A new family of Constitutive Artificial Neural Networks towards automated model discovery

Kevin Linka, Ellen Kuhl*

*Department of Mechanical Engineering, Stanford University, Stanford, CA, United States*

Dataset link: https://github.com/LivingMatterLab/CANN

We dedicate this manuscript to our Continuum Mechanics teachers, whose insights and passion for Continuum Mechanics have stimulated the ideas of this work, Wolfgang Ehlers, Mikhail Itskov, Christian Miehe, Michael Ortiz, Jörg Schröder, Erwin Stein, and Paul Steinmann

Graphical Abstract
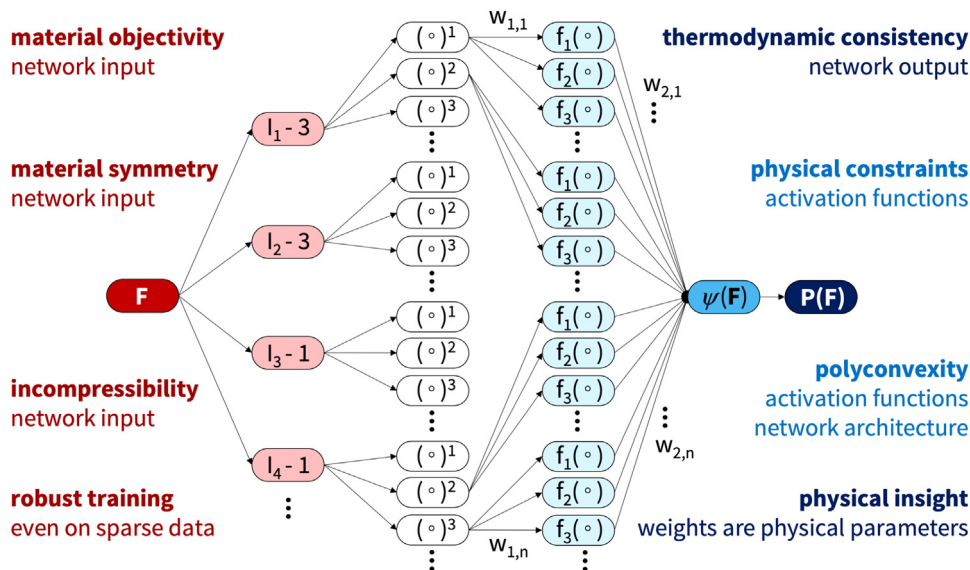
Abstract

For more than 100 years, chemical, physical, and material scientists have proposed competing constitutive models to best characterize the behavior of natural and man-made materials in response to mechanical loading. Now, computer science offers a

* Corresponding author.
  E-mail address: ekuhl@stanford.edu (E. Kuhl).

universal solution: Neural Networks. Neural Networks are powerful function approximators that can learn constitutive relations from large data without any knowledge of the underlying physics. However, classical Neural Networks entirely ignore a century of research in constitutive modeling, violate thermodynamic considerations, and fail to predict the behavior outside the training regime. Here we design a new family of Constitutive Artificial Neural Networks that inherently satisfy common kinematical, thermodynamical, and physical constraints and, at the same time, constrain the design space of admissible functions to create robust approximators, even in the presence of sparse data. Towards this goal we revisit the non-linear field theories of mechanics and reverse-engineer the network input to account for material objectivity, material symmetry and incompressibility; the network output to enforce thermodynamic consistency; the activation functions to implement physically reasonable restrictions; and the network architecture to ensure polyconvexity. We demonstrate that this new class of models is a generalization of the classical neo Hooke, Blatz Ko, Mooney Rivlin, Yeoh, and Demiray models and that the network weights have a clear physical interpretation in the form of shear moduli, stiffness-like parameters, and exponential coefficients. When trained with classical benchmark data for rubber under uniaxial tension, biaxial extension, and pure shear, our network autonomously selects the best constitutive model and learns its set of parameters. Our findings suggest that Constitutive Artificial Neural Networks have the potential to induce a paradigm shift in constitutive modeling, from user-defined model selection to automated model discovery. Our source code, data, and examples is available at https://github.com/LivingMatterLab/CANN.

1. Motivation

"What can your Neural Network tell you about the underlying physics?" is the most common question when we apply Neural Networks to study the behavior of materials and *"Nothing".* is the honest and disappointing answer. This manuscript challenges the notion that Neural Networks can teach us nothing about the physics of a material. It seeks to integrate more than a century of knowledge in continuum mechanics [1–8] and modern machine learning [9–11] to create a new family of Constitutive Artificial Neural Networks that inherently satisfy kinematical, thermodynamical, and physical constraints, and constrain the space of admissible functions to train robustly, even when data are space. While this general idea is by no means new and builds on several important recent discoveries [12–15], the true novelty of our Constitutive Artificial Neural Networks is that they *autonomously discover a constitutive model*, and, at the same time, *learn a set of physically meaningful parameters* associated with it.

Interestingly, the first Neural Network for constitutive modeling approximates the incremental principal strains in concrete from known principal strains, stresses, and stress increments and is more than three decades old [16]. In the early days, Neural Networks served merely as regression operators and were commonly viewed as a black box. The lack of transparency is probably the main reason why these early approaches never really generated momentum in the constitutive modeling community. More than 20 years later, data-driven constitutive modeling gained new traction, in part powered by a new computing paradigm, which directly uses experimental data and bypasses constitutive modeling altogether [17]. While data-driven elasticity builds on a transparent and rigorous mathematical foundation [18], it can also become fairly complex, especially when expanding the theory to anisotropic [19] or history-dependent [20] materials. Rather than following this path and eliminate the constitutive model entirely, here we attempt to build our prior physical knowledge into the Neural Network and learn something about the constitutive response [21].

Two successful but fundamentally different strategies have emerged to integrate physical knowledge into network modeling, *Physics-Informed Neural Networks* that add physics equations as additional terms to the loss function [9] and *Constitutive Artificial Neural Networks* that explicitly modify the network input, output, and architecture to hardwire physical constraints into the network design [14]. The former approach is more general and typically works well for incorporating ordinary [10] or partial [11] differential equations, while the latter is specifically tailored towards constitutive equations [22]. In fact, one such Neural Network, with strain invariants as input, free energy functions as output, and a single hidden layer with logistic activation functions in between, has been proposed for rubber materials almost two decades ago [23] and recently regained attention in the constitutive modeling community [24]. While these Constitutive Artificial Neural Networks generally provide excellent fits to experimental data [25–27], exactly how they should integrate thermodynamic constraints remains a question of ongoing debate.

Thermodynamics-based Artificial Neural Networks a priori build the first and second law of thermodynamics into the network architecture and select specific activation functions to ensure compliance with thermodynamic constraints [15]. Recent studies suggest that this approach can successfully reproduce the constitutive behavior of rubber-like materials [28]. Alternative approaches use a regular Artificial Neural Network and ensure thermodynamic consistency a posteriori via a pseudo-potential based correction in a post processing step [29]. To demonstrate the versatility of these different approaches, several recent studies have successfully embedded Neural Networks within a Finite Element Analysis, for example, to model plane rubber sheets [14], sheets with holes [30], or entire tires [23], the numerical homogenization of discrete lattice structures [31], the deployment of parachutes [12], or the anisotropic response of skin in reconstructive surgery [32]. Regardless of all these success stories, one limitation remains: the lack of an intuitive interpretation of the model and its parameters [13].

The general idea of this manuscript is to reverse-engineer a new family of Constitutive Artificial Neural Networks that are, by design, a generalization of widely used and commonly accepted hyperelastic constitutive models [25,26,33–36] with well-defined physical parameters [37,38]. Towards this goal, we review the underlying kinematics in Section 2 and discuss constitutive constraints in Section 3. We then introduce classical Neural Networks in Section 4 and our new family of Constitutive Artificial Neural Networks in Section 5. In Section 6, we briefly review the three special homogeneous deformation modes that we use to train our model in Section 7. We discuss our results, limitations, and future directions in Section 8 and close with a brief conclusion in Section 9.

2. Kinematics

We begin by characterizing the motion of a body and introduce the deformation map φ that, at any point in time t, maps material particles X from the undeformed configuration to particles, $x = \varphi(X, t)$, in the deformed configuration [1]. To characterize relative deformations within the body, we introduce the deformation gradient F, the gradient of the deformation map φ with respect to the undeformed coordinates X, and its Jacobian J,

$$F = \nabla_X \varphi \qquad \text{with} \qquad J = \det(F) > 0 \,. \tag{1}$$

Multiplying F with its transpose F^t, either from the left or the right, introduces the right and left Cauchy Green deformation tensors C and b,

$$C = F^t \cdot F \qquad \text{and} \qquad b = F \cdot F^t \,. \tag{2}$$

In the undeformed state, all three tensors are identical to the unit tensor, $F = I$, $C = I$, and $b = I$, and the Jacobian is one, $J = 1$. A Jacobian smaller than one, $0 < J < 1$, denotes compression and a Jacobian larger than one, $1 < J$, denotes extension.

Isotropy. To characterize an *isotropic* material, we introduce the three principal invariants I_1, I_2, I_3, either in terms of the deformation gradient F,

$$
\begin{aligned}
I_1 &= F : F & \partial_F I_1 &= 2F \\
I_2 &= \tfrac{1}{2}[I_1^2 - [F^t \cdot F] : [F^t \cdot F]] & \text{with} \qquad \partial_F I_2 &= 2[I_1 F - F \cdot F^t \cdot F] \\
I_3 &= \det(F^t \cdot F) = J^2 & \partial_F I_3 &= 2 I_3 F^{-t} \,,
\end{aligned} \tag{3}
$$

or, equivalently, in terms of the right or left Cauchy Green deformation tensors C or b,

$$
\begin{aligned}
I_1 &= \text{tr}(C) = C : I & \partial_C I_1 &= I & I_1 &= \text{tr}(b) = b : I & \partial_b I_1 &= I \\
I_2 &= \tfrac{1}{2}[I_1^2 - C : C] & \partial_C I_2 &= I_1 I - C & \text{or} \quad I_2 &= \tfrac{1}{2}[I_1^2 - b : b] & \partial_b I_2 &= I_1 I - b \\
I_3 &= \det(C) = J^2 & \partial_C I_3 &= I_3 C^{-t} & I_3 &= \det(b) = J^2 & \partial_b I_3 &= I_3 b^{-t} \,.
\end{aligned} \tag{4}
$$

In the undeformed state, $F = I$, and the three invariants are equal to three and one, $I_1 = 3$, $I_2 = 3$, and $I_3 = 1$.

Near incompressibility. To characterize an *isotropic*, *nearly incompressible* material, we perform a multiplicative decomposition of deformation gradient, $F = J^{1/3} I \cdot \bar{F}$, into a volumetric part, $J^{1/3} I$, and an isochoric part, $\bar{F}$ [39],

$$\bar{F} = J^{-1/3} F \qquad \text{and} \qquad \bar{J} = \det(\bar{F}) = 1 \,, \tag{5}$$

and introduce the isochoric right and left Cauchy Green deformation tensors $\bar{C}$ and $\bar{b}$,

$$\bar{C} = \bar{F}^{\mathrm{t}} \cdot \bar{F} = J^{-2/3}\, C \qquad \text{and} \qquad \bar{b} = \bar{F} \cdot \bar{F}^{\mathrm{t}} = J^{-2/3}\, b \,. \tag{6}$$

We can then introduce an alternative set of invariants for nearly incompressible materials, $\bar{I}_1$, $\bar{I}_2$, J, in terms of the deformation gradient $\bar{F}$,

$$
\begin{aligned}
\bar{I}_1 &= I_1/J^{2/3} = F : F/J^{2/3} & \partial_F \bar{I}_1 &= 2/J^{2/3}\, F - \tfrac{2}{3}\, \bar{I}_1\, F^{-\mathrm{t}} \\
\bar{I}_2 &= I_2/J^{4/3} = \tfrac{1}{2}[\bar{I}_1 - [F^{\mathrm{t}} \cdot F] : [F^{\mathrm{t}} \cdot F]/J^{4/3}] \quad \text{with} \quad & \partial_F \bar{I}_2 &= 2/J^{2/3}\, \bar{I}_1\, F - 2/J^{4/3}\, F \cdot F^{\mathrm{t}} \cdot F - \tfrac{4}{3}\, \bar{I}_2\, F^{-\mathrm{t}} \\
J &= \det(F) & \partial_F J &= J\, F^{-\mathrm{t}} \,,
\end{aligned}
\tag{7}
$$

or, equivalently, in terms of the right and left Cauchy Green deformation tenors C or b,

$$
\begin{aligned}
\bar{I}_1 &= I_1/J^{2/3} = C : I/J^{2/3} & \bar{I}_1 &= I_1/J^{2/3} = b : I/J^{2/3} \\
\bar{I}_2 &= I_2/J^{4/3} = \tfrac{1}{2}[\bar{I}_1 - C : C/J^{4/3}] \quad \text{or} \quad & \bar{I}_2 &= I_2/J^{4/3} = \tfrac{1}{2}[\bar{I}_1 - b : b/J^{4/3}] \\
J &= \det^{1/2}(C) & J &= \det^{1/2}(b) \,.
\end{aligned}
\tag{8}
$$

Perfect incompressibility. To characterize an *isotropic, perfectly incompressible* material, we recall that the third invariant always remains identical to one, $I_3 = J^2 = 1$. This implies that the principal and isochoric invariants are identical, $I_1 = \bar{I}_1$ and $I_2 = \bar{I}_2$, and that the set of invariants reduces to only these two.

Transverse isotropy. To characterize a *transversely isotropic* material with one constant pronounced direction with unit normal vector n, we introduce a fourth and fifth invariant [6],

$$
\begin{aligned}
I_4 &= n \cdot F^{\mathrm{t}} \cdot F \cdot n & &= C : N = \lambda_n^2 & \partial_C I_4 &= n \otimes n = N \\
I_5 &= n \cdot F^{\mathrm{t}} \cdot F \cdot F^{\mathrm{t}} \cdot F \cdot n &= [C \cdot C] : N \quad & \text{with} \quad & \partial_C I_5 &= n \otimes C \cdot n + C \cdot n \otimes n \,.
\end{aligned}
\tag{9}
$$

Here $N = n \otimes n$ denotes the structural tensor associated with the pronounced direction n, with a unit length of $\| n \| = 1$ in the reference configuration and a stretch of $\lambda_n = \| F \cdot n \|$ in the deformed configuration. In the undeformed state, $F = I$, and the stretch, the fourth and fifth invariants are one, $\lambda_n = 1$ and $I_4 = 1$ and $I_5 = 1$.

3. Constitutive equations

In the most general form, constitutive equations in solid mechanics are tensor-valued tensor functions that define the relation between a stress, for example the Piola or nominal stress, $P = \lim_{\mathrm{d}A \to 0} (\mathrm{d}f/\mathrm{d}A)$, as the force $\mathrm{d}f$ per undeformed area $\mathrm{d}A$, and a deformation measure, for example the deformation gradient F [3,7],

$$P = P(F) \,. \tag{10}$$

Conceptually, we could use any Neural Network as a function approximator to simply learn the functional relation between P and F and many approaches in the literature actually do exactly that [15,16,40]. However, the functions $P(F)$ that we learn through this approach might be too generic and violate well-known thermodynamical arguments and widely-accepted physical constraints [28]. Also, for limited amounts of data, the tensor-valued tensor function $P(F)$ can be difficult to learn and there is a high risk of overfitting [13]. Our objective is therefore to design a Constitutive Artificial Neural Network that a priori guarantees thermodynamic consistency of the function $P(F)$, and, at the same time, conveniently limits the space of admissible functions to ensure robustness and prevent overfitting when available data are sparse.

