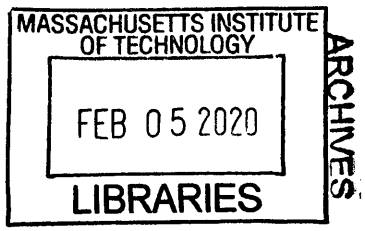
On the Potential of Convolutional Neural Networks for Estimating the Structural Response of Two-Material Structures

by

Julian Heidenreich



M.Sc. Karlsruhe Institute of Technology (2017)

Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2020

Author: Signature redacted

Department of Mechanical Engineering
January 15th, 2020

Certified by: Signature redacted

Tomasz Wierzbicki
Professor of Applied Mechanics
Thesis Supervisor

Certified by: Signature redacted

Nicolas Hadjiconstantinou
Chairman, Committee on Graduate Students
Department of Mechanical Engineering

On the Potential of Convolutional Neural Networks for Estimating the Structural Response of Two-Material Structures

by

Julian Heidenreich

Submitted to the Department of Mechanical Engineering on January 15th, 2020, in Partial Fulfillment of the Requirements for the Degree of Master of Science in Mechanical Engineering

Abstract

During the past years, the neural networks drew more and more interest from the solid mechanics community. In particular, they have proven to provide a powerful framework for constitutive modeling. This work takes first tentative steps towards the use of convolutional neural network based modelling techniques to estimate structural response.

The present work investigates the initial yield of two-dimensional architected materials. The dataset is obtained either by physical or virtual tests. In future applications, the network will be trained based on physical experiments. In the course of this work, they have been replaced by virtual experiments relying on numerical simulations. The computational framework is modeled as an encoder-decoder network and leverages the effectiveness of convolutional neural networks to estimate the structural response of two-material structures. More specifically, the constructed network manages to replicate shape distortions of the yield surface for numerous hole configurations as well as various types of perforation. Furthermore, it accurately predicts the orientation dependent material response for varying degrees of anisotropy. The fact that the network directly translates geometrical data into mechanically significant quantities leads to the strong conjecture that

the neural network is capable to extract and encapsulate all geometrical and mechanically significant information into a small number of scalar variables.

Thesis Supervisor: Tomasz Wierzbicki
Title: Professor of Applied Mechanics

Acknowledgements

First and foremost, I would like to express my deepest acknowledgments and gratitude to Professor Tomasz Wierzbicki, my thesis advisor, for offering me the opportunity to pursue this research and for his continuous guidance during my years at MIT.

I would like to sincerely thank my colleagues at the Impact and Crashworthiness Lab. Special thanks are due to Dr. Maysam Gorji for sharing the passion for solid mechanics as well as taking his time for countless, very fruitful discussions and sharing his knowledge. Special thanks are also due to Dr. Juner Zhu and Dr. Thomas Tancogne-Dejean for creating an invaluable working atmosphere in the lab as well as to Ms. Barbara Smith for her endless and indispensable support.

I greatly thank the MIT Industrial Fracture Consortium for the financial support.

My deepest gratitude goes to my family for their love, moral support and encouragements throughout this thesis. Last but not least, Catherine, for your love and invaluable support.

Contents

List of Figures

List of Tables

Chapter 1 General Introduction

1.1 Motivation and Objectives

Comprehensive studies on the theory of homogenization methods have been carried out throughout the last decades. There exists an increasing demand because of a need for more advanced materials and structures in terms of their weight and strength. Due to their superior specific mechanical properties and fatigue behavior, composite materials are becoming more and more attractive for a vast number of applications in automotive, aerospace, marine, as well as biomedical industry. Other applications of homogenization theory include modelling of granular materials used in lithium-ion batteries, metamaterials and other porous structures.

Material heterogeneities arise in these materials by combining different materials such as particle inclusions or fiber reinforcements. Deriving the macroscopic deformation as well as mechanical properties of the composite based on the microscopic stress states as well as local deformation is computationally expensive, even on modern high-speed computers. The characterization of interface behavior and failure between the combined materials poses an additional challenge, which needs to be addressed on atomic or particle level describing the interaction of individual pendant groups of molecules. Therefore, and especially in structural large-scale applications, it is infeasible to use microscale material models and to describe the exact microscale material structure in numerical simulations.

The first efforts of determining the effective elastic moduli of porous cellular materials date back to forty years ago. Gibson and Ashby (1982, 1988), as well as Christenson (1986), published a comprehensive list of scaling laws connecting the evolution of the apparent Young's modulus as a function of relative density for different load cases and structures like beam bending and axial compression. More recent approaches to approximate the effective elastic properties are summarized in the so-called self-consistent methods. Starting from an initially homogenous material, the second material is introduces as inclusion according to different mathematical rules and assumptions (e.g. Mori and Tanaka, 1973; Willis, 1977).

Each of these approaches is physically motivated but leaves a certain degree of uncertainty that prevents these approximations from being generally applicable. In this regard, several attempts have been made to derive general upper and lower bounds on the effective strain energy that later allows to derive upper and lower bounds on the stiffness tensor as well as engineering constants like the Young's or bulk modulus for anisotropic composites (Kröner, 1977). Linear formulations lead to the well-known Voigt (1889) and Reuss (1929) bounds for composite materials. Based on this approach, the upper and lower boundaries for the bulk and shear modulus result in being the harmonic and arithmetic mean of the properties of the base materials. Following this procedure, one obtains bounds that describe the effective mechanical properties accurately for low phase-contrasts but is not practical in the higher phase-contrast regime. In order to obtain stricter bounds on the effective strain energy, Hashin and Shtrikman (1963) relied on variational principles and deployed the concept of polarization stress allowing to reduce the mechanical problem to a Lippmann–Schwinger type of equation. The derived bounds on the elastic energy for macroscopically isotropic composites are often referred to when benchmarking optimal mechanical properties of porous or two-phase materials.

Later efforts refined the general bounds to more specific application and use cases based no experimental observations. Semi-empirical models were proposed by Halpin and Tsai (1967) incorporating an additional parameter accounting for the microstructure packaging, e.g. hexagonal close packing or quadratic arrangements. The experimentally determined factor scales the estimate of the effective moduli between the Voigt and Reuss bounds. A few decades later, Baxter et al. (2001) carefully develop a technique to extract the effective elastic moduli of particle reinforced metal matrix composites from three-dimensional random microstructure. Around the same time, Byström (2003) investigates the influence of the inclusions distribution on the measured effective conductivity of composite materials. Within this framework, a large amount of research effort focuses on determining the optimal representative volume element (RVE) to accurately capture the material response of a random material property field (Drugan and Willis, 1996; Gusev, 1997; Ostoja-Starzewski, 1998). Trias et al. (2006) clearly demonstrate the advantage of random models over periodic models for fiber reinforced composites concerning fracture initiation and other local phenomena.

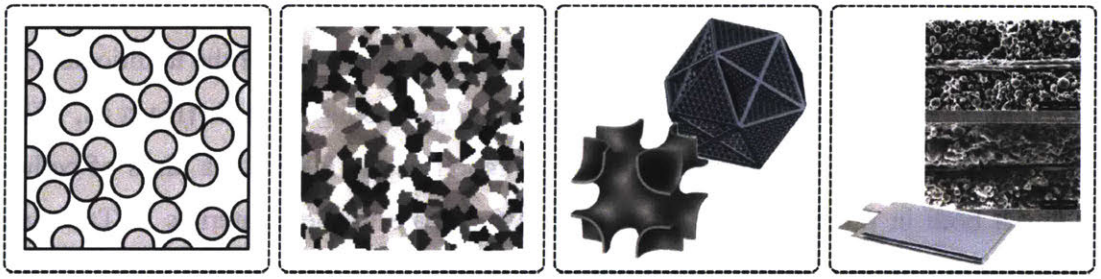
Suquet (1987) adopted the modern term of *homogenization*. The theory of homogenization uses the asymptotic expansion to approximate the relevant field variables like displacements, strains and stresses via an additive split in an average, constant value and perturbations in the field variable as a result from heterogeneity on a microscopic level. The assumption of the existence of a certain degree of microstructure symmetry allows to substitute the on a local level rapidly varying coefficients with constant or slowly varying coefficients on a global scale within the differential equations defining the physical problem. The resulting partial differential equations can be solved analytically or by deploying numerical methods like the boundary element method or spectral method. In recent times, the finite element method (FEM) is the most popular method in numerical homogenization. While providing a convenient and generally accurate way to generate predictions, the FEM also suffers from inherent drawbacks originating from the very nature of the underlying numerical procedure. The predictive accuracy is tied to the mesh quality and exact element formulation used to discretize the geometry as well as microstructure. As a result, especially with ongoing element distortion under large local deformations the methods suffers a significant accuracy loss and tends to provide an overall too stiff response (Liu, 2002). To overcome these issues, Liu (2010) developed multiple numerical methods summarized under the name smoothed finite element methods (S-FEMs). The strain smoothing technique helps to overcome some of the shortcomings of the traditional FEM and is believed to yield a higher accuracy compared to existing methods (He et al., 2013). In the above context, of interest is the construction of the RVE for a class of cellular material both in the elastic and plastic range (compare to metamaterials in Tancogne-Dejean et al. (2018) or Bonatti and Mohr (2017)).

Despite all the comprehensive studies carried out and the considerable progress that has been made during the last decades, the homogenization of material properties still remains a challenge nowadays. The preceding paragraphs should not leave the reader with the impression that the problem at hand has been solved, the reality is far from it. Every new class of materials and category of physical interaction of components requires a rigorous investigation and formulation of new mathematical formulations. Additionally, material sciences still advance much faster than the research regarding the material characterization. Our suggested approach does provide a robust solution to the problem at

hand providing a fully anisotropic, three dimensional material model capable of covering all physical material properties of metallic, polymeric or composite nature. The machine learning based procedure promises to be generally applicable to any material class and can be paired with any numerical method.

1.2 Problem statement

In the scope of this work, the author would like to propose a novel approach to bridge the gap between geometrical, structural information and effective mechanical properties. The introduced framework is capable of predicting effective material properties as well as full field quantities, like displacements or stresses. At the same time, it can be easily applied to a large variety of different applications (compare Fig. 1). These use cases include, but are not limited to particle and fiber composites (Fig. 1a), microstructural information of metallic materials (Fig. 1b), architected optimal metamaterial (Fig. 1c) and lithium-ion batteries (Fig. 1d).



a) b) c) d)

Figure 1. Illustration of possible use-cases of the proposed modelling framework: a) particle and fiber composites ; b) microstructural information of metallic materials; c) architected optimal metamaterial; d) lithium-ion batteries

For simplicity, the capabilities of the new modelling technique will be demonstrated by the means of a well-defined two-dimensional problem originated in the world of composite materials. Starting out with a particle or fiber reinforced basis material with a

random distribution of inclusions (see Fig. 2a), the investigations are limited to a regular square of five-by-five elementary unit cells (Fig. 2b). In order to avoid the numerically challenging description of the interactions between the different materials via the interface, the problem is reduced to a single, porous material (compare Fig. 2c). Circular holes, schematically representing the porosity, are randomly distributed on 13 out of the 25 regular grid positions. To limit the number of virtual experiments and simulation costs, we introduce a four-fold symmetry by rotating the structure around one of its corner points (Fig. 2d). A more detailed description of the geometries can be found in section 2.1.

