A mixture of experts approach to handle ambiguities in parameter identification problems in material modeling

Lukas Morand[a,*], Dirk Helm[a]

[a] *Fraunhofer Institute for Mechanics of Materials IWM, Wöhlerstraße 11, Freiburg 79108, Germany*

Abstract

To simulate the mechanical behavior of a material, it is essential to calibrate the internal parameters of the used material model to experimental measurements. This is typically done in a trail-and-error approach by hand or automatically using optimization algorithms. As an alternative to trial-and-error, neural network-based approaches can be used to imitate the inverse mapping. This is usually realized in a grey-box model, combining neural networks, deterministic models, and domain knowledge. However, the proposed neural network-based approaches found in literature do not address the challenge that is posed when the parameter identification problem is non-unique. In the present paper this problem is discussed and an improved approach is proposed using a mixture of experts model. Mixture of experts is an ensemble technique based on a dynamically structured framework of submodels aiming to partition the non-unique problem into unique subtasks.

Keywords: Grey-box model, direct inverse model, inverse problem, machine learning, material modeling, mixture of experts, neural networks, parameter identification

1. Introduction

Mechanical problems of deformable solids are characterized by universal relations for kinematics and balance equations of mass, linear momentum, and angular momentum. To close the set of equations, constitutive relations are

*Corresponding author

Email address: lukas.morand@iwm.fraunhofer.de (Lukas Morand)

required to represent stress-strain relationships. These constitutive relations form the material model. The material model can be seen as an operator $\boldsymbol{g}$ describing the stress response $\boldsymbol{\sigma}$ for an applied strain $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\sigma} = \boldsymbol{g}(\boldsymbol{\varepsilon}, \boldsymbol{m}). \tag{1}$$

In order to represent the material of interest, the internal parameters of the material model have to be calibrated to experimental measurements. The aim of this parameter identification problem is to minimize a certain misfit measure between the stress response of the material model and an experimentally measured stress $\boldsymbol{\sigma}_{\text{exp}}$:

$$\boldsymbol{m} = \text{argmin}_{\boldsymbol{m}} \left(||\boldsymbol{\sigma}_{\text{exp}} - \boldsymbol{g}(\boldsymbol{\varepsilon}, \boldsymbol{m})||_{\mathcal{L}^p} \right), \tag{2}$$

where $\mathcal{L}^p$ defines the norm.

Solving such problems is everyday work for engineers, which is why lots of solution approaches exist [1]. In the field of mechanical engineering, mostly optimization algorithms are used [2, 3]. Optimization algorithms perform a search in the space of material model parameters aiming to minimize the misfit measure that is for example described in Eq. (2). The success of optimization algorithms is highly depending on the chosen starting point, which is usually not known in advance. To find appropriate parameters for complex material models, typically many iterations are needed and the computational costs increase. As a consequence, engineers often have to use non-optimal parameters.

In contrast to the iterative procedure of optimization algorithms, neural network-based models can be used to directly estimate parameters that fit the material model to experimental data. In general, neural networks are mathematical models that approximate functions for given training data [4, 5]. The training data is generated by the material model itself and consists of material model parameters and their corresponding responses. By training a neural network-based model on the reversed input/output relations, a so-called direct inverse model is set up [6]. The direct inverse model can be described by an operator $\boldsymbol{G}$ that maps the space of material model responses to parameter space:

$$\boldsymbol{m} = \boldsymbol{G}(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}). \tag{3}$$

In material modeling literature, a commonly used approach is to train feed-forward neural networks using a squared loss function to generate the operator $\boldsymbol{G}$. This approach is first described in [7] to identify parameters of a

visco-plastic material model. Since then, the approach has been successfully applied to identify parameters of many kinds of different material models [8, 9, 10, 11, 12]. Improvements have been proposed in [13] using a Bayesian neural network [14] to additionally obtain confident intervals for the identified parameters.

