration.

The rest of this paper is organized as follows. The mechanistic parameterization and rectification strategy are first presented in Section 2 (together with a toy example). Applications to standard hyperelastic models relevant to both isotropic and anisotropic materials are then

* Corresponding author.
E-mail address: johann.guilleminot@duke.edu (J. Guilleminot).

https://doi.org/10.1016/j.mechrescom.2022.103993
Received 1 July 2022; Received in revised form 9 September 2022; Accepted 14 September 2022
Available online 19 September 2022

discussed in Section 3. Conclusions and avenues for future research are finally given in Section 4.

2. Rectified neural network representations

2.1. Background in elasticity

Let Ω be a collection of material points identified with their vector of coordinates $\boldsymbol{X}$ in $\mathbb{R}^3$, and denote by $\partial\Omega$ the boundary of Ω. For any material point $\boldsymbol{X} \in \Omega$, the spatial point $\boldsymbol{x}$ in the deformed configuration Ω^φ is given by $\boldsymbol{x} = \varphi(\boldsymbol{X})$, where φ is the deformation map. For any $\boldsymbol{X} \in \Omega$, the deformation gradient $\boldsymbol{F}$ is a second-order tensor defined as $\boldsymbol{F} = \boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{x}$. The left and right Cauchy–Green deformation tensors are given by $\boldsymbol{B} = \boldsymbol{F}\boldsymbol{F}^T$ and $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$, respectively.

We seek to construct a neural network surrogate that satisfies physical axioms and mathematical requirements arising in existence theorem in finite elasticity. From a theoretical standpoint, strain energy density functions are required to satisfy [19,20]:

1. Principle of material frame indifference (objectivity), stated as: $\forall \boldsymbol{Q} \in \mathrm{SO}(3)$,
$$w(\boldsymbol{Q}\boldsymbol{F}) = w(\boldsymbol{F}), \quad \boldsymbol{P}(\boldsymbol{Q}\boldsymbol{F}) = \boldsymbol{Q}\boldsymbol{P}(\boldsymbol{F}), \tag{1}$$
where $\boldsymbol{P}$ is the first Piola–Kirchhoff stress tensor;
2. Proper convexity conditions;
3. Some asymptotic behavior as $\det(\boldsymbol{F}) \to 0^+$; and
4. A coerciveness inequality (growth conditions).

In particular, the requirements (2–4) are fundamental to ensure the existence of (at least) one minimizer for the energy functional, see Chapter 7 in [19] (see also [21,22]).

Material frame-indifference is, in general, achieved by defining w in terms of the right Cauchy–Green deformation tensor $\boldsymbol{C}$, which is an *a priori* objective kinematic variable [20]. Following the work by Ball [23], polyconvexity is often imposed in lieu of convexity, as (i) it does not conflict with any physical constraints; (ii) it is generally satisfied by commonly employed models; and (iii) it enables the derivation of powerful existence results [19]. To proceed with the construction of the model, it is instructive at this point to recall the definition of polyconvexity. A strain energy density function $w : \mathbb{M}^3_+ \to \mathbb{R}$ is polyconvex if there exists a convex function $w^* : \mathbb{M}^3 \times \mathbb{M}^3 \times \mathbb{R}$ such that

$$w(\boldsymbol{F}) = w^*(\boldsymbol{F}, \mathrm{Cof}(\boldsymbol{F}), \det(\boldsymbol{F})), \tag{2}$$

for all $\boldsymbol{F} \in \mathbb{M}^3_+$, where (i) $\mathrm{Cof}(\boldsymbol{F})$ and $\det(\boldsymbol{F})$ are the cofactor matrix and determinant of $\boldsymbol{F}$; (ii) $\mathbb{M}^3$ and $\mathbb{M}^3_+$ denote the sets of real square matrices of order 3 with arbitrary and strictly positive determinants, respectively. The requirements (3) and (4) above, related to volume annihilation and coercivity, are crucial in the analytical derivation of functional forms for w. They are, however, less relevant to surrogate modeling which only involves bounded intervals, by construction. In this context, the choice of a proper parameterization (in terms of $\boldsymbol{C}$) and the satisfaction of the polyconvexity requirement are sufficient to ensure well-posedness and physical consistency.

Following the previous discussion, we then consider the construction of a polyconvex surrogate, that is $w(\boldsymbol{C}) = w^*(I_1, I_2, I_3)$ owing to a slight abuse of notation, where $I_1 = \mathrm{tr}\,\boldsymbol{C}$, $I_2 = \mathrm{tr}[\mathrm{Cof}\,\boldsymbol{C}]$, and $I_3 = \det \boldsymbol{C}$ are the polyconvex invariants of the right Cauchy–Green tensor $\boldsymbol{C}$. We assume an additive decomposition and define w^* as

$$w^*(I_1, I_2, I_3) := \sum_{i=1}^{3} w_i^*(I_i), \tag{3}$$

where $\{w_i^*\}_{i=1}^3$ are convex in the associated variables, hence ensuring the polyconvexity of w^* [24,25]. Notice that the above formulation can readily be extended to model anisotropic behaviors, by including mixed invariants that involve structural tensors [26] (see Sections 3.3 and 3.4).

Our aim now is to construct the set of convex functions $\{w_i^*\}_{i=1}^3$ using *fully unconstrained* neural networks.

2.2. Rectification of unconstrained neural networks for constitutive modeling in finite elasticity

In order to define the functions $\{w_i^*\}_{i=1}^3$, we start by recalling the integral representation of convex functions, using generic notation.

Let $f : I \to \mathbb{R}$ be a convex function. Then f admits the representation

$$f(x) = f(a) + \int_a^x \phi(t)\,dt, \tag{4}$$

for $a < x$ in the interval I, where $\phi : I \to \mathbb{R}$ is a nondecreasing function. The constant $f(a)$ in the right-hand side of Eq. (4) can be derived by fixing the value of f at some point $x^\star \geq a$ in I, that is

$$f(a) = f(x^\star) - \int_a^{x^\star} \phi(t)\,dt. \tag{5}$$

In the case of a strain energy density function, the point $x^\star$ is associated with the normalization condition $w(\boldsymbol{I}) = 0$.