Thermodynamic consistency. As a first step towards this goal, we ensure *thermodynamically consistency* and guarantee that the Piola stress P inherently satisfies the second law of thermodynamics, the entropy or Clausius–Duhem inequality [5], $\mathcal{D} = P : \dot{F} - \dot{\psi} \geq 0$. It states that, for any thermodynamic process, the total change in entropy, the dissipation $\mathcal{D}$, should always remain greater than or equal to zero, $\mathcal{D} \geq 0$. To a priori satisfy the dissipation inequality, we introduce the Helmholtz free energy as a function of the deformation gradient, $\psi = \psi(F)$ such that $\dot{\psi} = \partial\psi(F)/\partial F : \dot{F}$, and rewrite the dissipation inequality following the Coleman–Noll entropy principle [7] as $\mathcal{D} = [P - \partial\psi/\partial F] : \dot{F} \geq 0$. For the *hyperelastic* case with $\mathcal{D} \doteq 0$, for all possible $\dot{F}$, the entropy inequality reduces to $P - \partial\psi/\partial F \doteq 0$. The condition of thermodynamically consistency implies that the Piola stress P of a

hyperelastic, Green-elastic material is a thermodynamically conjugate function of the deformation gradient $\boldsymbol{F}$ [8],

$$\boldsymbol{P} = \frac{\partial \psi(\boldsymbol{F})}{\partial \boldsymbol{F}} \ . \tag{11}$$

For our Neural Network, this implies that, rather than approximating the nine stress components $\boldsymbol{P}(\boldsymbol{F})$ as nine generic functions of the nine components of the deformation gradient $\boldsymbol{F}$, we train the network to learn the free energy function $\psi(\boldsymbol{F})$ and derive the stress $\boldsymbol{P}$ in a post-processing step to a priori satisfy the second law of thermodynamics. As such, satisfying thermodynamic consistency according to Eq. (11) directly affects the *output* of the Neural Network.

Material objectivity and frame indifference. Second, we further constrain the choice of the free energy function ψ to satisfy *material objectivity* or *frame indifference* to ensure that the constitutive laws do not depend on the external frame of Ref. [41]. Mathematically speaking, the Helmholtz free energy has to be invariant under rigid body motions, $\psi(\boldsymbol{F}) = \psi(\boldsymbol{Q} \cdot \boldsymbol{F})$, for all proper orthogonal tensors $\boldsymbol{Q} \in \mathrm{SO}(3)$. The condition of objectivity implies that the arguments of the free energy function are independent of rotations and must be functions of the right Cauchy Green deformation tensor $\boldsymbol{C}$ [7],

$$\boldsymbol{P} = \frac{\partial \psi(\boldsymbol{C})}{\partial \boldsymbol{F}} = \frac{\partial \psi(\boldsymbol{C})}{\partial \boldsymbol{C}} : \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{F}} = 2\, \boldsymbol{F} \cdot \frac{\partial \psi(\boldsymbol{C})}{\partial \boldsymbol{C}} \ . \tag{12}$$

For our Neural Network, this implies that rather than using the nine independent components of the deformation gradient $\boldsymbol{F}$ as input, we constrain the input to the six independent components of the symmetric right Cauchy Green deformation tensor, $\boldsymbol{C} = \boldsymbol{F}^{\mathrm{t}} \cdot \boldsymbol{F}$. As such, satisfying material objectivity according to Eq. (12) directly affects the *input* of the Neural Network.

Material symmetry and isotropy. Third, we further constrain the choice of the free energy function ψ to include constraints of *material symmetry*, which implies that the material response remains unchanged under transformations of the reference configuration, $\psi(\boldsymbol{F}) = \psi(\boldsymbol{F} \cdot \boldsymbol{Q})$. Here we consider the special case of *isotropy* for which the material response remains unchanged under proper orthogonal transformations of the reference configuration, $\psi(\boldsymbol{F}^{\mathrm{t}} \cdot \boldsymbol{F}) = \psi(\boldsymbol{Q}^{\mathrm{t}} \cdot \boldsymbol{F}^{\mathrm{t}} \cdot \boldsymbol{F} \cdot \boldsymbol{Q})$, for *all* proper orthogonal tensors $\boldsymbol{Q} \in \mathrm{SO}(3)$ [1]. The condition of isotropy implies that the stress response functions, $\psi(\boldsymbol{C}) = \psi(\boldsymbol{b})$, must be functions of the left Cauchy Green deformation tensor, $\boldsymbol{b} = \boldsymbol{F} \cdot \boldsymbol{F}^{\mathrm{t}}$, and, together with the condition of objectivity, $\psi(\boldsymbol{b}) = \psi(\boldsymbol{Q}^{\mathrm{t}} \cdot \boldsymbol{b} \cdot \boldsymbol{Q})$, that the stress response functions must be functions of the *invariants* of $\boldsymbol{C}$ and $\boldsymbol{b}$, for example $\psi(I_1, I_2, I_3)$ using the set of invariants from Eq. (3). The Piola stress for hyperelastic isotropic materials then becomes

$$\boldsymbol{P} = \frac{\partial \psi(I_1, I_2, I_3)}{\partial \boldsymbol{F}} = \frac{\partial \psi}{\partial I_1}\frac{\partial I_1}{\partial \boldsymbol{F}} + \frac{\partial \psi}{\partial I_2}\frac{\partial I_2}{\partial \boldsymbol{F}} + \frac{\partial \psi}{\partial I_3}\frac{\partial I_3}{\partial \boldsymbol{F}} = 2\left[\frac{\partial \psi}{\partial I_1} + I_1\frac{\partial \psi}{\partial I_2}\right]\boldsymbol{F} - 2\frac{\partial \psi}{\partial I_2}\boldsymbol{F}\cdot\boldsymbol{F}^{\mathrm{t}}\cdot\boldsymbol{F} + 2I_3\frac{\partial \psi}{\partial I_3}\boldsymbol{F}^{-\mathrm{t}} \ . \tag{13}$$

For the case of *near incompressibility*, instead of using the invariants I_1, I_2, I_3, we can express the energy and stress as functions of the invariants $\bar{I}_1$, $\bar{I}_2$, J from Eq. (7) [3],

$$\boldsymbol{P} = \frac{\partial \psi(\bar{I}_1, \bar{I}_2, J)}{\partial \boldsymbol{F}} = 2\frac{1}{J^{2/3}}\left[\frac{\partial \psi}{\partial \bar{I}_1} + \bar{I}_1\frac{\partial \psi}{\partial \bar{I}_2}\right]\boldsymbol{F} - 2\frac{1}{J^{4/3}}\frac{\partial \psi}{\partial \bar{I}_2}\boldsymbol{F}\cdot\boldsymbol{F}^{\mathrm{t}}\cdot\boldsymbol{F} - \frac{2}{3}\left[\bar{I}_1\frac{\partial \psi}{\partial \bar{I}_1} + 2\bar{I}_2\frac{\partial \psi}{\partial \bar{I}_2}\right]\boldsymbol{F}^{-\mathrm{t}} + J\frac{\partial \psi}{\partial J}\boldsymbol{F}^{-\mathrm{t}} \ . \tag{14}$$

For our Neural Network, this implies that rather than using the six independent components of the symmetric right Cauchy Green deformation tensor $\boldsymbol{C}$ as input, we constrain the input to a set of three invariants of the right and left Cauchy Green deformation tensors $\boldsymbol{C}$ and $\boldsymbol{b}$, either I_1, I_2, I_3 or $\bar{I}_1$, $\bar{I}_2$, J. In essence, considering materials with known symmetry classes according to Eqs. (13) or (14) directly affects, and ideally reduces, the *input* of the Neural Network.

Perfect incompressibility. Fourth, we can further constrain the choice of the free energy function ψ for the special case of *perfect incompressibility* for which the Jacobian remains one, $I_3 = J^2 = 1$. The condition of perfect incompressibility implies that Eqs. (13) and (14) simplify to an expression in terms of only the first two invariants I_1 and I_2,

$$\boldsymbol{P} = \frac{\partial \psi}{\partial I_1}\frac{\partial I_1}{\partial \boldsymbol{F}} + \frac{\partial \psi}{\partial I_2}\frac{\partial I_2}{\partial \boldsymbol{F}} - p\,\boldsymbol{F} = 2\left[\frac{\partial \psi}{\partial I_1} + I_1\frac{\partial \psi}{\partial I_2}\right]\boldsymbol{F} - 2\frac{\partial \psi}{\partial I_2}\boldsymbol{F}\cdot\boldsymbol{F}^{\mathrm{t}}\cdot\boldsymbol{F} - p\,\boldsymbol{F} \ . \tag{15}$$

Here, $p = -\frac{1}{3} \boldsymbol{P} : \boldsymbol{F}$ is the hydrostatic pressure; it represents an additional unknown that we typically determine from the boundary conditions. For our Neural Network, perfect incompressibility implies that rather than using the set of three invariants of the right and left Cauchy Green deformation tensors, either I_1, I_2, I_3 or $\bar{I}_1, \bar{I}_2, J$ as input, we reduce the input to a set of only two invariants, I_1 and I_2. Considering materials with perfect incompressibility according to Eq. (15) further reduces the *input* of the Neural Network.

Physically reasonable constitutive restrictions. Fifth, in addition to systematically reducing the parameterization of the free energy ψ from the nine components of the non-symmetric deformation gradient $\boldsymbol{F}$, via the six components of the symmetric right Cauchy Green deformation tensor $\boldsymbol{C}$, to three or even two scalar-valued invariants I_1, I_2, I_3 and possibly I_1, I_2, we can restrict the functional form of the free energy ψ by including additional constitutive restrictions that are both physically reasonable and mathematically convenient [1]:

(i) The free energy ψ is *non-negative* for all deformation states,

$$\psi(\boldsymbol{F}) \geq 0 \quad \forall \quad \boldsymbol{F} . \tag{16}$$

This is a condition that we can easily satisfy by adding any constant to ψ without altering the stress function.

(ii) The free energy ψ is *zero* in the reference configuration, also known as the *growth condition*, and it a priori ensures a *stress-free reference configuration*,

$$\psi(\boldsymbol{F}) \doteq 0 \quad \text{for} \quad \boldsymbol{P}(\boldsymbol{F}) \doteq \boldsymbol{0} \quad \text{at} \quad \boldsymbol{F} = \boldsymbol{I} . \tag{17}$$

This is a condition that can become particularly challenging in the context of residual stresses [42], which we do not consider in the present manuscript.

(iii) The free energy ψ is *infinite* at the extreme states of infinite compression, $J \to 0$, and infinite expansion, $J \to \infty$,

$$\psi(\boldsymbol{F}) \to \infty \quad \text{for} \quad J \to 0 \quad \text{or} \quad J \to \infty . \tag{18}$$

In addition, it seems reasonable to require that an increase in a component of the strain should be accompanied by an increase in the corresponding component of the stress and that extreme deformations for which an eigenvalue of the strain is zero or infinite should result in infinite stresses. For our Neural Network, to facilitate a stress-free reference configuration according to Eq. (17), instead of using the invariants I_1, I_2, I_3 themselves as input, we use their deviation from the energy- and stress-free reference state, $[I_1 - 3], [I_2 - 3], [I_3 - 1]$, as input. In addition, from all possible activation functions, we select functional forms that comply with conditions (i), (ii), and (iii). As such, satisfying physical considerations according to Eqs. (16), (17), and (18) directly affects the *activation functions* of the Neural Network, especially those between the last hidden layer and the output layer.

Polyconvexity. Sixth, to guide the selection of the functional forms for the free energy function ψ, and ultimately the selection of appropriate activation functions for our Neural Network, we consider *polyconvexity* requirements [2]. From the general representation theorem we know that in its most generic form, the free energy of an isotropic material can be expressed as an infinite series of products of powers of the invariants [43], $\psi(I_1, I_2, I_3) = \sum_{j,k,l=0}^{\infty} a_{jk} [I_1 - 3]^j [I_2 - 3]^k [I_3 - 1]^l$, where a_{jkl} are material constants. Importantly, mixed products of convex functions are generally not convex, and it is easier to show that the sum of specific convex subfunction usually is [44]. This motivates a special subclass of free energy functions in which the free energy is the sum of three individual polyconvex subfunctions ψ_1, ψ_2, ψ_3, such that $\psi(\boldsymbol{F}) = \psi_1(I_1) + \psi_2(I_2) + \psi_3(I_3)$, is polyconvex by design and the stresses take the following form,

$$\boldsymbol{P} = \frac{\partial \psi}{\partial \boldsymbol{F}} = \frac{\partial \psi_1}{\partial I_1} \frac{\partial I_1}{\partial \boldsymbol{F}} + \frac{\partial \psi_2}{\partial I_2} \frac{\partial I_2}{\partial \boldsymbol{F}} + \frac{\partial \psi_3}{\partial I_3} \frac{\partial I_3}{\partial \boldsymbol{F}} . \tag{19}$$

Popular polyconvex subfunctions are the power functions, $\psi_1(I_1) = [I_1^k - 3^k]^i$ and $\psi_2(I_2) = [I_2^{3k/2} - (3\sqrt{3})^k]^i$ and $\psi_3(I_3) = [I_3 - 1]^k$, the exponential functions, $\psi_1(I_1) = \exp(\varphi_1(I_1)) - 1$ and $\psi_2(I_2) = \exp(\varphi_2(I_2)) - 1$, and the logarithmic function, $\psi_3(I_3) = I_3 - 2\ln((I_3)^{1/2}) + 4(\ln(I_3)^{1/2})^2$, for *non-negative coefficients*, $i, k \geq 1$. For our Neural Network, this implies that we can either select polyconvex activation functions from a set of algorithmically predefined activation functions [13] or custom-design our own activations functions from known polyconvex subfunctions ψ_1, ψ_2, ψ_3 [12]. In addition, polyconvexity requirements suggest that we should carefully
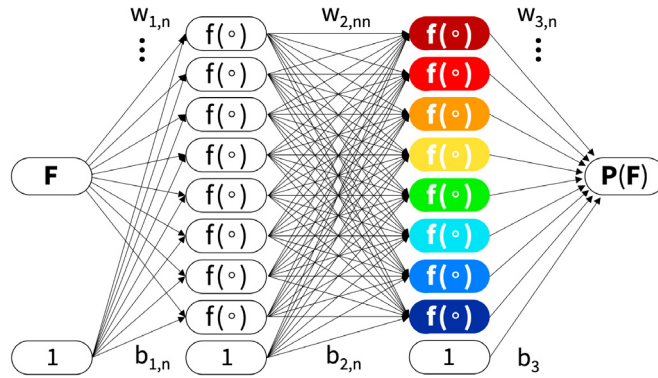
Fig. 1. Classical Neural Network. Example of a fully connected feed forward Neural Network with two hidden layers and eight nodes per layer to approximate the nine components of the tensor-valued Piola stress $\boldsymbol{P}(\boldsymbol{F})$ as a function of the nine components of the tensor-valued deformation gradient $\boldsymbol{F}$. The upper arrows originate from the network nodes and are associated with the weights $\mathbf{w}$, the lower arrows originate from the values one and are associated with the biases $\mathbf{b}$. The total number of arrows defines the number of network parameters we need to learn during the training process. The network in this example has $n_{\mathrm{w}} = 80$ weights, $n_{\mathrm{b}} = 17$ biases, and a total number of $n_\theta = 97$ parameters.

consider using a fully-connected network architecture, in which mixed products of the invariants I_1, I_2, I_3 emerge naturally. Rather, polyconvexity points towards network architectures in which the three inputs I_1, I_2, I_3 are decoupled and only combined additively when we collect the entries of last hidden layer into the free energy function, $\psi = \psi_1 + \psi_2 + \psi_3$. As such, satisfying polyconvexity, for example according to Eq. (19), generally enforces *non-negative network weights* [12] and directly affects the *architecture* and connectedness of the Neural Network [13].

4. Classical Neural Networks

Classical Neural Networks are versatile function approximators that are capable of learning any nonlinear function [45]. However, as we will see, conventional off-the-shelf Neural Networks may violate the conditions of thermodynamic consistency, material objectivity, material symmetry, incompressibility, constitutive restrictions, and polyconvexity. In this section, we briefly summarize the *input*, *output*, *architecture*, and *activation functions* of classical Neural Networks to then, in the following section, modify these four elements as we design a new family of Constitutive Artificial Neural Networks that a priori satisfy the fundamental laws of physics.

Neural Network input and output. In the constitutive modeling of hyperelastic materials, we can use Neural Networks as universal function approximators to map a second order tensor, the deformation gradient $\boldsymbol{F}$ or any other strain measure, onto another second order tensor, the Piola stress $\boldsymbol{P}$ or any other stress measure, according to Eq. (10). Fig. 1 illustrates a classical Neural Network with the nine components of the deformation gradient $\boldsymbol{F}$ as input and the nine components of the nominal or Piola stress $\boldsymbol{P}$ as output.