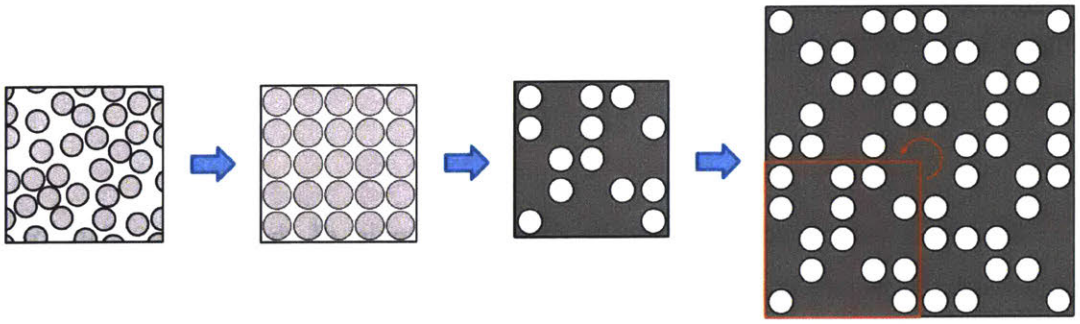


Figure 2. Refining the problem statement starting from a composite structure with a random particle or fiber distribution

In this study, the matrix material is taken to be a dual phase steel, the primary goal of this investigation is to predict the anisotropic yield surface of these architected, two-dimensional structures based on geometrical information. The geometrical input is diversified by different hole configurations (Fig. 3) as well as three different hole types: circular, square and triangular (see detail in Fig. 3). The author would like to showcase a neural network based modelling framework. After a successful training with the help of physical or virtual experiments, the only input to the system, an architecture relying on a convolutional neural network (CNN), is the geometry (a detailed description follows in section 2.1) and it directly outputs the structure's yield surface (compare Fig. 4).

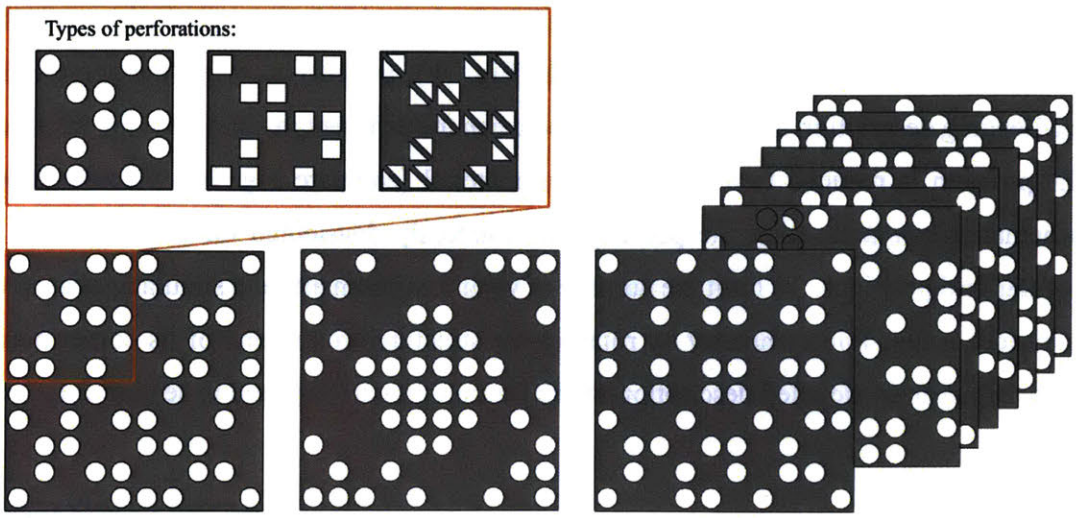


Figure 3. Selected examples of different hole configurations; detail depicts the three hole types

This is a very specific, precisely defined example of the general CNN based framework. However, this illustrative, simplified showcase should not diminish the fact that the proposed modelling procedure can be deployed for arbitrary geometric structures and full three dimensional investigations targeting effective material properties or field quantities.

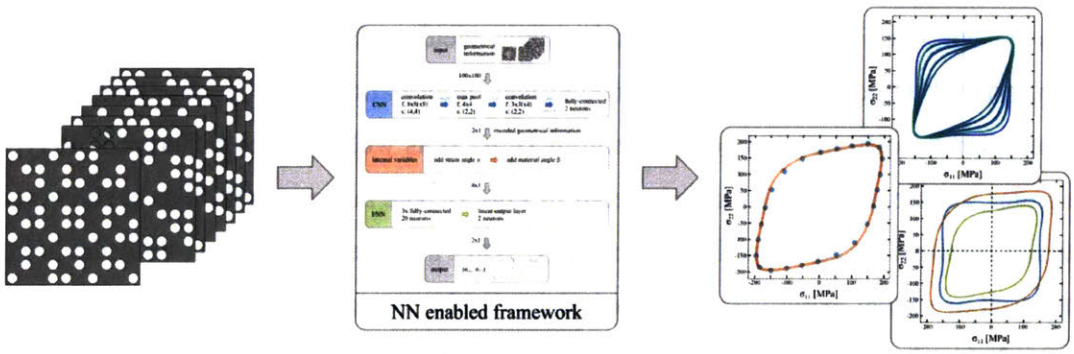


Figure 4. Problem statement: Prediction of anisotropic yield surface based on structural information only

1.3 Structure of the thesis

In order to tackle the problem pictured in the preceding paragraphs, the thesis will be divided in five main chapters. Each of these chapters will be dedicated to a detailed description of sub-tasks leading to the development of a CNN-based computational framework to predict the yield surface for the structures at hand.

After the motivation and the brief problem statement in the previous paragraphs, the second chapter is centered on the virtual experiments in order to create the required input data for the neural network. As substitution of physical experiments, we preform numerical simulations relying on an isotropic, elastic-plastic behavior of the base material. The simulations are carried out using the software package Abaqus/Standard. Periodic boundary conditions are deployed to generate the macroscopic material response based on the discretized reference volume element. A subsequent post-processing step is conducted to extract adequate data for adjusting the free parameters of the neural network model via the training procedure described in the next chapter.

Two neural network architectures are gently introduced to the reader covering the basic mathematical foundation of each structure. Furthermore, these two paragraphs on feedforward and convolutional neural networks help to establish common terminology and techniques that will be used throughout the work. The last section will be devoted to the training procedure. The described problem set-up results in a high-dimensional optimization problem that is very difficult to solve with a standard gradient descent algorithm. Thus, this work relies on both, more sophisticated training schemes as well as suitable transformations of input variables.

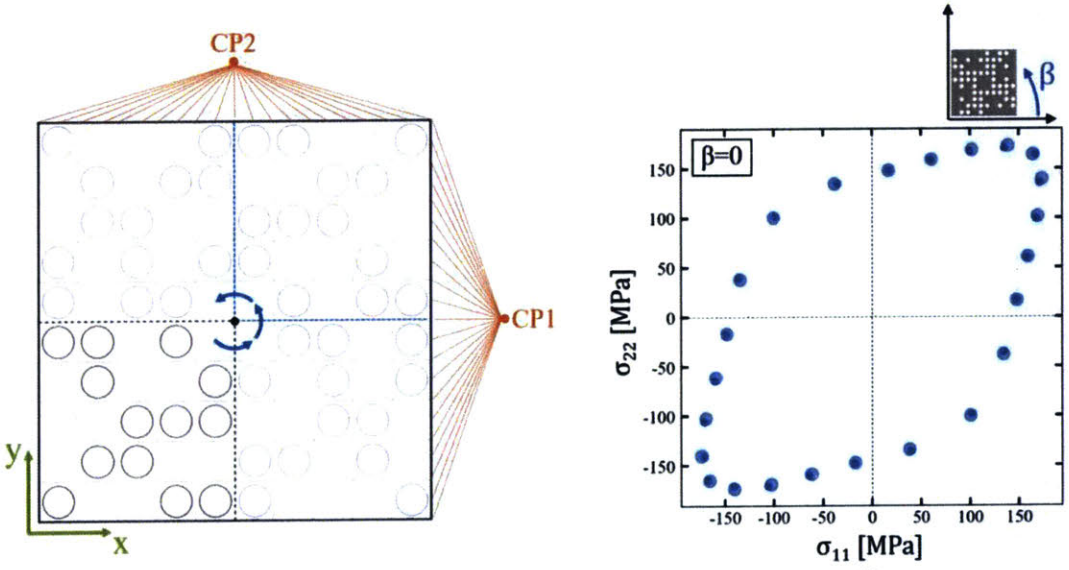
Combining all previous aspects, the forth chapter will explain the final architecture of the numerical framework in detail. It will elaborate on the importance of the appropriate choice of various model parameters in order to allow for a successful training procedure. In addition to that, an array of diverse model capabilities is demonstrated. The presented results include the prediction of the anisotropic material response of a single structure. Besides this material orientation dependent behavior, the modelling framework is capable to replicate shape distortions for different perforation types as well as hole configurations. The neural network based model exhibits a large amount of flexibility and agility to

accurately predict the unique characteristics and varying degrees of anisotropy of numerous architected materials.