The central idea is to define ϕ in terms of an arbitrary function, soon to be taken as an unconstrained NN. In order to enforce the monotonicity of ϕ, we rely on the representation proposed in [18] to enforce monotonicity on transport maps. Specifically, we define ϕ as

$$\phi(t) := \Phi(0) + \int_0^t g\left(\frac{d\Phi(z)}{dz}\right) dz, \tag{6}$$

where $\Phi : \mathbb{R} \to \mathbb{R}$ is *any* smooth function and $g : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a positive function. The operator defined by Eq. (6) maps any function Φ into a nondecreasing function ϕ and was called, for this reason, a rectifier in [18]. We use this terminology below, and write

$$\phi = \mathcal{R}_{\mathrm{inc}}\{\Phi\}, \quad \phi(t) = \mathcal{R}_{\mathrm{inc}}\{\Phi\}(t) \quad \forall t \in \mathbb{R}. \tag{7}$$

Similarly, Eq. (4) can be written as

$$f = \mathcal{R}_{\mathrm{cvx}}\{\phi\}, \quad f(x) = \mathcal{R}_{\mathrm{cvx}}\{\phi\}(x) \quad \forall x \in I, \tag{8}$$

where $\mathcal{R}_{\mathrm{cvx}}$ is seen as a second rectifier. Consequently, the convex function f can be defined as

$$f = \mathcal{R}\{\Phi\}, \quad \mathcal{R} := \mathcal{R}_{\mathrm{cvx}} \circ \mathcal{R}_{\mathrm{inc}}, \tag{9}$$

where the composite rectifier $\mathcal{R}$ implicitly depends on the function g. Several choices were proposed and studied in the literature, including the exponential, modified soft-plus, or square functions [18]. It follows that each function w_i^*, $1 \leq i \leq 3$, can be defined as

$$w_i^*(I_i) := \mathcal{R}\{\psi_i(\{\boldsymbol{W}_j^{(i)}, \boldsymbol{b}_j^{(i)}\}_{j=1}^{n_i})\}(I_i), \tag{10}$$

where ψ_i is the *unconstrained* neural network associated with input variable I_i, with weights and biases gathered in $\{\boldsymbol{W}_j^{(i)}\}_{j=1}^{n_i}$ and $\{\boldsymbol{b}_j^{(i)}\}_{j=1}^{n_i}$, respectively, and n_i is the number of layers (including the hidden and output layers). The neural network ψ_i associated with I_i is written as

$$\psi_i(I_i) := \boldsymbol{W}_{n_i}^{(i)}(\ldots A_1^{(i)}(\boldsymbol{W}_1^{(i)} I_i + \boldsymbol{b}_1^{(i)})\ldots) + \boldsymbol{b}_{n_i}^{(i)}, \tag{11}$$

where $\{A_j^{(i)}\}_{j=1}^{n_i-1}$ are $(n_i - 1)$ vector-valued, component-wise acting activation functions (note that no activation function is used for the outer layer). The rectified neural network surrogate for the strain energy function is finally obtained as

$$w^*(I_1, I_2, I_3) = \sum_{i=1}^{3} \mathcal{R}\{\psi_i(\{\boldsymbol{W}_j^{(i)}, \boldsymbol{b}_j^{(i)}\}_{j=1}^{n_i})\}(I_i). \tag{12}$$

While the neural networks $\{\psi_i\}_{i=1}^3$ are left undefined at this stage, it should be noticed that the architectures must be such that the surrogate w^* is twice differentiable. The integral representation makes this requirement weaker, as the neural network only needs to be differentiable. Finally, w^* must satisfy the normalization condition

$$w^*(3, 3, 1) = 0, \tag{13}$$

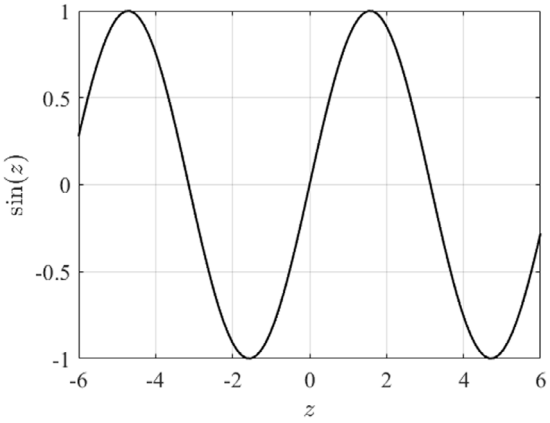

Fig. 1. Graph of Φ (unconstrained function).

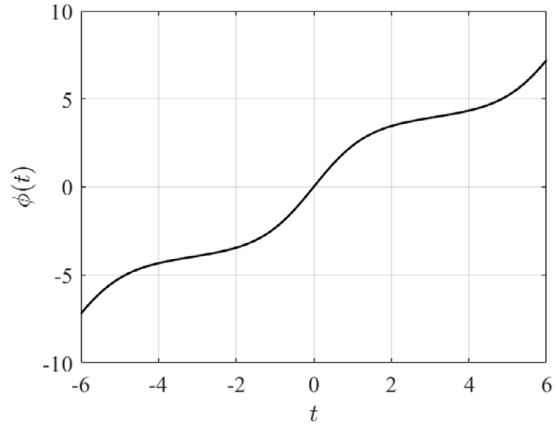

Fig. 2. Graph of $\phi = \mathcal{R}_{\text{inc}}\{\Phi\}$ (after first rectification).

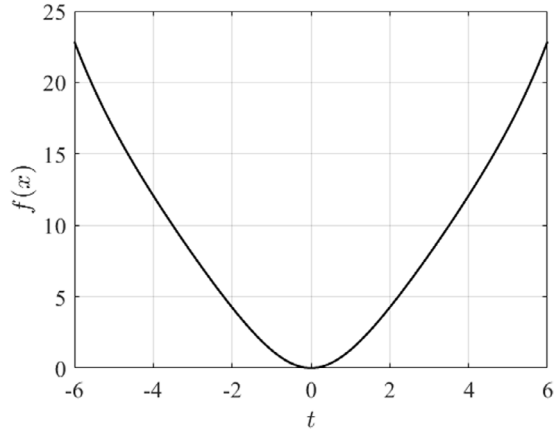

Fig. 3. Graph of $f = \mathcal{R}_{\text{cvx}}\{\phi\} = \mathcal{R}\{\Phi\}$ (fully rectified function).

as well as the constraint

$$\left.\frac{\partial w_1^*(I_1)}{\partial I_1}\right|_{I_1=3} + 2\left.\frac{\partial w_2^*(I_2)}{\partial I_2}\right|_{I_2=3} + \left.\frac{\partial w_3^*(I_3)}{\partial I_3}\right|_{I_3=1} = 0\,, \tag{14}$$

stemming from the stationarity of the (isotropic) strain energy density function at $\boldsymbol{C} = \boldsymbol{I}$.

