

Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning

Kevin Linka^{a,*}, Markus Hillgärtner^b, Kian P. Abdolazizi^{a,b}, Roland C. Aydin^c, Mikhail Itskov^b, Christian J. Cyron^{a,c}

^a Department of Continuum and Materials Mechanics, Hamburg University of Technology, Eißendorfer Straße 42, 21073 Hamburg, Germany

^b Department of Continuum Mechanics, RWTH Aachen University, Eilfschornsteinstr. 18, 52062 Aachen, Germany

^c Institute of Materials Research, Materials Mechanics, Helmholtz-Zentrum Geesthacht, Max-Planck-Straße 1, 21502 Geesthacht, Germany

ARTICLE INFO

Article history:

Received 13 March 2020

Received in revised form 26 September 2020

Accepted 16 November 2020

Available online 24 November 2020

Dataset link:

<https://github.com/ConstitutiveANN/CANN>

Keywords:

Deep learning

Data-driven

Constitutive modeling

Hyperelasticity

ABSTRACT

In this paper we introduce constitutive artificial neural networks (CANNs), a novel machine learning architecture for data-driven modeling of the mechanical constitutive behavior of materials. CANNs are able to incorporate by their very design information from three different sources, namely stress-strain data, theoretical knowledge from materials theory, and diverse additional information (e.g., about microstructure or materials processing). CANNs can easily and efficiently be implemented in standard computational software. They require only a low-to-moderate amount of training data and training time to learn without human guidance the constitutive behavior also of complex nonlinear and anisotropic materials. Moreover, in a simple academic example we demonstrate how the input of microstructural data can endow CANNs with the ability to describe not only the behavior of known materials but to predict also the properties of new materials where no stress-strain data are available yet. This ability may be particularly useful for the future in-silico design of new materials. The developed source code of the CANN architecture and accompanying example data sets are available at <https://github.com/ConstitutiveANN/CANN>.

© 2021 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Over the last decades, data acquisition technologies in science and engineering have substantially advanced. These advances include devices for faster and more accurate mechanical tests, imaging technologies such as nanotomography or magnetic resonance imaging (MRI), and high-precision sensor and actuator systems harvesting by the way large amounts of diverse data during materials processing and handling [1,2]. How these data can effectively be used to model the mechanical constitutive behavior of materials, that is, the relation between mechanical stress and strain, is a key question. This question is addressed within the emerging field of data-driven constitutive modeling, which can be divided into two major branches.

* Corresponding author.

E-mail address: kevin.linka@tuhh.de (K. Linka).

The first branch mainly focuses on the question how to represent the relation between mechanical stress and strain in a material. In materials modeling, there has been a long-standing controversy about the functional relations that are most suitable to this end. In particular, the development of new materials typically brings the challenge of developing also tailor-made functional relations to represent their elastic and inelastic constitutive behavior. The first branch of data-driven constitutive modeling tries to resolve this controversy by developing methods for describing the constitutive behavior of materials which do not assume any specific functional form a priori. This problem can be addressed either in a mathematically rigorous manner [3–9] or by some kind of soft computing such as artificial neural networks (ANNs) [10]. Focusing mainly on stress-strain data and neglecting advanced principles from materials theory or thermodynamics naturally increases the amount of data required for constitutive modeling. To overcome this problem, [11,12] introduced invariant-based ANNs that can be trained by stress-strain data but incorporate already by their very structure at least some theoretical knowledge. This work was, however, limited to simple isotropic materials only. Moreover, this and all the above mentioned data-driven methods can be used to describe the mechanical behavior of a known material based on given stress-strain data. None of these methods allows, however, predictions how new materials yet to be developed should behave.

The second branch of data-driven constitutive modeling uses geometric information about the microstructure of a material for creating representative volume elements (RVEs) whose constitutive behavior resembles the one of the whole material. Computational homogenization of these RVEs yields then the macroscopic constitutive behavior of the material [13–16]. This approach allows not only descriptive but also predictive constitutive modeling. However, its computational cost can be very high - and often unfortunately even prohibitively high - even if it is accelerated by model order reduction techniques [17] or machine learning [18–21]. Moreover, it can by its very design not handle data which may be correlated with the constitutive behavior of a material but which lack a direct mechanical or geometric interpretation. Examples for such data are temperature or feed rate data from measurements during the production or processing of materials.

In this paper, we introduce a new machine learning architecture called constitutive artificial neural networks (CANNs). CANNs are able to learn from given training data how to represent the relation between mechanical stress and strain. Thereby, they fall into the first above delineated branch of data-driven constitutive modeling. However, they distinguish themselves from the previously published methods in this field by their ability to incorporate systematically substantial knowledge from materials theory. This allows CANNs to combine advantages from data-driven modeling such as generality and low bias with the advantages of classical analytical constitutive equations, which are in particular the ability to extrapolate reasonably beyond available data, the ability to filter out noise, and the low computational cost.

CANNs are not only able to learn from stress-strain data to describe the constitutive behavior of existing materials. They can also learn from additional relevant information provided to predict the constitutive behavior of new materials for which no experimental data are available yet. This predictive ability distinguishes CANNs from all so far proposed methods of the above delineated first branch of data-driven constitutive modeling.

We will demonstrate that a combined application of CANNs and RVE simulations is particularly promising. CANNs are able to learn in such a setting how to extrapolate the results of RVE-based simulations in a smart manner and help thereby to reduce significantly the number of RVE simulations required to predict the mechanical properties of new materials. This makes CANNs a tailor-made tool for the future in silico development of new materials where the extreme cost of brute-force RVE simulations currently significantly slows down scientific progress.

2. Mechanical background

2.1. Hyperelastic materials

The deformation of a continuum body is characterized by the mapping of its material points from their reference position $\mathbf{X}$ to their respective current position $\mathbf{x}$. The so-called deformation gradient $\mathbf{F} = \text{Grad} \mathbf{x}$, often also written as $\mathbf{F} = \partial \mathbf{x} / \partial \mathbf{X}$, can be used to define the right Cauchy-Green tensor $\mathbf{C} = \mathbf{F}^T \mathbf{F}$, which is uniquely related to the Green-Lagrange strain tensor

$$\mathbf{E} = \frac{1}{2} (\mathbf{C} - \mathbf{I}), \quad (1)$$

with the second order identity tensor $\mathbf{I}$. Both $\mathbf{C}$ and $\mathbf{E}$ can be equivalently used to characterize strain, that is, local deformation of a continuum body. Such a body consists at each point of a specific material. A material is said to be hyperelastic if it can be described by a strain energy per unit referential volume $\Psi(\mathbf{C})$. Strain energy can be used to compute the mechanical stresses in the material. Different stress measures are common in continuum mechanics, for example, the second Piola-Kirchhoff stress tensor

$$\mathbf{S} = 2 \frac{\partial \Psi}{\partial \mathbf{C}}, \quad (2)$$

while the first Piola-Kirchhoff (nominal) and the Cauchy stress tensor are expressed, respectively, by $\mathbf{P} = \mathbf{F} \mathbf{S}$ and $\boldsymbol{\sigma} = 1/(\det \mathbf{F}) \mathbf{F} \mathbf{S} \mathbf{F}^T$.

A mechanical constitutive law relates a measure of strain like $\mathbf{C}$ or $\mathbf{E}$ to a measure of stress like $\mathbf{S}$, $\mathbf{P}$ or $\boldsymbol{\sigma}$. Without loss of generality we focus in this paper on the combination $\mathbf{C}$ and $\mathbf{S}$.

The derivative of the second Piola-Kirchhoff stress tensor with respect to the right Cauchy-Green tensor yields the fourth order tangent tensor (in the material description)

$$\mathbb{C} = 4 \frac{\partial^2 \Psi}{\partial \mathbf{C} \partial \mathbf{C}}. \quad (3)$$

It characterizes the stiffness of a material and is often required for the computational (e.g. finite element) implementation of hyperelastic constitutive laws.

For a material with an internal kinematic constraint $\gamma(\mathbf{C}) = 0$ the constitutive relation (2) is modified to

$$\mathbf{S} = 2 \frac{\partial \Psi}{\partial \mathbf{C}} + q \frac{\partial \gamma}{\partial \mathbf{C}}, \quad (4)$$

where the scalar parameter q represents the Lagrange multiplier associated with the constraint. In particular, for incompressible materials characterized by the condition $\det \mathbf{F} = 1$ we have

$$\mathbf{S} = 2 \frac{\partial \Psi}{\partial \mathbf{C}} - p \mathbf{C}^{-1}, \quad (5)$$

where p represents a hydrostatic pressure.

2.2. Strain energy of anisotropic materials

Many materials of practical interest exhibit different mechanical properties in different directions. This so-called anisotropy complicates the way how strain energy of such materials can be formulated. At the same time, for machine learning of the constitutive behavior of materials it is important to describe their strain energy in as simple terms as possible so that the number of degrees of freedom to be adjusted during machine learning becomes minimal. In this section, we briefly summarize how invariant theory can help us to express strain energy functions of anisotropic materials in a very general and yet highly compact form.

We start by introducing the so-called full orthogonal group in three dimensions

$$\text{Orth} = \{\mathbf{Q} \in \mathbb{R}^3 \otimes \mathbb{R}^3 : \mathbf{Q}^{-1} = \mathbf{Q}^T\}. \quad (6)$$

Any rotation and any reflection from a plane in three dimensions can be represented by a second-order tensor $\mathbf{Q} \in \text{Orth}$. A set $\mathcal{G} \subseteq \text{Orth}$ is called the *symmetry group* of a hyperelastic material with strain energy function Ψ if and only if

$$\Psi(\mathbf{Q}\mathbf{C}\mathbf{Q}^T) = \Psi(\mathbf{C}), \quad \forall \mathbf{Q} \in \mathcal{G}. \quad (7)$$

Geometrically, this means that the strain energy and thereby the mechanical constitutive behavior of the material as a whole do not change if the reference configuration of a material is rotated or reflected by any $\mathbf{Q} \in \mathcal{G}$ before straining it.