Chapter 2 Virtual experiments

2.1 Detailed FE model

As briefly mentioned in section 1, the present illustrations are based on architected, two-dimensional structure as briefly shown in figure 2 and 3. The basis material will represented by a dual phase steel. The material porosity is introduced by randomly distributing 13 holes onto a five-by-five grid. By choosing this ratio we limit the number of possible hole configuration to 5,200,300. All cut-outs, both circular and square as well as triangular, are designed to reduce the surface are of an elementary cell by half; yielding a relative density of $\rho^* = 0.74$ for the entire structure. The final structure is created by three successive 90°-rotations of the five-by-five base array of about the upper right corner (compare Fig. 5a). Introducing this four-fold symmetry significantly reduces the number of numerical simulations to construct the yield surface. The simulations are carried out using the software package Abaqus/Standard. The structures are discretized using approximately 37,500 linear, quadrilateral S4R-elements. The four basic unit cells are depicted in Fig. 6.



a) b)

Figure 5. Virtual experiments - a) Detailed FE model; b) 24 (σ_{11}, σ_{22})-pairs obtained from individual numerical simulations

In the following, we define the initial yield strength to be the corresponding stress values as the plastic dissipation reaches 10% of the elastic strain energy observed for the entire unit cell. Each yield surface is constructed based on 24 different loading scenarios (compare Fig. 6b). The linear, monotonic radial loadings are applied to the structures via periodic boundary conditions. The linear kinematic constraints between the control nodes and the created quadrilateral mesh allow to prescribe the macroscopic deformation gradient $\boldsymbol{F}$ to the unit cell. Imposing the symmetry of the deformation gradient, the Hencky strain tensor $\boldsymbol{\varepsilon}$ can be directly obtained from the input fields.

$$\boldsymbol{\varepsilon} = \frac{1}{2}\ln\left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}\right) \tag{1}$$

The components of the first Piola-Kirchhoff stress $\boldsymbol{P}^{\mathrm{I}}$ tensor can be directly extracted from the reaction forces at the control nodes. For convenience, we define a ratio between the normal strain components ε_{11} and ε_{22} to differentiate the different loading scenarios, in the following referred to as strain angle α.

$$\tan\alpha = \frac{\varepsilon_{22}}{\varepsilon_{11}} \tag{2}$$

Due to the introduced symmetry of the investigated structures, we only require 13 different strain angles to span the entire yield surface. Thus, we can limit the strain angle α to the interval $[\pi/4, 5\pi/4]$. These loading scenarios range from biaxial tension to biaxial compression covering the first to the third quadrant of figure 5b. Despite the four-fold rotation symmetry, the structures still exhibit an anisotropic response for angles between 0 and $\pi/2$. Therefore, we define a second measure, the material angle β, tracking deviation of the material orientation relative to the loading direction (Fig. 14). We construct an entirety of six different yield surfaces for one structure, using the material angles $\beta = \{0, \pi/12, \pi/6, \pi/4, \pi/3, 5\pi/12\}$. Considering both strain and material angles, this scheme leads to a total of 78 numerical simulations per structure.

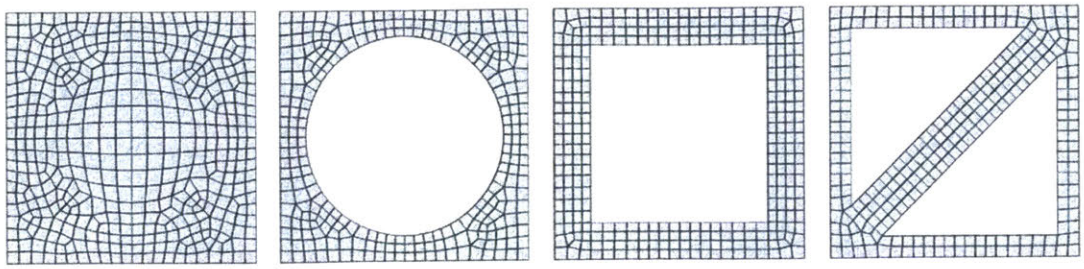


Figure 6. FE discretization of the four basic unit cells

2.2 Basis material

We have chosen the DP980 dual phase steel to function as the basis material for the architected structures. Rigorous investigations of Pack (2017) revealed that the anisotropy of DP980 as a sheet metal with a thickness of 0.82mm is almost negligible. Therefore, we use the isotropic von Mises yield criterion combined with an associated flow rule.

$$f[\boldsymbol{\sigma}, \bar{\varepsilon}_p] = \bar{\sigma} - k[\bar{\varepsilon}_p] = \sqrt{\frac{3}{2}\boldsymbol{\sigma}' : \boldsymbol{\sigma}'} - k[\bar{\varepsilon}_p] = 0 \tag{3}$$

The material resistance k only depends on the equivalent plastic strain $\bar{\varepsilon}_p$. The isotropic hardening behavior is defined using a weighted sum of a power law $k_s[\bar{\varepsilon}_p]$ (Swift, 1952) and an exponential law $k_v[\bar{\varepsilon}_p]$ (Voce, 1948).

$$\begin{aligned} k[\bar{\varepsilon}_p] &= \omega k_s[\bar{\varepsilon}_p] + (1-\omega) k_v[\bar{\varepsilon}_p] \\ &= \omega A(\varepsilon_0 + \bar{\varepsilon}_p)^n + (1-\omega)\left(k_0 + Q\left(1 - \exp\left(-b\bar{\varepsilon}_p\right)\right)\right) \end{aligned} \tag{4}$$

The calibrated material parameters are taken from Pack (2017). The resulting values are summarized in table 1.

Table 1. Calibrated parameters of the mixed Swift-Voce hardening law

ω [-]	A [MPa]	ε_0 [-]	n [-]	k_0 [MPa]	Q [MPa]	b [-]
0.58	1570	2.56E-05	0.129	548	531	92.8

2.3 Simulation post-processing: Determine initial yield strength

The resulting yield surfaces, as shown in Fig. 5b, are drawn using the Cauchy stress tensor. This decision has been made with regard to the future anticipation to be implement this scheme inside the framework of numerical simulations. However, the simulation results, as described in paragraph 2.1, will be obtained in terms of deformation gradient and first Piola-Kirchhoff tensor. Therefore, we convert the obtained field values to calculate the components of the Cauchy stress tensor

$$\boldsymbol{\sigma} = \det \boldsymbol{F}^{-1} \boldsymbol{F} \boldsymbol{P}^{\mathrm{I}}, \tag{5}$$

where det ... denotes the determinant of a tensor. In order to identify the thickness, normal strain ε_{33}, which is not part of our input, we monitor the Hencky strain in thickness-direction at each integration point. An effective measure of thickness strain is determined by area-averaging these strain components considering the current surface area of each FE-element. This effective strain is transformed to the engineering strain space and allows to complete the three-dimensional deformation gradient. Through this, all stress components at the onset of yielding, determined by the energy ratio discussed in the previous paragraph 2.1, can be obtained.

Chapter 3 Neural network architectures

The design of the neural network architecture deployed in the present work is inspired by the working principle of encoder-decoder networks. These network types are most commonly used for time-series input and thus predestinated for natural language processing tasks, like machine translation and speech recognition. The first encoder part of the network takes the inputs and transforms them into a vectorized feature map. This encoded hidden state is transferred into a decoder network. This second network use the encoded features and maps them onto the desired output. The present studies are carried out using a combination of a convolutional neural network and a feedforward neural network. The convolutional part is presented with the structural input and encodes the geometrical information into a hidden state. The feedforward neural network functions as a decoder network, translating the input vectors to the final network output.

3.1 Feedforward neural networks (FNN)

Mathematically, feedforwards neural networks can be expressed as vectorial non-linear function mapping N_{in}-dimensional input vector $\boldsymbol{x}$ to N_{out}-dimensional output vector $\boldsymbol{y}$:

$$f_{\text{FNN}} : \boldsymbol{x} \rightarrow \boldsymbol{y} . \tag{6}$$

The structure of a FNN model is composed of multiple hidden layers N_{HL}, each layer characterized by a hidden state vector $\boldsymbol{h}^{[i]}$ and potentially featuring a different number of neurons $N_{\text{NPL},i}$ (see Fig. 7a). The input vector $\boldsymbol{x}$ is sequentially passed through the network's hidden layers $\boldsymbol{h}^{[i]}$ until resulting in the desired output vector $\boldsymbol{y}$: $\boldsymbol{x} \rightarrow \boldsymbol{h}^{[1]} \rightarrow \boldsymbol{h}^{[2]} \rightarrow \ldots \rightarrow \boldsymbol{h}^{[N_{\text{HL}}]} \rightarrow \boldsymbol{y}$. The hidden state of the i-th layer is calculated by the following computation:

$$\boldsymbol{h}^{[i]} = g(\boldsymbol{W}^{[i]}\boldsymbol{h}^{[i-1]} + \boldsymbol{b}^{[i]}) \tag{7}$$

The trainable parameters of the FNN model are represented by the weight matrix $\boldsymbol{W}^{[i]}$ of dimension $N_{\mathrm{NPL},i} \times N_{\mathrm{NPL},i-1}$ and the bias vector $\boldsymbol{b}^{[i]}$ of dimension $N_{\mathrm{NPL},i}$. The function $g(...)$ is formally known as activation function and introduces the required non-linearity into the NN model. In the scope of this paper, we will have a closer look at four activation functions: hyperbolic tangent function (tanh), sigmoid function as well as rectified linear unit (ReLU).

The output layer follows a standard linear combination of activations of the activations of the last hidden layer. In other words, the output layer itself features a linear activation function:

$$\boldsymbol{y} = \boldsymbol{W}^{[N_{\mathrm{HL}}+1]} \boldsymbol{h}^{[N_{\mathrm{HL}}]} + \boldsymbol{b}^{[N_{\mathrm{HL}}+1]}. \tag{8}$$

Summarizing all described mathematical operations a FNN model results in the a total number of trainable variables $N_{\mathrm{trainable}}$.

$$N_{\mathrm{trainable}} = (N_{\mathrm{in}}+1)N_{\mathrm{NPL},1} + \sum_{i=2}^{N_{\mathrm{HL}}} (N_{\mathrm{NPL},i-1}+1)N_{\mathrm{NPL},i} + (N_{\mathrm{NPL},N_{\mathrm{HL}}}+1)N_{\mathrm{out}}. \tag{9}$$

3.2 Convolutional neural network (CNN)

The second basic neural network architecture we would like to rely on in the course of our investigations are convolutional neural networks. These networks have proven to be very efficient and reliable in detecting patterns and extracting feature maps of two-dimensional input data. Therefore, they are widely used in a variety of different image recognition tasks ranging from classical classification problems to object detection and localization as well as semantic segmentation.

The core building block of every CNN is based on the mathematically convolution operation. Hereby, a smaller weight matrix $\boldsymbol{W}$, known as filter or kernel, is used to transform the two- or three-dimensional array into a feature map (compare Fig. 7b). The modified discrete, two-dimensional version of the convolution operation for each element $h^{[i]}_{<m,n>}$ can be summarized as:

$$h^{[i]}_{<m,n>} = (\boldsymbol{h}^{[i-1]} * \boldsymbol{W}^{[i]})_{<m,n>} = \sum_q \sum_r W^{[i]}_{<q,r>} h^{[i-1]}_{<m-q,n-r>} . \tag{10}$$

Each element $h^{[i]}_{<m,n>}$ is determined by means of a double contraction between a filter $\boldsymbol{W}$ of size $f^{[i]} \times f^{[i]}$ and a subset of the input array or the previously obtained feature map at hand.