Generating a direct inverse model generally suffers from two drawbacks [6]:

1. Sampling is not goal-directed: To generate training data, the parameter space of the material model is sampled and the responses are calculated. To create a direct inverse model, a neural network-based model learns the reversed parameter/response relations. Consequently, the actual sampling is done in the output space of the neural network-based model. In this case, it is not guaranteed that the input space is covered sufficiently. Therefore, one has to sample in a wide range and rely on interpolation [6].
2. Direct inverse models have difficulties to approximate an inverse mapping, if the mapping is non-unique: The parameter identification problem is non-unique, when a certain material model response is caused by more than one parameter set. When this is the case, many supervised learning algorithms, such as feedforward neural networks trained with a squared loss function, end up averaging across the potential parameter sets [15].

For the feedforward neural network approaches found in literature, the latter is not a problem, because the therein analyzed material models underlie a unique relation between model responses and internal parameters. However, such a unique relation is not necessarily the case. Armstrong-Frederick-type models for example do not hold a unique relation between stress response and their internal parameters, when the hardening model is described via a series expansion [16]. Often such material models are used for sheet metal forming simulations. Another example is the modeling of visco-elasticity through a series of Maxwell-elements (Prony series [17]). Such models can be used to simulate the behavior of glass, polymer materials or filled rubber.

The present paper addresses the application of neural network-based approaches to parameter identification problems that are non-unique. Therefore, we analyze and validate the applicability of a mixture of experts model [18]. Mixture of experts is an ensemble technique based on a dynamically structured framework of submodels, i.e. neural networks. The mixture of ex-

perts model dynamically partitions the input space into unique subtasks [19]. In contrast to the feedforward neural networks trained with a squared loss function, mixture of experts is able to handle non-unique problems, as long as the number of possible solutions is known [20]. To the authors knowledge, this is the first approach that uses a mixture of experts model to identify material model parameters in a non-unique parameter identification problem.

The structure of the paper is as follows: First, we introduce an elasto-plastic material model, for which the identification problem is non-unique. Then, we describe the neural network-based approach in terms of a grey-box model and explain the structure of a mixture of experts model. Afterwards, two numerical examples are given addressing non-unique parameter identification problems: A simplified one, to compare feedforward neural networks trained with a squared loss function with the mixture of experts approach; and a parameter identification problem for tensile test data of three metallic materials to show the applicability of mixture of experts to real data. Finally, concluding remarks are given, as well as an outlook on future research.

2. Methods

2.1. Elasto-plastic material model and non-uniqueness

In this section, we describe an elasto-plastic material model, for which the parameter identification problem is non-unique. The model is described in its one-dimensional form for infinitesimal strains. This is sufficient for the use in the present work, as the numerical examples consider only tensile tests and therefore a uniaxial stress state is predominant.

The strain is split in an additive manner into elastic and plastic parts. The linear relation between stress and elastic strain is defined by the Young's modulus E:

$$\sigma = E\ (\varepsilon - \varepsilon_{\mathrm{pl}}). \tag{4}$$

The scalar yield function defines when plastic flow takes place and is expressed as

$$f = ||\sigma|| - k_0 - R = 0, \tag{5}$$

where k_0 is the initial yield strength and R the hardening stress. The evolution equation for plastic deformation is given by the flow rule

$$\dot{\varepsilon}_{\mathrm{pl}} = \lambda \frac{\mathrm{d}f}{\mathrm{d}\sigma}, \tag{6}$$

where λ underlies the Karush-Kuhn-Tucker conditions and is therefore greater than zero when the yield point is reached ($f = 0$) and equals zero otherwise. To obtain R, a hardening model has to be introduced. In this paper, an exponential hardening model is used [21]:

$$R = \sum_i \left[\frac{\gamma_i}{\beta_i} (1 - \exp(-\beta_i s_\mathrm{p})) \right], \tag{7}$$

where s_p denotes the accumulated plastic strain and $\{\beta_i, \gamma_i\}$ are internal parameters. If more than one term is considered ($i > 1$), the identification of the internal parameters is a non-unique problem.