For most engineering materials it is possible to define a symmetry group $\mathcal{G}$ by a set $\mathcal{L} = \{\mathbf{L}_j : j = 1, 2, \dots, J\}$ of second-order tensors $\mathbf{L}_j$, called structure tensors, as

$$\mathcal{G} = \{\mathbf{Q} \in \text{Orth} : \mathbf{Q}\mathbf{L}_j\mathbf{Q}^T = \mathbf{L}_j, \quad \forall \mathbf{L}_j \in \mathcal{L}\}. \quad (8)$$

A common way of defining structure tensors is

$$\mathbf{L}_j = \mathbf{l}_j \otimes \mathbf{l}_j, \quad j = 1, \dots, J, \quad \|\mathbf{l}_j\| = 1, \quad (9)$$

where the $\mathbf{l}_j \in \mathbb{R}^3$ are unit direction vectors. These specify preferred directions of the material. Focusing on structure tensors of the above kind only does not include any arbitrary kind of anisotropic material. It includes, however, at least a very large set of such materials, in particular all the practically most relevant types such as isotropic, transversely isotropic, orthotropic or (thin) fiber-reinforced materials. This level of generality appears sufficient for the purposes of this introductory paper on constitutive artificial neural networks. More general settings may be discussed in future work.

Isotropic materials exhibit the same material properties in all directions. This maximal symmetry is associated with an empty set of preferred directions, that is, $J = 0$, so that $\mathcal{G} = \text{Orth}$. For transversely isotropic materials, $J = 1$ and the single existing preferred direction (called the material principal direction) is the axis of rotational symmetry of the material.

When focusing on structure tensors of the type $\mathbf{L}_j = \mathbf{l}_j \otimes \mathbf{l}_j$, it can be shown that the strain energy of the associated anisotropic materials can always be expressed as a function of the following so-called invariants [22,23]

$$\text{tr}[\mathbf{C}], \text{tr}[\mathbf{C}^2], \text{tr}[\mathbf{C}^3], \text{tr}[\mathbf{C}\mathbf{L}_j], \text{tr}[\mathbf{C}^2\mathbf{L}_j], \quad j = 1, \dots, J \quad (10)$$

and

$$\text{tr}[\mathbf{CL}_i\mathbf{L}_j], \text{tr}[\mathbf{L}_i\mathbf{L}_j], \text{tr}[\mathbf{L}_i\mathbf{L}_j\mathbf{L}_k], \quad 1 \leq i < j < k \leq J. \quad (11)$$

The latter two invariants are constant and therefore not required for expressing Ψ as a function of $\mathbf{C}$. The first term in Eq. (11) captures the effect of the change of the angle between two preferred directions during the deformation [23]. In many situations of practical interest, it plays only a minor role and is thus skipped in the following. This leads to strain energy functions of the form

$$\Psi = \Psi \left(\text{tr}[\mathbf{C}], \text{tr}[\mathbf{C}^2], \text{tr}[\mathbf{C}^3], \text{tr}[\mathbf{CL}_1], \text{tr}[\mathbf{C}^2\mathbf{L}_1], \dots, \text{tr}[\mathbf{CL}_J], \text{tr}[\mathbf{C}^2\mathbf{L}_J] \right). \quad (12)$$

It has been noted [23,24] that (12) can be written equivalently in the form

$$\Psi = \Psi \left(\tilde{I}_1, \tilde{J}_1, \dots, \tilde{I}_R, \tilde{J}_R, \text{III}_{\mathbf{C}} \right) \quad (13)$$

using $2R + 1$ so-called generalized invariants

$$\tilde{I}_r = \text{tr}[\tilde{\mathbf{C}}_r], \quad \tilde{J}_r = \text{tr}[(\det \mathbf{C} \mathbf{C}^{-T}) \tilde{\mathbf{L}}_r], \quad \text{III}_{\mathbf{C}} = \det \mathbf{C}, \quad r = 1, 2, \dots, R \quad (14)$$

defined on the basis of $2R + 1$ generalized structure tensors

$$\tilde{\mathbf{L}}_0 = \frac{1}{3} \mathbf{I}, \quad \tilde{\mathbf{L}}_r = \sum_{j=1}^{J_r} w_{rj} \mathbf{L}_{rj}, \quad \sum_{j=1}^{J_r} w_{rj} = 1, \quad w_{rj} \in \mathbb{R}^+, \quad r = 1, 2, \dots, R. \quad (15)$$

These generalized structure tensors represent weighted sums of the standard structure tensors $\mathbf{L}_{rj} = \mathbf{l}_{rj} \otimes \mathbf{l}_{rj}$ introduced in (9). We use a double subscript rj to underline that in principle each generalized structure tensor $\tilde{\mathbf{L}}_r$ can rely on a different subset of J_r preferred directions $\mathbf{l}_{rj}$, $j = 1, \dots, J_r$.

In theory, (12) and (13) are equivalent. However, (13) appears to be more suitable in the context of machine learning for the following reason. In general, the information about the material anisotropy is divided into two parts. The first part is encoded in the structure tensors (i.e., in (10) or (15)) and the second part in the functional relation between invariants and strain energy (i.e., in (12) or (13)). As the weights w_{rj} in (15) allow a greater variety of structure tensors (for the same number of preferred directions), the generalized structure tensors in (15) can capture a larger part of the information about the material anisotropy than the standard structure tensors in (10). Therefore, using (15) and (13) instead of (12) and (10) allows us to isolate the specific aspect of material anisotropy to a larger extent into a single equation (i.e., (15)). Such a divide-and-conquer approach often turns out to be favorable in machine learning where it makes it easier to apply effectively separate sub-ANNs to different aspects of a problem. For this reason, we will use in the remainder of this article (13) to represent the strain energy of anisotropic materials.

Using (13), the second Piola-Kirchhoff stress tensor (2) can be expressed [23,24] for unconstrained materials as

$$\mathbf{S} = 2 \sum_{r=1}^R \left[\frac{\partial \Psi}{\partial \tilde{I}_r} \tilde{\mathbf{L}}_r - \text{III}_{\mathbf{C}} \frac{\partial \Psi}{\partial \tilde{J}_r} \mathbf{C}^{-1} \tilde{\mathbf{L}}_r \mathbf{C}^{-1} + \tilde{J}_r \frac{\partial \Psi}{\partial \tilde{J}_r} \mathbf{C}^{-1} \right] + 2 \frac{\partial \Psi}{\partial \text{III}_{\mathbf{C}}} \mathbf{C}^{-1}, \quad (16)$$

and for incompressible materials as

$$\mathbf{S} = 2 \sum_{r=1}^R \left[\frac{\partial \Psi}{\partial \tilde{I}_r} \tilde{\mathbf{L}}_r - \frac{\partial \Psi}{\partial \tilde{J}_r} \mathbf{C}^{-1} \tilde{\mathbf{L}}_r \mathbf{C}^{-1} \right] - p \mathbf{C}^{-1}. \quad (17)$$

3. Constitutive artificial neural networks (CANNs)

3.1. General

Artificial neural networks (ANNs) map some kind of input to some kind of output. If they are provided in an initial training stage, a sufficient number of input values with the associated correct output values, they can learn from these training data the underlying input-output relation. This process requires training data but not necessarily specific knowledge about the mechanisms that govern the relations between input and output. For the reader not yet familiar with ANNs, Appendix A summarizes the foundations of this computational technique. In this article we focus on so-called feed-forward ANNs, one of the most widely used ANN architectures where signals are passed only in one direction through the different layers of neurons.

According to the universal approximation theorem [25], feed-forward ANNs with a sufficient level of complexity and non-constant activation functions can approximate any continuous function. Therefore, it appears reasonable to replace the numerous individual functional relations that have been proposed in the past for different materials by a general ANN-based

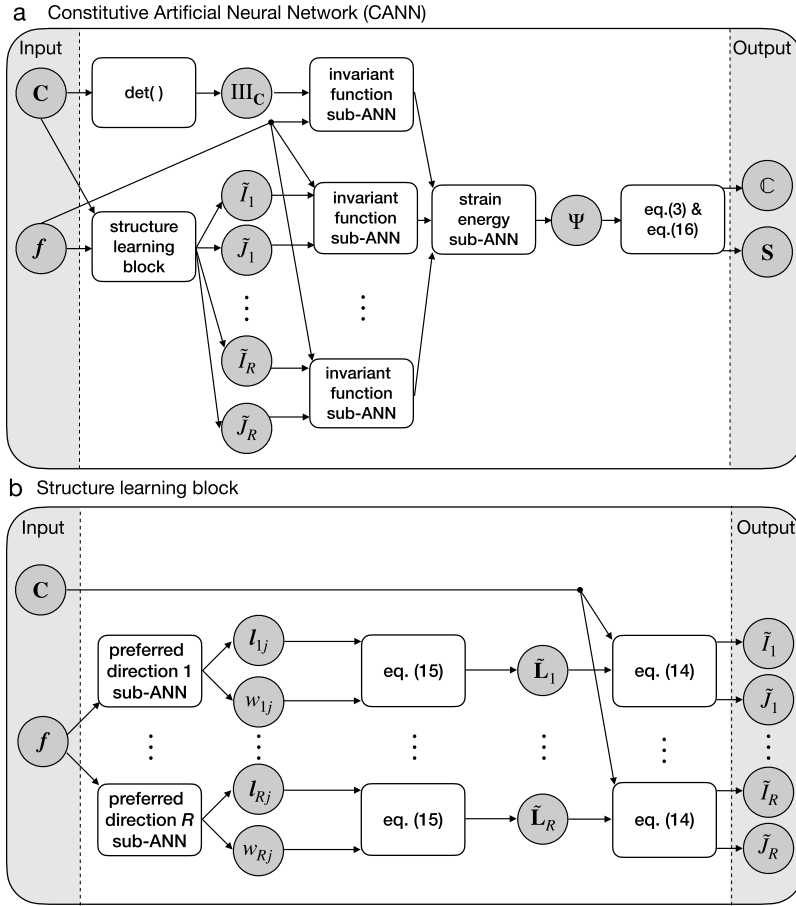


Fig. 1. (a) Schematic illustration of the general architecture of constitutive artificial neural networks (CANNs): the right Cauchy-Green tensor $\mathbf{C}$ together with a feature vector $\mathbf{f}$ that characterizes the properties of the material in some way are fed into a structure learning block that learns the generalized structure tensors of the material considered. This yields the set of generalized invariants $\tilde{I}_1, \tilde{J}_1, \text{III}_{\mathbf{C}}$ on which strain energy may depend. Subsequently, another type of sub-ANN learns the sub-functions by which the individual invariants affect strain energy, respectively. These sub-functions are finally composed by a third type of sub-ANN to a strain energy function Ψ . (b) Schematic illustration of the structure learning block used for computation of the preferred directions and generalized invariants of the material: the feature vector $\mathbf{f}$ is mapped to the proper characteristic material directions I_{ij} and their corresponding weighting coefficients w_{rj} . Using (15) and (14) together with $\mathbf{C}$, one obtains the generalized invariants $\tilde{I}_r$ and $\tilde{J}_r$. All depicted sub-ANNs represent fully connected feed-forward ANNs associated with their illustrated in- and outputs and are explained in detail in Appendix A.