The output dimension $\hat{n}^{[i]}_{\text{out}}$ of a convolution operation is governed by two parameters, namely the stride $s^{[i]}$ and the padding $p^{[i]}$. The filter is shifted over the feature map during convolving. The stride determines how many units the filter is moved each time. Thus, it directly characterizes the overlap of two filter passes. The stride is comparable to the step size in the context of digital image correlation that determines the successive shift of the pixel subset ('filter'). The two most commonly used padding options are referred to as 'same' or 'valid'. A same convolution results in a feature map of the same dimension as the input. Since the input shrinks while performing the convolution operation the input has to be padded with additional boundary layer of zero values. In contrast, a valid convolution does not alter the respective input array. Consequently, the output dimension can be directly calculated based on filter size and stride, only. Using the floor function $\lfloor ... \rfloor$, the output size based on the two-dimensional convolution operation follows the equation:

$$\hat{n}^{[i]}_{\text{out}} = \left\lfloor \frac{\hat{n}^{[i]}_{\text{in}} + 2p^{[i]} - f^{[i]}}{s^{[i]}} + 1 \right\rfloor . \tag{11}$$

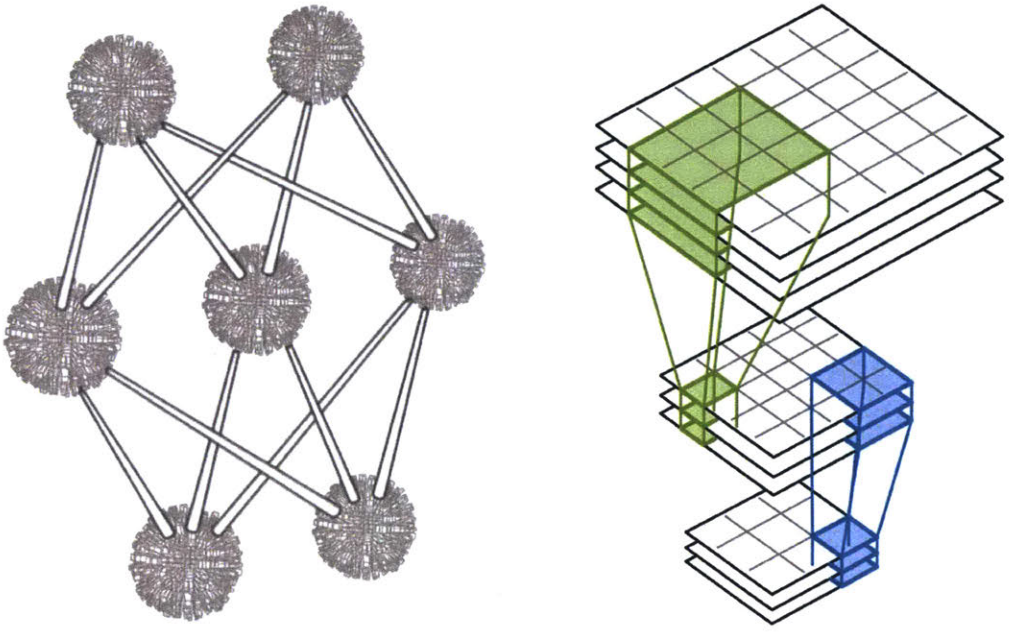
Similar to the hidden state in a FNN, the non-linearity is introduced by utilization of appropriate activation functions. According to a general notion in the pattern recognition and image processing community, we use ReLU activation functions for all convolutional layers as they proof to be extremely reliable and show a good performance in this framework. In the final network structure, we add one more level of complexity by performing the convolution operations over volume. A third dimension is added to weight matrices, referred to as channels $n^{[i]}_c$, and filters are assigned a certain depth $n^{[i]}_f$. Mathematically, this new, three-dimensional operation can be broken down into stacked two-dimensional convolutions as described above. In close analogy to a hidden layer

definition in FNN architectures, a bias vector $\boldsymbol{b}^{[i]}$ is added after performing the convolution operation with the dimension $n_f^{[i]}$. Summarizing the explanations given in the previous paragraphs, the total number of trainable variables for each convolutional layer adds up to:

$$N_{\text{trainable}}^{[i]} = n_f^{[i]} \times \left(f^{[i]} \times f^{[i]} \times n_c^{[i]} + 1 \right). \tag{12}$$

Note that the total number of parameters of a convolutional layer does not depend on the width and height of the previous layer and therefore size of the structural input.

This gives rise to one of the main advantages of CNNs over FNN for the task at hand. The architecture of CNNs allows to shift comparably small filters over larger input images and/or feature maps. This parameter sharing significantly reduces the total number of trainable parameters. In addition, each entry of the current convolutional layer only depends on a small number of pixels originating from the previous layer, whereas in a fully connected layer each neuron depends on each of the previous activations. Both features, parameter sharing as well as the sparsity of connections enable a good level of generalization and help to avoid overfitting.



a) b)

Figure 7. Schematic drawing of the internal connections for different NN architectures: a) feedforward neural network (FNN); b) convolutional neural network (CNN)

3.3 Model parameter identification (supervised learning)

In order to evaluate the performance of the neural network during the training phase we introduce the mean squared error (MSE) based on the difference between the targeted output values and the network's predictions.

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 \tag{13}$$

For each of the n training examples, we refer to the respective target value as $y^{(i)}$ and the current prediction by $\hat{y}^{(i)}$. The minimization of the loss function and incremental adjustments of the weights and biases is carried out using the Adam optimization algorithm (Kingma and Ba, 2015). Adam combines the ideas of momentum (Qian, 1999) and RMSprop (Hinton, 2012). The extension of the standard gradient descent with momentum helps to damp out oscillations in the parameter updates of the network relying on the exponentially weighted average of gradients to tune weights and biases. In contrast, RMSprop tunes the parameters with the help of an exponentially weighted average of the past squared gradients. Similarly, this helps to smooth the training process and generally allows to choose a larger step size to speed up the minimization scheme.

We would like to highlight two features that significantly contribute to further facilitate a successful training, namely the normalization of input data as well as the weight initialization. Both strain angles as well as material angles are transformed according to the Z-score normalization described by the following equation:

$$\bar{x}^{(i)} = \frac{x^{(i)} - \mu}{\sigma}. \tag{14}$$

Here, $x^{(i)}$ denotes a single example of the respective input feature, strain or material angle, and $\bar{x}^{(i)}$ its normalized value. In contrast, μ represents the mean value over all examples and σ corresponding the standard deviation. In case the input feature experiences a large standard deviation the normalized values will be projected to values close to zero, with a mean of zero and a standard deviation of one. The Z-score normalization transforms all input features to approximately the same scale putting equal importance and emphasis on all features helping with the gradient based optimization process.

Closely related to the normalization of the input features is the initialization of weights matrices. This helps to prevent vanishing or exploding gradients while passing the inputs through multiple neural network layers and to produce a meaningful output. A large number of commonly activation functions, like sigmoid or tanh, are symmetric. Furthermore, these functions asymptotic approach a certain threshold value for large or small input values. These properties entail a number of numerical problems and challenges. For input values in close proximity to the origin, these functions exhibit a linear behavior. Due to the loss of non-linearity in this range, the network degrades to a linear combination of input values and there is no use of concatenating multiple layers in the network anymore. On the other side, large as well as small input values to these activation functions leads to vanishing gradients which significantly slows down the parameter updates during training and worst case prevents the optimization from reaching a satisfactory minimum. In order to prevent the layer activations to take on unreasonably large or small values, we rely on the Xavier initialization scheme (Glorot and Bengio, 2010) for layers using symmetric activation functions. The initial values are drawn from a random uniform distribution U bounded by the values scaling with the size of the previous $n_{\text{in}}^{[i]}$ and current layer $n_{\text{out}}^{[i]}$:

$$W^{[i]} \sim U\left[-\sqrt{\frac{6}{n_{\text{in}}^{[i]}+n_{\text{out}}^{[i]}}}, \sqrt{\frac{6}{n_{\text{in}}^{[i]}+n_{\text{out}}^{[i]}}}\right]. \tag{15}$$

Resulting in a mean of zero and a standard deviation around one. For layers introducing the non-linearity based on non-symmetric activation functions, like ReLU, we utilize the Kaiming initialization (He et al., 2015). Here, the weight matrices are populated by random samples drawn from a normal distribution and each multiplied by $\sqrt{2/n_{\text{in}}^{[i]}}$. For a convolutional layer, we take the input size $n_{\text{in}}^{[i]}$ to be the product of the number of incoming channels $n_c^{[i]}$ and the square of the filter size $f^{[i]} \times f^{[i]}$. Both initialization schemes help to keep the standard deviation of the individual layer activations around one on average which significantly helps in the challenges discussed above.

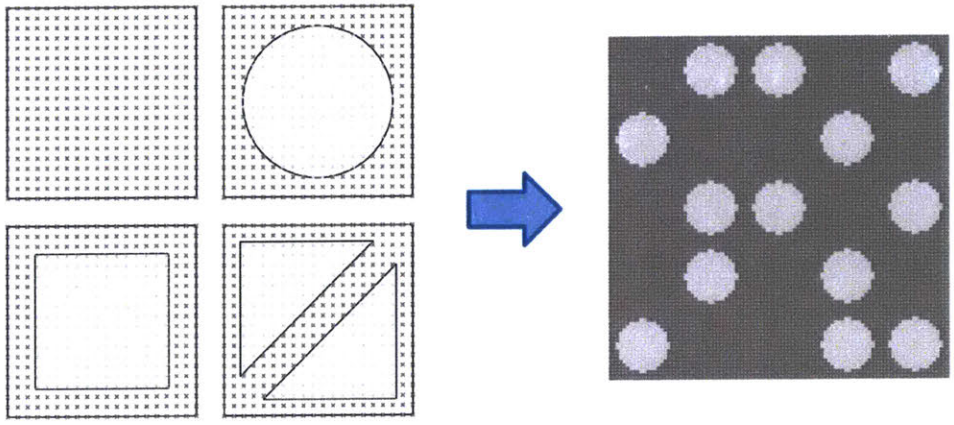
All of the features described above are implemented in python making use of the tensorflow library.

Chapter 4 Results and Discussion

4.1 Final network architecture

The final network architecture can be considered to act as a large black-box converting two dimensional structural data into pairs of normal stress components $(\sigma_{11}, \sigma_{22})$ characterizing the structure's global yield surface. This first paragraphs of chapter 5 focuses on the encoding of the input data as well as sheds some light on the techniques hidden inside the black-box structure that allow this architecture to produce accurate predictions.