A typical scenario is for example the use of two expansion terms ($i = 2$). This yields the hardening model

$$R = \frac{\gamma_1}{\beta_1} (1 - \exp(-\beta_1 s_\mathrm{p})) + \frac{\gamma_2}{\beta_2} (1 - \exp(-\beta_2 s_\mathrm{p})). \tag{8}$$

As the ranges of β_i and γ_i are generally not restricted, one specific model response can be caused by two parameter sets, due to the relation

$$R(s_\mathrm{p}, \beta_1, \gamma_1, \beta_2, \gamma_2) \equiv R(s_\mathrm{p}, \beta_2, \gamma_2, \beta_1, \gamma_1). \tag{9}$$

It is to be mentioned, that the non-unique parameter identification problem can be transformed into a unique one, by strictly separating the parameter ranges of $\{\beta_1, \beta_2\}$ and $\{\gamma_1, \gamma_2\}$:

$$\beta_1 < \overline{\beta} < \beta_2 \ , \quad \gamma_1 < \overline{\gamma} < \gamma_2. \tag{10}$$

However, there is no clear method to set such limits adequately and, when set, the input/output space is cut and certain combinations of model parameters and responses are excluded from the identification problem. Therefore, one should allow $\{\beta_1, \beta_2\}$ and $\{\gamma_1, \gamma_2\}$ to exist in the same parameter ranges.

2.2. Grey-box models

Grey-box models combine deterministic models, domain knowledge, and data-driven approaches in order to increase model accuracy. The neural network-based approaches discussed in this paper can be understood as grey-box models, as the training data is generated via a deterministic model, namely the material model. Generating data via a deterministic model generally has

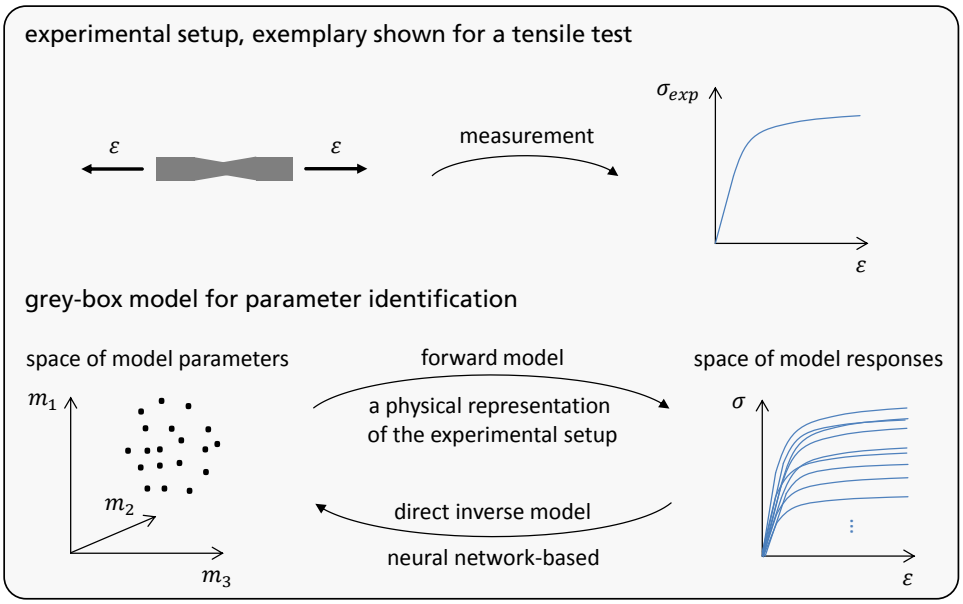


Figure 1: Workflow to set up a grey-box model that identifies material model parameters for given experimental measurements

two advantages [22]: First, the amount of data is restricted only by computation power, whereas the experimentation of complex systems can rapidly grow very expensive. Second, deterministic models have no random error, which is why the data is clean and noiseless. The workflow to generate such a grey-box model is schematically depicted in Fig. 1 and can be described by two steps:

1. Setting up the forward model: The experimental setup is analyzed and the boundary conditions of the problem are defined, as well as the parameter space to cover. Based on that, a physical model is used to generate training data following a certain sampling strategy. The physical model includes the material model and represents the experimental setup.
2. Setting up the direct inverse model: A neural network-based model is trained with the generated data by reversing the role of inputs and outputs. Then, it is applied to the experimental data of interest in order to identify a set of material model parameters.