architecture, which can simply be trained with specific data to represent the constitutive behavior of a specific material. This approach is, however, facing two key problems. First, ANNs are by their very nature highly general and do by default not incorporate any information about principles of mechanics and materials theory. As a consequence, ANNs often need a lot of training data to learn the ability to represent the constitutive behavior of a specific material appropriately. Second, standard ANNs by default do not constrain their output to a physically reasonable range, which is, among other factors, massively limiting their ability to extrapolate beyond their training data in a proper way. To overcome these limitations of standard ANNs as data-driven constitutive models, we introduce herein a novel ANN-based machine learning architecture which we refer to as constitutive artificial neural networks (CANNs). The general architecture of CANNs is illustrated in Fig. 1.

CANNs accept as an input strain data in form of the Cauchy-Green tensor $\mathbf{C}$ as well as (optionally) also non-kinematic data in the form of a feature vector $\mathbf{f}$. This input is processed in the interior of CANNs into a strain energy, from which stress and stiffness can be directly calculated using (3), (16), and (17). CANNs do not assume a specific functional form for the strain energy a priori as analytical constitutive equations do. On the other hand CANNs do also not start into learning the constitutive properties of a material without any prior knowledge as standard ANNs would do and as would require excessive amounts of data. Rather CANNs incorporate the knowledge from materials theory that is summarized in the above section 2. That is, the architecture of CANNs directly reflects the structure of the equations in this section but leaves the underlying material-specific functional relations open to be learned from specific material data. This way, CANNs are highly general but incorporate at the same time substantial theoretical prior knowledge that reduces the amount of training data required for learning the constitutive behavior of a specific material significantly. How exactly CANNs incorporate the theoretical knowledge from section 2 in their architecture and how CANNs can be implemented and applied in practice is discussed in more detail in the three following subsections.

3.2. Input

To yield a strain energy Ψ , CANNs, of course, require as an input a kinematic measure of strain such as $\mathbf{C}$. Moreover, CANNs process as an input optionally also a feature vector $\mathbf{f}$. This vector encodes in some way - that may change from material to material - non-kinematic information about the constitutive behavior of a material in the form of real-valued components. These may, for example, characterize the microstructure or processing conditions of a material. If no such information is available, the dimension of $\mathbf{f}$ is zero and the feature vector effectively drops out of the CANN architecture. For the sake of visualization, the complete CANN architecture as schematic implemented implementation is shown in Fig. B1. In such cases, CANNs are only able to describe the constitutive behavior of existing materials for which stress-strain data is available. By contrast, if $\mathbf{f}$ provides some information about certain features of the material or its processing conditions, the CANN automatically learns to relate this information to the constitutive behavior represented by the given stress-strain data. As a consequence, a trained CANN is then able to describe not only the constitutive behavior of materials with known features but also to predict how the constitutive behavior of the material changes if these features encoded in $\mathbf{f}$ change.

3.3. Internal architecture

The internal architecture of CANNs reflects the theoretical knowledge condensed in the equations of section 2. These equations imply that even for very (though not fully) general anisotropic materials strain energy can always be computed in two distinct steps. In the first step the preferred directions and thereby generalized structure tensors of the material of interest are computed via (9) and (15), which yields the generalized invariants of (14). In the second step, strain energy is computed from the generalized invariants following (13).

The first step is reflected by a computation block that is specialized on structure learning and whose topology is illustrated in Fig. 1. This block captures the mechanical knowledge condensed in the equations (9), (14), and (15). As the equations (14) and (15) equally hold for any of the anisotropic materials considered herein, they are directly implemented as algebraic calculations. By contrast, the preferred directions from equation (9) may change from material to material. Therefore, sub-ANNs are used to represent this equation by a functional form that can be learned for each material of interest from the available training data.

The second step could in principle also be captured by a single sub-ANN that learns from the given training data the functional relation between generalized invariants and strain energy by adjusting its internal weights. This would make it, however, hard to quantify the precise role of the different invariants. As the different invariants have a very specific geometric interpretation, respectively - e.g., III_C describes the volumetric deformation - this would make it hard to interpret the trained state of a CANN physically. Indeed physical interpretability is a frequently discussed issue in the field of machine learning. To ensure it at least to some extent, we split the second step of mapping the generalized invariants to a strain energy into two sub-steps. In the first one, the different invariants are individually mapped to sub-functions. In the second one these sub-functions are combined into a strain energy function. Both sub-steps are represented by separate sub-ANNs where the ones for the first sub-step ('invariant function sub-ANN' in Fig. 1) are typically chosen considerably more complex than the one for the second sub-step ('strain energy sub-ANN' in Fig. 1). In the trained state of the CANN, the sub-ANNs for the first sub-step reveal how different geometric deformation modes (such as volumetric deformation) affect strain energy. By contrast, the sub-ANN for the second sub-step reveals how different deformation modes interact with each other with respect to strain energy.

The technical details for implementation and training of the sub-ANNs described above are summarized in Appendix B.

4. Results

4.1. Rubber elasticity

One of the greatest challenges for constitutive models is capturing multi-axial loading. To test their ability to do so for isotropic incompressible materials, Treloar [26] suggested sampling the so-called invariant plane $\text{I}_C\text{-II}_C$, which represents all admissible strain states for isotropic incompressible materials, where with respect to equation (14) we have only two generalized invariants $\text{I}_C = 3 \tilde{\mathbf{J}}_1(\tilde{\mathbf{L}}_0)$ and $\text{II}_C = 3 \tilde{\mathbf{J}}_2(\tilde{\mathbf{L}}_0)$ with a single generalized structure tensor $\tilde{\mathbf{L}}_0 = \frac{1}{3}\mathbf{I}$. For sampling the invariant plane $\text{I}_C\text{-II}_C$, usually three different loading protocols are applied, uniaxial tension, equi-biaxial tension and pure shear. The first two deformation states form the lower and upper bounds of the domain of admissible deformation states in the invariant plane.

To test the ability of CANNs to learn from a small amount of experimental data to represent the mechanical constitutive behavior of the underlying material under multi-axial loading, we used the experimental data from rubber specimens reported by Treloar [27]. These data cover a large part of the $\text{I}_C\text{-II}_C$ plane and are not straightforward to capture by constitutive models [28]. Training data for uniaxial loading, equi-biaxial loading and pure shear under large strains [27] were provided to the CANN. Due to the known isotropy and incompressibility of rubber and the lack of any non-mechanical information, we chose a CANN design with $J = 0$ preferred directions and removed the input III_C and $\mathbf{f}$. The resulting network topology and CANN hyperparameters are given in Appendix B. As shown in Fig. 2, the CANN is able to learn from just 15 data points per load case to represent the complex constitutive behavior underlying Treloar's data. It is worth mentioning that CANNs

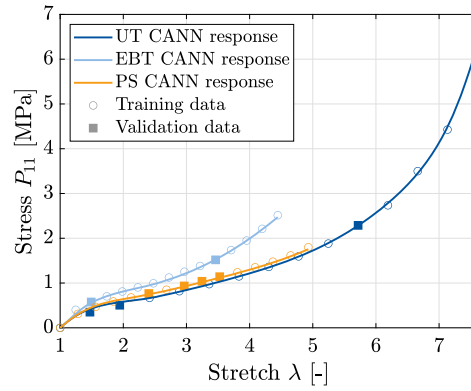


Fig. 2. Experimental data collected by [27] for rubber specimens with three different loading protocols (symbols) vs. constitutive behavior of CANNs trained with these data (solid lines): trained CANNs capture very well the constitutive behavior of the rubber specimens studied in [27]. The abbreviations uniaxial tension (UT), equi-biaxial tension (EBT) and pure shear (PS) are used in the figure legend.

Table 1
Material parameters used for Mooney-Rivlin strain energy function from (18).

c_{10} [MPa]	c_{20} [MPa]	c_{30} [MPa]	c_{01} [MPa]	c_{02} [MPa]	c_{03} [MPa]
$1.6 \cdot 10^{-1}$	$-1.4 \cdot 10^{-3}$	$3.9 \cdot 10^{-5}$	$1.5 \cdot 10^{-2}$	$-2.0 \cdot 10^{-6}$	$1.0 \cdot 10^{-10}$

are able to reach this descriptive capability without any human guidance within just a few minutes of machine learning on a standard desktop computer. By contrast, historically it took human researchers months to years to develop analytical constitutive equations such as the generalized Mooney-Rivlin model that perform equally well.