Eq. (13) can be enforced by the shift defined by Eq. (5). The constraint given by Eq. (14) can be accounted for in two ways. Weak enforcement can be achieved by adding a penalty term in the loss function during training. This strategy may, however, lead to spurious behaviors that were reported in [11] for example. Alternatively, the stress-free constraint may be integrated in the strong sense either by enforcing an algebraic equation on hyperparameters, or by simply shifting the stress value at the origin [11]. The former approach leads to nonlinear constraints and was found to affect expressiveness in numerical experiments. In contrast, the latter strategy usually enables good accuracy and can easily be implemented. For these reasons, the stress shift strategy will be used in the examples discussed in Section 3. Note that this amounts to adding a term in the strain energy density function that does not affect its properties in terms of theoretical requirements (*e.g.*, convexity); see the discussion in [11].

Remark. To illustrate the approach, let us consider the rectification of the function $\Phi(z) = \sin(z)$ on $I = [-6, 6]$, and take $g(x) = \exp(x)$. We have

$$\phi(t) = \mathcal{R}_{\text{inc}}\{\Phi\}(t) = \sin(0) + \int_0^t \exp\left(\cos(z)\right)\, dz \tag{15}$$

and

$$f(x) = \mathcal{R}_{\text{cvx}}\{\phi\}(x) = f(-6) + \int_{-6}^{x} \phi(t)\, dt\,. \tag{16}$$

Here, we enforce the constraint $f(0) = 0$ (that is, $x^\star = 0$), so that

$$f(x) = -\int_{-6}^{0} \phi(t)\, dt + \int_{-6}^{x} \phi(t)\, dt\,, \quad \forall x \geq -6\,. \tag{17}$$

The rectified function $f = \mathcal{R}\{\Phi\}$, together with the latent functions, are shown in Figs. 1, 2 and 3. Notice that in this example, the derivative of the rectified function f cannot be required to vanish at an arbitrary point (as in Eq. (14) for example), since the primary function Φ is fixed (as opposed to the case where it can be trained).

3. Applications

We consider a standard setting where data are provided in the form stress–strain responses. For an incompressible material for instance, the Cauchy stress associated with the rectified NN is evaluated as

$$\boldsymbol{\Sigma}^*(\boldsymbol{F}) = 2\boldsymbol{F}\left(\sum_{i=1}^{3} \frac{\partial w_i^*(I_i)}{\partial I_i}\frac{\partial I_i}{\partial \boldsymbol{C}}\right)\boldsymbol{F}^T - p\boldsymbol{I}\,, \tag{18}$$

where the left-hand side depends on the parameters of the NN, w_i^* is defined by Eq. (10), p is a Lagrange multiplier arising from the incompressibility condition (in practice, p is evaluated by imposing a stress-free condition).

Training with respect to data can be achieved using the cost function

$$\mathcal{L}_d = \frac{\sum_{i=1}^{N}\left(\Sigma^*(\boldsymbol{F}(\lambda_i)) - \Sigma^{\text{data}}(\boldsymbol{F}(\lambda_i))\right)^2}{\sum_{i=1}^{N}\Sigma^{\text{data}}(\boldsymbol{F}(\lambda_i))^2}\,, \tag{19}$$

along a loading path $\lambda \mapsto \boldsymbol{F}(\lambda)$, discretized with N points. Here, Σ^* denotes the relevant stress component (*e.g.*, along testing direction), and $\{(\lambda_i, \Sigma^{\text{data}}(\boldsymbol{F}(\lambda_i)))\}_{i=1}^N$ constitutes the dataset. Note that the above loss function can be readily extended to cases where several loading conditions are considered (see Section 3.4).

In the applications presented below, no attempt was made to fully optimize network architectures and training strategies. The numbers of layers and neurons per layer were determined through a standard parametric analysis on the validation loss defined by Eq. (19). No activation function was used for the toy problem presented in Section 3.1, while the sigmoid activation function was selected for all hidden layers in all other examples (in Section 3.2, 3.3, and 3.4).

The Adaptive Moment Estimation (ADAM) algorithm was used for training, with an implementation in JAX [27].

Table 1
Results for the validation step: L^2-norm errors for the rectified neural network model (toy problem).

Function g	Error ϵ
Square	1.5850×10^{-7}
Exponential	2.0018×10^{-5}
Soft-plus	1.6069×10^{-7}

3.1. Toy problem

We first consider a toy example where the target convex function is given as

$$f^{\text{target}}(x) = k_1 \exp\left(\frac{(x-k_2)^2}{2k_3^2}\right), \quad \forall x \in \mathbb{R}, \tag{20}$$

where $k_1 = 10$, $k_2 = 5$, and $k_3 = 20$. We seek to construct an approximation over the interval $I = [0, 10]$, centered without loss of generality around the abscissa $x = k_2$ at which f^{target} reaches its minimum, with $f(k_2) = k_1$. A set of 100 equidistant datapoints is used for fitting, and 20% datapoints are used as the validation set. Adopting the generic notation ψ for the neural network, the rectified model reads as

$$f(x) = k_1 - \int_0^{k_2} \phi(t)\, dt + \int_0^x \phi(t)\, dt, \quad \forall x \in [0, 10], \tag{21}$$

where

$$\phi(t) = \psi(0) + \int_0^t g\left(\frac{d\psi(z)}{dz}\right) dz. \tag{22}$$

Three different choices for g were considered, namely the square function $g(x) = x^2$, the exponential function $g(x) = \exp(x)$, and the modified soft-plus function $g(x) = \log(2^x + 1)/\log(2)$.

In this example, a simple neural network architecture with one hidden layer and two neurons is used (without activation functions). The learning rate for the ADAM optimizer is set to 0.01. Mean square validation errors for the rectified functions are reported in Table 1 for the three choices of g. The square and modified soft-plus functions provide fairly similar validation errors, smaller than the one obtained with the exponential function. In addition, the model rectified with the square function converged in 3,000 epochs, while the rectified model with the modified soft-plus function converged in about 8,000 epochs. The model with the exponential function converged in more than 10,000 epochs, which is slower than with the other two positive functions. In order to qualitatively assess the accuracy, the predictions obtained with the rectified neural network for the validation dataset are shown in Fig. 4 (see Table 1 for validation metrics). Recall that no restrictions are imposed on weights and activation functions in the proposed formulation.