Neural Network architecture. The architecture of the Neural Network determines how we approximate the relation between network input and output, in our case deformation gradient $\boldsymbol{F}$ and Piola stress $\boldsymbol{P}$. The simplest architecture is a *feed forward* Neural Network in which information moves only in one direction – forward – from the input nodes, without any cycles or loops, to the output nodes. Between input and output, the information passes through one or multiple *hidden layers*. Each hidden layer consists of multiple *nodes or neurons*. In the simplest case of a *fully connected* feed forward Neural Network, all nodes of a layer receive information from all nodes of the previous layer, each multiplied by an individual *weight*, all summed up and modulated by a *bias*. Fig. 1 illustrates the example of a fully connected feed forward Neural Network with an input layer composed of the deformation gradient $\boldsymbol{F}$, two hidden layers with eight nodes per layer, and an output layer composed of the Piola stress $\boldsymbol{P}$. Let us denote the input as $\mathbf{z}_0$, the nodal values of the hidden layer k as $\mathbf{z}_k$, and the output as $\mathbf{z}_{k+1}$. For the example in Fig. 1 with two hidden layers, $k = 1, 2$, we calculate the values of each new layer from the values of the previous
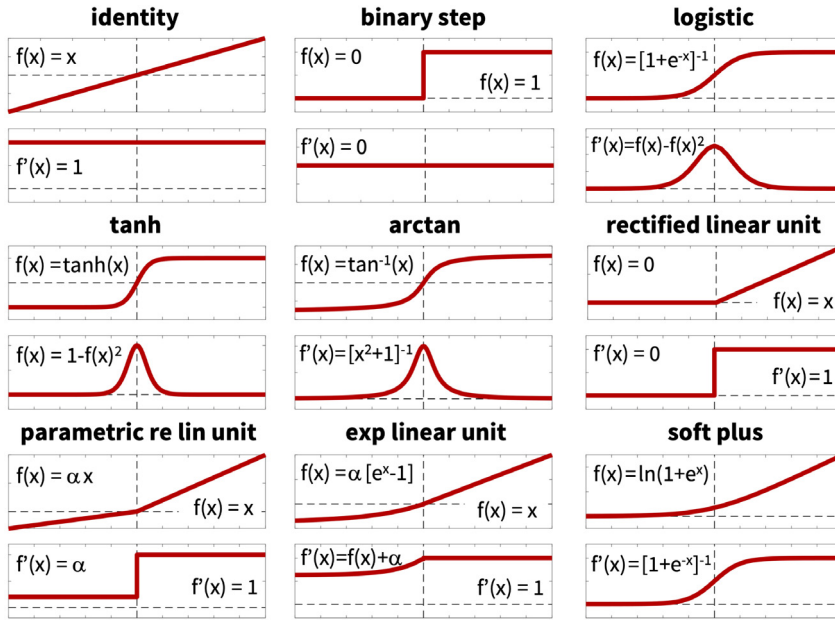
Fig. 2. Activation functions for Classical Neural Networks. Popular activation functions $f(x)$ along with their derivatives $f'(x)$ include the identity, binary step, logistic or soft step, hyperbolic tangent, inverse tangent, rectified linear unit or ReLU, parametric rectified linear unit or PReLU, exponential linear unit or ELU, and soft plus functions. Activation functions can be continuous or discontinuous, linear or nonlinear, and bounded or unbounded.

layer according to the following set of equations,

$$
\begin{aligned}
\mathbf{z}_0 &= \mathbf{F} \\
\mathbf{z}_1 &= f(\mathbf{w}_1 \mathbf{z}_0 + \mathbf{b}_1) \\
\mathbf{z}_2 &= f(\mathbf{w}_2 \mathbf{z}_1 + \mathbf{b}_2) \\
\mathbf{z}_3 &= \mathbf{w}_3 \mathbf{z}_2 + \mathbf{b}_3 \approx \mathbf{P}(\mathbf{F}).
\end{aligned}
\tag{20}
$$

Here, $\mathbf{w}$ are the set of network weights, $\mathbf{b}$ are the network biases, and $f(\circ)$ are the activation functions. In Fig. 1, the upper arrows that originate from the nodes of the previous layer and are associated with the weights $\mathbf{w}$, the lower arrows that originate from the values one and are associated with the biases $\mathbf{b}$. The total number of arrows defines the number of network parameters we need to learn during the training process. For the fully connected feed forward Neural Network in Fig. 1 with two hidden layers with eight nodes each, $\mathbf{w}_1 \in \mathbb{R}^{1 \times 8}$, $\mathbf{w}_2 \in \mathbb{R}^{8 \times 8}$, $\mathbf{w}_3 \in \mathbb{R}^{8 \times 1}$, and $\mathbf{b}_1 \in \mathbb{R}^8$, $\mathbf{b}_2 \in \mathbb{R}^8$, $\mathbf{b}_3 \in \mathbb{R}^1$, resulting in $n_w = 8 + 8 \times 8 + 8 = 80$ weights and $n_b = 8 + 8 + 1 = 17$ biases, and a total number of $n_\theta = 97$ network parameters.

Activation functions. Activation functions translate the sum of the weighted inputs to each node into an output signal that will be fed into the next layer [45]. In analogy to the brain that processes input signals and decides whether a neuron should fire or not [46], activation functions decide whether the nodal input is important or not in the process of approximating the final function, in our case the stress $\boldsymbol{P}(\boldsymbol{F})$.

Fig. 2 illustrates the nine most popular activation functions $f(x)$ in Neural Network modeling along with their derivatives $f'(x)$. Depending on the final function we want to approximate, we can select from continuous or discontinuous, linear or nonlinear, and bounded or unbounded activation functions. In classical Neural Networks, all hidden layers typically use the same activation function, whereas the final output layer often uses a different activation function. For the simple example of a feed forward fully connected Neural Network similar to Fig. 1, with one input $z_0 = F_1$, one output $z_3 = P_1$, and two hidden layers with two nodes per layer, $\mathbf{z}_1 = [z_{11}, z_{12}]$ and $\mathbf{z}_2 = [z_{21}, z_{22}]$, the system of Eqs. (20) with activation functions of hyperbolic tangent type, $f(x) = \tanh(x)$,

results in the following explicit expressions,

$$
\begin{aligned}
z_0 &= F_{11} \\
z_{11} &= \tanh (\, w_{111} \cdot F_{11} + b_{11} \,) \\
z_{12} &= \tanh (\, w_{112} \cdot F_{11} + b_{12} \,) \\
z_{21} &= \tanh (\, w_{211} \cdot z_{11} + w_{212} \cdot z_{12} + b_{21} \,) \\
z_{22} &= \tanh (\, w_{221} \cdot z_{11} + w_{222} \cdot z_{12} + b_{22} \,) \\
z_3 &= \quad w_{321} \cdot z_{21} + w_{322} \cdot z_{22} + b_{31} \\
P_{11} &\approx \quad w_{321} \cdot (\tanh(w_{211} \cdot \tanh(w_{111} \cdot F_{11} + b_{11})) + \tanh(w_{212} \cdot \tanh(w_{112} \cdot F_{11} + b_{12})) + b_{21}) \\
&+ \quad w_{322} \cdot (\tanh(w_{221} \cdot \tanh(w_{111} \cdot F_{11} + b_{11})) + \tanh(w_{222} \cdot \tanh(w_{112} \cdot F_{11} + b_{12})) + b_{22}) + b_{31} \, ,
\end{aligned}
\tag{21}
$$

where the output of the last layer z_3 approximates the true solution, $P_1 \approx z_3$. This specific Neural Network has $\mathbf{w}_1 = [\, w_{111}, w_{112} \,]$, $\mathbf{w}_2 = [\, w_{211}, w_{212}, w_{221}, w_{222} \,]$, $\mathbf{w}_3 = [\, w_{321}, w_{322} \,]$, and $\mathbf{b}_1 = [\, b_{11}, b_{12} \,]$, $\mathbf{b}_2 = [\, b_{21}, b_{22} \,]$, $\mathbf{b}_3 = [\, b_{31} \,]$, resulting in $n_{\mathrm{w}} = 2 + 2 \times 2 + 2 = 8$ weights and $n_{\mathrm{b}} = 2 + 2 + 1 = 5$ biases, and a total number of $n_\theta = 13$ network parameters. The set of Eqs. (21) illustrates that, for every hidden layer, we add one more level of nested activation functions, in this case $\tanh(\circ)$. The final approximated stress stretch relation $P_1(F_1)$ is fairly complex, inherently nonlinear, and difficult if not impossible to invert explicitly. From the set of Eqs. (21), it is clear that the network weights and biases have *no clear physical interpretation*.

The selection of appropriate activation functions depends on the type of prediction we expect from our model. In constitutive modeling, where we seek to approximate the stress $\boldsymbol{P}$ as a function of the deformation gradient $\boldsymbol{F}$, we can immediately rule out some of the common activation functions in Fig. 2 – at least for the final output layer – when considering the physically reasonable constitutive restrictions (16), (17), and (18) from Section 3: (i) the binary step function is *discontinuous* at the origin, $f(-0) \neq f(+0)$, which violates our general understanding that the energy ψ and the stress $\boldsymbol{P}$ should be smooth and continuous for all hyperelastic deformations; (ii) the binary step function and rectified linear unit are *constant* over part or all of the domain, $f(x) = 0$ or $f(x) = 1$, which violates our general understanding that the stress $\boldsymbol{P}$ should not be constant, but rather increase with increasing deformation $\boldsymbol{F}$; (iii) the binary step, logistic, hyperbolic tangent, and inverse tangent functions are *horizontally asymptotic*, $f(-\infty) = 0$ and $f(+\infty) = 1$, which violates the physically reasonable constitutive restriction (18) that the energy and stress should not be bounded, but rather become infinite, $\boldsymbol{P} \to \infty$, for extreme deformations, $\boldsymbol{F} \to \infty$; (iv) the rectified linear unit, parametric rectified linear unit, and exponential linear unit are continuous but *non-differentiable* at zero, $f'(-0) \neq f'(+0)$, which could be useful to model tension–compression asymmetry, but is not the most elegant choice to model the tension–compression transition at the origin. At the same time, the identity, $f(x) = x$, and the left branch of the exponential linear unit, $f(x) = \alpha\,[\exp(x) - 1]$, remind us of the classical linear neo Hooke [35] and exponential Holzapfel [47] models. Together with the soft plus function, $f(x) = \ln(1 + \exp(x))$, they are the only three functions that are *continuous*, *differentiable*, and *polyconvex* [13]. This motivates the question, can we identify existing activation functions or design our own set of activation functions that mimic known constitutive models, or contributions to them, and, ideally, satisfy polyconvexity requirements by design?

Loss function. The objective of a classical Neural Network is to learn the network parameters, $\boldsymbol{\theta} = \{\mathbf{w}_k, \mathbf{b}_k\}$, the network weights and biases, by minimizing a loss function L that penalizes the error between model and data. We commonly characterize this error as the mean squared error, the L_2-norm of the difference between model $\boldsymbol{P}(\boldsymbol{F}_i)$ and data $\hat{\boldsymbol{P}}_i$, divided by the number of training points n_{trn},

$$
L(\boldsymbol{\theta}; \boldsymbol{F}) = \frac{1}{n_{\mathrm{trn}}} \sum_{i=1}^{n_{\mathrm{trn}}} \| \boldsymbol{P}(\boldsymbol{F}_i) - \hat{\boldsymbol{P}}_i \|^2 \to \min .
\tag{22}
$$

We train the network by minimizing the loss function (22) and learn the network parameters, $\boldsymbol{\theta} = \{\mathbf{w}_k, \mathbf{b}_k\}$, in our case using the ADAM optimizer, a robust adaptive algorithm for gradient-based first-order optimization. With appropriate training data, classical Neural Networks can interpolate data well, without any prior knowledge of the underlying physics. However, they typically fail to extrapolate and make informed predictions [21]. Since they usually have many degrees of freedom, they are inherently at risk of overfitting, especially if the available data are sparse [48]. In addition, they may violate the thermodynamic restrictions of Section 3. This motivates the question,
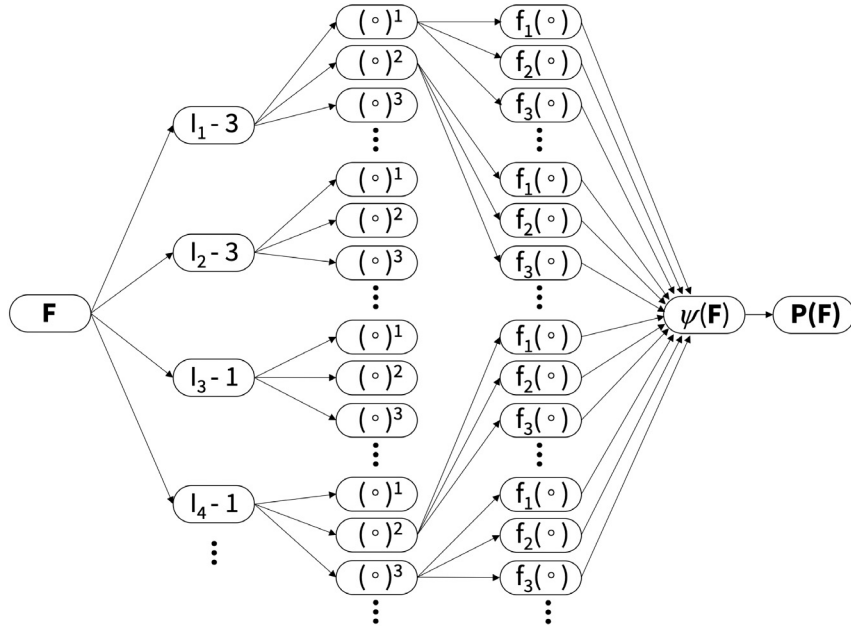
Fig. 3. Constitutive Artificial Neural Network. Family of a feed forward Constitutive Artificial Neural Networks with two hidden layers to approximate the single scalar-valued free energy function $\psi(I_1, I_2, I_3, I_4, \ldots)$ as a function of the scalar-valued invariants $I_1, I_2, I_3, I_4, \ldots$ of the deformation gradient $\boldsymbol{F}$. The first layer generates powers $(\circ)$, $(\circ)^2$, $(\circ)^3$, $\ldots$ of the network input and the second layer applies thermodynamically admissible activation functions $f(\circ)$ to these powers. Constitutive Artificial Neural Networks are typically not fully connected by design to a priori satisfy the condition of polyconvexity.

can we integrate physical information we already know to constrain the function $\boldsymbol{P}(\boldsymbol{F})$, prevent overfitting, and make the model more predictive?

5. Constitutive Artificial Neural Networks

We now propose a new family of Constitutive Artificial Neural Networks that satisfy the conditions of thermodynamic consistency, material objectivity, material symmetry, incompressibility, constitutive restrictions, and polyconvexity by design. In the following, we discuss how this guides our selection of network *input*, *output*, *architecture*, and *activation functions* to a priori satisfy the fundamental laws of physics. We also demonstrate that, for special cases, members of this family reduce to well-known constitutive models, including the neo Hooke [35], Blatz Ko [25], Mooney Rivlin [26,34], Yeoh [36], Demiray [33] and Holzapfel [47] models, and that the network weights have a clear physical interpretation.

Constitutive Artificial Neural Network input and output. To ensure thermodynamical consistency, rather than directly approximating the stress $\boldsymbol{P}$ as a function of the deformation gradient $\boldsymbol{F}$, we use the Constitutive Artificial Neural Network as a universal function approximator to map the scalar-valued invariants I_1, I_2, I_3, I_4, I_5 onto the scalar-valued free energy function ψ according to Eqs. (13). The Piola stress $\boldsymbol{P}$ then follows naturally from the second law of thermodynamics as the derivative of the free energy ψ with respect to the deformation gradient $\boldsymbol{F}$ according to Eqs. (10) and (13). Fig. 3 illustrates a Constitutive Artificial Neural Network with the invariants I_1, I_2, I_3, I_4 as input and the free energy ψ as output.