4.1.1. Generalizability, extrapolation and robustness

During their training, ANNs basically acquire the ability to interpolate between their training data. Hence well-trained ANNs are typically able to yield for any input in the range of their training data a reasonable output. This property of ANNs is often called generalizability. Generalizability is considered one of the most important properties for the practical application of ANNs [29]. Unfortunately, generalizability does not necessarily imply the ability of ANNs to extrapolate, that is, to provide a reasonable output also for input outside the range covered by their training data. In this section, we demonstrate that CANNs indeed have an excellent ability both to generalize and extrapolate, most likely due to the physical knowledge incorporated in their architecture a priori.

We consider a fictitious (isotropic incompressible) rubber-like material with a Mooney-Rivlin strain energy function

$$\Psi_{MR} = \sum_{i=1}^3 c_{i0} (I_C - 3)^i + c_{0i} (II_C - 3)^i. \quad (18)$$

For this material, we generated training and validation stress data with c_{0i} and c_{i0} as specified in Table 1.

We used exactly the same CANN architecture as in the previous section and trained it with just 15 data points per load case. Again training data covered only the three standard load protocols uniaxial tension, equi-biaxial tension and pure shear. All the training data was located within the triangle with the corners $(I_C, II_C) = (50, 15)$, $(I_C, II_C) = (32, 256)$, and $(I_C, II_C) = (3, 3)$. The generalizability of the CANN can be quantified by the difference between the CANN output and the stress computed from (18) and (17) at those points in this triangle that do not form part of the training data. Fig. 3(d) reveals that already for the modest amount of training data used here this error is nowhere larger than around one percent, mostly even much smaller. This clearly reveals an excellent generalizability of CANNs, comparable to the one of analytical constitutive equations. Discussing generalizability, it is also instructive to compare the performance by the amount of data for CANNs and data-driven constitutive models using the distance-minimizing methods introduced by [3]. To yield approximation errors on the order of a few percent for cases similar to the one studied here, the distance-minimizing method was reported in [30] to require on the order of 10^4 – 10^5 stress-strain data pairs, that is, around three order of magnitude more than CANNs. In particular in the context of practical applications, this is a substantial argument in favor of CANNs. In terms of generalizability, CANNs seem to outperform also competing deep learning approaches that have recently been proposed in the area of computational homogenization [18]. These have been reported to require on the order of 10^4 sample points to approximate the three-dimensional mechanical behavior of a single nonlinear spherical inclusion composite with an error of 0.1%. By contrast, CANNs seem to reach such a low approximation error already with around 10^2 training samples.

Fig. 3(d) reveals that also outside the range of training data, that is, on the upper right of the line between the two points $(I_C, II_C) = (50, 15)$ and $(I_C, II_C) = (32, 256)$, the relative error of the CANN is still mostly not higher than around 0.5% and

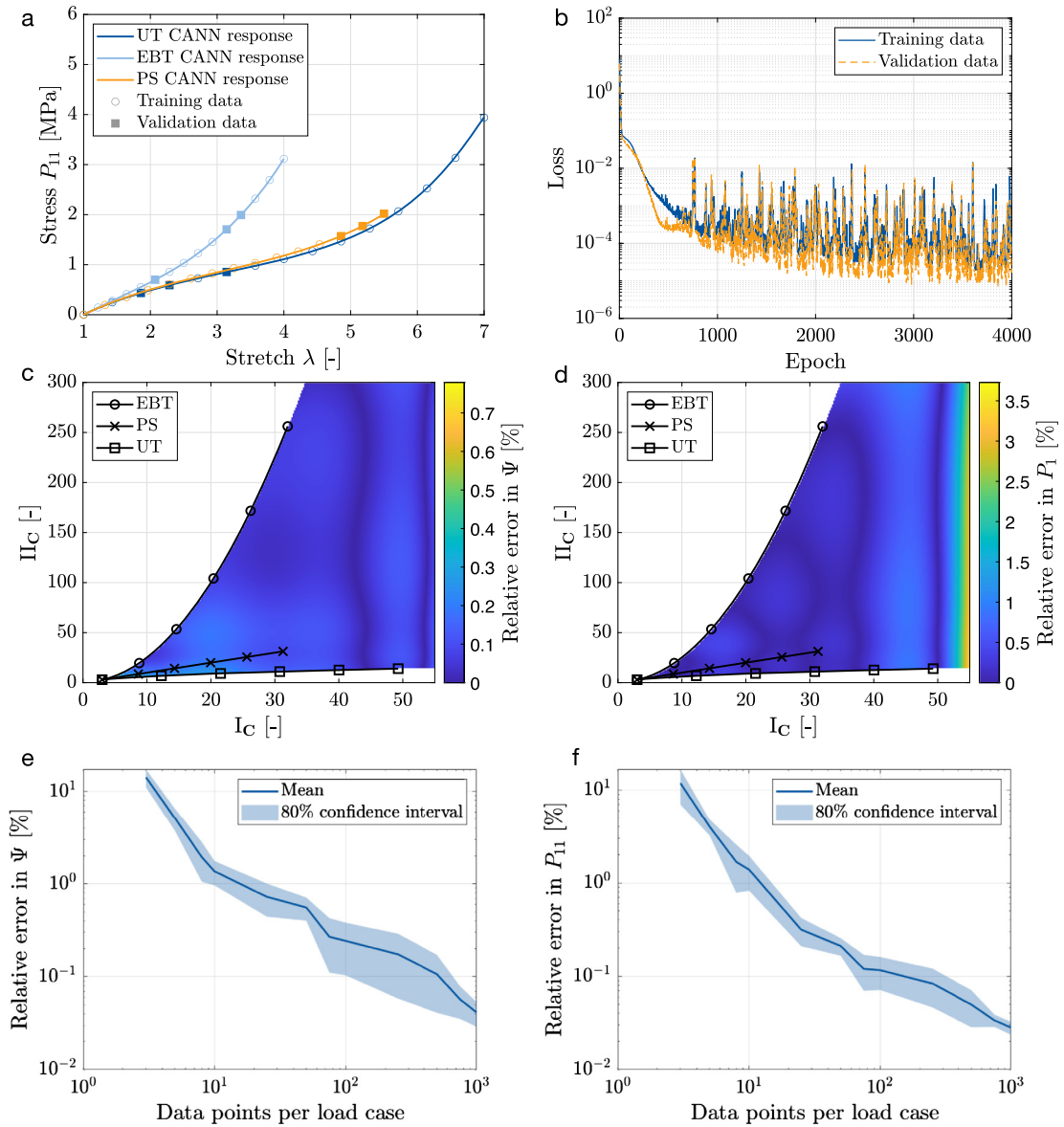


Fig. 3. (a) Training data generated with generalized Mooney-Rivlin model (solid lines) for study of generalizability; (b) convergence of mean square error (MSE) of stresses during CANN training; (c, d) difference between generalized Mooney-Rivlin model from which CANN training data were generated and trained CANN across the whole I_C - I_{II} invariant plane with respect to strain energy and nominal stress. (e, f) Convergence of CANN approximation error (average across the whole invariant plane) as number of stress-strain training data points increases with respect to strain energy and nominal stress. The abbreviations uniaxial tension (UT), equi-biaxial tension (EBT) and pure shear (PS) are used in the figure legends.

nowhere higher than 3.5%. This demonstrates that CANNs exhibit not only an excellent generalizability but also an excellent ability to extrapolate beyond their training data, likely due to the substantial physical knowledge they incorporate a priori in their architecture. Detailed information on the architecture and hyperparameters of the CANN architecture utilized in this subsection are provided in Appendix B in Fig. B2.

It is instructive to study how the CANN approximation across the invariant plane improves as the amount of training data increases. Fig. 3e and f reveal a monotonous convergence of the relative error of the CANN approximation, underlining the reliability of CANNs as data-driven constitutive models.

Not only the performance for ideal data sets is important for a constitutive model but also its robustness against noise as present in many experimental data. To examine this robustness of CANNs we again generated training data with (18), however, before providing the training data to the CANN we added artificial noise. As shown in Appendix C in Fig. C1, the trained CANN demonstrates an excellent ability to filter out the noise in the training data and reproduce the underlying ‘true’ constitutive behavior. Such robustness against noise is a well-known property of ANNs in general [29], which CANNs apparently inherit.

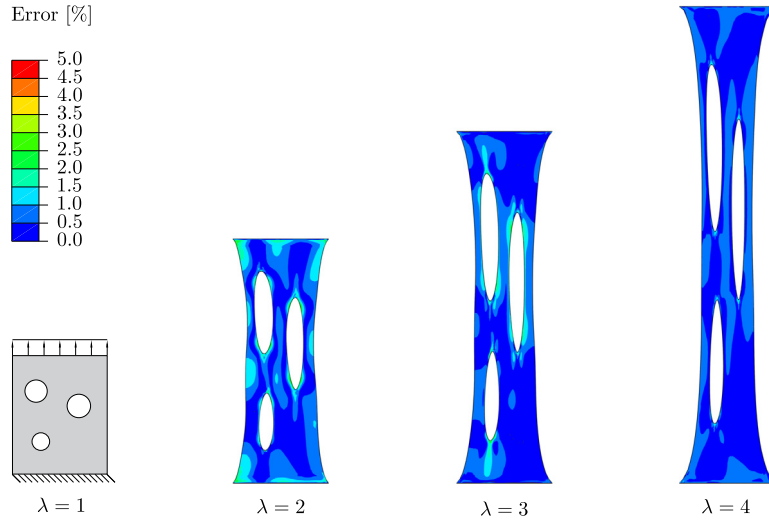


Fig. 4. Abaqus simulation of a plate with circular holes stretched by a factor of $\lambda = 2, 3, 4$, respectively: relative difference in von-Mises stress between the simulation based on trained CANN vs. simulation based on analytical generalized Mooney-Rivlin law (18).