3.2. Mooney–Rivlin model

Here we address the case of a Mooney–Rivlin material, defined by the stored energy function

$$w^{\text{MR}}(I_1, I_2) = C_1(I_1 - 3) + C_2(I_2 - 3), \tag{23}$$

where C_1 and C_2 are strictly positive material parameters (see [19], p. 189). The rectified model is written as

$$w^*(I_1, I_2) = w_1^*(I_1) + w_2^*(I_2), \tag{24}$$

with

$$w_1^*(I_1) = \mathcal{R}\{\psi_1(\{\boldsymbol{W}_j^{(1)}, \boldsymbol{b}_j^{(1)}\}_{j=1}^{n_1})\}(I_1) \tag{25}$$

and

$$w_2^*(I_2) = \mathcal{R}\{\psi_2(\{\boldsymbol{W}_j^{(2)}, \boldsymbol{b}_j^{(2)}\}_{j=1}^{n_2})\}(I_2). \tag{26}$$

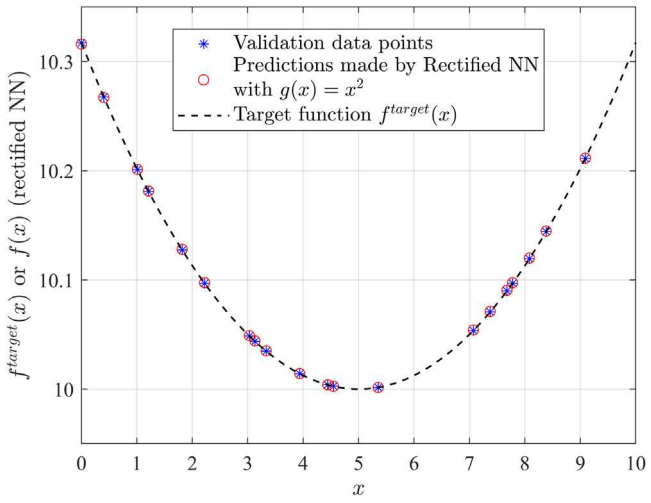

Fig. 4. Target function (black dashed line), reference values (blue star), and rectified neural network predictions (red circles) for the validation dataset (random selection).

The Cauchy stress is given by (see [28], p. 224)

$$\boldsymbol{\Sigma} = 2C_1\boldsymbol{B} - 2C_2\boldsymbol{B}^{-1} - p\boldsymbol{I}. \tag{27}$$

We consider uniaxial tension along the first direction for training purposes, in which case the uniaxial Cauchy stress writes

$$\Sigma(\lambda) = 2\left(C_1 + \frac{C_2}{\lambda}\right)\left(\lambda^2 - \frac{1}{\lambda}\right), \tag{28}$$

where λ is the driving principal stretch. The uniaxial Cauchy stress associated with the rectified model can be evaluated as

$$\Sigma^*(\lambda) = 2\left(\frac{\partial w_1^*(I_1)}{\partial I_1} + \frac{\partial w_2^*(I_2)}{\partial I_2}\frac{1}{\lambda}\right)\left(\lambda^2 - \frac{1}{\lambda}\right), \tag{29}$$

with

$$\frac{\partial w_i^*(I_i)}{\partial I_i} = \left(\psi_i(0) + \int_0^{I_i} g\left(\frac{d\psi_i}{dz}\right) dz\right). \tag{30}$$

Derivatives are computed using automatic differentiation. In this example, we consider an approximation for $\lambda \in [1.0, 1.4]$ ($I_1 \geq 3$ and $I_2 \geq 3$). The material parameters are arbitrarily chosen as $C_1 = 10$ and $C_2 = 5$. Normalization condition is imposed by taking $w_1^*(3) = w_2^*(3) = 0$ with the constant terms (see Eq. (4), with $a = 3$) set to 0. Note that the stress free condition is automatically satisfied owing to the definition of p.

Two hundreds datapoints are generated using Eq. (28), with 20% of samples allocated for validation. Parametric studies on validation error were conducted to identify the NN architecture, using a learning rate set to 0.01, 5,000 epochs, and the square function (in the first rectifier); see Fig. 5. Results predicted by the calibrated rectified neural network model (with 5 layers and 40 neurons per layer) on the validation dataset is shown in Fig. 6.

Remark. As indicated in Section 1, Input-Convex Neural Networks (ICNNs) can be obtained by forcing all weights, except those connected to the input, to be non-negative, and by using only convex non-decreasing activation functions [10–12,14]. In order to evaluate the performance of such constrained neural networks in terms of training results, we consider a model of the form

$$\psi_{\text{ICNN}}(I_1, I_2) = \psi_{\text{ICNN}}^{(1)}(I_1) + \psi_{\text{ICNN}}^{(2)}(I_2), \tag{31}$$

where each ICNN involves unconstrained weights that are mapped to positive weights in the loss function, using the modified soft-plus activation function proposed in [10,11]. The validation loss history for each approach is shown in Fig. 7 for the Mooney–Rivlin dataset. Reported results were obtained with 2 hidden layers and 20 neurons

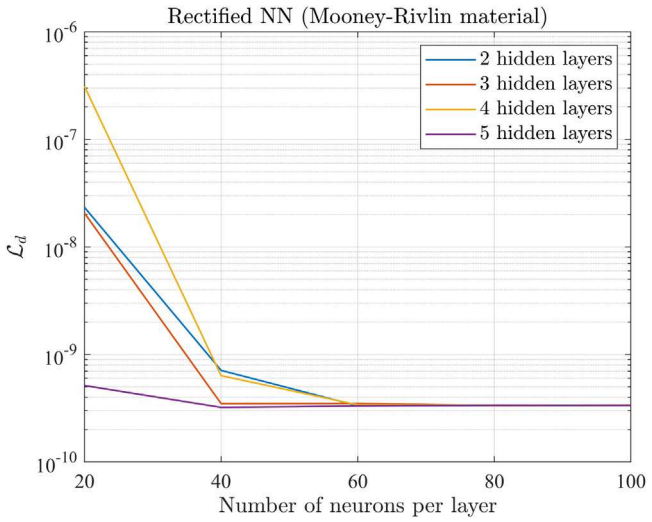

Fig. 5. Loss function $\mathcal{L}_d$ for different neural network architectures on the validation dataset (Mooney–Rivlin material).