Additionally, domain knowledge can be integrated in grey-box models for

example via feature engineering and sampling. Feature engineering is the process of finding new features that better represent the underlying problem as the raw data. Having meaningful features can significantly simplify the parameter identification problem and improve the interpretability of the grey-box model. Furthermore, choosing the right sampling strategy that sufficiently covers the input space is essential for neural network-based models. A commonly used sampling strategy is Latin hypercube sampling [23, 22]. It ensures that the parameters to analyze are represented in a fully stratified manner. However, in an inverse problem such a sampling strategy is not goal-directed [6]. In this case, it cannot be guaranteed that the input space of the neural network-based model is covered sufficiently. The key to handle this problem is to design a knowledge-based sampling strategy that focuses sensitive regions of the problem. This can be realized for example by densely sampling more sensitive regions and sparsely sampling less sensitive regions. In any case, the generated training data should be reviewed critically from domain sight.

2.3. Mixture of experts

Mixture of experts is a machine learning method that is originally introduced in [18] and belongs to the class of ensemble techniques. The concept of ensemble techniques is to combine several weak submodels into a global model in order to achieve better performance. The main idea behind mixture of experts is to partition the global problem into subtasks and assign expert models to each subtask. The partitioning is performed via a gating model that determines mixture coefficients indicating the active expert. The experts and the gate are typically designed as neural networks. It is also possible to use a Bayesian formulation [24, 25] or design the experts as Gaussian processes [26]. An alternative, but closely related approach is mixture density networks [27], in which the mixture coefficients and the expert outputs are determined in one and the same neural network model. The structure of mixture of experts is illustrated in Fig. 2. This structure can be called a dynamic structure, as the model dynamically determines which expert runs on a given input [19]. To make the mixture of experts model more flexible, it can be extended to an hierarchical structure [28].

In this work, we use a mixture of experts model, in which each expert is realized by a feedforward neural network following a Gaussian distribution [20]. For the i^{th} expert, the local probability distribution for the output $\boldsymbol{y}$

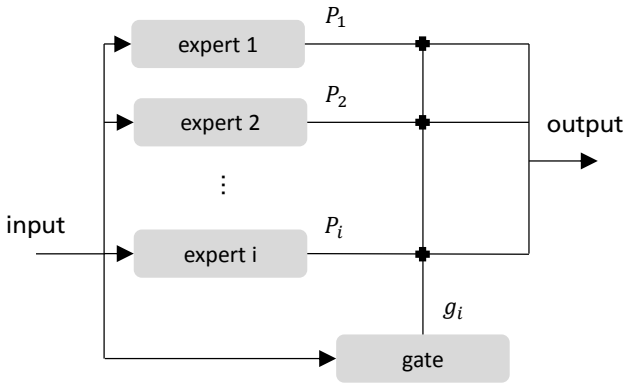


Figure 2: Scheme of a mixture of experts model with n experts and one gate. The input signal is fed to the experts and the gate. The model output is defined by the sum of the expert outputs P_i weighted by the mixture coefficients g_i.

given the input signal $\boldsymbol{X}$ is expressed as

$$P(\boldsymbol{y}|\boldsymbol{X}, \theta_i) = \mathcal{N}(\boldsymbol{y}|\overline{\boldsymbol{y}}_i(\boldsymbol{X}, \boldsymbol{w}_i), \boldsymbol{\Gamma}_i), \tag{11}$$

where $\mathcal{N}$ denotes a Gaussian distribution and θ_i includes the network weights $\boldsymbol{w}_i$ and covariance $\boldsymbol{\Gamma}_i$. The mixture coefficients are determined by the gate via the softmax function

$$g_i(\boldsymbol{X}, \boldsymbol{v}) = \frac{\exp(z_i(\boldsymbol{X}, \boldsymbol{v}))}{\sum_j^n \exp(z_j(\boldsymbol{X}, \boldsymbol{v}))}, \tag{12}$$

where n is the number of experts and $z_i(\boldsymbol{X}, \boldsymbol{v})$ the i^{th} output of the gating network including the weights $\boldsymbol{v}$. Combining Eq. (11) with Eq. (12), the global probability distribution can be written as a sum over all experts:

$$P(\boldsymbol{y}|\boldsymbol{X}, \theta_i, \boldsymbol{v}) = \sum_i^n g_i(\boldsymbol{X}, \boldsymbol{v}) P(\boldsymbol{y}|\boldsymbol{X}, \theta_i). \tag{13}$$