Generally, as described in chapter 3.2, a two-dimensional array or matrix serves as input of a CNN. In order to distinguish the four basic unit cells, shown in figure 6, we introduce a 20-by-20 points grid of pixel values. This grid is fitted onto the elementary cells (see Fig. 8a). The dark grey points coincide with the base material. These pixels will be represented by a value of one in the final input array. The light greys points lie on top of void areas. Hence, the corresponding matrix values are chosen as minus one. To describe the complete structure of interest, we assemble the basic five-by-five base array (compare Fig. 5a). Thus, the final input of the neural network is a matrix of dimension 100-by-100. Figure 8b features an exemplary input using a round hole perforation on one of the random hole configurations.



a) b)

Figure 8. Mathematical encoding of the structural information in a two-dimensional matrix of zeros and ones (blue represents pixel values of one; red depicts values of minus one)

As indicated in previous paragraphs, the layout of the neural network relies on two parts - a CNN-based encoder and a subsequent FNN-based decoder. Figure 9 shows the final computational framework as well as summarizes the detailed values of some of the underlying hyperparameters like filter sizes, filter depths, strides, activations functions and hidden layer sizes.

The encoder part consists of two convolutional layers, which are separated by one maximum pooling layer. The first convolutional layer features a rather large filter size of 8-by-8. In combination with a valid padding, this filter extracts key features of the geometrical structures. Using a stride of four, the filter is slowly shifted across the input image to gather every detail of the input structure as the transformation of the first convolutional layer is crucial to later distinguish different hole types. The following max pooling layer does not feature any additional trainable parameters but significantly reduces the dimension of the encoded feature map from an image of 24-by-24(-by-8) pixels to 11-by-11(-by-8). Similar to convolutional layers, pooling layers are also characterized by filter size, stride and padding. However in case of a max pooling, for each pass the corresponding component of the output is selected based on the maximum value within the filter's boundaries. The subsequent, second convolutional layer turns the feature map into a matrix of size 5-by-5(-by-4). Both convolutional layers rely on ReLU based on the notions mentioned in chapter 3.2. After the first processing through convolutional layers, the obtained feature map is flattened to a one-dimensional array of length 100. The last layer of the encoder part of the framework is a fully-connected layer with 2 neurons. In contrast to the first layers, the fully-connected layer is completed by a symmetric sigmoid activation function, putting equal emphasis on positive and negative pixel values. The output of this layer is considered to represent the compressed version of all information hidden in the geometrical input. In other words, all geometrical as well as mechanically significant information describing the entirety of the structures at hand as well as the differences in mechanical behavior is contained in the combination of two scalar values, the outputs of this first CNN-based neural network.

Before processing the compressed data by the FNN, we introduce the strain and material angle to the modelling framework. For this purpose, we concatenate the geometrical data with the two angles, yielding a four-dimensional vector. Both, strain and

material angle are considered to be *internal variables*. Unlike, the geometrical information which are different for every structure, the strain and material angles stay constant for each structure. Therefore, these two quantities are not considered to be part of the actual input to the overall model and consequentially labeled as *internal variables*. The core of the FNN decoder network is composed of three hidden layers featuring 20 neurons each. The mathematical formulation works based on the principles explained in chapter 3.1. Once more, the activations of the fully-connected layers are generated using the sigmoid activation function. The last layer reduces the 20-dimensional array carried through the FNN to the output pair of normal stress components $(\sigma_{11}, \sigma_{22})$. Additionally, this output layer is linear. That is, it does not introduce any further non-linearities or dilatations to the output vector. This overall size of the FNN-based part of the modelling framework might seem large for the task at hand. Indeed, a smaller FNN decoder does achieve comparable results in terms of MSE and visual comparison of the prediction initial yield surfaces. However, the explained network architecture has been chosen in regard to future extensions to this framework as well as the enrichment of the input by a variety of additional, different structures. On top of that, even the larger FNN decoder, used to create the predictions that will be presented in the following paragraphs, does not show signs of overfitting if the remaining hyperparameters targeting the learning procedure itself are chosen appropriately.

For a more comprehensive study of the network's architecture and the hyperparameters aiding in the overall training procedure the reader may refer to section 4.4. These paragraphs will elaborate on the thought process that went into creating the network structure and setting its parameters. The following subchapters are dedicated to the demonstration of different aspects and capabilities of the fully-trained framework. The training of these particular results was performed based on 14 different hole configuration for each of the three different hole types – using 13 strain angles and six material angles each. Here, the 14 different hole configurations have not been chosen randomly, but they have been handpicked out of a pool of 1000 different instances. The selected configurations represent the largest variety of possible stress values and different degrees of anisotropy with regard to both the shape of the individual yield surfaces themselves as well as the sensitivity to the material angle. At this point, the author would like to kindly emphasize once more that all predictions are based on geometrical information only.

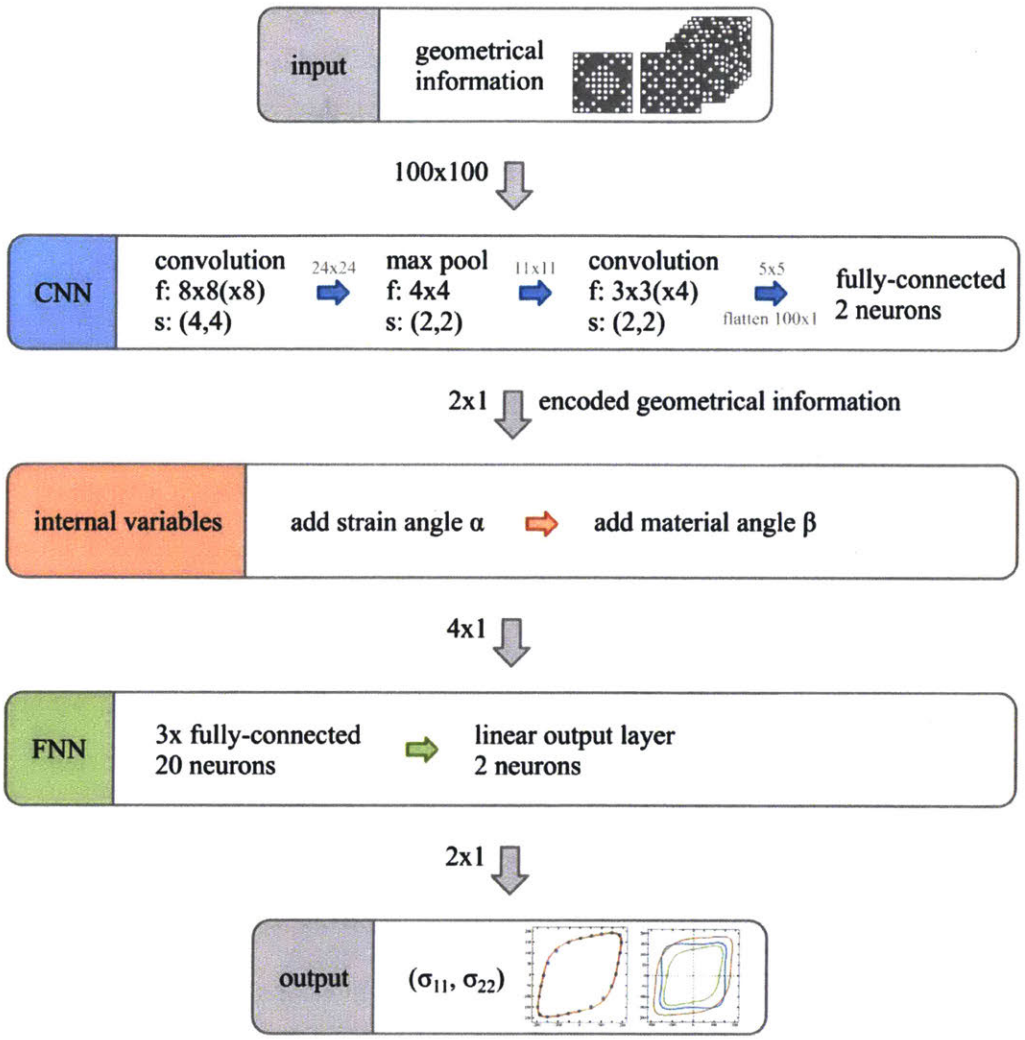


Figure 9. Schematic illustration of final neural network based modelling framework

4.2 Model's predictive capabilities: Degrees of anisotropy

The following lines aim to showcase two aspects of the model's predictive capabilities. On the one hand, the network is able to accurately predict the influence of the hole type, circular-, square- as well as triangular-shaped perforations. On the other hand, the network is capable of reproducing the influence of the different hole configurations. During all

considerations in this chapter, the material angle stays constant. The material anisotropy will be subject of the investigations of the chapter 4.3.

Figure 10 shows three different initial yield surfaces in σ_{11}-σ_{22} - space corresponding to the different perforation types - circular (a), square (b) and triangular (c) - of one specific hole configuration (in the following referred to as configuration 900). The individual points in the first to the third quadrant correspond to the 13 strain angles used during the training procedure of the network. Due to material symmetry as well as loading scenarios, these points can be used to complete the yield surfaces that now consist of 24 individual points, respectively. The comparison of the three graphs highlights the differing magnitudes of the yield contours. Both the circular as well as the square-holed structure reach maximum stress component values around 200 MPa. Whereas, the triangular perforation features a maximum component value at around 170 MPa, about 15% lower compared to the other hole types. Another major characteristic is the difference in shape of the resulting yield surfaces. Considering configuration 900, the triangular perforation yields the most narrow envelope. As for all surfaces, the structure exhibits the highest load bearing capacity until initial yield under equibiaxial straining. Whereas, the performance significantly decreases to only 71% compared to its equibiaxial equivalent under confined tension and to roughly 52% under pure shear loading. The square-holed structure shows the least variation due to a change in the loading scenario. Generally, the circular perforation lies in-between these two extremes. However, a closer inspection reveals an additional peculiarity of this yield surface. While moving along the envelope starting from biaxial straining via confined tension towards a pure shear loading scenario we observe a change in curvature. Based on the chosen criteria of initial yield of the architected two-dimensional structure, the obtained surface does not exhibit convexity. Despite all of these inherent differences and subtleties, the NN-based computational framework succeeds in predicting all of these diverse aspects. In order to compare the network's predictions and the results of the numerical simulations, Fig. 10 features both datasets. The large circles depict numerical simulations, whereas the small circles represent the network's predictions. For all graphs, every of these corresponding pairs of numerical simulation and network's predictions is in close agreement, which is evidence of the good network performance. The color-coding used in Fig. 10 gives more insight into the training procedure of the free network parameters. The

color blue indicates that these points belong to the training dataset. In contrast, the green and red points are part of the validation and testing dataset, respectively. This implies that only 80% of the entire dataset (blue points) is used for determining the weights and biases of the neural network as validation and testing dataset are excluded from gradient calculations during the training procedure itself. As all of these points are randomly assigned to training, validation or testing data, each structure features a different composition of points from these three datasets. Thus, for one structure, all or rather most of the 13 points might be used for training or as it is the case for the triangular perforation many are solely used for validation and testing purposes. Hence, these points are results of the network's generalization and illustrate once more its capabilities.