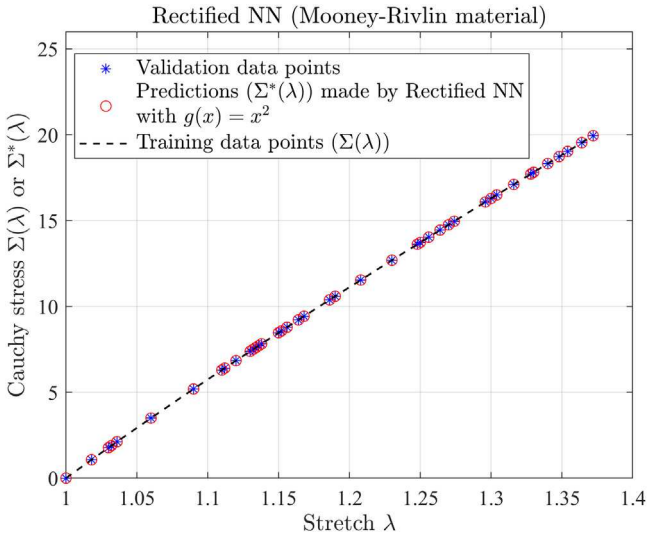

Fig. 6. Reference stress response $\lambda \mapsto \Sigma(\lambda)$ (black dashed line), reference values (blue star), and rectified neural network predictions (red circle) for the validation dataset (random selection). Here, 5 hidden layers and 40 neurons per layer are used.

per layer for both strategies, based on a parametric convergence study on the validation metric. It is seen that the rectified NN converges much faster than the ICNN and leads to a lower training error. While similar results were obtained for a wide range of architectures, it should be pointed out that the computational cost per iteration is greater in the case of rectified NNs, which require numerical integration to be performed. In addition, opposite trends with respect to learning rates were observed: ICNNs tend to perform slightly better at large learning rates (*e.g.*, 1.5) but require a very large number of iterations (greater than 40,000) at standard learning rates (*e.g.*, 0.001). These observations are most likely imputable to the transformation of negative weights, which generates large regions with "flat" gradients in the optimization process. In contrast, rectified NNs were found to perform steadily in terms of training cost, regardless of the learning rate, and typically performs (much) better at smaller rates. An extensive comparison of the tradeoffs between these approaches is left for future work.

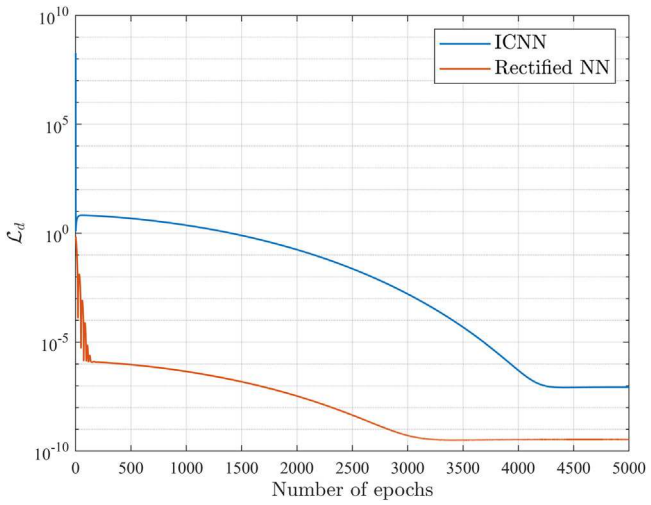

Fig. 7. Loss history for ICNN and the proposed rectified NN.

3.3. Anisotropic model for digital dataset

We next consider an anisotropic strain energy density function, relevant to the modeling of soft biological tissues such as arterial vessels [29,30]. It should be noticed such materials are often modeled as nearly-incompressible in a computational setting, in which case the strain energy density function is typically expressed in terms of isochoric invariants. The reference function is defined as

$$w(I_1, I_2, J_4^{(1)}, J_4^{(2)}) = w^{\mathrm{MR}}(I_1, I_2) + w^{\mathrm{A}}(J_4^{(1)}, J_4^{(2)}), \tag{32}$$

where w^{MR} is given by Eq. (23) and the anisotropic term is defined as

$$w^{\mathrm{A}}(J_4^{(1)}, J_4^{(2)}) = \sum_{k=1}^{2} w^{\mathrm{ti}}(J_4^{(k)}), \tag{33}$$

with

$$w^{\mathrm{ti}}(J_4^{(k)}) = \frac{\mu_4}{\beta_4}\left\{\exp\left(\beta_4 w^B(I_1, J_4^{(k)})\right) - 1\right\} \tag{34}$$

where $w^B(I_1, J_4^{(k)}) = (1-\rho)(I_1-3)^2 + \rho\langle J_4^{(k)} - 1\rangle_m^2$, and $J_4^{(k)} = \mathrm{tr}(\boldsymbol{C}\boldsymbol{M}^{(k)})$. The structural tensors $\boldsymbol{M}^{(k)} = \boldsymbol{a}^{(k)} \otimes \boldsymbol{a}^{(k)}$ are defined in terms of the unit vectors

$$\boldsymbol{a}^{(1)} = \cos(\alpha)\boldsymbol{e}^{(1)} + \sin(\alpha)\boldsymbol{e}^{(2)}, \tag{35a}$$

$$\boldsymbol{a}^{(2)} = \cos(\alpha)\boldsymbol{e}^{(1)} - \sin(\alpha)\boldsymbol{e}^{(2)}, \tag{35b}$$

where $\boldsymbol{e}^{(1)}$ and $\boldsymbol{e}^{(2)}$ are unit basis vectors and α is the angle defining the directions of anisotropy. In Eq. (34), μ_4, β_4, and ρ are material parameters, and $\langle\cdot\rangle_m$ denotes the Macaulay bracket. Note that the angle α is also considered as a trainable parameter.

The rectified neural network is sought as

$$w_1^*(I_1) + w_2^*(I_2) + w_3^*(J_4^{(1)}) + w_4^*(J_4^{(2)}), \tag{36}$$

where

$$w_i^*(I_i) = \mathcal{R}\{\psi_i(\{\boldsymbol{W}_j^{(i)}, \boldsymbol{b}_j^{(i)}\}_{j=1}^{n_i})\}(I_i), \quad i = 1, 2, \tag{37}$$

$$w_3^*(J_4^{(1)}) = \mathcal{R}\{\psi_3(\{\boldsymbol{W}_j^{(3)}, \boldsymbol{b}_j^{(3)}\}_{j=1}^{n_3})\}(J_4^{(1)}), \tag{38}$$

and

$$w_4^*(J_4^{(2)}) = \mathcal{R}\{\psi_4(\{\boldsymbol{W}_j^{(4)}, \boldsymbol{b}_j^{(4)}\}_{j=1}^{n_4})\}(J_4^{(2)}). \tag{39}$$

Biaxial tension is used for training purposes. The Cauchy stress associated with the reference model is obtained as

$$\Sigma(\lambda) = \Sigma^{\mathrm{MR}}(\lambda) + \Sigma_{(1)}^{\mathrm{ti}}(\lambda) + \Sigma_{(2)}^{\mathrm{ti}}(\lambda), \tag{40}$$