For exact predictions, the mean of the Gaussian distribution of the active expert is evaluated:

$$\overline{\boldsymbol{y}}(\boldsymbol{X}) = g_i(\boldsymbol{X}, \boldsymbol{v}) \overline{\boldsymbol{y}}_i(\boldsymbol{X}, \boldsymbol{w}_i). \tag{14}$$

The active expert is determined by the largest mixture coefficient. The mixture of experts model is trained by minimizing the negative logarithm of the likelihood described in Eq. (13). During training, the neural network

weights are adjusted via error backpropagation using Adam optimizer [29]. The mixture of experts model is implemented via python package tensorflow [30].

In the following it is explained briefly how a mixture of experts model can solve non-unique problems. Therefore, we assume to have a set of data points derived from a non-unique mapping. For each input parameter, there are n equivalently valid but unequal outputs. To model this one-to-n mapping, a mixture of experts model with at least n experts is considered. During training, the gate softly partitions the input data into n distinct subsets while every expert gets specialized to one subset. This is achieved by multiplying the expert outputs by the mixture coefficients g_i. As a consequence the intensity of the backpropagated error increases for possibly active experts ($g_i \to 1$) and decreases for possibly non-active experts ($g_i \to 0$). After training, the mixture of all experts forms the desired one-to-n mapping.

Examples for successfully applied mixture of experts models to non-unique problems can be found in other fields of research. In [25] and [31] it is used to solve inverse kinematics problems and in [32] for a problem of 3D pose reconstruction. Furthermore, in [33] a method to detect ambiguities in regression problems is introduced, following the idea to split an ambiguous data set into distinct subsets, however using a fuzzy rule-based system.

3. Numerical examples

3.1. A simplified non-unique parameter identification problem

In a first study, we compare the mixture of experts approach to a feedforward neural network trained with a squared loss function. For a proof of concept, we apply both models to a non-unique parameter identification problem. The feedforward neural network is set up via python package scikit-learn [34].

3.1.1. Problem description

We focus on the hardening model described in Eq. (8). The aim is to identify the internal parameters of the hardening model for given stress-strain curves. The parameter identification problem can be described as follows: Find a set of parameters $\{\beta_1, \gamma_1, \beta_2, \gamma_2\}$, for which $R(s_\mathrm{p}, \beta_1, \gamma_1, \beta_2, \gamma_2)$ approximates a given curve $\overline{R}(s_\mathrm{p})$. The accumulated plastic strain s_p is an independent variable. Due to the relation described in Eq. (9), the parameter identification problem is non-unique.

Table 1: Ranges and units of parameters to identify

parameter	min	max	unit
β_1	5	200	-
γ_1/β_1	100	400	MPa
β_2	5	200	-
γ_2/β_2	100	400	MPa

3.1.2. Data base generated by forward model

Based on Latin hypercube sampling, $10,000$ training samples are generated. Additionally, $2,500$ samples are generated randomly for validation and further $2,500$ for testing. To define the inputs for the neural network-based models, the stress-strain curves have to be discretized. Therefore, $R(s_\mathrm{p})$ is evaluated at 20 equidistantly distributed points on the strain axis inside the limits $s_\mathrm{p} = [0, 0.2]$. To increase the variety of information in the data, domain knowledge is integrated in the sampling process.

First, we introduce the parameters $\{\beta_i, \frac{\gamma_i}{\beta_i}\}$ instead of using $\{\beta_i, \gamma_i\}$. The term $\frac{\gamma_i}{\beta_i}$ determines the limit of R, when s_p grows large:

$$\lim_{s_\mathrm{p}\to\infty} R = \sum_i \gamma_i/\beta_i. \tag{15}$$

Using this description, the maximal hardening stress in the data set can be better regulated and the identification task becomes easier to interpret.