4.1.2. Implementation and performance in computational software

Two important criteria for the suitability of a constitutive model are the effort which their implementation takes and their computational cost in realistic problem settings. To examine the suitability of CANNs with respect to these criteria, we implemented the already trained CANN, whose error plot is given in Fig. 3, in the finite element software Abaqus 2017 (Simulia Corp., Providence, USA) as a user subroutine (UHYPER). As a reference, we also implemented the analytical form of the generalized Mooney-Rivlin law (18) as a UHYPER. Subsequently, we used both implementations to simulate a plate with asymmetrically arranged circular holes, clamped at one end and stretched at the opposite end to a maximal stretch ratio of $\lambda = 4$. The exact geometry and boundary conditions of the plate are defined in Appendix D in Fig. D1. To account for the incompressibility of the material, we discretized the plate with the Abaqus element type C3D8H. This element type is based on a hybrid finite element formulation with bilinear displacement and constant pressure interpolation. Fig. 4 reveals contour plots of our simulations for different stretch ratios λ with $\lambda = 1$ in the stress-free reference configuration. The color code of the contour plots illustrates the relative difference in the von-Mises stress between the simulation with a trained CANN vs. a simulation based on the analytical generalized Mooney-Rivlin model from which the CANN training data was generated. Apparently, this relative difference is very small across the whole domain for all stretch ratios studied.

The implementation of the trained CANN in Abaqus was straightforward. Moreover, the overall computational cost of the CANN-based simulations was only 3% higher than the one of simulations based on the analytical constitutive law. This clearly demonstrates that the implementation of CANNs as constitutive models in standard commercial software is easy and computationally efficient.

4.2. Prediction of constitutive properties

In the previous sections we demonstrated that CANNs are able to learn from stress-strain training data how to describe the constitutive behavior of materials. In this section, we demonstrate that by proper utilization of the feature vector $\mathbf{f}$ CANNs have in addition to that also the ability to predict the constitutive properties of materials not known to them from any stress-strain training data.

We demonstrate this using an example that extends our scope also in another important aspect. Whereas the previous sections focused on isotropic materials only, in this section we study the more general example of an anisotropic matrix-inclusion composite material. Both matrix and inclusions are assumed to consist of an incompressible Neo-Hookean material with a strain energy of the type $\Psi_{\text{NH}} = c_{\text{NH}} (I_{\text{C}} - 3)$. The inclusions are of spheroidal shape. Their centers are placed on a regular cubic grid with periodicity L . The behavior of the compound material is governed by four scalar parameters, which characterize its microstructure and can be summarized in a feature vector $\mathbf{f}$: the material parameter c_{NH} of the matrix and inclusion material, respectively, and the two ratios between the semi-major and semi-minor axis of the spheroid and the periodicity L of the cubic grid.

Against this background, we now imagine the following situation: the homogenized macroscopic mechanical behavior is known for a variety of compound materials of the above type in the form of stress-strain data. In our case we imagine that these stress-strain data have been produced by simulations of RVEs [31]. In other similar scenarios these data might, however, also come from experiments. The key point is only that we assume that for each material where stress-strain data are available, also the microstructure is known in the form of an associated feature vector $\mathbf{f}$.

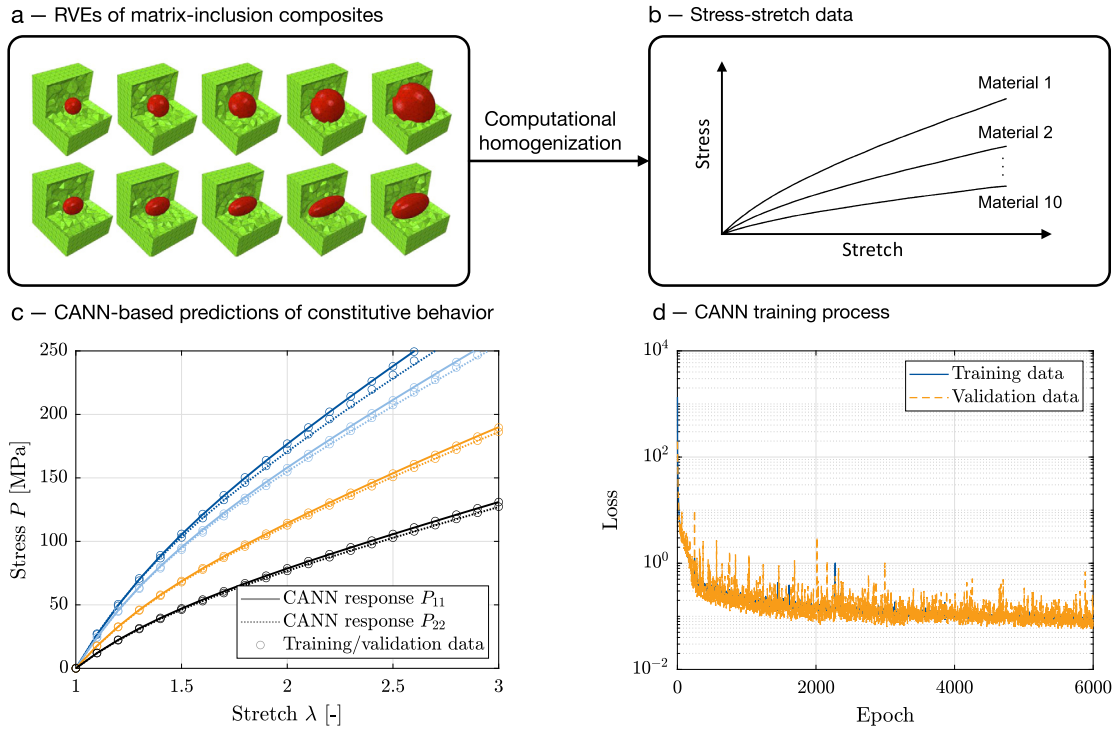


Fig. 5. (a) 10 examples of RVEs of matrix-inclusion composite materials; (b) stress-strain data from computational homogenization of RVEs used for CANN training; (c) CANN predictions (solid and dashed lines) of constitutive behavior of four materials (color-coded) for which the CANN was not provided any stress-strain training data vs. results of RVE simulations of the same materials (symbols); (d) convergence of mean square error (MSE) of stresses during CANN training.

Table 2

Components $[a_1, a_2, a_3]$ of symmetry axes $\mathbf{a}_i$ of inclusion spheroids (given in terms of a Cartesian coordinate system aligned with RVE edges).

i	1	2	3	4	5
$\mathbf{a}_i$	$[0, 0, 1]$	$[0.5, 0, \sqrt{3}/2]$	$[\sqrt{3}/4, 0.25, \sqrt{3}/2]$	$[0.25, \sqrt{3}/4, \sqrt{3}/2]$	$[\sqrt{3}/2, 0, 0.5]$
i	6	7	8	9	10
$\mathbf{a}_i$	$[0.75, \sqrt{3}/4, 0.5]$	$[\sqrt{3}/4, 0.75, 0.5]$	$[1, 0, 0]$	$[\sqrt{3}/2, 0.5, 0]$	$[0.5, \sqrt{3}/2, 0]$

Our objective is now to predict with a CANN the macroscopic mechanical behavior of other compound materials of the same type for which, however, no stress-strain data from RVE simulations (or experiments) are available yet. Such problems are very common in the area of materials research where one often faces the question how the microstructure of a material should be designed to obtain desired mechanical properties. To address this challenge, we train a CANN with the (macroscopic homogenized) stress-strain data of the known materials and provide them also the respective feature vectors $\mathbf{f}$ as an additional input. This way, the CANN can learn the relation between the features quantified in $\mathbf{f}$ and the macroscopic mechanical behavior. The trained CANN should then be able to predict the mechanical behavior also of new materials for which no stress-strain training data are available and about which only their microstructure is known in the form of a feature vector $\mathbf{f}$.

To test, whether CANNs indeed have this predictive ability we first generated training data from 4000 different matrix-inclusion composite materials of the above type. To this end, we performed computer simulations with cubical RVEs with periodic boundary conditions subjected to uniaxial tension up to a stretch of $\lambda = 3$. The micromechanical features of the RVEs were varied in the following way. The Neo-Hookean material parameter for matrix and inclusion material was varied within the set $\{10, 20, 30, 40\}$ MPa and $\{10, 50, 100, 150\}$ MPa, respectively. The two ratios between the lengths of the semi-major and semi-minor axis of the spheroid and the RVE edge length were varied within the set $\{0.15, 0.2, 0.25, 0.3, 0.4\}$. Moreover, we generated for the symmetry axis of the inclusion spheroid, represented by a unit vector $\mathbf{a}_i$, 10 different variations specified in Table 2. These variations sampled both the polar and the azimuthal angle in a spherical coordinate system in $\pi/8$ steps. Examples of the resulting RVEs are illustrated in Fig. 5a.

The homogenized stress-strain behavior of the RVEs was simulated with the commercial finite element software Abaqus 2017 (Simulia Corp., Providence, USA). For each compound material simulated, 20 equidistant stress-stretch sample points

were generated by RVE simulations. For each compound material stress responses in the two directions P_{11} and P_{22} were computed. The results of these simulations are illustrated in Fig. 5b. They form the training data base for a CANN.

Regarding the topology of this CANN, we note that one preferred direction and one associated pair of generalized invariants would be sufficient to represent the compound materials studied in this section. One aim of our study is, however, demonstrating how CANNs handle materials for whose anisotropy their network topology is not tailor-made, which will in practice most often be the case. Therefore, we used a CANN architecture with three randomly initialized preferred directions and three pairs of generalized invariants. The detailed network topology and the associated activation functions are given in Fig. B3. We did not provide III_c as an input to the CANN, endowing it thereby with the a-priori knowledge that the material was incompressible.

After around 6000 training epochs (which took around 8 hours on a standard desktop computer) the CANN had acquired not only the ability to describe the constitutive behavior of the compound materials represented in the training data but also to predict the one of materials not represented in the training data. This ability is illustrated in Fig. 5c for three examples of such materials. In these cases, the CANN was able to predict from a given feature vector $\mathbf{f}$ the constitutive behavior (i.e., the stress-strain curve) of such materials without ever having been provided stress-strain data for materials with exactly these features. Furthermore, we observed that the weighting coefficients of two of the three preferred directions of each invariant pair vanish to nearly zero, demonstrating that the CANN had learned without any external guidance the simplest possible representation of the material symmetry.