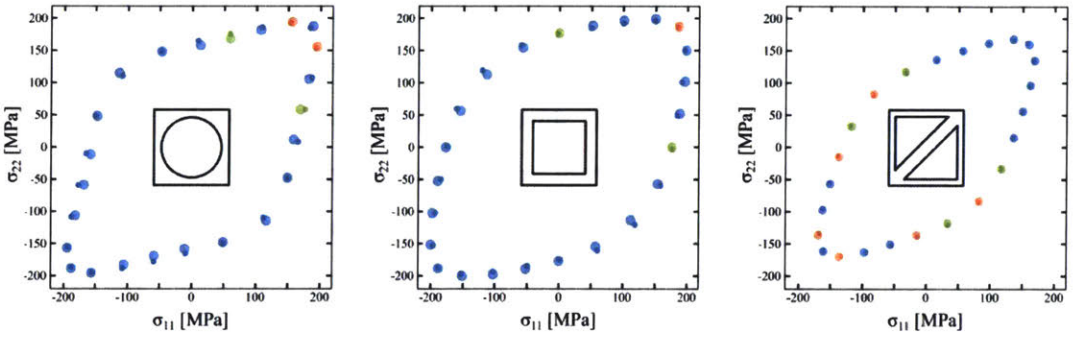
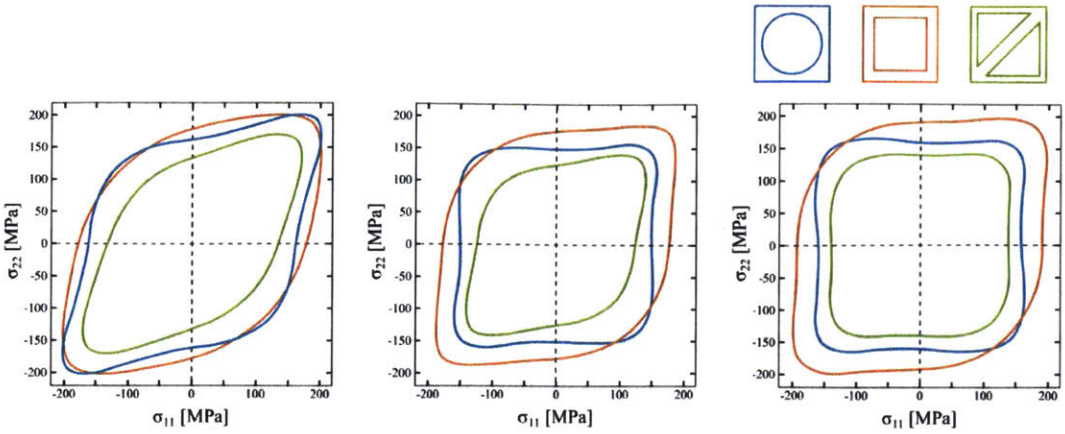


Figure 10. Model's predictive capabilities: Influence of hole type: a) circular, b) square and c) triangular perforation of hole configuration 900 (large circles depict the numerical simulations, small circles show the network's predictions; colors represent training (blue), validation (green) and testing dataset (red))

On top of the influence of the individual hole types, the network possess the agility to predict the influence of different hole configurations on the mechanical response of the created architected material. Figure 11 displays the yield surfaces of three different structures (from left to right: structure 900, 964 and 300). Each of the subplots contains the RVE material response of all three hole types considered in the present study. Note, that these curves are not based on any interpolation scheme based on the 13 points highlighted in e.g. Fig. 10 but correspond to the direct, smooth network output. The left subplot shows

structure 900, which is subject of discussion in the previous paragraph and hence serves as comparison. Whereas the yield surfaces of structure 900 are of a particularly sharp nature, resembling an almost rhombic shape, the envelopes of structure 964 and especially structure 300 tend to be square-shaped. Figure 11 implicitly contains three major challenges that have to be handle by the neural network at the same time. Firstly, there exists a notable difference in magnitude for the different hole configurations. Whereas as simple scaling of the yield envelopes could be taken care of by two scalar factors, the envelopes additionally exhibit a change in their overall shape. This observation goes hand in hand with the next and second challenge. The comparison of the graphs of the circular perforation reveals a transformation from a rhombic (structure 900) to a square-shaped surface (structure 964 and 300) only by varying the hole configuration. Lastly, but rather the most delicate and demanding nuance, the distinct characteristics of each structure upon combining both of the previously mentioned differences. In case of structure 900, all yield surfaces tend towards a rhombic shape. In contrast, structure 964 exhibits a variation from rhombic to square shaped yield surfaces. Additionally, the specificities of this variation might be different from structure to structure. While structure 964 features a rhombic-shaped envelope in case of a triangular perforation, the graph's shape for structure 300 shifts towards a square-type of yield surface.

Despite this multitude of different facets and subtleties, the neural network based framework has the capability to accurately reproduce any shift and transformation of the yield surfaces with regard to changes in magnitude as well as shape.



a) structure 900 b) structure 964 c) structure 300

Figure 11. Model's predictive capabilities: Influence of hole configurations: a) structure 900, b) structure 964 and c) structure 300. Solid lines represent the network's predictions

4.3 Model's predictive capabilities: Material anisotropy

The investigations of the preceding section 5.2 did not incorporate any considerations in regard to material anisotropy. Therefore, the following analysis is targeted at the examination of the material angle and its influence on the effective material response.

The fundamental training of the neural network structure has been carried out using six distinct material angles $\beta = \{0, \pi/12, \pi/6, \pi/4, \pi/3, 5\pi/12\}$. The graphs in Fig. 12 summarize the results for structure 946 starting from $\beta = 0$ at the top left corner to $\beta = 5\pi/12$ at the bottom right corner. The rotated structure in the center of each plot schematically shows the current material orientation for each individual graph. Analogous to Fig. 10 each diagram includes 24 data points based on numerical simulations in blue as well as the network's predictions in orange. Each of the six network's predictions is densely sampled and comprised of 400 individual points. One major achievement of the modelling framework is the smooth output in all cases. For all material angles, the predictions are in very good agreement with the numerical counterparts. The model is able to predict the continuous change in shape of the yield surfaces as the material angle β progresses from

$\beta = 0$ to $\beta = \pi / 2$. Furthermore, it is capable to capture the change in curvature of the yield envelope when necessary (e.g. $\beta = \pi / 6$ at the top right corner).

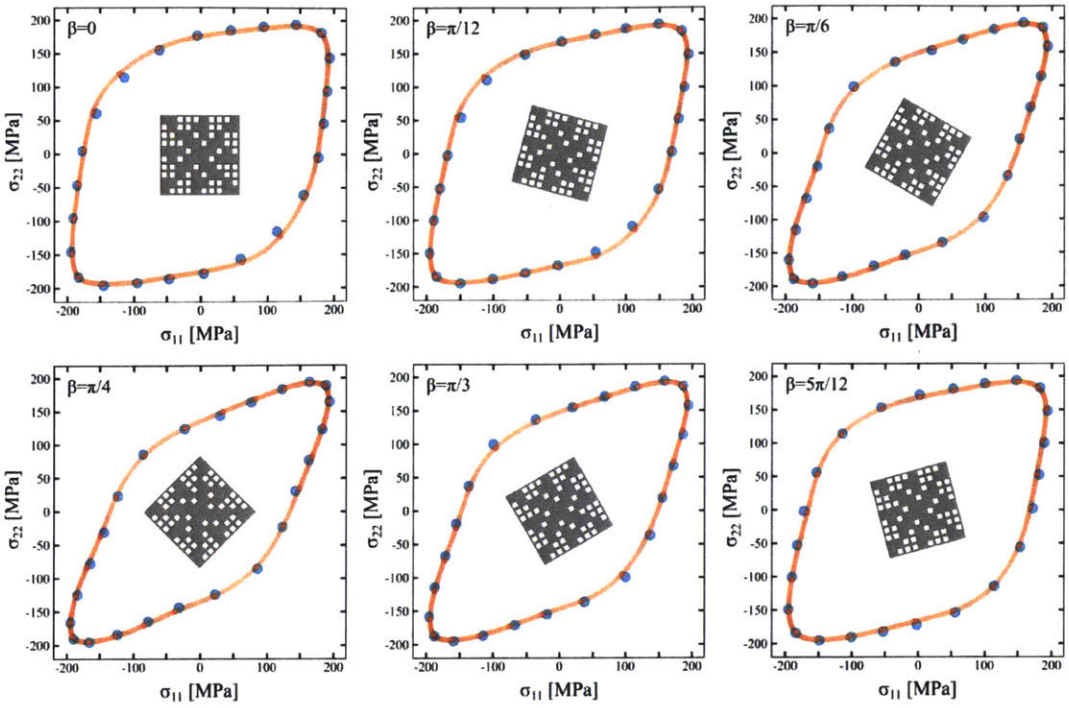
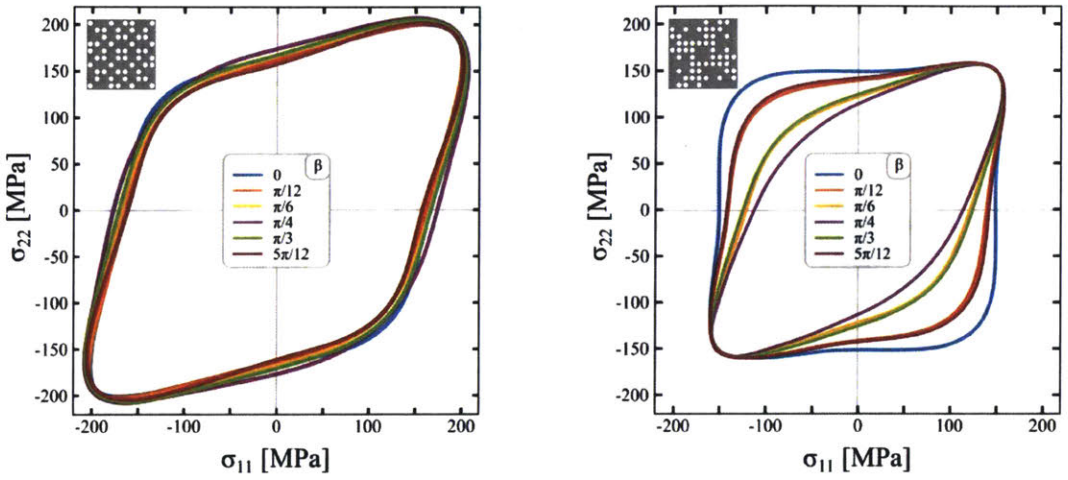