Second, we focus sampling on regions where β_i is small, as the curvature of $R(s_\mathrm{p})$ is more sensitive to variations of small β_i values as for large β_i-values, inside the given limits. This is caused by the exponential form of the hardening model and the relation shown in Eq. (15). A dense sampling in regions where β_i is small is realized by rearranging the β_i parameters given by the Latin hypercube sampling strategy on a logarithmic scale. As a consequence, this leads to stress-strain curves with a more realistic curvature. The considered parameter ranges are listed in Table 1.

3.1.3. Comparing mixture of experts with feedforward neural networks

The feedforward neural network is designed as a three-layer neural network, in which the hidden layer contains 100 neurons. For training, a squared loss function is used. The mixture of experts model consists of two expert networks, as there are two possible parameter sets for each curve to identify.

Both experts are realized by three-layer neural networks with 100 neurons in the hidden layers. The gating network is also designed as a three-layer neural network, but with ten neurons in the hidden layer.

To evaluate both models, it is not suitable to compare the true and the identified parameters, because first, multiple parameter sets are correct and second, we are actually interested in how close the material model response fits the target response. Therefore, we recalculate $R(s_\mathrm{p})$ using the identified parameters and compare the reproduced curve $R_\mathrm{repr}(s_\mathrm{p})$ with the target curve $\overline{R}(s_\mathrm{p})$. To measure the misfit between the two curves, we calculate the mean absolute error:

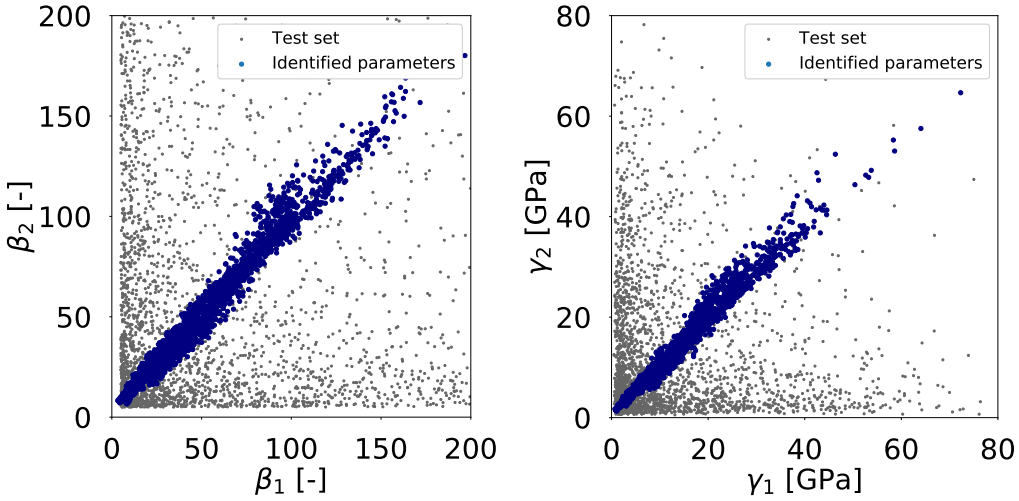
$$MAE = \frac{1}{s_\mathrm{p1} - s_\mathrm{p0}} \int_{s_\mathrm{p0}}^{s_\mathrm{p1}} |\overline{R}(s_\mathrm{p}) - R_\mathrm{repr}(s_\mathrm{p})| \mathrm{d}s_\mathrm{p}, \tag{16}$$

where s_p0 and s_p1 describe the limits, inside which the curves are evaluated.

3.1.4. Results and discussion

The mean and the maximal MAE for both models applied to the samples in the test set is given in Table 2. The results show that the mixture of experts model achieves a higher accuracy in the given non-unique parameter identification problem than the feedforward neural network trained with a squared loss function. The latter ends up averaging across the possible targets. This behavior can be observed in Fig. 3. Therein, the identified parameters are depicted, as well as the ground truth of the test set. The identified parameters lie close to the β-space diagonal and γ-space diagonal. This behavior is caused by the relation given in Eq. (9) and the averaging tendency of the used feedforward neural network model trained with a squared loss function. In contrast, the mixture of experts model partitions the parameter identification problem dynamically into two distinct subtasks. While the partitioning is rather unsystematic in the input space, it has a clear effect on the output space; see Fig. 4. The expert outputs are separated along the β-space diagonal and γ-space diagonal and thereby cover the whole parameter space. Consequently, the expert networks are able to solve the assigned subtasks.