Constitutive Artificial Neural Network architecture. Since we seek to model a hyperelastic history-independent material, we select a feed forward architecture in which information only moves in one direction, from the input nodes, without any cycles or loops, to the output nodes. To control polyconvexity, rather than choosing a fully connected feed forward network, we select a network architecture in which nodes only receive an input from selected nodes of the previous layer. Specifically, according to Eq. (19), the nodes of the individual invariants are not connected, such that the free energy function does not contain mixed terms in the invariants. Fig. 3
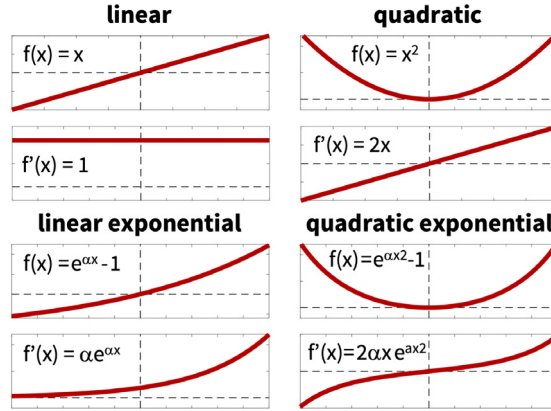
Fig. 4. Activation functions for Constitutive Artificial Neural Networks. We use custom-designed activation functions $f(x)$ along with their derivatives $f'(x)$ that include linear and quadratic mappings, either as final activation functions themselves, top rows, or combined with exponential functions, bottom rows, to reverse engineer a free energy function that captures popular functional forms of constitutive terms.

illustrates one possible architecture that attempts to a priori satisfy the polyconvexity condition (19) by decoupling the information of the individual invariants. For this particular network architecture, the free energy function that we seek to approximate takes the following format,

$$
\begin{aligned}
\psi(I_1, I_2, I_3, I_4) = {} & w_{2,1} \; f_1(w_{1,1} \, [\, I_1 - 3\,]^1) + w_{2,2} \; f_2(w_{1,2} \, [\, I_1 - 3\,]^1) + w_{2,3} \; f_3(w_{1,3} \, [\, I_1 - 3\,]^1) \\
& + w_{2,4} \; f_1(w_{1,4} \, [\, I_1 - 3\,]^2) + w_{2,5} \; f_2(w_{1,5} \, [\, I_1 - 3\,]^2) + w_{2,6} \; f_3(w_{1,6} \, [\, I_1 - 3\,]^2) \\
& + w_{2,7} \; f_1(w_{1,7} \, [\, I_1 - 3\,]^3) + w_{2,8} \; f_2(w_{1,8} \, [\, I_1 - 3\,]^3) + w_{2,9} \; f_3(w_{1,9} \, [\, I_1 - 3\,]^3) \\
& + w_{2,10} f_1(w_{1,10} [\, I_2 - 3\,]^1) + w_{2,11} f_2(w_{1,11} [\, I_2 - 3\,]^1) + w_{2,12} f_3(w_{1,12} [\, I_2 - 3\,]^1) \\
& + w_{2,13} f_1(w_{1,13} [\, I_2 - 3\,]^2) + w_{2,14} f_2(w_{1,14} [\, I_2 - 3\,]^2) + w_{2,15} f_3(w_{1,15} [\, I_2 - 3\,]^2) \\
& + w_{2,16} f_1(w_{1,16} [\, I_2 - 3\,]^3) + w_{2,17} f_2(w_{1,17} [\, I_2 - 3\,]^3) + w_{2,18} f_3(w_{1,18} [\, I_2 - 3\,]^3) \\
& + w_{2,19} f_1(w_{1,19} [\, I_3 - 1\,]^1) + w_{2,20} f_2(w_{1,20} [\, I_3 - 1\,]^1) + w_{2,21} f_3(w_{1,21} [\, I_3 - 1\,]^1) \\
& + w_{2,22} f_1(w_{1,22} [\, I_3 - 1\,]^2) + w_{2,23} f_2(w_{1,23} [\, I_3 - 1\,]^2) + w_{2,24} f_3(w_{1,24} [\, I_3 - 1\,]^2) \\
& + w_{2,25} f_1(w_{1,25} [\, I_3 - 1\,]^3) + w_{2,26} f_2(w_{1,26} [\, I_3 - 1\,]^3) + w_{2,27} f_3(w_{1,27} [\, I_3 - 1\,]^3) + \cdots.
\end{aligned}
$$

(23)

This specific network has $5 \times 3 \times 3 + 4 \times 3 \times 3 = 90$ weights for the transversely isotropic case with all five invariants I_1, I_2, I_3, I_4, I_5 and $3 \times 3 \times 3 + 3 \times 3 \times 3 = 54$ weights for the isotropic case with only three invariants I_1, I_2, I_3.

Activation functions. To ensure that our network satisfies basic physically reasonable constitutive restrictions, rather than selecting from the popular pre-defined activation functions in Fig. 2, we custom-design our own activation functions to *reverse-engineer* a free energy function that captures popular forms of constitutive terms. Specifically, we select from linear, quadratic, cubic, and higher order powers for the first layer of the network, and from linear, exponential, or logarithmic functions for the second layer.

Fig. 4 illustrates the four activation functions $f(x)$ along with their derivatives $f'(x)$ that we use throughout the remainder of this work. Notably, in contrast to the activation functions for classical Neural Networks in Fig. 2, all four functions are not only *monotonic*, $f(x + \varepsilon) \geq f(x)$ for $\varepsilon \geq 0$, such that increasing deformations result in increasing stresses, but also *continuous* at the origin, $f(-0) = f(+0)$, *continuously differentiable* and *smooth* at the origin, $f'(-0) = f'(+0)$, zero at the origin, $f(0) = 0$, to ensure an energy- and stress-free reference configuration according to Eq. (17), and *unbounded*, $f(-\infty) \to \infty$ and $f(+\infty) \to \infty$, to ensure an infinite energy and stress for extreme deformations according to Eq. (18).

Fig. 5 illustrates an example of an isotropic incompressible Constitutive Artificial Neural Network with two hidden layers and four and eight nodes. The first layer generates powers ($\circ$) and $(\circ)^2$ of the network input and the second layer applies the identity, ($\circ$), and the exponential function, $(\exp(\alpha(\circ)) - 1)$, to these powers. As such, the
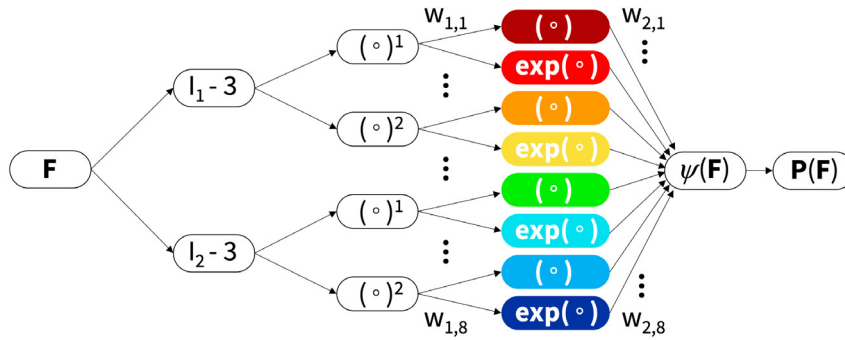
Fig. 5. Constitutive Artificial Neural Network. Example of an isotropic perfectly incompressible Constitutive Artificial Neural Network with two hidden layers to approximate the single scalar-valued free energy function $\psi(I_1, I_2)$ as a function of the first and second invariants of the deformation gradient $\boldsymbol{F}$ using eight terms. The first layer generates powers $(\circ)$ and $(\circ)^2$ of the network input and the second layer applies the identity $(\circ)$ and exponential functions $(\exp(\alpha(\circ)) - 1)$ to these powers. The networks is not fully connected by design to a priori satisfy the condition of polyconvexity.

first and fifths dark red and green inputs to the free energy in Fig. 5 correspond to the linear activation function in Fig. 4, the second and sixths red and light blue inputs correspond to the quadratic activation function, the third and sevenths orange and blue inputs correspond to the linear exponential function, and the fourth and eights yellow and dark blue inputs correspond to the quadratic exponential function. The set of equations for this networks takes the following explicit form,

$$
\begin{aligned}
\psi(I_1, I_2) = {}& w_{2,1}w_{1,1}\,[\,I_1 - 3\,] \;\; + w_{2,2}\,[\,\exp(\,w_{1,2}\,[\,I_1 - 3\,]\,) - 1\,] \\
& + w_{2,3}w_{1,3}\,[\,I_1 - 3\,]^2 + w_{2,4}\,[\,\exp(\,w_{1,4}\,[\,I_1 - 3\,]^2) - 1\,] \\
& + w_{2,5}w_{1,5}\,[\,I_2 - 3\,] \;\; + w_{2,6}\,[\,\exp(\,w_{1,6}\,[\,I_2 - 3\,]\,) - 1\,] \\
& + w_{2,7}w_{1,7}\,[\,I_2 - 3\,]^2 + w_{2,8}\,[\,\exp(\,w_{1,8}\,[\,I_2 - 3\,]^2) - 1\,].
\end{aligned}
\tag{24}
$$

For this particular format, one of the first two weights of each row becomes redundant, and we can reduce the set of network parameters to twelve, $\mathbf{w} = [\,(w_{1,1}w_{2,1}), w_{1,2}, w_{2,2}, (w_{1,3}w_{2,3}), w_{1,4}, w_{2,4}(w_{1,5}w_{2,5}), w_{1,6}, w_{2,6}, (w_{1,7}w_{2,7}), w_{1,8}, w_{2,8}\,]$. Using the second law of thermodynamics, we can derive an explicit expression for the Piola stress from Eq. (11), $\boldsymbol{P} = \partial\psi/\partial\boldsymbol{F}$, or, more specifically, for the case of perfect incompressibility from Eq. (15), $\boldsymbol{P} = \partial\psi/\partial I_1 \cdot \partial I_1/\partial\boldsymbol{F} + \partial\psi/\partial I_2 \cdot \partial I_2/\partial\boldsymbol{F}$,

$$
\begin{aligned}
\boldsymbol{P} = [& \quad\quad w_{2,1}w_{1,1} \;\; + w_{2,2}w_{1,2}\exp(\,w_{1,2}\,[\,I_1 - 3\,]\,) \\
+ 2\,[\,I_1 - 3\,] & [w_{2,3}w_{1,3} \;\; + w_{2,4}w_{1,4}\exp(\,w_{1,4}\,[\,I_1 - 3\,]^2)]\,\partial I_1/\partial\boldsymbol{F} \\
+ [& \quad\quad w_{2,5}w_{1,5} \;\; + w_{2,6}w_{1,6}\exp(\,w_{1,6}\,[\,I_2 - 3\,]\,) \\
+ 2\,[\,I_2 - 3\,] & [w_{2,7}w_{1,7} \;\; + w_{2,8}w_{1,8}\exp(\,w_{1,8}\,[\,I_2 - 3\,]^2)]\,\partial I_2/\partial\boldsymbol{F}
\end{aligned}
\tag{25}
$$

Compared to the stress stretch relation $\boldsymbol{P}(\boldsymbol{F})$ of classical Neural Networks (21), the stress stretch relation of Constitutive Artificial Neural Networks (25) is fairly simple by design. More importantly, the particular form (25) represents a *generalization* of many popular constitutive models for incompressible hyperelastic materials. It seems natural to ask whether and how our network parameters $\mathbf{w}$ relate to common well-known material parameters.

Special types of constitutive equations. To demonstrate that the family of Constitutive Artificial Neural Networks in Fig. 3 and the specific example in Fig. 5 are a *generalization* of popular constitutive models, we consider several widely used models and systematically compare their material parameters to our network weights $\mathbf{w}$:

The *neo Hooke model* [35], the simplest of all models, has a free energy function that is a constant function of only the first invariant, $[\,I_1 - 3\,]$, scaled by the shear modulus μ,

$$
\psi = \tfrac{1}{2}\mu\,[\,I_1 - 3\,] \qquad \text{where} \qquad \mu = 2\,w_{1,1}w_{2,1} \text{ in Eq. (24)}.
\tag{26}
$$

The *Blatz Ko model* [25], has a free energy function that depends only the second and third invariants, $[\,I_2 - 3\,]$ and $[\,I_3 - 1\,]$, scaled by the shear modulus μ, $\psi = \tfrac{1}{2}\mu\,[\,I_2/I_3 + 2\sqrt{I_3} - 5\,]$. For perfectly incompressible materials,

$I_3 = 1$, it simplifies to the following form,

$$\psi = \tfrac{1}{2}\,\mu\,[\,I_2 - 3\,] \qquad \text{where} \qquad \mu = 2\,w_{1,5}w_{2,5} \text{ in Eq. (24)}. \tag{27}$$

The *Mooney Rivlin model* [26,34] is a combination of both and accounts for the first and second invariants, $[\,I_1 - 3\,]$ and $[\,I_2 - 3\,]$, scaled by the moduli μ_1 and μ_2 that sum up to the overall shear modulus, $\mu = \mu_1 + \mu_2$,

$$\psi = \tfrac{1}{2}\,\mu_1\,[\,I_1 - 3\,] + \tfrac{1}{2}\,\mu_2\,[\,I_2 - 3\,] \qquad \text{where} \qquad \mu_1 = 2\,w_{1,1}w_{2,1} \text{ and } \mu_2 = 2\,w_{1,5}w_{2,5} \text{ in Eq. (24)}. \tag{28}$$

The *Yeoh model* [36] considers linear, quadratic, and cubic terms of only the first invariant, $[\,I_1 - 3\,]$, as

$$\psi = a_1\,[\,I_1 - 3\,] + a_2\,[\,I_1 - 3\,]^2 + a_3\,[\,I_1 - 3\,]^3 \text{ where } a_1 = 2\,w_{1,1}w_{2,1} \text{ and } a_2 = 2\,w_{1,3}w_{2,3} \text{ and } a_3 = 0 \text{ in Eq. (23)}. \tag{29}$$

The *Demiray model* [33] or *Delfino model* [49] uses linear exponentials of the first invariant, $[\,I_1 - 3\,]$, in terms of two parameters a and b,

$$\psi = \frac{1}{2}\frac{a}{b}\,[\,\exp(b\,[\,I_1 - 3\,]) - 1\,] \qquad \text{where} \qquad a = 2\,w_{1,2}w_{2,2} \text{ and } b = w_{1,2} \text{ in Eq. (24)}. \tag{30}$$

The *Treloar model* [35] and *Mooney Rivlin model* [26,34] for nearly incompressible materials both consider a quadratic term of the third invariant, $[\,J - 1\,]$, scaled by the bulk modulus κ, to additionally account for the bulk behavior,

$$\bar{\psi} = \tfrac{1}{2}\,\kappa\,[\,J - 3\,]^2 \qquad \text{where} \qquad \kappa = 2\,w_{1,13}w_{2,13} \text{ in Eq. (23)}. \tag{31}$$

The *Holzapfel model* [47] uses quadratic exponentials of the fourth invariant, $[\,I_4 - 1\,]$, in terms of two parameters a and b to additionally account for a transversely isotropic behavior,

$$\bar{\psi} = \frac{1}{2}\frac{a}{b}\,[\,\exp(b\,[\,I_4 - 1\,]^2) - 1\,] \qquad \text{where} \qquad a = 2\,w_{1,22}w_{2,22} \text{ and } b = w_{1,22} \text{ in Eq. (23)}. \tag{32}$$

These simple examples demonstrate that we can recover popular constitutive functions for which the network weights gain a well-defined physical meaning.

Loss function. The objective of a Constitutive Artificial Neural Network is to learn the network parameters $\boldsymbol{\theta} = \{\mathbf{w}_k\}$, the network weights, by minimizing a loss function L that penalizes the error between model and data. Similar to classical Neural Networks, we characterize this error as the mean squared error, the L_2-norm of the difference between model $\boldsymbol{P}(\boldsymbol{F}_i)$ and data $\hat{\boldsymbol{P}}_i$, divided by the number of training points n_{trn},

$$L(\boldsymbol{\theta}; \boldsymbol{F}) = \frac{1}{n_{\text{trn}}}\sum_{i=1}^{n_{\text{trn}}} \|\,\boldsymbol{P}(\boldsymbol{F}_i) - \hat{\boldsymbol{P}}_i\,\|^2 \to \min. \tag{33}$$

While this is not the focus of the present work, in the spirit of Physics Informed Neural Networks, we could add additional thermodynamic constraints to the loss function [9,10]. For the perfectly incompressible hyperelastic materials we consider here, the thermodynamics are already well represented and hardwired into the network through input, output, architecture and activation functions, and we do not need to consider this extra step. We train the network by minimizing the loss function (33) and learn the network parameters $\boldsymbol{\theta} = \{\mathbf{w}\}$ using the ADAM optimizer, a robust adaptive algorithm for gradient-based first-order optimization, and constrain the network weights to always remain non-negative, $\mathbf{w} \geq \mathbf{0}$. While we could equally well solve the optimization problem (33) using a different optimization solver, we capitalize on the power and robustness of optimizers developed for machine learning and opt for the widely used ADAM optimizer, rather than implementing this minimization ourselves.