Figure 12. Model's predictive capabilities: Material anisotropy. Continuous shape change of the yield surfaces of structure 946

While this first glance at the effect of the material orientation implies a relatively simple shape-change of the yield surfaces, the comparison of various hole configurations reveals an additional level of complexity. Figure 13 depicts the effective material response of structure 900 on the left ad structure 964 in the right hand side. Both structures feature a circular perforation and are shown for all six material angles. Structure 900 on the left features a very homogenous perforation of the base material. Consequently, the effective material response appears to be virtually isotropic. The envelopes of all six different material orientations lie almost on top of each other. On the other hand, structure 964 depicted on the right exhibits an even more pronounced anisotropic material response compared to structure 946 shown in Fig. 12. While a material angle of $\beta = \pi / 4$ results in a narrow, rhombic yield surface, the yield surface gradually adjusts to a square shape as

the material angle approaches $\beta = 0$ and $\beta = \pi / 2$, respectively. In other words, the performance under a pure shear loading scenario successively increases as the material angle converges to $\beta = 0$ as well as $\beta = \pi / 2$. The comparison of the both structures' material responses highlights the different degrees of anisotropy. Both extreme cases as well as all intermediate forms and combinations occurring for various hole configurations and perforation types are accurately handled by the neural network.



a) structure 900 b) structure 964

Figure 13. Model's predictive capabilities: Varying degrees of material anisotropy for different structures. Solid lines represent the network's predictions

As stated before, the smooth predictions for arbitrary strain angles make this modelling framework very attractive. On top of that, the network manages to give accurate results for unseen material angles. These additional material angles were not part of either training nor validation or testing dataset during parameter identification. Figure 14 illustrates this capability with the help of structure 964. The plot on the left hand side depicts the curves based on the original six material angles. The right hand side shows nine different material angles in the interval $\beta = [0, \pi / 2[$. Note, that this request of additional material angles is simply reflected in a change of the internal variables within the computational framework and can be adjusted arbitrarily. That is, any desired number of material angles can be

retrieved whilst the actual input to the network remains unchanged. Once more, the network provides accurate, smooth predictions and a seamless transition between the different material angles even for unseen material angles within the original training interval.

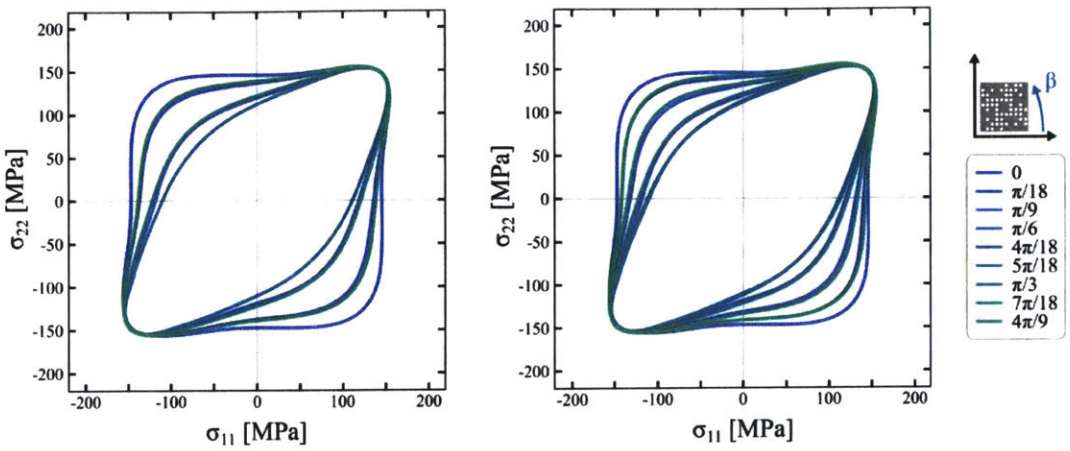


Figure 14. Model's predictive capabilities: Network's prediction for unseen material angles

4.4 Remarks on training the CNN based architecture

So far, the study only showcased the model's capabilities with the help of polished results during the previous chapters and did not elaborate on the thought process that went into creating the final network architecture. Thus, the following paragraphs will give more insight into selected aspects of the framework as well as the comprehensive training procedure allowing for a successful parameter identification.

At such an early stage of research into the use of convolutional neural networks in mechanics, it is difficult to draw general conclusions based on the conducted investigations. However, the present study and results provide enough evidence to give first tentative guidance and allow the author to share a collection of first hunches. One of the key aspects to a fruitful application of neural network based modelling frameworks is to provide clean

input data that can later be successfully interpreted by the machine learning algorithm. The individual examples have to be distinct and contain just the right amount of independent information in order to be correctly processed, either be means of classification or regression. This very general notion is interlinked with the choice concerning the filter of the first convolutional layer. Here, the filter size of 8-by-8 has been chosen with respect to the exact input of the network, more precisely the network's task to accurately distinguish between the elementary unit cells (see Fig. 6). In case the filter size is set out to be too large, the filter evaluation would inevitably smooth out the differences between the three perforations. Consequentially, the different types of perforations would not be distinguishable by the final network architecture. In contrast, a very small filter size in connection with the currently chosen representation of the geometrical information many filter passes would be identical comparing different hole configurations as well as hole types. Hence, this is expected to make training significantly more difficult because it would involve an additional sorting of relevant and irrelevant information.

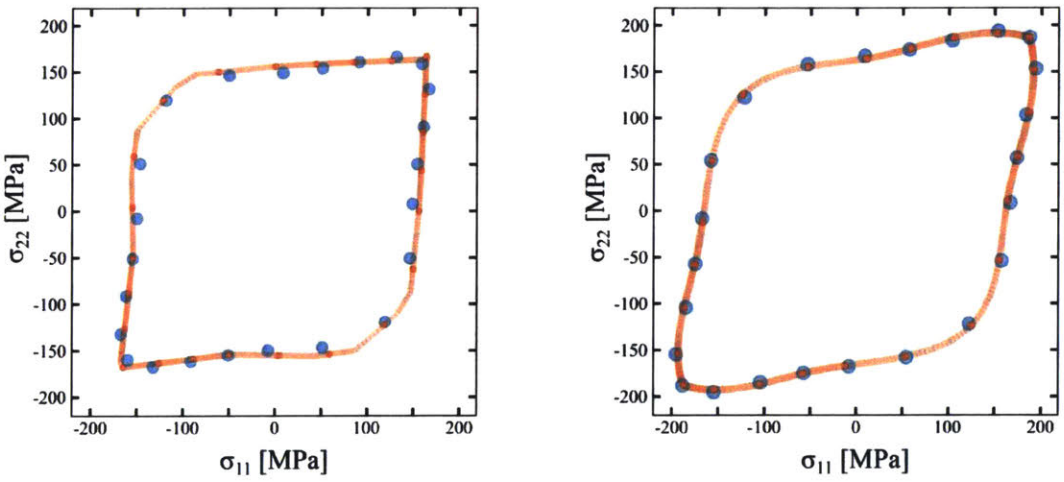
Training a neural network is equivalent to a classical mathematical optimization problem aiming at minimizing the cost function. In most practical applications, this problem tends to be ill-defined that is the optimization landscape exhibits a multitude of local minima that make it very difficult to identify suitable weights and biases which is crucial for the networks later performance. In certain ways, you could consider the decoder part of the modelling framework to be a single, isolated FNN featuring its own set of input and output variables. This is of special importance since the input of the decoder network combines the compressed geometrical information as well as the introduced internal variables. As mentioned in the previous sections of chapter 4, two scalar variables seem to be sufficient to incorporate all geometrically and mechanically significant information to characterize and distinguish all considered structures. Another notion concerning the choice of exactly two scalar values as geometrical representations focuses around creating a well-defined input to the decoder network. This input consists of a concatenation of two structural and two internal variables. Both aspect are equally important for determining the effective material response. Therefore, instead of compressing the geometrical information in a larger number, e.g. eight, parameters the author specifically chose two scalars in order to provide a well-balanced internal input after introducing the internal quantities. Overall,

this architecture puts equal emphasis on the geometrical information identifying the individual structure as well as on the strain and material angle defining the exact loading scenario and material orientation. This idea is somewhat comparable to the well-established concept of input normalization. By transforming the individual inputs onto the same numerical interval, the normalization helps to allow for equal importance of every input feature and every pixel of the input image, which contributes towards a successful parameter identification.

At this points, the author would like to emphasis once more that the preceding paragraphs are by no means meant to represent general statements. Nevertheless, they illustrate tentative hunches that considerably helped to construct and identify a suitable network for estimating structure-property relationships in this present study.