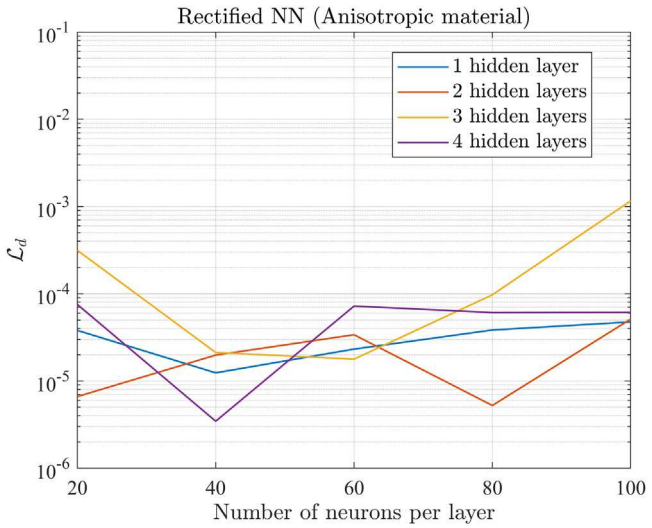

Fig. 8. Parametric study of the mean squared error for different NN architectures on the validation dataset (Anisotropic material).

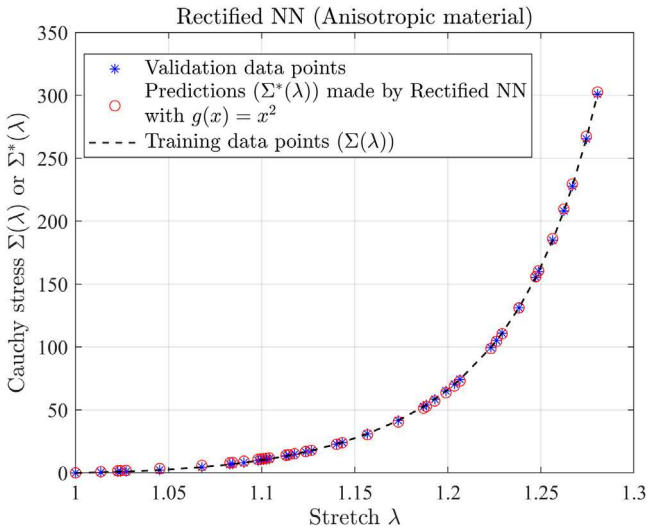

Fig. 9. Reference stress response $\lambda \mapsto \Sigma(\lambda)$ (black dashed line), reference values (blue star), and rectified neural network predictions (red circle) for the validation dataset (random selection). Here, the NN involves 4 hidden layers and 40 neurons per layer.

with

$$\Sigma^{\mathrm{MR}}(\lambda) = 2C_1(\lambda^2 - \frac{1}{\lambda^4}) - 2C_2(\frac{1}{\lambda^2} - \lambda^4)\,, \tag{41}$$

and

$$\begin{aligned}\Sigma^{\mathrm{ti}}_{(k)}(\lambda) = 4\lambda^2\mu_4 \Big\{ &(1-\rho)(I_1-3)\left(1-\frac{1}{\lambda^6}\right) \\ &+ \rho\langle J_4^{(k)} - 1\rangle_m \cos^2(\alpha)\Big\} \\ &\times \exp(\beta_4 w^B(I_1, J_4^{(k)}))\,, \quad k = 1,2\,.\end{aligned} \tag{42}$$

with a slight abuse of notation. The Cauchy stress for the rectified model defined by Eq. (36) is given by

$$\Sigma^* = \Sigma_1^*(\lambda) + \Sigma_2^*(\lambda) + \Sigma_3^*(\lambda) + \Sigma_4^*(\lambda)\,, \tag{43}$$

where terms in the right-hand side are defined as

$$\Sigma_1^*(\lambda) = 2\frac{\partial w_1^*(I_1)}{\partial I_1}\left(\lambda^2 - \frac{1}{\lambda^4}\right)\,, \tag{44}$$

$$\Sigma_2^*(\lambda) = -2\frac{\partial w_2^*(I_2)}{\partial I_2}\left(\frac{1}{\lambda^2} - \lambda^4\right)\,, \tag{45}$$

$$\Sigma_3^*(\lambda) = 2\lambda^2\frac{\partial w_3^*(J_4^{(1)})}{\partial J_4^{(1)}}\cos^2(\alpha)\,, \tag{46}$$

$$\Sigma_4^*(\lambda) = 2\lambda^2\frac{\partial w_4^*(J_4^{(2)})}{\partial J_4^{(2)}}\cos^2(\alpha)\,. \tag{47}$$

Since $\cos(\alpha) \neq 0$ in practice, the stress free constraint reduces to $\Sigma_3^*(1) + \Sigma_4^*(1) = 0$. In the numerical example below, the material parameters correspond to the values identified in [31] for sample #10 in the media layer: $C_1 = 0.7071$ [kPa], $C_2 = 0.0531$ [kPa], $\alpha = 0.2740$ [rad], $\mu_4 = 15.5753$ [kPa], $\beta_4 = 2.5561$, and $\rho = 0.0986$. Similarly to the previous case, we generated 200 datapoints, 20% of which are used as the validation set. The learning rate is set to 0.005 for the first 3,000 epochs and then to 0.001 for 3,000 epochs, and presented results were obtained using the exponential function in the rectifier (i.e., $g(x) = \exp(x)$). The validation errors for different neural network architectures are shown in Fig. 8.

Results predicted with the fitted rectified neural network model on the validation dataset are shown in Figs. 9. The validation metric in this example is 3.46×10^{-6}.

3.4. Anisotropic model for experimental dataset

We finally apply the proposed rectification method to the experimental dataset presented in [32], corresponding to uniaxial extension tests on human illiac arterial walls. In those experiments, two different strips were harvested along the circumferential and longitudinal directions on each specimen to capture anisotropic effects. For the sake of illustration, two samples are randomly selected as target data for each layer defining the artery (adventitia, media, intima), and 10% of the data is used as the validation dataset. The rectified neural network is similar to the one used in Section 3.3 (see Eq. (36)).