With only small amounts of training data, Constitutive Artificial Neural Networks can both interpolate and extrapolate well and make informed predictions within the range of validity of the underlying thermodynamic assumptions. Since they limit the number of degrees of freedom, they are less likely to overfit, especially if the available data are sparse. By design, Constitutive Artificial Neural Networks are compliant with the thermodynamic restrictions of Section 3. Most importantly, for practical applications, they do not operate as a black box; rather they are a *generalization* of existing constitutive models and their parameters have a clear *physical interpretation*.
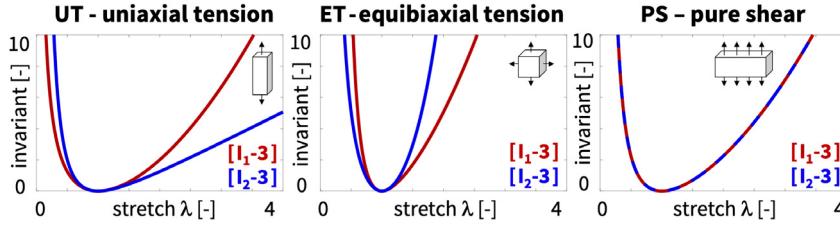
Fig. 6. Special homogeneous deformation modes. Invariant-stretch relations for the special modes of perfectly incompressible uniaxial tension with $\boldsymbol{F} = \mathrm{diag}\{\lambda, \lambda^{-1/2}, \lambda^{-1/2}\}$, equibiaxial tension with $\boldsymbol{F} = \mathrm{diag}\{\lambda, \lambda, \lambda^{-2}\}$, and pure shear with $\boldsymbol{F} = \mathrm{diag}\{\lambda, 1, \lambda^{-1}\}$.

6. Special homogeneous deformation modes

To demonstrate the features of our thermodynamically consistent Constitutive Artificial Neural Networks, we consider an *isotropic*, *perfectly incompressible* material for which the principal stretches λ_i and nominal stresses P_i are related via

$$P_i = \frac{\partial \psi}{\partial \lambda_i} - \frac{1}{\lambda_i} p \qquad \forall \qquad i = 1, 2, 3, \tag{34}$$

where p denotes the hydrostatic pressure. Using the chain rule, we can reparameterize Eq. (34) in terms of the invariants I_1 and I_2, recalling the incompressibility constraint $I_3 = 1$, such that

$$P_i = \frac{\partial \psi}{\partial I_1} \frac{\partial I_1}{\partial \lambda_i} + \frac{\partial \psi}{\partial I_2} \frac{\partial I_2}{\partial \lambda_i} - \frac{1}{\lambda_i} p \qquad \forall \qquad i = 1, 2, 3. \tag{35}$$

In the following, we summarize the deformation gradients $\boldsymbol{F}$, the invariants I_1 and I_2, their derivatives $\partial I_1/\partial \lambda$ and $\partial I_2/\partial \lambda$, and the resulting nominal stress $\boldsymbol{P}$ for the special homogeneous deformation modes of incompressible *uniaxial tension*, *equibiaxial tension*, and *pure shear* [4]. Fig. 6 summarizes the stretch-invariant relationship for all three cases.

Uniaxial tension. For the special case of incompressible uniaxial tension, we stretch the specimen in one direction, $\lambda_1 = \lambda$. From isotropy and incompressibility, $I_3 = \lambda_1^2 \lambda_2^2 \lambda_3^2 = 1$, we conclude that the stretches orthogonal to this direction are the same and equal to the square root of the stretch, $\lambda_2 = \lambda_3 = \lambda^{-1/2}$. The deformation gradient $\boldsymbol{F}$ and Piola stress $\boldsymbol{P}$ for incompressible uniaxial tension follow as

$$\boldsymbol{F} = \mathrm{diag}\{\lambda, \lambda^{-1/2}, \lambda^{-1/2}\} \quad \text{and} \quad \boldsymbol{P} = \mathrm{diag}\{P_1, 0, 0\}. \tag{36}$$

We can use the explicit expressions of the first and second invariants and their derivatives,

$$I_1 = \lambda^2 + \frac{2}{\lambda} \quad \text{and} \quad I_2 = 2\lambda + \frac{1}{\lambda^2} \quad \text{with} \quad \frac{\partial I_1}{\partial \lambda} = 2\left[\lambda - \frac{1}{\lambda^2}\right] \quad \text{and} \quad \frac{\partial I_2}{\partial \lambda} = 2\left[1 - \frac{1}{\lambda^3}\right], \tag{37}$$

to determine the pressure p from the zero stress condition in the transverse directions, $P_2 = 0$ and $P_3 = 0$, using Eq. (35),

$$p = \frac{2}{\lambda} \frac{\partial \psi}{\partial I_1} + 2\left[\lambda + \frac{1}{\lambda^2}\right] \frac{\partial \psi}{\partial I_2}, \tag{38}$$

and obtain an explicit analytical expression for the nominal stress P_1 in terms of the stretch λ from Eq. (35),

$$P_1 = 2\left[\frac{\partial \psi}{\partial I_1} + \frac{1}{\lambda} \frac{\partial \psi}{\partial I_2}\right]\left[\lambda - \frac{1}{\lambda^2}\right]. \tag{39}$$

Equibiaxial tension. For the special case of incompressible equibiaxial tension, we stretch the specimen equally in two directions, $\lambda_1 = \lambda_2 = \lambda$. From the incompressibility condition, $I_3 = \lambda_1^2 \lambda_2^2 \lambda_3^2 = 1$, we conclude that the stretch in the third direction is $\lambda_3 = \lambda^{-2}$. The deformation gradient $\boldsymbol{F}$ and Piola stress $\boldsymbol{P}$ for incompressible equibiaxial tension follow as

$$\boldsymbol{F} = \mathrm{diag}\{\lambda, \lambda, \lambda^{-2}\} \quad \text{and} \quad \boldsymbol{P} = \mathrm{diag}\{P_1, P_2, 0\}. \tag{40}$$
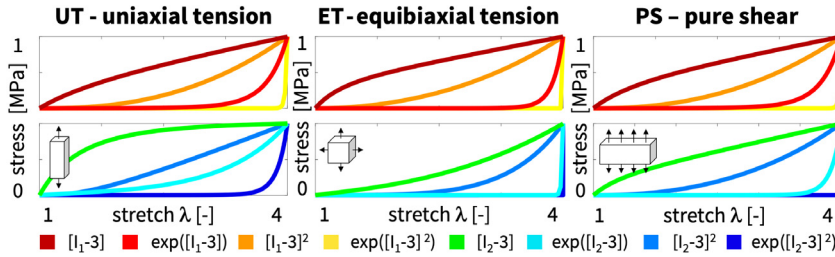
Fig. 7. Special homogeneous deformation modes. Stress–stretch relations for the example of the free energy function ψ in Eq. (25) for the special modes of perfectly incompressible uniaxial tension, equibiaxial tension, and pure shear. The eight curves highlight the linear, quadratic, linear exponential, and quadratic exponential contributions of the first and second invariants I_1 and I_2 to the final stress function $P_1(\lambda)$ in Eqs. (39), (43), and (47). The color-code agrees with the nodes of the Constitutive Artificial Neural Network in Fig. 5.

Using the explicit expressions of the first and second invariants and their derivatives,

$$I_1 = 2\lambda^2 + \frac{1}{\lambda^4} \quad \text{and} \quad I_2 = \lambda^4 + \frac{2}{\lambda^2} \quad \text{with} \quad \frac{\partial I_1}{\partial \lambda} = 2\left[\lambda - \frac{1}{\lambda^5}\right] \quad \text{and} \quad \frac{\partial I_2}{\partial \lambda} = 2\left[\lambda^3 - \frac{1}{\lambda^3}\right], \tag{41}$$

we determine the pressure p from the zero stress condition in the third direction, $P_3 = 0$, using Eq. (35),

$$p = \frac{2}{\lambda^4}\frac{\partial \psi}{\partial I_1} + \frac{4}{\lambda^2}\frac{\partial \psi}{\partial I_2} \tag{42}$$

and obtain an explicit analytical expression for the nominal stresses P_1 and P_2 in terms of the stretch λ from Eq. (35),

$$P_1 = P_2 = 2\left[\frac{\partial \psi}{\partial I_1} + \lambda^2\frac{\partial \psi}{\partial I_2}\right]\left[\lambda - \frac{1}{\lambda^5}\right]. \tag{43}$$

Pure shear. For the special case of incompressible pure shear, we stretch a long rectangular specimen along its short axis, $\lambda_1 = \lambda$, and assume that it remains undeformed along its long axis, $\lambda_2 = 1$. From the incompressibility condition, $I_3 = \lambda_1^2\lambda_2^2\lambda_3^2 = 1$, we conclude that the stretch in the third direction is $\lambda_3 = \lambda^{-1}$. The deformation gradient $\boldsymbol{F}$ and Piola stress $\boldsymbol{P}$ for incompressible pure shear are

$$\boldsymbol{F} = \text{diag}\{\lambda, 1, \lambda^{-1}\} \quad \text{and} \quad \boldsymbol{P} = \text{diag}\{P_1, P_2, 0\}. \tag{44}$$

Using the explicit expressions of the first and second invariants and their derivatives,

$$I_1 = I_2 = \lambda^2 + 1 + \frac{1}{\lambda^2} \quad \text{with} \quad \frac{\partial I_1}{\partial \lambda} = \frac{\partial I_2}{\partial \lambda} = 2\left[\lambda - \frac{1}{\lambda^3}\right] \tag{45}$$

we determine the pressure p from the zero stress condition in the third direction, $P_3 = 0$, using Eq. (35),

$$p = \frac{2}{\lambda^2}\frac{\partial \psi}{\partial I_1} + 2\left[1 + \frac{1}{\lambda^2}\right]\frac{\partial \psi}{\partial I_2}, \tag{46}$$

and obtain explicit analytical expressions for the nominal stresses P_1 and P_2 in terms of the stretch λ from Eq. (35),

$$P_1 = 2\left[\frac{\partial \psi}{\partial I_1} + \frac{\partial \psi}{\partial I_2}\right]\left[\lambda - \frac{1}{\lambda^3}\right] \quad \text{and} \quad P_2 = 2\left[\frac{\partial \psi}{\partial I_1} + \lambda^2\frac{\partial \psi}{\partial I_2}\right]\left[1 - \frac{1}{\lambda^2}\right]. \tag{47}$$

Fig. 7 illustrates the stress–stretch relations for the example of the free energy function $\psi(\lambda)$ in Eq. (24) for the special homogeneous deformation modes of perfectly incompressible uniaxial tension, equibiaxial tension, and pure shear. The eight curves highlight the linear, quadratic, linear exponential, and quadratic exponential contributions of the first invariant I_1, top row, and second invariant I_2, bottom row, to the final stress function $P_1(\lambda)$ in Eqs. (39), (43), and (47). For comparison, all curves are scaled to unity. Their color code corresponds to the eight nodes of the Constitutive Artificial Neural Network in Fig. 5. The stress contributions of the first invariant take a comparable shape for all three deformation modes: The linear term, $[I_1 - 3]$, is concave for all three modes, whereas the other three terms are convex. The terms of the second invariant behave similarly under uniaxial tension and pure shear:

Table 1

Benchmark stress-stretch data for single-mode training. Uniaxial tension (UT) experiments for rubber at 20° and 50° [27], gum stock and tread stock [26], and polymeric foam and rubber [25]. All reported stresses are converted from their initial units [kg/cm^2] [27], [kg/2.5 · 3.2mm^2] [26], and [Psi] [25] into the unified unit [MPa].

UT rubber 20° Treloar [27]		UT rubber 50° Treloar [27]		UT gum stock Mooney [26]		UT tread stock Mooney [26]		UT foam Blatz Ko [25]		UT rubber Blatz Ko [25]	
λ [-]	P [MPa]	λ [-]	P [MPa]	λ [-]	P [MPa]	λ [-]	P [MPa]	λ [-]	P [MPa]	λ [-]	P [MPa]
1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
1.01	0.00	1.11	0.17	1.46	0.31	1.16	0.31	1.05	0.04	1.05	0.03
1.13	0.14	1.23	0.29	2.30	0.61	1.50	0.61	1.10	0.06	1.10	0.07
1.23	0.24	1.57	0.54	4.66	1.23	2.56	1.23	1.15	0.07	1.16	0.10
1.41	0.33	2.12	0.80	6.45	1.84	3.30	1.84	1.20	0.09	1.22	0.13
1.61	0.43	2.73	1.03	6.77	2.45	3.53	2.45	1.30	0.12	1.27	0.16
1.89	0.52	3.36	1.30	6.96	3.06	3.63	3.06	1.40	0.14	1.31	0.18
2.17	0.59	3.95	1.57			3.71	3.68	1.50	0.16	1.37	0.20
2.45	0.68	4.39	1.79					1.60	0.16	1.41	0.22
3.06	0.87	5.29	2.29					1.70	0.17	1.47	0.24
3.62	1.06	6.11	2.80					1.80	0.18	1.52	0.26
4.06	1.24	6.54	3.75					1.90	0.19	1.57	0.27
4.82	1.60	6.95	5.27					2.00	0.20	1.62	0.29
5.41	1.95	7.43	7.73					2.10	0.20		
5.79	2.30	7.76	10.21					2.20	0.21		
6.23	2.68							2.30	0.21		
6.46	3.03							2.34	0.21		
6.67	3.40										
6.96	3.78										
7.14	4.16										
7.25	4.49										
7.36	4.86										
7.49	5.24										
7.60	5.60										
7.69	6.33										

The linear term, $[I_2 - 3]$, is concave and the other three terms are convex. For equibiaxial tension, however, all four terms, including the $[I_2 - 3]$ term, are convex. Notably, both quadratic exponential terms increase rapidly for all six cases. In the following section, when we train our Constitutive Artificial Neural Network with real data, we will explore how linear combinations of these eight terms, scaled by the learnt twelve network weights **w**, make up the free energy function $\psi(\lambda)$, and with it the stress $P(\lambda)$ that best approximates the data $\hat{P}$.

7. Results

To demonstrate the performance of our new family of Constitutive Artificial Neural Networks, we perform a systematic side-by-side comparison of classical Neural Networks and Constitutive Neural Networks using widely-used benchmark data for rubber elasticity. Specifically, we train and compare the fully connected two-layer eight-term Neural Network from Fig. 1 and the two-layer eight-term Constitutive Artificial Neural Networks for isotropic perfectly incompressible materials from Fig. 5. We consider two training scenarios, *single-mode training* and *multi-mode training*, for the special deformation modes of uniaxial tension, biaxial tension, and pure shear.

Table 1 summarizes our benchmark data for *single-mode training* from uniaxial tension experiments for rubber at 20° and 50° [27], for gum stock and tread stock [26], and for polymeric foam and rubber [25]. For comparison, we converted all reported stresses from their initial units [kg/cm^2] [27], [kg/2.5 · 3.2 mm^2] [26], and [Psi] [25] into the unified unit [MPa]. Table 2 summarizes our benchmark data for *multi-mode training* from uniaxial tension, equibiaxial tension, and pure shear experiments for rubber at 20° and 50° [27]. For comparison, we multiplied the equibiaxial stresses by their stretches and converted all reported stresses from their initial unit [kg/cm^2] into the unified unit [MPa].

Table 2

Benchmark stress-stretch data for multi-mode training. Uniaxial tension (UT), equibiaxial tension (ET), and pure shear (PS) experiments for rubber at 20° and 50° [27]. Equibiaxial stresses are multiplied by their stretches and all stresses are converted from their initial unit [kg/cm^2] into the unified unit [MPa].