As also in all the previous examples, when training the CANN, we strictly split the available data into 80% training data and 20% validation data. The so-called generalization error, that is, the difference between the error for the training and validation data, was consistently small during the whole training process as illustrated in Fig. 5d. This indicates that no overfitting occurred and the CANN indeed learned a generalized ability rather than just an adaptation to peculiarities of the specific training data provided.

5. Conclusions

In this paper we introduced constitutive artificial neural networks (CANNs), which have the ability to learn the constitutive relation of materials from given stress-strain data. This learning process is equivalent to the calibration of the parameters in an analytical strain energy function such as the Neo-Hookean or Mooney-Rivlin strain energy function. However, such analytical models constrain the admitted function space from the beginning on to very specific functional forms so that each of them is applicable only to a very limited set of materials. Therefore, new materials typically require the development of new analytical constitutive equations. In opposition to that CANNs - like ANNs in general - are able to represent nearly any practically relevant functional form. The same CANN architecture may thus in principle be used to handle a large range of very different materials.

CANNs share this advantage of a high generality with other data-driven constitutive models, for example, the ones based on the distance-minimizing method [3–9]. The latter are mathematically clearly more rigorous than CANNs because CANNs rely on ANNs, which have attracted fast rising attention over the last years, but for whose convergence behavior and accuracy rigorous and general mathematical theorems still largely remain pending. Moreover, the distance-minimizing method for data-driven constitutive modeling is also slightly more general than CANNs in that it starts with (almost) no bias coming from the method itself, whereas CANNs introduce at least some bias by their topology and the initialization of their hyperparameters.

While therefore from a theoretical perspective the distance-minimizing method is clearly superior, CANNs surpass this method in several important practical aspects. The first and most important one is the amount of data required for describing the constitutive behavior of a material. As discussed in section 4.1.1, CANNs excel by their ability to generalize provided training data and thus need around three orders of magnitude less data points for reaching a practically acceptable accuracy. This advantage of CANNs results from the substantial prior knowledge from materials theory which they incorporate. Moreover, CANNs inherit the robustness against noise that is a generally known property of ANNs. Finally, we demonstrated that CANNs can easily be implemented in commercial computational software and used there at a computational cost comparable to the one of classical analytical constitutive models. Although we have not yet run a rigorous comparison of performance, one can at least expect that the computational cost of CANNs at runtime is considerably lower than the one of the distance-minimizing method since the latter requires the solution of a non-trivial search problem for the evaluation of the constitutive properties in a specific state whereas CANNs require only the evaluation a standard algebraic equation. These advantages of CANNs compared to other data-driven approaches to constitutive modeling are of great importance for practical applications. There the ability to operate on the basis of a limited amount of data as well as empirical robustness are often considered more important than mathematical rigor.

In summary, the ability to represent the constitutive behavior of a material based on just a very limited amount of data and the ability to extrapolate reasonably beyond are typical advantages of analytical strain energy functions. At the same time, general applicability of one and the same formalism to a wide range of different materials is a typical advantage of modern data-driven modeling as introduced by [3,9]. One can conclude that CANNs favorably combine the advantages of both.

Moreover, CANNs distinguish themselves also in one other important aspect from previously proposed analytical and data-driven constitutive models. They can process as an input not only stress-strain data but also diverse additional infor-

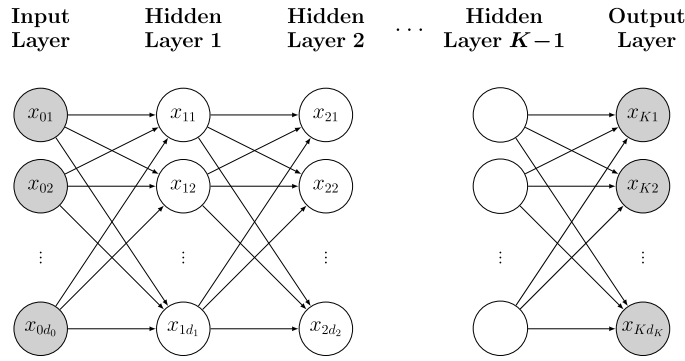


Fig. A1. Illustration of (densely connected feed-forward) artificial neural network (ANN) with neurons illustrated as circles and connections between neurons as lines. The input vector $\mathbf{x}_0$ with components x_{0i} is mapped through $K - 1$ hidden layers of neurons to the output vector $\mathbf{x}_K$. The output value produced by the j -neuron in the k -th layer is generally denoted by x_{kj} with $k = 1, \dots, K$ and $j = 1, \dots, d_K$.

mation provided in a feature vector $\mathbf{f}$ in order to predict the behavior of new materials. We demonstrated this ability only by way of one simple academic example that should be understood in the first place as a proof of concept. This example illustrated in particular how CANNs can learn from RVE simulations and help thereby to reduce the number of such simulations that is required for the computer-aided prediction and optimization of new materials. It is important to note that the feature vector $\mathbf{f}$ that can be processed by CANNs is not limited to any specific kind of data. Thus CANNs should be able to use in principle also information without any direct mechanical or geometric interpretation to predict the properties of new materials. This is in principle not possible with standard RVE-based simulations and it could turn out in future work to be a unique and powerful capability of CANNs in the area of predictive data-driven constitutive modeling. As this general capability has, however, not yet been studied in specific examples herein, strong claims about its practical relevance should still be avoided at this point. Another interesting avenue of future research may be related to the question how to combine the practical advantages of CANNs with the mathematical rigor of the distance-minimizing data-driven constitutive models following [3].

CRedit authorship contribution statement

K.L. and C.J.C. devised the study design and mainly drafted the manuscript. K.L., C.J.C., M.H. and K.P.A. established and refined the computational model. M.H., K.L. and K.P.A. implemented the methods and performed the simulations. M.I. and R.C.A. further refined the study design and helped coordinate its performance. All authors read and approved the final manuscript.

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

The developed source code of the CANN architecture and accompanying example data sets are available at <https://github.com/ConstitutiveANN/CANN>.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer 257981274; Projektnummer 192346071 – SFB 986. Moreover, K. P. Abdolazizi and C. J. Cyron greatly acknowledge financial support from TUHH within the I³-Lab ‘Modellgestütztes maschinelles Lernen fuer die Weichgewebsmodellierung in der Medizin’.

Appendix A. Artificial neural networks

Artificial neural networks (ANNs) can be understood as a form of signal processing unit. They pass the components x_{0i} of an input vector $\mathbf{x}_0$ through several layers of so-called neurons (Fig. A1). Neurons can be understood as computing units. The j -th neuron in the k -th layer receives several input signals and converts them by a so-called activation function g_{kj} into an output value x_{kj} . Herein we focus on densely connected feed-forward neural networks only. In such networks, the j -th neuron in the k -th layer receives as input signals the sum of the output values $x_{(k-1)i}$ of all neurons in the previous $(k - 1)$ -th layer multiplied by some scalar weights $v_{kij} \in \mathbb{R}^+$, respectively. Mathematically, this means

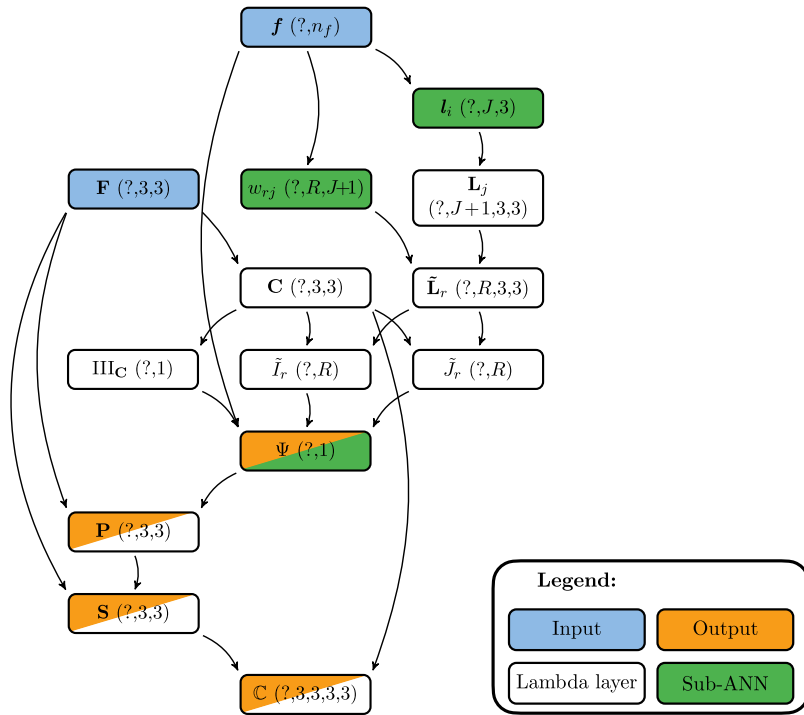


Fig. B1. Schematic implementation of the CANN framework. The tensor dimensions are given in parentheses. Following the conventions of Keras, a question mark refers to an existing dimension of unknown size which here denotes the batch size and Lambda layers refer to user defined functions inside the network architecture. Here, the input is given by the deformation gradient $\mathbf{F}$.

$$x_{kj} = g_{kj} \left(\sum_{i=0}^{d_{k-1}} v_{kij} x_{(k-1)i} \right) \quad (\text{A.1})$$

Here d_{k-1} is the number of neurons in layer $k-1$. In practice, often the same activation function g is chosen for all neurons and all layers so that $g_{kj} = g$. The g_{kj} and v_{kij} are often referred to as hyperparameters. While the activation functions are chosen initially when the ANN is set up and are typically not modified thereafter, the weights v_{kij} are altered during the process of machine learning, that is, during the process where an ANN learns to resemble a certain desired input-output behavior from given training data. These training data contain pairs of input vectors and associated (correct) output vectors. Before an ANN can be used, it requires an initial training stage. During this training stage the input vectors from the training data are successively provided to the ANN and its respective output is compared with the known associated (correct) output from the training data. Based on the difference between both, the weights v_{kij} of the ANN are gradually adjusted. After this procedure has been performed for all training data sufficiently often, one reaches a state where the ANN can successfully resemble the input-output relation underlying to the training data.