This shows that typically used feedforward neural network models trained with a squared loss function cannot be applied to non-unique parameter identification problems. To identify parameters in a non-unique parameter identification problem alternative learning methods should be used, such as the mixture of experts model. When the number of possible solutions is known in advance, the mixture of experts model shows satisfying results.

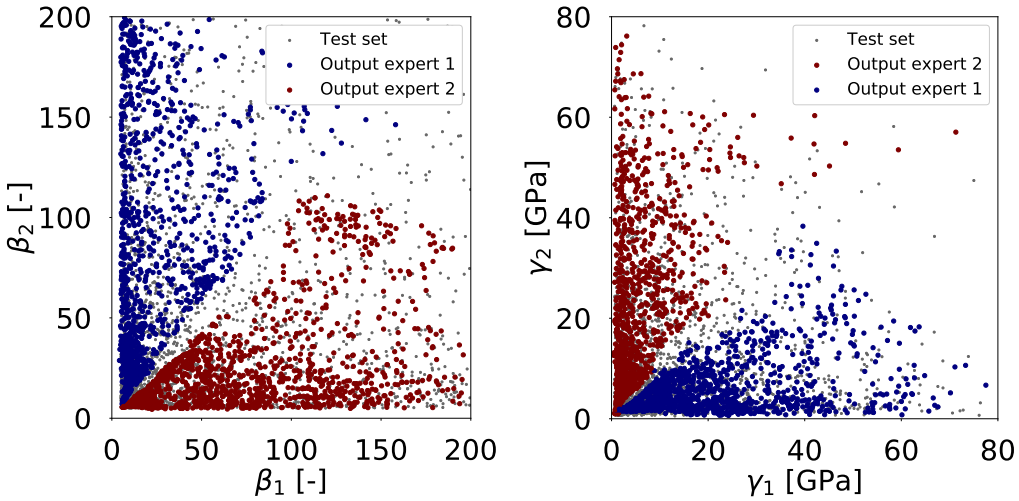


(a) β-space (b) γ-space

Figure 3: The space of model parameters: Distribution of parameters in the test set and the parameters identified by the feedforward neural network using a squared loss function.

Table 2: Mean and maximum MAE of the results obtained by the feedforward neural network trained via a squared loss function and the mixture of experts model.

method	mean MAE	max MAE	unit
feedforward neural network	32.1	197.6	MPa
mixture of experts	4.8	32.8	MPa



(a) β-space (b) γ-space

Figure 4: Space of model parameters: Distribution of parameters in the test set and the parameters identified by the mixture of experts model. The colors indicate the assigned experts.

3.2. Using mixture of experts to identify material model parameters for tensile test data

In a second example it is shown that the mixture of experts model can be used to identify parameters in a non-unique parameter identification problem for given experimentally measured data. The experimental data is extracted from tensile tests of three metallic materials, namely AA6014 aluminum alloy[1], DC04 deep drawing steel[2], and DP600 dual-phase steel[3]; cf. Fig. 5.

3.2.1. Problem description

To represent the behavior of the three metallic materials, the isotropic elasto-plastic material model described in Section 2.1 is used. The parameters to identify are therefore $\{E, k_0, \beta_1, \gamma_1, \beta_2, \gamma_2\}$. The parameter identification problem can be described as follows: Find a set of parameters $\{E, k_0, \beta_1, \gamma_1, \beta_2, \gamma_2\}$

[1]Tests performed at Fraunhofer IWU during AiF project 18810BG [35]
[2]Tests performed at IUL Dortmund during DFG project Graduate School 1483 [36]
[3]Tests performed at IFU Stuttgart during DFG project Graduate School 1483 [37]