UT rubber 20° Treloar [27]		ET rubber 20° Treloar [27]		PS rubber 20° Treloar [27]		UT rubber 50° Treloar [27]		ET rubber 50° Treloar [27]		PS rubber 50° Treloar [27]	
λ [-]	P [MPa]	λ [-]	P [MPa]	λ [-]	P [MPa]	λ [-]	P [MPa]	λ [-]	P [MPa]	λ [-]	P [MPa]
1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
1.01	0.00	1.04	0.09	1.05	0.06	1.11	0.17	1.02	0.15	1.04	0.17
1.13	0.14	1.08	0.16	1.13	0.16	1.23	0.29	1.08	0.30	1.23	0.40
1.23	0.24	1.12	0.24	1.20	0.24	1.57	0.54	1.16	0.48	1.48	0.63
1.41	0.33	1.15	0.26	1.33	0.33	2.12	0.80	1.37	0.74	2.52	1.03
1.61	0.43	1.21	0.33	1.45	0.42	2.73	1.03	1.57	0.92	3.51	1.49
1.89	0.52	1.32	0.44	1.86	0.59	3.36	1.30	1.96	1.17	4.33	1.90
2.17	0.59	1.43	0.51	2.40	0.77	3.95	1.57	2.46	1.49	5.07	2.36
2.45	0.68	1.70	0.66	2.99	0.95	4.39	1.79	2.79	1.78	5.74	2.74
3.06	0.87	1.95	0.77	3.50	1.13	5.29	2.29	3.14	2.04	6.24	3.22
3.62	1.06	2.50	0.97	3.98	1.29	6.11	2.80	3.45	2.33	6.36	3.63
4.06	1.24	3.04	1.26	4.39	1.48	6.54	3.75	3.60	2.53	6.65	4.49
4.82	1.60	3.44	1.47	4.72	1.65	6.95	5.27	3.86	2.96	6.91	5.34
5.41	1.95	3.76	1.73	4.99	1.82	7.43	7.73	4.11	3.24	7.06	6.23
5.79	2.30	4.03	1.97			7.76	10.21	4.60	4.24	7.26	7.00
6.23	2.68	4.26	2.23					5.06	6.15	7.42	7.89
6.46	3.03	4.45	2.45					5.28	6.99	7.56	9.18
6.67	3.40							5.42	8.18	7.83	10.90
6.96	3.78							5.59	9.87		
7.14	4.16							5.67	11.59		
7.25	4.49										
7.36	4.86										
7.49	5.24										
7.60	5.60										
7.69	6.33										

**Classical Neural Networks can describe data well but cannot predict beyond the training regime.** Fig. 8 illustrates the effect of network depth and breadth for six classical fully connected feed forward Neural Networks with one and two layers and two, four, and eight nodes. The number of network weights and biases increases with increasing number of layers and nodes: The simplest model with one hidden layer and two nodes has $n_w = 2+2 = 4$ weights and $n_b = 2+1 = 3$ biases and a total number of $n_\theta = 7$ network parameters; the most complex model with two hidden layers and eight nodes has $n_w = 8+8 \times 8+8 = 80$ weights and $n_b = 8+8+1 = 17$ biases and a total number of $n_\theta = 97$ network parameters. For this example, for all nodes, we use the hyperbolic tangent activation function according to Fig. 2. Specifically, the network with two layers and two nodes uses the set of Eqs. (21). The networks learn the approximation of the Piola stress $P(\lambda)$ as a function of the stretch λ using the uniaxial tension data $\hat{P}$ for rubber at 20° [27] from Tables 1 and 2. The dots illustrate the training data $\hat{P}$ and the color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$. First and foremost, all six networks robustly approximate the stress $P(\lambda)$ as a function of the stretch λ with virtually no error compared to the dots of the experimental data $\hat{P}$. In general, the cost of training a Neural Network increases with the number of nodes per layer and with the number of layers. Similar to a mesh refinement in a finite element analysis, in the spirit of h-adaptivity, we expect the approximation to improve with increasing network breadth and depth. The dots in Fig. 8 indicate that the behavior of rubber under uniaxial tension is nonlinear, but monotonic and fairly smooth [27]. As a result, all six networks perform exceptionally well at *describing* or *interpolating* the data within the training regime of $1 \leq \lambda \leq 8$, even the simplest network with only one layer and two nodes. However, all six networks do a poor job at *predicting* or *extrapolating* the behavior outside the training regime for $\lambda > 8$.

Classical Neural Networks perform well for big data but tend to overfit sparse data. Fig. 9 illustrates the performance of classical Neural Networks for different uniaxial tension data. For this example, we use a fully
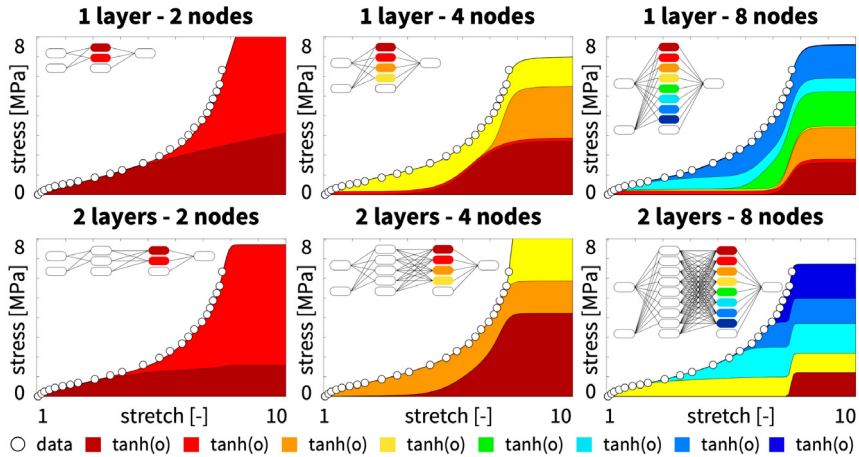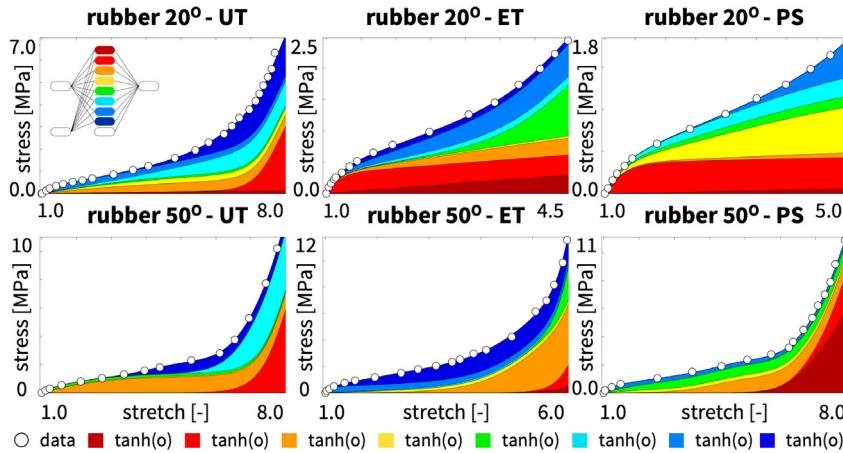
Fig. 8. Classical Neural Networks. Effect of network depth and breadth. Piola stress $P(\lambda)$ as a function of the stretch λ for six fully connected feed forward Neural Networks with one and two layers and two, four, and eight nodes using the hyperbolic tangent activation function from Fig. 2. Dots illustrate the uniaxial tension data $\hat{P}$ for rubber at 20° [27] from Tables 1 and 2; color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$.
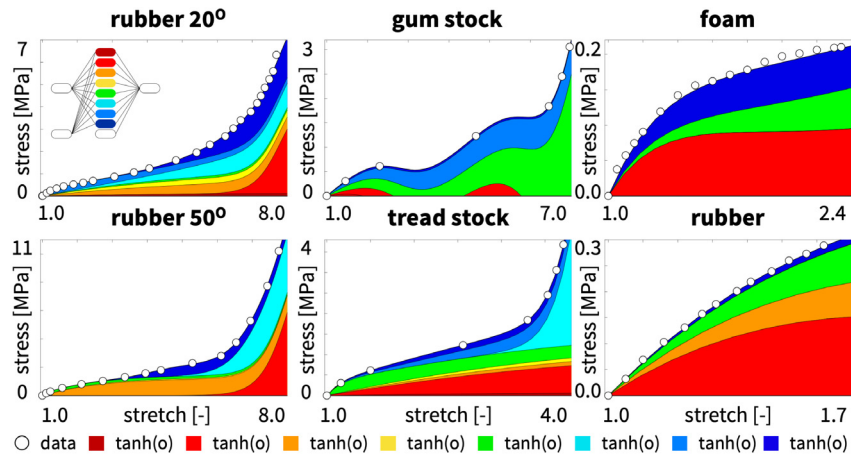
Fig. 9. Classical Neural Network. Uniaxial tension. Piola stress $P(\lambda)$ as a function of the stretch λ for a fully connected feed forward Neural Network with one layer, eight nodes, 16 weights, and nine biases, using the hyperbolic tangent activation function from Fig. 2. Dots illustrate the uniaxial tension data $\hat{P}$ for rubber at 20° and 50° [27], gum stock and tread stock [26], and polymeric foam and rubber [25] from Table 1; color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$.

connected feed forward Neural Network with one layer, eight nodes, 16 weights, nine biases, and the hyperbolic tangent activation function from Fig. 2 for all nodes. The network learns the approximation of the Piola stress $P(\lambda)$ as a function of the stretch λ using the uniaxial tension data $\hat{P}$ for rubber at 20° and 50° [27], gum stock and tread stock [26], and polymeric foam and rubber [25] from Table 1. The dots illustrate the training data $\hat{P}$ and the color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$. In general, our observations agree with Fig. 8 and suggest that classical Neural Networks robustly interpolate uniaxial tension data for rubber for all six experiments. However, for the example of gum stock with only seven data points and $n_\theta = 25$ network parameters, we observe oscillations in the approximated stress function $P(\lambda)$ in the center region between $2.4 \leq \lambda \leq 6.4$, where we only have one data point. These oscillations are a result of *negative weights* in the final output layer that make the approximated function *non-convex*. While this single example is by no means a rigorous mathematical proof, it supports the general notion that classical Neural Networks *fit big data well* but *tend to overfit sparse data*.

Fig. 10. Classical Neural Network. Uniaxial tension, equibiaxial tension, and pure shear. Piola stress $P(\lambda)$ as a function of the stretch λ for a fully connected feed forward Neural Network with one layer, eight nodes, 16 weights, and nine biases, using the hyperbolic tangent activation function from Fig. 2. Dots illustrate the uniaxial tension, equibiaxial tension, and pure shear data $\hat{P}$ for rubber at 20° and 50° [27] from Table 2; color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$ for individual single-mode training.

Classical Neural Networks perform well for multi-mode data but provide no physical insight. Fig. 10 illustrates the performance of classical Neural Networks for multi-mode data, trained *individually* for each mode. Similar to the previous example, we use a fully connected feed forward Neural Network with one layer, eight nodes, 16 weights, nine biases, and the hyperbolic tangent activation function from Fig. 2 for all nodes. The network learns the approximation of the Piola stress $P(\lambda)$ as a function of the stretch λ and trains *individually* on the uniaxial tension, equibiaxial tension, and pure shear data for rubber at 20° and 50° [27] from Table 2. The dots illustrate the training data $\hat{P}$ and the color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$. The network performs robustly on all six training sets and generates stress approximations $P(\lambda)$ that fit the stress–stretch data well, even for the S-shaped curves and in the presence of pronounced stretch stiffening. For all six cases, the loss function rapidly decreases by four orders of magnitude within less than 20,000 epochs and the error between model $P(\lambda)$ and data $\hat{P}$ is virtually invisible from the graphs. The full color spectrum in each graph suggests that all eight nodes contribute to the final stress approximation and that all weights between the last hidden layer and the output layer are non-zero. We conclude that we can robustly learn the $n_\theta = 25$ network weights and biases from multi-modal training data; yet, these parameters have no physical meaning and do not contribute to *interpreting* or *explaining* the physics of rubber under uniaxial tension, equibiaxial tension, or pure shear.

Constitutive Artificial Neural Networks describe and predict well and prevent overfitting. Fig. 11 demonstrates the performance of our new class of Constitutive Artificial Neural Networks for different uniaxial tension data. For this example, we use the feed forward Constitutive Artificial Neural Network from Fig. 5 with two layers, eight nodes, and twelve weights using the custom-designed activation functions from Fig. 4. The network learns the approximation of the free energy as a function of the invariants $\psi(I_1, I_2)$, where pre-processing generates the invariants as functions of the stretch $I_1(\lambda)$, $I_2(\lambda)$, and post-processing generates the stress as a function of the free energy $P(\psi)$. The network trains on the uniaxial tension data $\hat{P}$ for rubber at 20° and 50° [27], gum stock and tread stock [26], and polymeric foam and rubber [25] from Table 1. The dots illustrate the training data $\hat{P}$ and the color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$. First and foremost, similar to the classical Neural Network in Fig. 9, the new Constitutive Artificial Neural Network in Fig. 11 performs robustly on all six training sets and learns stress functions $P(\lambda)$ that approximate the stress–stretch data well, even for S-shaped curves and in the presence of pronounced stretch stiffening. For all six cases, the loss function rapidly decreases by four orders of magnitude within less than 10,000 epochs and the error between model $P(\lambda)$ and data $\hat{P}$ is virtually invisible from the graphs. In contrast to the Neural Network example in Fig. 8 where
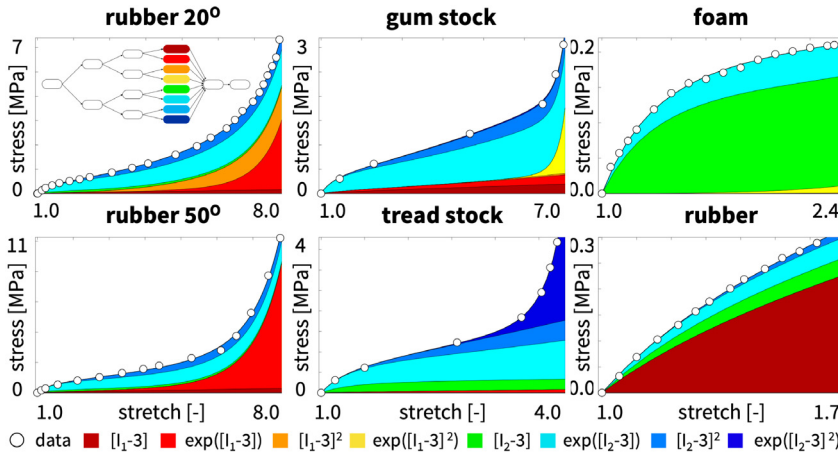
Fig. 11. Constitutive Artificial Neural Network. Uniaxial tension. Piola stress $P(\lambda)$ as a function of the stretch λ for the feed forward Constitutive Artificial Neural Network from Fig. 5 with two layers, eight nodes, and twelve weights using the custom-designed activation functions from Fig. 4. Dots illustrate the uniaxial tension data $\hat{P}$ for rubber at 20° and 50° [27], gum stock and tread stock [26], and polymeric foam and rubber [25] from Table 1; color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$.
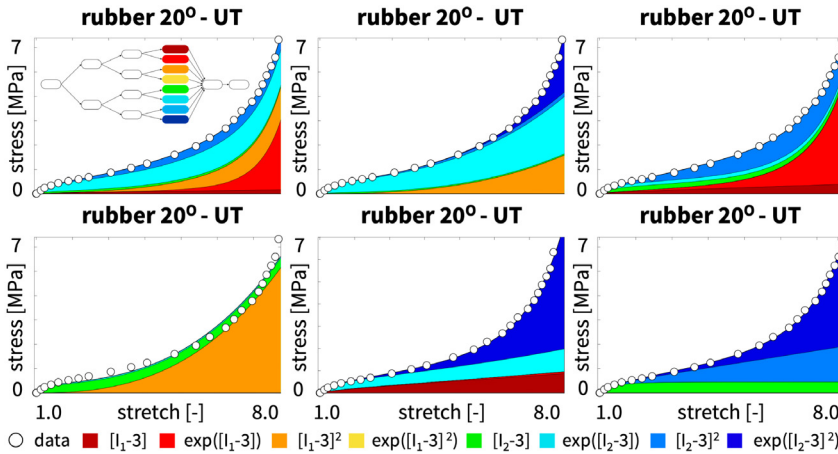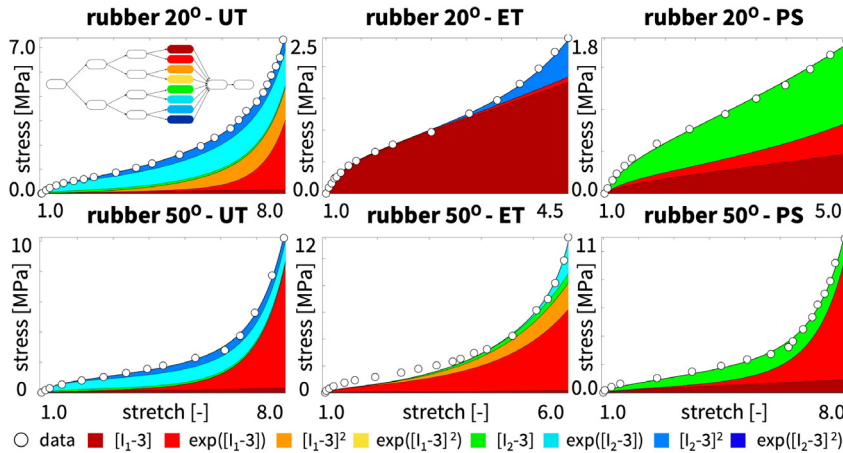

Fig. 12. Constitutive Artificial Neural Network. Effect of initial conditions and non-uniqueness. Six Piola stresses $P(\lambda)$ as functions of the stretch λ for the feed forward Constitutive Artificial Neural Network from Fig. 5 with two layers, eight nodes, and 12 weights, initialized with six different sets of initial conditions. Dots illustrate the uniaxial tension data $\hat{P}$ for rubber at 20° [27] from Table 1; color-coded areas highlight the contributions of the color-coded nodes to the stress functions $P(\lambda)$ for six different sets of initial conditions.

the learned stresses flatline abruptly outside the training regime, all six stress approximations continue smoothly beyond the initial training regime. In contrast to the gum stock example with only seven data points in Fig. 9, the Constitutive Artificial Neural Network generates smooth non-oscillatory stresses $P(\lambda)$, even in regions with sparse data. These observations suggest that our new Constitutive Artificial Neural Networks succeed at *describing*, *predicting*, and *preventing overfitting*, even in regions where data are sparse.