In a network with $K+1$ layers, the first layer is called input layer, the layers $1 \leq k \leq K-1$ are called hidden layers, and the last, K -th layer is called output layer. The signals x_{Kj} produced by the neurons of the output layer are the output of the ANN and not processed by any other layer of the ANN (Fig. A1). In general, the choice of the activation function as well as the initialization of the weights are crucial for the problem-specific learning success of ANNs. For a more detailed description of ANNs and their properties, the reader is referred to [29].

Appendix B. Implementation and training of CANNs

We implemented the above delineated general CANN architecture in the machine learning library Keras-TensorFlow [32]. Weights of the different sub-ANNs were generally initialized following [33].

The hyperparameters of CANNs are the number of hidden layers, the number of neurons per layer and the type of activation function of the neurons in the different sub-ANNs.

The upstream sub-ANNs from Fig. 1 for computing the preferred directions uses sigmoidal activation functions at the output neurons of the weights w_{rj} to ensure $0 \leq w_{rj} \leq 1$ and normalizing activation functions for the preferred directions $\mathbf{l}_{rj}$ to ensure unit direction vectors. The condition (15)₃ is enforced in these sub-ANNs by a subsequent normalization operation before the generalized structural tensors are assembled. For the sub-ANNs involved in the computation of the strain energy from the generalized invariants, softplus activation functions $g(x) = \log[1 + \exp(x)]$ [34] were used. The complete schematic implementation is shown in Fig. B1.

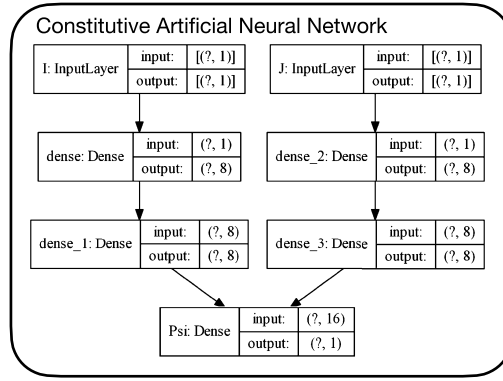


Fig. B2. Detailed architecture of the CANN used for isotropic materials throughout the article. The network architecture is represented in accordance with the Keras visualization scheme [36].

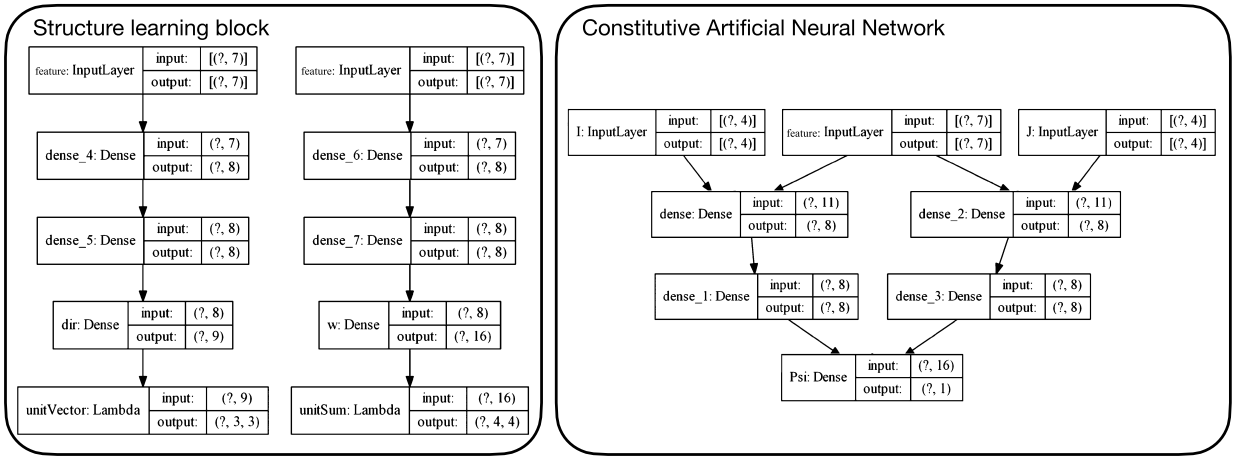


Fig. B3. Detailed architecture of the CANN employed in section 4.2. The network architecture is represented in accordance with the Keras visualization scheme [36].

The precise number of neurons and layers in the different sub-ANNs can be varied from application to application, depending on the complexity of the material to be studied. We found that the results are not overly sensitive to these hyperparameters as long as they are varied within reasonable bounds. In Figs. B3 and B2 the number of layers and the number of neurons per layer are specified for all sub-ANNs in the CANNs used for the different examples in section 4. In addition, the training algorithm settings are provided in Table B1.

It is worth mentioning that, if one chooses upstream sub-ANNs with more preferred directions than actually present in the material, CANNs seem to have the favorable property - as discussed also in section 4.2 - to use only the minimal number of preferred directions that is required to capture the material symmetry and to suppress the other preferred directions by assigning to them (almost) zero weights w_{rj} in the generalized structure tensors in (15). Therefore, no precise knowledge about the anisotropy of a material is required when applying CANNs. The user only has to specify in the beginning a number of preferred directions that definitely appears to be sufficient, which in practice does not pose any major problem. In case the feature vector $\mathbf{f}$ is empty, the preferred directions and associated weights are initialized as trainable variables, otherwise these values are defined as output of the preferred directions sub-ANNs, depicted in Fig. 1.

To compute the derivatives required for stress and stiffness tensors at the output of the CANNs, we used the symbol-to-symbol automatic differentiation [29] provided by TensorFlow [32].

Training of the CANNs was accomplished by a backpropagation algorithm using Adam [35] as an optimizer for the gradient descent. The error function minimized by the training process was the mean squared error of the stress response as

$$\text{MSE} = \frac{1}{n} \sum_{k=1}^n (\tilde{\mathbf{S}}^k - \mathbf{S}^k) : (\tilde{\mathbf{S}}^k - \mathbf{S}^k), \quad (\text{B.1})$$

Table B1
Detailed information on the CANN training algorithm.

Learning rate:	0.001
Optimizer:	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-7}$)
Regularization:	-
Weight & Bias initializer:	Glorot
Training/test data points:	80/20%
Training epochs:	4000 / 6000 (Sec. 4.2)
Batch size:	4 / 64 (Sec. 4.2)
Loss:	Mean squared error
Epoch selection:	Best set of all training epochs

where n denotes the number of stress training data points S^k and the colon a double tensor contraction. As usual, a strict separation between training and validation data was maintained, where 80% of all available data were used for training and 20% for validating the predictive capabilities.

Appendix C. Noisy data

Since measurements are always blurred by interference signals, we evaluated the CANN's ability to cope with noisy data. To do so, we added Gaussian noise with zero mean and standard deviation of 5% related to the reference stress data derived from the analytical generalized Mooney-Rivlin model (18). The noisy stress-stretch training data is depicted in Fig. C1(a). By Fig. C1(b) it becomes evident that the CANN can deal very well with polluted data as the CANN resembles closely the analytical stress-stretch relationship. The error maps at the bottom of Fig. C1 emphasize these findings and demonstrate that the CANN maintains a high generalizability even when trained with sparse noisy data.

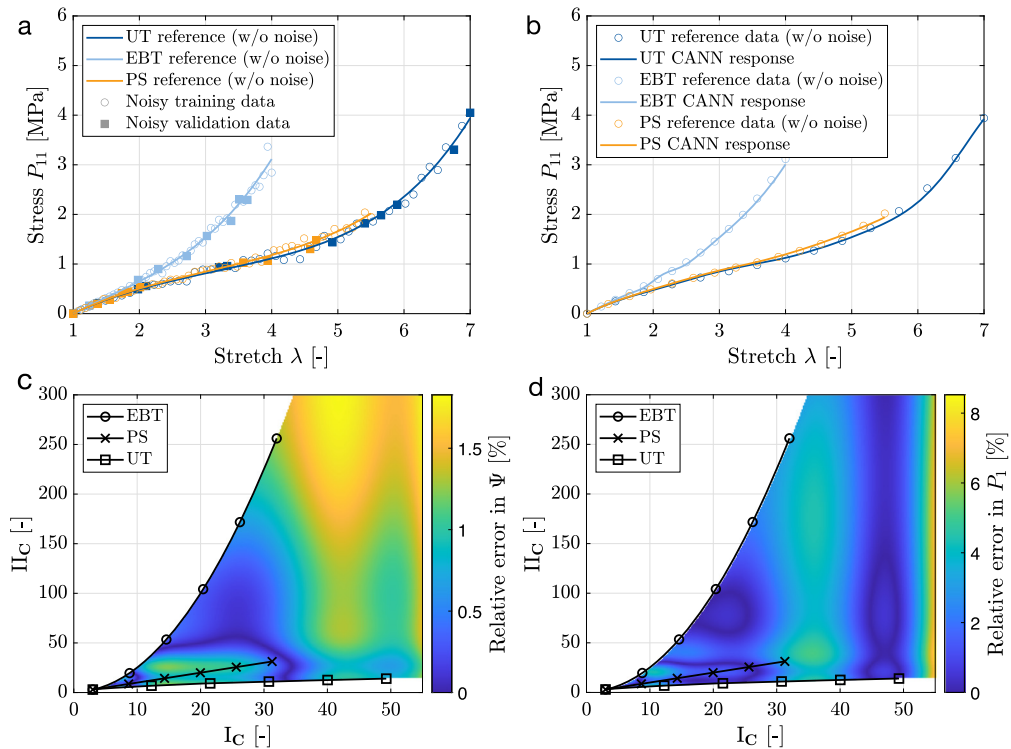


Fig. C1. 50 data points per load case polluted with Gaussian noise: (a) Training data (symbols) generated with generalized Mooney-Rivlin model (solid lines) for study of generalizability; (b) constitutive behavior of trained CANN (solid lines) and corresponding validation data (symbols); (c, d) difference between generalized Mooney-Rivlin model from which CANN training data were generated and trained CANN across the whole I_C - II_C invariant plane with respect to strain energy and nominal stress. The abbreviations uniaxial tension (UT), equi-biaxial tension (EBT) and pure shear (PS) are used in the figure legends.