During the making of the final neural network architecture, the author considered a variety of different activation functions for the individual layers. In general, activation functions belong to the few possibilities to introduce non-linearities to neural network based architectures. Without any form of non-linearity, the modelling framework comes down to an architecture of nested linear combinations based on the input features. As stated in section 3.1, four different functions, namely the hyperbolic tangent function (tanh), the sigmoid function as well as the rectified linear unit (ReLU), were explored. As a general notion and based on successful implementation in a variety of different use-cases, ReLU are the classical choice for convolutional layers. Since this choice has proven to be very effective in this present study as well, every convolutional layer relies on ReLU activations. However, the later selection of an activation function for the fully-connected layers has a direct impact on the shape of the predicted yield surfaces. Figure 15 shows a comparison of two possible shapes that are adopted based on the choice of activation functions. The graph on the left side depicts the results originating from the use of ReLU. In contrast, the envelope on the right hand side is generated with the help of sigmoid or tanh activation functions. A neural network architecture that is solely comprised of ReLU activation functions yields a piecewise linear approximation of the target values. The left diagram in Fig. 15 shows the predicted yield surface of structure 958 in orange on top of the numerical simulations in blue. Due to the use of ReLU, the orange envelope is constructed by a series of linear elements that bears a great resemblance to a linear interpolation of the 24 target

data points. In contrast, the use of sigmoid or tanh functions allows for a yield surface that nicely adapts to the specific contour of the underlying numerical simulations, making it possible to recreate any convex or non-convex shape. In this particular setting, considering input representation and normalization as well as initialization of the network's parameters and network architecture, the sigmoid activation function worked best as it was a little bit easier to train the final network. Regardless, to the best of the author's knowledge there is no clear and definite reason as to why the sigmoid function should outperform tanh. Additionally, the latter is equally successful in delivering accurate predictions when considering a fully-trained neural network. However, due to the fact that the sigmoid function does allow for an easier and simpler model parameter identification, all fully-connected layers are based on sigmoid activations.



a) ReLU - structure 958

b) sigmoid/ tanh - structure 74

Figure 15. Influence of the activation functions on the resulting shape of the yield surfaces

In order to identify suitable weights and biases a rigorous training scheme is employed. The results of the parameter identification of one single neural network architecture are summarized in Fig. 16. The diagram depicts the MSE for numerous optimization runs carried out using different sets of hyperparameters. Every optimization has been performed using the Adam optimization algorithm. Additionally, every set of hyperparameters has been initialized and fully-trained five times. For each set, only the best out of those five

runs is shown in Fig. 16. The MSE is reported separately for training (blue), validation (orange) and testing dataset (green). Generally, the validation and testing errors are higher than the training errors. This is to be expected since both datasets are excluded from the underlying optimization procedure. Nevertheless, all three error measures for each set of hyperparameters lie within close proximity to each other which is a first indication of good generalization capabilities of the trained network. Along the abscissa, there are 36 different combinations of different initial learning rates and batch sizes, the number of examples fed to the network during one gradient update. The learning rate has been gradually varied between 0.05 and 0.001, whereas the batch size takes values ranging from 24 to 1024. These are just two of the tunable hyperparameters. However, the neural networks are most sensitive to these two parameters (alongside with the general structure, i.e. number of layers as well as layer sizes. In case of this specific network, a small initial learning rate in combination with a small batch size yielded the lowest MSE. Nevertheless, this does not hold true for every network. Hence, it is necessary and essential to subject every new architecture to this rigorous and comprehensive training procedure in order to search for the best possible distribution of weights and biases. This approach allows to effectively, yet tentatively, compare the performance of different neural network architectures and to ultimately identify the structure best suited for the task at hand.

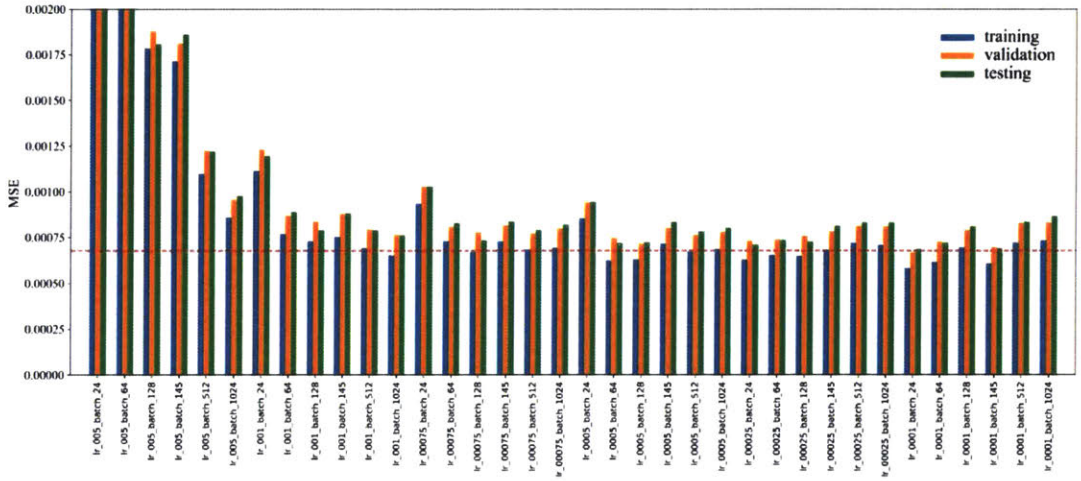


Figure 16. Parameter identification using different sets of hyperparameters. (depicted values correspond to the best run out of five random initializations of the neural network)

Chapter 5 Conclusions

During the past years, neural networks drew more and more interest from the metal forming community. They have proven to provide an effective and powerful framework, in particular for constitutive modeling. Hence, they are certainly able to compete with traditional physics-based models. This work takes first tentative steps towards the use of convolutional neural network based modelling techniques to estimate structure-property relationships.

The presented investigations clearly demonstrate the capabilities of such architectures to directly translate geometrical data into mechanically significant quantities. In future applications, the computational framework will be trained based on physical experiments. In the course of this work, they have been replaced by virtual experiments relying on numerical simulations. After successful training, i.e. parameter identification of the neural network, the model manages to replicate the initial yielding of architected porous materials based on geometrical information. The two-material structures differ in different perforation types as well as perforation layouts. Additionally, a change in loading direction leads to an increased level of complexity and require an even greater flexibility of the neural network. All of these factors culminate in numerous augmentations of the yield surfaces. These include but are not limited to magnifications, shape distortions as well as varying degrees of anisotropy. On top of that, all of these specificities are of a highly non-linear nature and unique to each structure, which creates a very intricate problem on its own. Despite this challenging task, the created modelling framework is capable of characterizing and distinguishing the individual structures and predicting their anisotropic material responses. The fact that the network solely relies on structural data as well as the network's encoder-decoder architecture strongly suggest that all geometrical and mechanically significant information can be stored in a small number of scalar variables.

In the future, the outlined approach can easily be augmented to other geometrical structures, extended to three dimensional considerations or used to predict a vast variety of diverse mechanical properties. In addition to the presented architected metamaterials, the modelling framework can be applied to particle and fiber composites as well as to interpret

microstructural information of metallic materials or at component level to model the effective material response of lithium-ion batteries.

Bibliography

Baxter, S., Hossain, M., & Graham, L. (2001). Micromechanics based random material property fields for particulate reinforced composites. *International Journal for Numerical Methods in Engineering, 38*, 9209-9220.

Bonatti, C., & Mohr, D. (2017). Large deformation response of additively-manufactured FCC metamaterials: From octet truss lattices towards continuous shell mesostructures. *International Journal of Plasticity, 92*, 122-147.

Byström, J. (2003). Influence of the inclusions distribution on the effective properties of heterogeneous media. *Composites Part B: Engineering, 34 (7)*, 587-592.

Christenson, R. (1986). Mechanics of low density materials. *Journal of the Mechanics and Physics of Solids 34*, 563-578.

Drugan, W., & Willis, J. (1996). A micromechanics-based nonlocal constitutive equation and estimates of representative volume element size for elastic composites. *Journal of Mechanics and Physics of Solids, 44 (4)*, 497-524.

Gibson, L., & Ashby, M. (1982). The mechanics of three-dimensional cellular materials. *Proceedings of the Royal Society of London A382*, 43-59.

Gibson, L., & Ashby, M. (1988). Cellular Solids: Structure and Properties. *Oxford: Pergamon Press*.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR 9*, 249-256.

Gusev, A. (1997). Representative volume element size for elastic composites: a numerical study. *Journal of the Mechanics and Physics of Solids, 45 (9)*, 1449-1459.

Halpin, J., & Tsai, S. (1967). *Effect of environmental factors on composite materials. Air Force Technical Report AFML-TR 67-423.* Dayton, OH: Wright Aeronautical Laboratories.

Hashin, Z., & S., S. (1963). A variational approach to the theory of composite elastic materials. *Journal of the Mechanics and Physics of Solids 11(2)*, 127-140.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *Published in Proceedings of IEEE International Conference on Computer Vision, arXiv preprint https://arxiv.org/abs/1502.01852.*

He, Z., Li, G., Zhong, Z., Cheng, A., Zhang, G. Y., Liu, G. R., . . . Zhou, Z. (2013). An edge-based smoothed tetrahedron finite element method (ES-T-FEM) for 3D static and dynamic problems. *Computational Mechanics 52(1)*, 221-236.

Hinton, G. (2012). Lecture 6.5 - Rmsprop: normalize the gradient. *Coursera online class - Neural Networks for Machine Learning*, University of Toronto.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *Published in Proceedings of 3rd International Conference for Learning Representations.* arXiv preprint https://arxiv.org/abs/1412.6980.

Kröner, E. (1977). Bounds for effective elastic moduli of disordered materials. *Journal of the Mechanics and Physics of Solids 25*, 137–155.

Liu, G. R. (2010). A G space theory and a weakened weak (W2) form for a unified formulation of compatible and incompatible methods: Part I theory. *International Journal for Numerical Methods in Engineering, 81*, 1093-1126.

Liu, G. R. (2002). *Meshfree methods: Moving beyond the Finite Element Method.* Boca Raton: CRC Press.

Mori, T., & Tanaka, K. (1973). Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta Metallurgica 231*, 571-574.

Ostoja-Starzewski, M. (2429-2455). Random field models of heterogeneous materials. *International Journal of Solids and Structures, 35 (19)*, 1998.

Pack, K. (2017). *Modeling of sheet metal fracture for shell finite elements with component level validation.* PhD thesis. Boston: Massachusetts Institute of Technology.

Qian, N. (145–151). On the momentum term in gradient descent learning algorithms. *Neural networks, 12(1)*, 1999.

Reuss, A. (1929). Berechnung der Fließgrenze von Mischkristallen auf Grund der Plastizitätbedingung für Einkristalle. *Journal of Applied Mathematics and Mechanics 9*, 49-58.

Suquet, P. (1987). Elements of homogenization theory for inelastic solid mechanics. In E. Sanchez-Palencia, & A. Zaoui, *Homogenization techniques for composite media* (pp. 194-278). Berlin, Heidelberg, New York: Springer.

Swift, H. (1952). Plastic instability under plane stress. *Journal of the Mechanics and Physics of Solids, 1 (1)*, 1-18.

Tancogne-Dejean, T., Diamantopoulou, M., Gorji, M., Bonatti, C., & Mohr, D. (2018). 3D Plate-Lattices: An Emerging Class of Low-Density Metamaterial Exhibiting Optimal Isotropic Stiffness. *Advanced Materials, 30*(45), 2018.

Trias, D., Costa, J., Mayugo, J., & Hurtado, J. (2016). Random models versus periodic models for fibre reinforced composites. *Computational Materials Science, 38 (2)*, 316-324.

Voce, E. (1948). The Relationship between Stress and Strain for Homogeneous Deformation. *Journal of the Institute of Metals, 74*, 537-562.

Voigt, W. (1889). Über die Beziehung zwischen den beiden Elastizität isotroper Körper. *Wied. Arm. 38*, 573-587.

Willis, J. R. (1977). Bounds and self-consistent estimates for the overall properties of anisotropic composites. *Journal of the Mechanics and Physics of Solids 25*, 185-202.