Both axial and circumferential tension data are used for training. The Cauchy stress for the rectified model defined by Eq. (36) in the circumferential direction reads as in Eq. (43), in which

$$\Sigma_1^*(\lambda) = 2\frac{\partial w_1^*}{\partial I_1}\left(\lambda^2 - \frac{1}{\lambda}\right)\,, \tag{48}$$

$$\Sigma_2^*(\lambda) = 2\frac{\partial w_2^*}{\partial I_2}\left(\lambda - \frac{1}{\lambda^2}\right)\,, \tag{49}$$

$$\Sigma_3^*(\lambda) = 2\lambda^2\frac{\partial w_3^*(J_4^{(1)})}{\partial J_4^{(1)}}\cos^2(\alpha)\,, \tag{50}$$

$$\Sigma_4^*(\lambda) = 2\lambda^2\frac{\partial w_4^*(J_4^{(2)})}{\partial J_4^{(2)}}\cos^2(\alpha)\,. \tag{51}$$

for uniaxial elongation.

The Cauchy stress for the tissue contribution in the longitudinal direction involves the terms

$$\Sigma_3^*(\lambda) = 2\lambda^2\frac{\partial w_3^*(J_4^{(1)})}{\partial J_4^{(1)}}\sin^2(\alpha) \tag{52}$$

and

$$\Sigma_4^*(\lambda) = 2\lambda^2\frac{\partial w_4^*(J_4^{(2)})}{\partial J_4^{(2)}}\sin^2(\alpha)\,. \tag{53}$$

The loss function in terms of datapoints is then defined as

$$\begin{aligned}\mathcal{L}_d = &\frac{\sum_{i=1}^{n_p^c}\left(\Sigma^{\exp}(\lambda_1^c) - \Sigma^*(\lambda_1^c;\boldsymbol{p})\right)^2}{\sum_{i=1}^{n_p^c}\Sigma^{\exp}(\lambda_1^c)^2} \\ &+ \frac{\sum_{i=1}^{n_p^a}\left(\Sigma^{\exp}(\lambda_1^a) - \Sigma^*(\lambda_1^a;\boldsymbol{p})\right)^2}{\sum_{i=1}^{n_p^a}\Sigma^{\exp}(\lambda_1^a)^2}\,,\end{aligned} \tag{54}$$

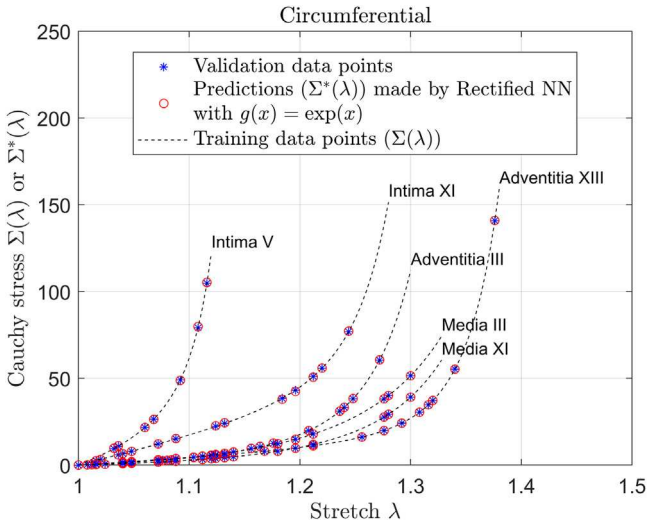

Fig. 10. Reference stress response $\lambda \mapsto \Sigma(\lambda)$ (black dashed line), reference values (blue star), and rectified neural network predictions (red circle) for the validation dataset in the circumferential direction (six experimental responses are considered for illustration purposes).

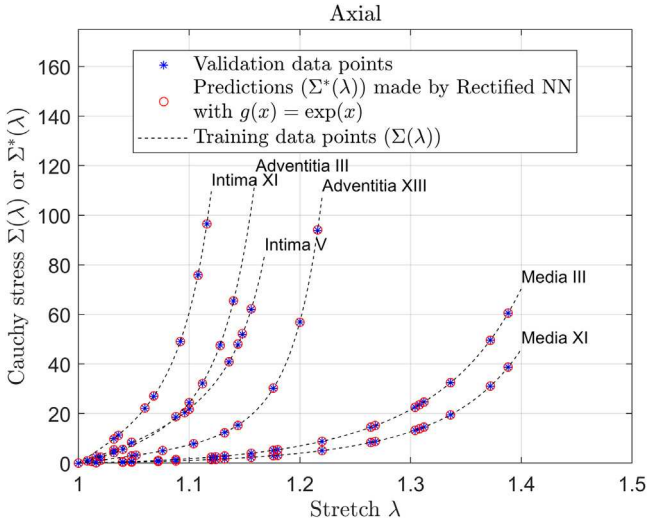

Fig. 11. Reference stress response $\lambda \mapsto \Sigma(\lambda)$ (black dashed line), reference values (blue star), and rectified neural network predictions (red circle) for the validation dataset in the axial direction (six experimental responses are considered for illustration purposes).

where the superscripts "c" and "a" refer to data obtained by stretching along the circumferential and axial directions, respectively, n_p^c and n_p^a are the associated numbers of datapoints. As in Section 3.3, the stress free constraint is enforced in a strong sense by shifting the Cauchy stress.

Predictions obtained with the rectified neural network can be qualitatively compared with reference values in Figs. 10 and 11, while validation errors can be found in Table 2. In this application, we used 2 hidden layers per neural network and 100 neurons per hidden layer. The rectified NN is trained for 2,000 epochs with a learning rate set to 0.01, then for 2,000 epochs with a learning rate taken as 0.001, and finally for 2,000 epochs at a learning rate set to 0.0001. With a maximal validation error equal to 1.0096×10^{-4} (obtained for sample #XI, intima layer), it is seen that the rectified neural network can reproduce the experimental data very well, for all different layers in the two directions.

Table 2
Validation errors for each sample. Specimen numbers are those reported in [32].

Layer/Specimen number	Error $\mathcal{L}_d$ $\times 10^{-4}$
Adventitia/#III	0.9972
Adventitia/#XIII	0.1331
Intima/#V	0.7994
Intima/#XI	1.0096
Media/#III	0.0252
Media/#XI	0.0355

4. Conclusion

A method to correct unconstrained neural networks for hyperelastic models was proposed. The approach relies on a composite mapping that transforms any function into a convex function, hence ensuring the polyconvexity of the neural network — without constraints on the weights and activation functions.

The strategy was first illustrated on a toy problem to characterize the impact of the positive function used to enforce monotonicity (in terms of accuracy and training effort). The rectified NN models were then deployed on digitally synthesized and experimental datasets, relevant to both isotropic and anisotropic materials. Good fitting capabilities were observed in all applications. It was shown that the proposed rectified models typically convergence faster than *a priori* constrained models, at the expense of a greater computational cost per iteration.