Constitutive Artificial Neural Networks generate non-unique solutions for insufficiently rich data. Fig. 12 illustrates the effect of the initial conditions for the feed forward Constitutive Artificial Neural Network from Fig. 5 with two layers, eight nodes, and twelve weights, and the custom-designed activation functions from Fig. 4. For this example, we initialize the twelve network weights with six different sets of randomly generated numbers and compare their contributions to the final stress approximation $P(\lambda)$ as an indicator for the magnitude
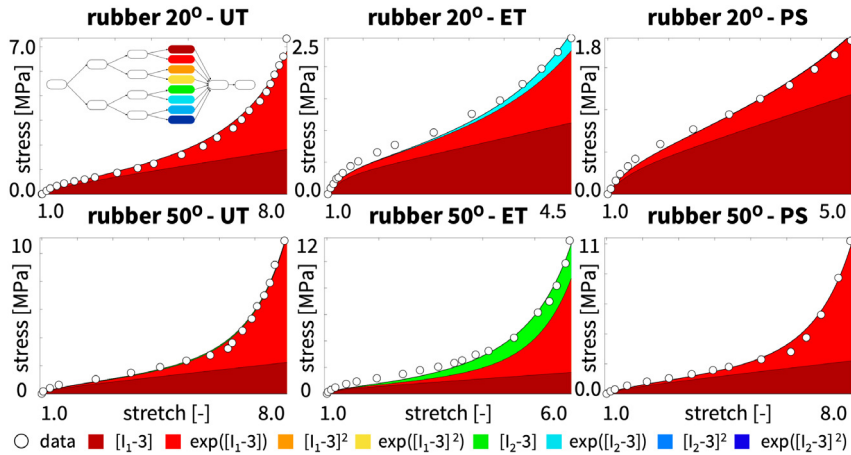
Fig. 13. Constitutive Artificial Neural Network. Uniaxial tension, equibiaxial tension, and pure shear. Piola stress $P(\lambda)$ as a function of the stretch λ for the feed forward Constitutive Artificial Neural Network from Fig. 5 with two layers, eight nodes, and twelve weights using the custom-designed activation functions from Fig. 4. Dots illustrate the uniaxial tension, equibiaxial tension, and pure shear data $\hat{P}$ for rubber at 20° and 50° [27] from Table 2; color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$ for individual single-mode training.

of the learned weights. The dots indicate the uniaxial tension data $\hat{P}$ for rubber at 20° [27] from Table 1, and the color-coded areas highlight the contributions of the color-coded nodes to the stress functions $P(\lambda)$ for the six different sets of initial conditions. First and most importantly, within less than 10,000 epochs, all six sets of initial conditions robustly converge towards a set of weights that reduce the loss function by more than four orders of magnitude and interpolate the training equally data well. Interestingly, in contrast to the classical Neural Network graphs in Figs. 9 and 10, none of the six graphs in Fig. 12 covers the full color spectrum. This suggests that only a subset of the eight nodes of the last hidden layer contribute to the final stress approximation, while most of the weights between the last hidden layer and the output layer train to zero. For example, the fourth graph approximates the stress exclusively in terms of the third and fifth terms, $[I_1 - 3]^2$ and $[I_2 - 3]$, whereas the fifth graph uses the first, sixths, and eights terms, $[I_1 - 3]$, $[\exp([I_2 - 3]) - 1]$, and, $[\exp([I_2 - 3]^2) - 1]$. From comparing the curves and the colored stress contributions in all six graphs, we conclude that the selection of weights that best approximate the stress–stretch relation is *non-unique*. While this is also true and well-known for classical Neural Networks, it is unfortunate for Constitutive Artificial Neural Networks since we attempt to correlate the network weights to constitutive parameters with a clear physical interpretation. It seems natural to ask whether this non-uniqueness is an inherent property of the Constitutive Artificial Neural Network itself or rather a result of insufficiently rich training data.

Constitutive Artificial Neural Networks are a natural generalization of existing constitutive models. Fig. 13 illustrates the performance of Constitutive Artificial Neural Networks for multi-mode data, trained *individually* for each mode. Similar to the previous two examples, we use the feed forward Constitutive Artificial Neural Network from Fig. 5 with two layers, eight nodes, and twelve weights using the custom-designed activation functions from Fig. 4. The network learns the approximation of the free energy as a function of the invariants $\psi(I_1, I_2)$ and trains *individually* on the uniaxial tension, equibiaxial tension, and pure shear data for rubber at 20° and 50° [27] from Table 2. Similar to the classical Neural Network in Fig. 10, the Constitutive Artificial Neural Network in Fig. 13 performs robustly on all six training sets and generates stress functions $P(\lambda)$ that approximate the stress–stretch data $\hat{P}$ well, even for the S-shaped curves and in the presence of pronounced stretch stiffening. Similar to the previous example, none of the six graphs in Fig. 13 covers the full color spectrum and most of the weights between the last hidden layer and the output layer train to zero. Interestingly, some of the non-zero terms correlate well with the widely-used constitutive models for rubber elasticity: The dominant dark red $[I_1 - 3]$ term for the 20° equibiaxial tension data correlates well with the classical *neo Hooke model* [35] in Eq. (26), the dominant green $[I_1 - 2]$ term for the 20° pure shear data correlates well with the *Blatz Ko model* [25] in Eq. (27),

Fig. 14. Constitutive Artificial Neural Network. Uniaxial tension, equibiaxial tension, and pure shear. Piola stress $P(\lambda)$ as a function of the stretch λ for the feed forward Constitutive Artificial Neural Network from Fig. 5 with two layers, eight nodes, and twelve weights using the custom-designed activation functions from Fig. 4. Dots illustrate the uniaxial tension, equibiaxial tension, and pure shear data $\hat{P}$ for rubber at 20° and 50° [27] from Table 2; color-coded areas highlight the contributions of the color-coded nodes to the final stress function $P(\lambda)$ for simultaneous multi-mode training.

the interacting $[\,I_1-1\,]$ and $[\,I_1-2\,]$ terms for the 20° and 50° pure shear data correlate well with the *Mooney Rivlin model* [26,34] in Eq. (28), and the dominant $[\exp([I_1-3])]$ term for the 50° uniaxial and equibiaxial tension data correlates well with the *Demiray model* [33] in Eq. (30). This suggests that Constitutive Artificial Neural Networks are a *generalization* of existing constitutive models that naturally self-select terms from subsets of well-known constitutive models that best explain the data.

Constitutive Artificial Neural Networks identify a single unique model and parameter set for sufficient data. Fig. 14 illustrates the performance of Constitutive Artificial Neural Networks for multi-mode data, trained *simultaneously* for all three modes. Similar to the previous examples, we use the feed forward Constitutive Artificial Neural Network from Fig. 5 with two layers, eight nodes, and twelve weights using the custom-designed activation functions from Fig. 4. The network learns the approximation of the free energy as a function of the invariants $\psi(I_1, I_2)$ and trains *simultaneously* on the uniaxial tension, equibiaxial tension, and pure shear data for rubber at 20° and 50° [27] from Table 2. Overall, the network trains robustly and uniquely for multi-mode data, both for the 20° and the 50° training sets. It is insensitive to the initial conditions and repeatedly converges towards the same set of weights to reduce the loss function by more than four orders of magnitude within less than 10,000 epochs. Similar to the other Constitutive Artificial Neural Network examples, and in contrast to the classical Neural Network, the final approximation uses only a subset of non-zero weights, while most of the weights are zero. Compared to the individual single-mode training in Fig. 13, the simultaneous multi-mode training in Fig. 14 seeks to approximate all three deformation modes simultaneously at the cost of a perfect fit: While the stress approximation $P(\lambda)$ slightly underestimates the training stress $\hat{P}$ in equibiaxial tension, it slightly overestimates the training stress $\hat{P}$ in the stiffening region in uniaxial tension and pure shear. Most importantly though, the Constitutive Artificial Neural Network robustly identifies one *unique model and parameter set* for rubber at 20° and one set for rubber at 50°. For the low-temperature regime, the free energy reduces to a three-term function in terms of the first invariant and the linear exponentials of the first and second invariants,

$$\psi(I_1, I_2) = \frac{1}{2}\mu_1\,[\,I_1 - 3\,] + \frac{1}{2}\frac{a_1}{b_1}\,[\,\exp(b_1[\,I_1 - 3\,]) - 1\,] + \frac{1}{2}\frac{a_2}{b_2}\,[\,\exp(b_2[\,I_2 - 3\,]) - 1\,]\,. \tag{48}$$

It introduces five network weights that translate into five physically meaningful parameters with well-defined physical units, the shear modulus, $\mu_1 = 2\,w_{1,1}w_{2,1} = 0.2370\,\mathrm{MPa}$, the stiffness-like parameters $a_1 = 2\,w_{1,2}w_{2,2} = 0.0582\,\mathrm{MPa}$ and $a_2 = 2\,w_{1,6}w_{2,6} = 0.0013\,\mathrm{MPa}$, and the unit-less exponential coefficients $b_1 = w_{1,2} = 0.0387$ and $b_2 = w_{1,6} = 0.0022$. For the high-temperature regime, the free energy reduces to a three-term function in terms of

the first and second invariants and the linear exponential of the first invariant,

$$\psi(I_1, I_2) = \frac{1}{2} \mu_1 [I_1 - 3] + \frac{1}{2} \mu_2 [I_1 - 3] + \frac{1}{2} \frac{a_1}{b_1} [\exp(b_1[I_1 - 3]) - 1]. \tag{49}$$

It introduces four network weights that translate into four physically meaningful parameters with well-defined physical units, the shear moduli, $\mu_1 = 2 w_{1,1} w_{2,1} = 0.2830$ MPa and $\mu_2 = 2 w_{1,5} w_{2,5} = 0.0141$ MPa, the stiffness-like parameter $a_1 = 2 w_{1,2} w_{2,2} = 0.0434$ MPa, and the unit-less exponential coefficient $b_1 = w_{1,2} = 0.0541$. This example suggests that the non-uniqueness of the fit in Fig. 12 is not an inherent problem of Constitutive Artificial Neural Networks per se, but rather a problem of insufficiently rich data to appropriately train the network. With multi-mode data from uniaxial tension, biaxial tension, and pure shear, our Constitutive Artificial Neural Network *trains robustly* and *uniquely and simultaneously learns both model and parameters*. Interestingly, the training autonomously selects a subset of weights that activate the relevant terms to the free energy function, while the remaining weights train to zero. This suggests that Constitutive Artificial Neural Networks are capable of identifying a free energy function and its material parameters – out of a broad spectrum of functions and parameters – to best explain the data.

8. Discussion

Constitutive Artificial Neural Networks simultaneously learn both model and parameters. For decades, chemical, physical, and material scientists alike have been modeling the hyperelastic response of rubber under large deformations [4,25–27,50]. They have proposed numerous competing constitutive models to best characterize the behavior of artificial and biological polymers and calibrated the model parameters in response to different modes of mechanical loading [33,38,47,51–56]. Here we propose a radically different approach towards constitutive modeling and abandon the common strategy to *first* select a constitutive model and *then* tune its parameters by fitting the model to data. Instead, we propose a family of Constitutive Artificial Neural Networks that *simultaneously learn both* the constitutive model and its material parameters.

Classical Neural Networks ignore the underlying physics. In the most general form, constitutive equations in solid mechanics are tensor-valued tensor functions that define a second order stress tensor, in our case the Piola stress, as a function of a second order deformation or strain measure, in our case the deformation gradient [7,8]. Classical Neural Networks are universal function approximators that learn these functions [45], in our case the stress, from training data, in our case experimentally measured stress–strain data, by minimizing a loss function, in our case the mean squared error between model and data stress. Neural Networks have advanced as a powerful technology to interpolate or describe big data; yet, they fail to extrapolate or predict scenarios beyond the training regime [21]. They are an excellent choice when we have no information about the underlying data, but in constitutive modeling, they entirely ignore our prior knowledge and thermodynamic considerations [48].

Constitutive Artificial Neural Networks include kinematical, thermodynamical, and physical constraints. The general idea of this manuscript is to design a new family of Neural Networks that inherently satisfy common kinematical, thermodynamical, and physical constrains while, at the same time, constraining the design space of all admissible functions to make the network robust and reliable, even in the presence of small training data. Our approach is to *reverse-engineer* Constitutive Artificial Neural Networks that are, by design, a generalization of widely used and commonly accepted constitutive models with well-defined physical parameters [37,38]. Towards this goal we revisit the non-linear field theories of mechanics [1,7,8] and suggest to constrain the network *output* to enforce thermodynamic consistency; the network *input* to enforce material objectivity, and, if desired, material symmetry and incompressibility; the *activation functions* to implement physically reasonable constitutive restrictions; and the network *architecture* to ensure polyconvexity.

Constitutive Artificial Neural Networks are a generalization of popular constitutive models. We prototype the design of Constitutive Artificial Neural Networks for the example of an isotropic perfectly incompressible feed forward network with two hidden layers and twelve weights that takes the scalar-valued first and second invariants of the deformation gradient, $[I_1 - 3]$ and $[I_2 - 3]$, as input and approximates the scalar-valued free energy function, $\psi(I_1, I_2)$, as output. The first layer generates the first and second powers, $(\circ)$ and $(\circ)^2$, of the input and the second layer applies the identity and the exponential, $(\circ)$ and $(\exp(\alpha(\circ)) - 1)$, to these powers. This results in eight individual subfunctions that additively feed into the final free energy function ψ from which we

derive the Piola stress, $\boldsymbol{P} = \partial \psi / \partial \boldsymbol{F}$, following standard arguments of thermodynamics. We demonstrate that the approximated free energy function of our network is a *generalization* of popular constitutive models with the neo Hooke [35], Blatz Ko [25], Mooney Rivlin [26,34], Yeoh [36], and Demiray [33] models as special cases. Most importantly, through a direct comparison with these models, the twelve weights of the network gain a clear physical interpretation.

Classical Neural Networks can interpolate robustly, but fail to extrapolate. In a side-by-side comparison with a classical Neural Network, we demonstrate the features of our new Constitutive Artificial Neural Network for several classical benchmark data sets for rubber in uniaxial tension [25–27], equibiaxial tension [27], and pure shear [27]. Both methods robustly identify functions that approximate the data well and reduce the error between model and data within less than 10,000 epochs: The classical Neural Network, without any prior knowledge of the underlying physics, directly learns the stress as a function of the deformation gradient, $\boldsymbol{P}(\boldsymbol{F})$, while the Constitutive Artificial Neural Network learns the free energy as a function of the strain invariants, $\psi(I_1, I_2)$. Our results in Fig. 8 support the general notion that classical Neural Networks *describe* or *interpolate* data well, but cannot *predict* or *extrapolate* the behavior outside the training regime [21]. We also confirm in Fig. 9 that they *fit big data well*, but tend to *overfit sparse data* [48]. To quickly assess the importance of the individual nodes, we color-code their outputs and visually compare their contributions to the final output layer. From the color spectrum in Fig. 10, we conclude that classical Neural Networks tend to activate all nodes of the final layer with non-zero weights, but that these weights have no physical meaning and do not contribute to *interpret* or *explain* the underlying physics.