Appendix D. Finite element implementation

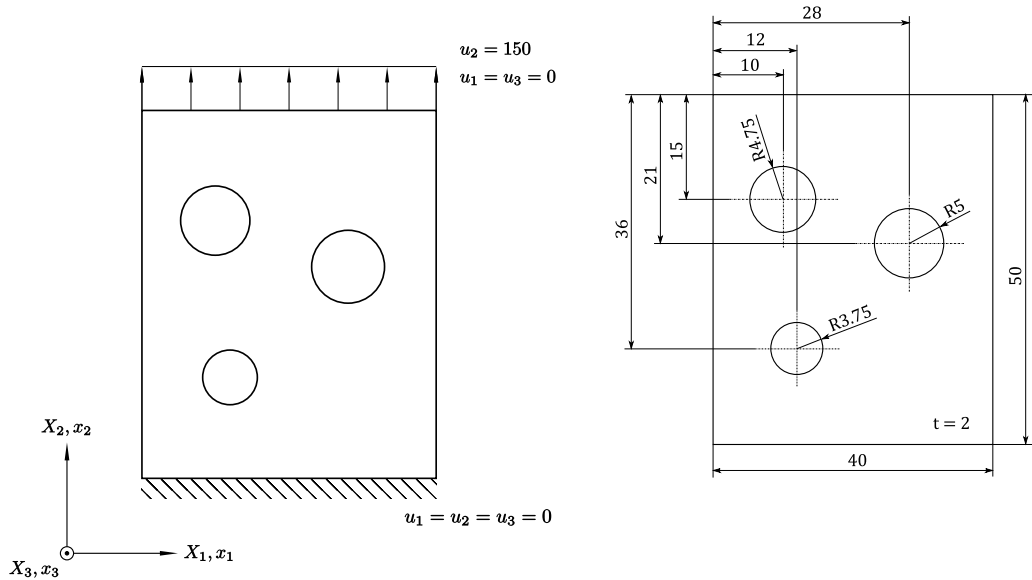


Fig. D1. Boundary conditions (left) and geometric details (right) of plate with circular holes used in the finite element example in section.

References

- [1] Yong Guan, Yunhui Gong, Gang Liu, Xiangxia Song, Xiaobo Zhang, Ying Xiong, Haiqian Wang, Yangchao Tian, Analysis of impact of sintering time on microstructure of LSM-YSZ composite cathodes by X-ray nanotomography, *Mater. Express* 3 (2) (2013) 166–170.
- [2] Sven Nebelung, Björn Sondern, Simon Oehrl, Markus Tingart, Björn Rath, Thomas Pufe, Stefan Raith, Horst Fischer, Christiane Kuhl, Holger Jahr, et al., Functional MR imaging mapping of human articular cartilage response to loading, *Radiology* 282 (2) (2017) 464–474.
- [3] Trenton Kirchdoerfer, Michael Ortiz, Data-driven computational mechanics, *Comput. Methods Appl. Mech. Eng.* 304 (2016) 81–101.
- [4] Ruben Ibañez, Domenico Borzacchiello, Jose Vicente Aguado, Emmanuelle Abisset-Chavanne, Elías Cueto, Pierre Ladevèze, Francisco Chinesta, Data-driven non-linear elasticity: constitutive manifold construction and problem discretization, *Comput. Mech.* 60 (5) (2017) 813–826.
- [5] Rubén Ibanez, Emmanuelle Abisset-Chavanne, Jose Vicente Aguado, David Gonzalez, Elías Cueto, Francisco Chinesta, A manifold learning approach to data-driven computational elasticity and inelasticity, *Arch. Comput. Methods Eng.* 25 (1) (2018) 47–57.
- [6] Lu Trong Khiem Nguyen, Marc-André Keip, A data-driven approach to nonlinear elasticity, *Comput. Struct.* 194 (2018) 97–115.
- [7] Adrien Leygue, Michel Coret, Julien Réthoré, Laurent Stainier, Erwan Verron, Data-based derivation of material response, *Comput. Methods Appl. Mech. Eng.* 331 (2018) 184–196.
- [8] Marcos Latorre, Francisco J. Montáns, Experimental data reduction for hyperelasticity, *Comput. Struct.* (2018).
- [9] Robert Eggersmann, Trenton Kirchdoerfer, Stefanie Reese, Laurent Stainier, Michael Ortiz, Model-free data-driven inelasticity, *Comput. Methods Appl. Mech. Eng.* 350 (2019) 81–99.
- [10] Y.M.A. Hashash, S. Jung, J. Ghaboussi, Numerical implementation of a neural network based material model in finite element analysis, *Int. J. Numer. Methods Eng.* 59 (7) (2004) 989–1005.
- [11] Yuelin Shen, K. Chandrashekhara, W.F. Breig, L.R. Oliver, Neural network based constitutive model for rubber material, *Rubber Chem. Technol.* 77 (2) (2004) 257–277.
- [12] Guanghui Liang, K. Chandrashekhara, Neural network based constitutive model for elastomeric foams, *Eng. Struct.* 30 (7) (2008) 2002–2011.
- [13] I. Temizer, T.I. Zohdi, A numerical method for homogenization in non-linear elasticity, *Comput. Mech.* 40 (2) (2007) 281–298.
- [14] Felix Fritzen, Samuel Forest, Djimedo Kondo, Thomas Böhlke, Computational homogenization of porous materials of Green type, *Comput. Mech.* 52 (1) (2013) 121–134.
- [15] B. Stier, J.-W. Simon, S. Reese, Comparing experimental results to a numerical meso-scale approach for woven fiber reinforced plastics, *Compos. Struct.* 122 (2015) 553–560.
- [16] Christian Miehe, Jörg Schröder, Jan Schotte, Computational homogenization analysis in finite plasticity simulation of texture development in polycrystalline materials, *Comput. Methods Appl. Mech. Eng.* 171 (3–4) (1999) 387–418.
- [17] Felix Fritzen, Matthias Leuschner, Reduced basis hybrid computational homogenization based on a mixed incremental formulation, *Comput. Methods Appl. Mech. Eng.* 260 (2013) 143–154.
- [18] B.A. Le, Julien Yvonnet, Q.-C. He, Computational homogenization of nonlinear elastic materials using neural networks, *Int. J. Numer. Methods Eng.* 104 (12) (2015) 1061–1084.
- [19] Zeliang Liu, C.T. Wu, M. Koishi, A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials, *Comput. Methods Appl. Mech. Eng.* 345 (2019) 1138–1168.
- [20] Xiaoxin Lu, Dimitris G. Giovanis, Julien Yvonnet, Vissarion Papadopoulos, Fabrice Detrez, Jinbo Bai, A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites, *Comput. Mech.* 64 (2) (2019) 307–321.
- [21] Denise Reimann, Kapil Nidadavolu, Hamad ul Hassan, Napat Vajragupta, Tobias Glasmachers, Philipp Junker, Alexander Hartmaier, Modeling macroscopic material behavior with machine learning algorithms trained by micromechanical simulations, *Front. Mater.* 6 (2019) 181.

- [22] Itskov Mikhail, *Tensor Algebra and Tensor Analysis for Engineers: With Applications to Continuum Mechanics*, 4th edition, Springer Publishing Company, Incorporated, 2015.
- [23] Alexander E. Ehret, Mikhail Itskov, A polyconvex hyperelastic model for fiber-reinforced materials in application to soft tissues, *J. Mater. Sci.* 42 (21) (2007) 8853–8863.
- [24] Mikhail Itskov, Nuri Aksel, A class of orthotropic and transversely isotropic hyperelastic constitutive models based on a polyconvex strain energy function, *Int. J. Solids Struct.* 41 (14) (2004) 3833–3848.
- [25] Kurt Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Netw.* 4 (2) (1991) 251–257.
- [26] Leslie Ronald George Treloar, *The Physics of Rubber Elasticity*, Oxford University Press, USA, 1975.
- [27] L.R.G. Treloar, Stress-strain data for vulcanized rubber under various types of deformation, *Rubber Chem. Technol.* 17 (4) (1944) 813–825.
- [28] Paul Steinmann, Mokarram Hossain, Gunnar Possart, Hyperelastic models for rubber-like materials: consistent tangent operators and suitability for treloar's data, *Arch. Appl. Mech.* 82 (9) (2012) 1183–1217.
- [29] Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press, 2016.
- [30] Laurent Stainier, Adrien Leygue, Michael Ortiz, Model-free data-driven methods in mechanics: material data identification and solvers, *Comput. Mech.* 64 (2) (2019) 381–393.
- [31] S.L. Omairey, P.D. Dunning, S. Sriramula, Development of an ABAQUS plugin tool for periodic RVE homogenisation, *Eng. Comput.* 35 (2019) 567–577, <https://doi.org/10.1007/s00366-018-0616-4>.
- [32] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, 2010, <https://arxiv.org/abs/1502.01852>.
- [34] Vinod Nair, Geoffrey E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.
- [35] D. Kinga, J. Ba Adam, A method for stochastic optimization, in: *International Conference on Learning Representations (ICLR)*, vol. 5, 2015.
- [36] François Chollet Keras, <https://github.com/fchollet/keras>, 2015.