Avenues for future research include the generalization to other types of strain energy density functions, as well as more extensive comparisons with *a priori* constrained representations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

Financial support of the National Science Foundation (NSF), USA under grants CMMI-1726403, CMMI-1942928, and DGE-2022040 is gratefully acknowledged.

References

[1] Moritz Flaschel, Siddhant Kumar, Laura De Lorenzis, Unsupervised discovery of interpretable hyperelastic constitutive laws, Comput. Methods Appl. Mech. Engrg. 381 (2021) 113852.
[2] Kailai Xu, Daniel Z. Huang, Eric Darve, Learning constitutive relations using symmetric positive definite neural networks, J. Comput. Phys. 428 (2021) 110072.
[3] Gerhard A Holzapfel, Kevin Linka, Selda Sherifova, Christian J Cyron, Predictive constitutive modelling of arteries by deep learning, J. R. Soc. Interface 18 (182) (2021) 20210411.
[4] Sungmoon Jung, Jamshid Ghaboussi, Neural network constitutive model for rate-dependent materials, Comput. Struct. 84 (15–16) (2006) 955–963.
[5] Jamshid Ghaboussi, David A Pecknold, Mingfu Zhang, Rami M Haj-Ali, Autoprogressive training of neural network constitutive models, Internat. J. Numer. Methods Engrg. 42 (1) (1998) 105–126.
[6] Jamshid Ghaboussi, Djoni Eka Sidarta, New nested adaptive neural networks (NANN) for constitutive modeling, Comput. Geotech. 22 (1) (1998) 29–52.
[7] Youssef Hashash, Sungmoon Jung, Jamshid Ghaboussi, Numerical implementation of a neural network based material model in finite element analysis, Internat. J. Numer. Methods Engrg. 59 (7) (2004) 989–1005.

[8] Tomonari Furukawa, Genki Yagawa, Implicit constitutive modelling for viscoplasticity using neural networks, Internat. J. Numer. Methods Engrg. 43 (2) (1998) 195–219.
[9] Akshay Joshi, Prakash Thakolkaran, Yiwen Zheng, Maxime Escande, Moritz Flaschel, Laura De Lorenzis, Siddhant Kumar, Bayesian-EUCLID: discovering hyperelastic material laws with uncertainties, 2022, arXiv preprint arXiv:2203.07422.
[10] Faisal As'ad, Philip Avery, Charbel Farhat, A mechanics-informed artificial neural network approach in data-driven constitutive modeling, in: AIAA Scitech 2022 Forum, 2022, p. 0100.
[11] Faisal As'ad, Philip Avery, Charbel Farhat, A mechanics-informed artificial neural network approach in data-driven constitutive modeling, Internat. J. Numer. Methods Engrg. 123 (12) (2022) 2738–2759.
[12] Dominik K. Klein, Mauricio Fernández, Robert J. Martin, Patrizio Neff, Oliver Weeger, Polyconvex anisotropic hyperelasticity with neural networks, J. Mech. Phys. Solids 159 (2022) 104703.
[13] Minliang Liu, Liang Liang, Wei Sun, A generic physics-informed neural network-based constitutive model for soft biological tissues, Comput. Methods Appl. Mech. Engrg. 372 (2020) 113402.
[14] Brandon Amos, Lei Xu, J. Zico Kolter, Input convex neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 146–155.
[15] Sarath Sivaprasad, Ankur Singh, Naresh Manwani, Vineet Gandhi, The curious case of convex neural networks, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 738–754.
[16] Filippo Masi, Ioannis Stefanou, Thermodynamics-based artificial neural networks (TANN) for multiscale modeling of materials with inelastic microstructure, 2021, arXiv preprint arXiv:2108.13137.
[17] Antoine Wehenkel, Gilles Louppe, Unconstrained monotonic neural networks, Adv. Neural Inf. Process. Syst. 32 (2019).
[18] Ricardo Baptista, Olivier Zahm, Youssef Marzouk, An adaptive transport framework for joint and conditional density estimation, 2020, arXiv preprint arXiv:2009.10303.
[19] Philippe G. Ciarlet, Mathematical Elasticity, Vol. 1, Three-dimensional Elasticity Elsevier Science Publishers BV, 1988.
[20] Clifford Truesdell, Walter Noll, The non-linear field theories of mechanics, in: The Non-Linear Field Theories of Mechanics, Springer, 2004, pp. 1–579.
[21] Pablo Pedregal, Variational Methods in Nonlinear Elasticity, SIAM, 2000.
[22] Bernard Dacorogna, Direct Methods in the Calculus of Varations, second ed., Springer, 1989.
[23] John M. Ball, Convexity conditions and existence theorems in nonlinear elasticity, Arch. Ration. Mech. Anal. 63 (4) (1976) 337–403.
[24] Stefan Hartmann, Patrizio Neff, Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility, Int. J. Solids Struct. 40 (11) (2003) 2767–2791.
[25] Jörg Schröder, Patrizio Neff, Invariant formulation of hyperelastic transverse isotropy based on polyconvex free energy functions, Int. J. Solids Struct. 40 (2) (2003) 401–445.
[26] Vera Ebbing, Jörg Schröder, Patrizio Neff, Construction of polyconvex energies for non-trivial anisotropy classes, in: Poly-, Quasi-and Rank-One Convexity in Applied Mechanics, Springer, 2010, pp. 107–130.
[27] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, Qiao Zhang, JAX: composable transformations of Python+NumPy programs, 2018.
[28] Gerhard A. Holzapfel, Nonlinear Solids Mechanics: A Continuum Approach for Engineering, Wiley, Chichester, 2000.
[29] Daniel Balzani, Patrizio Neff, Jörg Schröder, Gerhard A Holzapfel, A polyconvex framework for soft biological tissues. Adjustment to experimental data, Int. J. Solids Struct. 43 (20) (2006) 6052–6070.
[30] Brian Staber, Johann Guilleminot, A random field model for anisotropic strain energy functions and its application for uncertainty quantification in vascular mechanics, Comput. Methods Appl. Mech. Engrg. 333 (2018) 94–113.
[31] Peiyi Chen, Johann Guilleminot, Spatially-dependent material uncertainties in anisotropic nonlinear elasticity: Stochastic modeling, identification, and propagation, Comput. Methods Appl. Mech. Engrg. 394 (2022) 114897.
[32] Gerhard A Holzapfel, Gerhard Sommer, Christian T Gasser, Peter Regitnig, Determination of layer-specific mechanical properties of human coronary arteries with nonatherosclerotic intimal thickening and related constitutive modeling, Am. J. Physiol.-Heart Circul. Physiol. 289 (5) (2005) H2048–H2058.