Constitutive Artificial Neural Networks robustly learn both model and parameters, even for sparse data. Our new family of Constitutive Artificial Neural Network addresses the limitations of conventional classical Neural Networks by including thermodynamic considerations by design. Fig. 11 suggests that they are both *descriptive* and *predictive*, *without overfitting* the data. From the reduced color spectra in Figs. 12 and 13, we conclude that our networks self-select subsets of activation functions, while most of their weights remain zero. Fig. 13 also shows that, for insufficiently rich data, the network still approximates the overall function $\psi(I_1, I_2)$ robustly, but the distribution of the individual contributions of the I_1 and I_2 terms is non-unique. Enriching the training data by multi-mode data from uniaxial tension, equibiaxial tension, and pure shear in Fig. 14 eliminates these non-uniqueness. This suggests that, when trained with sufficiently rich data, Constitutive Artificial Neural Networks *simultaneously learn both a unique model and parameter set*.

Constitutive Artificial Neural Networks enable automated model discovery. For the example of rubber in the high and low temperature regimes, our new Constitutive Artificial Neural Network discovers two three-term models in terms of the first and second invariants, $\frac{1}{2} \mu_2 [I_1 - 3]$ and $\frac{1}{2} \mu_2 [I_2 - 3]$, similar to the classical Mooney Rivlin model [26,34], and in terms of their linear exponentials, $\frac{1}{2} a_1 [\exp(b_1 [I_1 - 3]) - 1]/b_1$ and $\frac{1}{2} a_2 [\exp(b_2 [I_2 - 3]) - 1]/b_2$, similar to the Demiray model [33]. The non-zero network weights take the interpretation of the shear moduli, μ_1 and μ_2, stiffness-like parameters, a_1 and a_2, and exponential coefficients, b_1 and b_2 of these models. Since the network autonomously self-selects the model and parameters that best approximate the data, the human user no longer needs to decide which model to choose. The underlying concept has elements that are more commonly seen in sparse system identification [57], for example, with applications to model discovery in developmental biology [58] or in the characterization of soft materials [59], where the identification process self-selects the relevant terms from a library of admissible terms. Yet, instead of designing the identification algorithms from scratch, our scientific model discovery capitalizes on existing robust algorithms that have been developed and proven successful for classical neural networks. This could have enormous implications, for example in finite element simulations: Instead of selecting a specific material model from a library of available models, finite element solvers could be built around a single generalized model, the Constitutive Artificial Neural Network *autonomously discovers the model* from data, populates the model parameters, and activates the relevant terms.

Current limitations and future applications. In the present work, we have shown the application of Constitutive Artificial Neural Networks for the special case of perfectly incompressible isotropic hyperelastic materials according to Fig. 5. It is easy to see that the general concept in Fig. 3 extends naturally to *compressible* or *nearly incompressible* materials with other symmetry classes, *transversely isotropic* or *orthotropic*, simply by expanding the network input to other sets of strain invariants. A more involved extension would be to consider

history-dependent, inelastic materials, for example by replacing the feed forward architecture through a *long short-term memory network* with feedback connections [60], while still keeping the same overall network input, output, activation functions, and basic architecture. In parallel, we could revisit the network architecture in Fig. 3 by expressing the free energy as a truncated infinite series of products of powers of the invariants, which would result in a *fully connected feed forward* network architecture. This would be critical to capture more complex materials for which the invariants are coupled, including arteries [47] or the brain [61]. One limitation we foresee for these more complex networks, is that the majority of weights might no longer train to zero. If the network learns a large set of non-zero weights, and with them, activates too many terms that feed into the final free energy function, we could reduce the model to the most relevant terms by *network pruning*, a neurologically inspired process in which the network gradually self-eliminates less relevant connections from its overall architecture [46]. Of course, we could also always enforce certain weights to zero, recover a popular subclasses of models, and use the Constitutive Artificial Neural Network for a plain inverse analysis and parameter identification. Finally, one important extension would be to embed the network in a Bayesian framework to supplement the analysis with *uncertainty quantification*. Instead of simple point estimates for the network parameters, a Bayesian Constitutive Artificial Neural Network would learn parameter distributions with means and credible intervals. In contrast to classical Bayesian Neural Networks, here, these distributions would have a clear physical interpretation, since our network weights have a well-defined physical meaning.

9. Conclusion

Constitutive Artificial Neural Networks are a new family of neural networks that satisfy kinematical, thermodynamical, and physical constraints by design, and, at the same time, constrain the space of admissible functions to train robustly, even for space data. In contrast to classical Neural Networks, they can describe, predict, and explain data and reduce the risk of overfitting. Constitutive Artificial Neural Networks integrate more than a century of knowledge in continuum mechanics and modern machine learning to create Neural Networks with specific network input, output, activation functions, and architecture to a priori guarantee thermodynamic consistency, material objectivity, material symmetry, physical restrictions, and polyconvexity. The resulting network is a generalization of widely used popular constitutive models with network weights that have a clear physical interpretation. When trained with sufficiently rich data, Constitutive Artificial Neural Networks can simultaneously learn both a unique model and set of parameters, while most of the network weights train to zero. This suggests that Constitutive Artificial Neural Networks have the potential to enable automated model discovery and could induce a paradigm shift in constitutive modeling, from user-defined to automated model selection and parameterization.

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

Our source code, data, and examples will be available at https://github.com/LivingMatterLab/CANN.

Acknowledgments

References

[1] S.S. Antman, Nonlinear Problems of Elasticity, Second ed., Springer-Verlag New York, 2005.

[2] J.M. Ball, Convexity conditions and existence theorems in nonlinear elasticity, Arch. Ration. Mech. Anal. 63 (1977) 337–403.

[3] G. Holzapfel, Nonlinear Solid Mechanics: A Continuum Approach to Engineering, John Wiley & Sons, Chichester, 2000.

[4] R.W. Ogen, Large deformation isotropic elasticity – on the correlation of theory and experiment for incompressible rubberlike solids, Proc. R. Soc. London Series A 326 (1972) 565–584.

[5] M. Planck, Vorlesungen Über Thermodynamik, Verlag von Veit & Comp, Leipzig, 1897.

[6] A.J.M. Spencer, Theory of invariants, in: A.C. Eringen (Ed.), Continuum Physics, Vol. 1, Academic Press, New York, 1971, pp. 239–353.

[7] C. Truesdell, W. Noll, Non-linear field theories of mechanics, in: S. Flügge (Ed.), Encyclopedia of Physics, III/3, Spinger, Berlin, 1965.

[8] C. Truesdell, Rational Thermodynamics, in: Lecture 5, McGraw-Hill, New York, 1969.

[9] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (2021) 422–440.

[10] K. Linka, A. Schafer, X. Meng, Z. Zou, G.E. Karniadakis, E. Kuhl, Bayesian physics-informed neural networks for real-world nonlinear dynamical systems, Comput. Methods Appl. Mech. Engrg. (2022) http://dx.doi.org/10.1016/j.cma.2022.115346.

[11] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[12] F. As'ad, P. Avery, C. Farhat, A mechanics-informed artificial neural network approach in data-driven constitutive modeling, Internat. J. Numer. Methods Engrg. 123 (2022) 2738–2759.

[13] D.K. Klein, M. Fernandez, R.J. Martin, P. Neff, O. Weeger, Polyconvex anisotropic hyperelasticity with neural networks, J. Mech. Phys. Solics 159 (2022) 105703.

[14] K. Linka, M. Hillgartner, K.P. Abdolazizi, R.C. Aydin, M. Itskov, C.J. Cyron, Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning, J. Comput. Phys. 429 (2021) 110010.

[15] F. Masi, I. Stefanou, P. Vannucci, V. Maffi-Berthier, Thermodynamics-based artificial neural networks for constitutive modeling, J. Mech. Phys. Solids 147 (04277) (2021).

[16] J. Ghaboussi, J.H. Garrett, X. Wu, Knowledge-based modeling of material behavior with neural networks, J. Eng. Mech. 117 (1991) 132–153.

[17] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, Comput. Methods Appl. Mech. Engrg. 304 (2016) 81–101.

[18] S. Conti, S. Müller, M. Ortiz, Data-driven problems in elasticity, Arch. Ration. Mech. Anal. 229 (2018) 79–123.

[19] F.A. Denli, M. Kaliske, A.K. Aan, M.E. Tüfekcioglu, H. Dal, Data driven constitutive modeling of rubberlike materials, in: Research Gate, 2022, Preprint 362157074.

[20] R. Eggersmann, L. Stainier, M. Ortiz, S. Reese, Model-free data-driven computational mechanics enhanced by tensor voting, Comput. Methods Appl. Mech. Engrg. 373 (2010) 113499.

[21] M. Alber, A. Buganza Tepole, W. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G.E. Karniadakis, W.W. Lytton, P. Perdikaris, L. Petzold, E. Kuhl, Integrating machine learning and multiscale modeling: Perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences, Npj Digit. Med. 2 (2019) 115.

[22] K. Linka, C. Cavinato, J.D. Humphrey, C.J. Cyron, Predicting and understanding arterial elasticity from key microstructural features by bidirectional deep learning, Acta Biomater. 147 (2022) 63–72.

[23] Y. Shen, K. Chandrashekhara, W.F. Breig, L.R. Oliver, Neural network based constitutive model for rubber material, Rubber Chem. Technol. 77 (2004) 257–277.

[24] C. Zopf, M. Kaliske, Numerical characterisation of uncured elastomers by a neural network based approach, Comput. Struct. 182 (2017) 504–525.

[25] P.J. Blatz, W.L. Ko, Application of finite elastic theory to the deformation of rubbery materials, Trans. Soc. Rheol. 6 (1962) 223–251.

[26] M. Mooney, A theory of large elastic deformations, J. Appl. Phys. 11 (1940) 582–590.

[27] L.R.G. Treloar, Stress–strain data for vulcanised rubber under various types of deformation, Trans. Faraday Soc. 40 (1944) 59–70.

[28] A. Ghaderi, V. Morovati, R. Dargazany, A physics-informed assembly for feed-forward neural network engines to predict inelasticity in cross-linked polymers, Polymers 12 (2628) (2020).

[29] K.A. Kalina, L. Linden, J. Brummund, P. Metsch, M. Kästner, Automated constitutive modeling of isotropic hyperelasticity based on artificial neural networks, Comput. Mech. 69 (2022) 213–232.

[30] P. Thakolkaran, A. Joshi, Y. Zheng, M. Flaschel, L. De Lorenzis, S. Kumar, NN-EUCLID: Deep-learning hyperelasticity without stress data, J. Mech. Phys. Solids 169 (2022) 105076.

[31] F. Masi, I. Stefanou, Multiscale modeling of inelastic materials with thermodynamics-based artificial neural networks (TANN), Comput. Methods Appl. Mech. Engrg. 398 (2022) 115190.

[32] V. Tac, F. Sahli Costabal, A. Buganza Tepole, Data-driven tissue mechanics with polyconvex neural ordinary differential equations, Comput. Methods Appl. Mech. Engrg. 398 (2022) 115248.

[33] H. Demiray, A note on the elasticity of soft biological tissues, J. Biomech. 5 (1972) 309–311.

[34] R.S. Rivlin, Large elastic deformations of isotropic materials. IV. Further developments of the general theory, Philos. Trans. R. Soc. London Series A 241 (1948) 379–397.

[35] L.R.G. Treloar, Stresses and birefringence in rubber subjected to general homogeneous strain, Proc. Phys. Soc. 60 (1948) 135–144.

[36] O.H. Yeoh, Some forms of the strain energy function for rubber, Rubber Chem. Technol. 66 (1993) 754–771.

[37] R. Mahnken, Strain mode-dependent weighting functions in hyperelasticity accounting for verification, validation, and stability of material parameters, Arch. Appl. Mech. 92 (2022) 713–754.

[38] P. Steinmann, M. Hossain, G. Possart, Hyperelastic models for rubber-like materials: consistent tangent operators and suitability for Treloar's data, Arch. Appl. Mech. 82 (2012) 1183–1217.

[39] P.J. Flory, Thermodynamic relations for high elastic materials, Trans. Faraday Soc. 57 (1961) 829–838.

[40] R. Schulte, C. Karca, R. Ostwald, A. Menzel, Machine learning-assisted parameter identification for constitutive models based on concatenated loading path sequences, Eur. J. Mech. A Solids (2022).

[41] W. Noll, A mathematical theory of the mechanical behavior of continuous media, Arch. Rational Mech. Anal. 2 (1958) 197–226.

[42] C. Polindara, T. Waffenschmidt, A. Menzel, Simulation of balloon angioplasty in residually stressed blood vessels, J. Biomech. 49 (2016) 2341–2348.

[43] R.S. Rivlin, D.W. Saunders, Large elastic deformations of isotropic materials. VII. Experiments on the deformation of rubber, Philos. Trans. R. Soc. London Series A 243 (1951) 251–288.

[44] S. Hartmann, P. Neff, Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility, Int. J. Solids Struct. 40 (2003) 2767–2791.

[45] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133.

[46] S. Budday, P. Steinmann, E. Kuhl, Physical biology of human brain development, Front. Cellular Neurosci. 9 (257) (2015).

[47] G.A. Holzapfel, T.C. Gasser, R.W. Ogden, A new constitutive framework for arterial wall mechanics and comparative study of material models, J. Elasticity 61 (2000) 1–48.

[48] G.C.Y. Peng, M. Alber, A. Buganza Tepole, W. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G.E. Karniadakis, W.W. Lytton, P. Perdikaris, L. Petzold, E. Kuhl, Multiscale modeling meets machine learning: What can we learn? Arch. Comput. Methods Eng. 28 (2021) 1017–1037.

[49] A. Delfino, N. Stergiopulos, J.E. Moore, J.J. Meister, Residual strain effects on the stress field in a thick wall finite element model of the human carotid bifurcation, J. Biomech. 30 (1997) 777–786.

[50] H.M. James, E. Guth, Theory of the elastic properties of rubber, J. Chem. Phys. 11 (1943) 455–481.

[51] M.C. Boyce, D.M. Parks, A.S. Argon, Large inelastic deformation of glassy polymers, Part I: Rate Depend. Constit. Model Mech. Mater. 7 (1988) 15–33.

[52] H. Dal, M. Kaliske, A micro-continuum-mechanical material model for failure of rubber-like materials, J. Mech. Phys. Solids 57 (2009) 1340–1356.

[53] A. Gent, A new constitutive relation for rubber, Rubber Chem. Technol. 69 (1996) 59–61.

[54] G.A. Holzapfel, J.C. Simo, Entropy elasticity of isotropic rubber-like solids at finite strains, Comput. Methods Appl. Mech. Engrg. 132 (1996) 17–44.

[55] C. Miehe, S. Göktepe, F. Lulei, A micro-macro approach to rubber-like materials?part I: the non-affine micro-sphere model of rubber elasticity, J. Mech. Phys. Solids 52 (2004) 2617–2660.

[56] S. Reese, S. Govindjee, A theory of finite viscoelasticity and numerical aspects, Int. J. Solids Struct. 35 (1998) 3455–3482.

[57] A.A. Kaptanoglu, BM. de Silva, U. Fasel, K. Kaheman, J.L. Callaham, C.B. Delahunt, K. Champion, J.C. Loiseau, J.N. Kutz, S.L. Brunton, PySINDy: A comprehensive python package for robust sparse system identification, 2021, arXiv preprint arXiv:2111.08481.

[58] Z. Wang, X. Huan, K. Garikipati, Variational system identification of the partial differential equations governing the physics of pattern-formation: Inference under varying fidelity and noise, Comput. Methods Appl. Mech. Engrg. 356 (2019) 44–74.

[59] Z. Wang, J.B. Estrada, E.M. Arruda, K. Garikipati, Inference of deformation mechanisms and constitutive response of soft material surrogates of biological tissue by full-field characterization and data-driven variational system identification, J. Mech. Phys. Solids 153 (2021) 104474.

[60] M.A. Bhouri, F. Sahli Costabal, H. Wang, K. Linka, M. Peirlinck, E. Kuhl, P. Perdikaris, COVID-19 dynamics across the US: A deep learning study of human mobility and social behavior, Comput. Methods Appl. Mech. Engrg. 382 (2021) 113891.

[61] S. Budday, G. Sommer, C. Birkl, C. Langkammer, J. Jaybaeck, Kohnert, M. Bauer, F. Paulsen, P. Steinmann, E. Kuhl, G.A. Holzapfel, Mechanical characterization of human brain tissue, Acta Biomater. 48 (2017) 319–340.