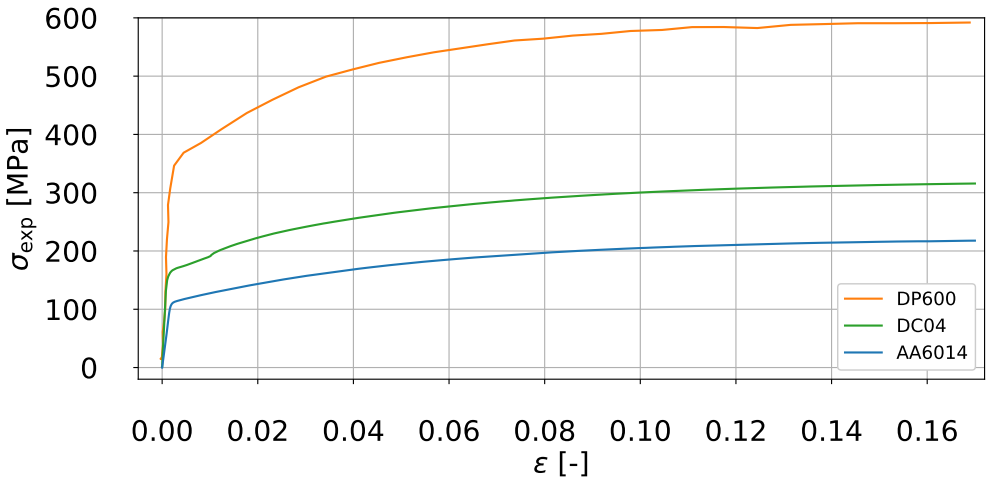


Figure 5: Engineering stress-strain curves of tensile tests of AA6014 aluminum alloy, DC04 deep drawing steel, and DP600 dual-phase steel.

for which the isotropic elasto-plastic material model approximates the experimentally measured stress-strain curve $\sigma_{\text{exp}}(\varepsilon)$. The strain ε is an independent variable.

Following the grey-box model approach described in Section 2.2, the training data is generated via the material model itself. As the material model is an idealization of real material behavior, the information contained in the training data is slightly different to the information contained in experimentally measured data. In this example, this is mainly caused by two phenomena:

1. Measurement errors: The experimental measurements of AA6014 and DC04 are quite accurate and result in rather smooth stress-strain curves, whereas the experimental measurements of DP600 underly a large fluctuation. In contrast, the response of the isotropic elasto-plastic material model is generally smooth (except for the transition between elastic and plastic deformation).
2. Idealization of discontinuous yielding: Note the transition from elastic to plastic behavior in the measured stress-strain curves of DP600 and DC04. Especially the DP600 specimen shows a broad zone of discontinuous yielding. The material model, in contrast, idealizes the transition as only one point, the yield point.

Table 3: Ranges and units of parameters to identify

parameter	min	max	unit
E	50000	200000	MPa
k_0	100	400	MPa
β_1	5	200	-
γ_1/β_1	50	400	MPa
β_2	5	200	-
γ_2/β_2	50	400	MPa

As a consequence, the accuracy of grey-box models decreases, when they are trained on ideal data and applied to real measurements. However, modifying the training data can handle this problem as we will describe in the following section.

3.2.2. Data base generated by forward model

Using Latin hypercube sampling, $50,000$ training samples are generated. Furthermore, $5,000$ randomly distributed samples are generated for validation. To test the model we use the tensile test data of the three materials. The parameter space of γ_i and β_i is sampled as described in Section 3.1. The parameter ranges are listed in Table 3. We assume, that the parameters to identify lie inside the chosen parameter ranges.

To increase the accuracy of the mixture of experts model when applied to experimentally measured stress-strain curves, the training data is modified as follows:

1. To make the mixture of experts model robust to measurement errors, a Gaussian noise with zero mean and variance s^2 is added to the material model response data [38]:

$$\sigma_{\text{noisy}} = \sigma \ (1 + \ \mathcal{N}(\mu = 0, s^2 = 10^{-4})). \tag{17}$$

2. It is ensured that the evaluation points (the inputs for the mixture of experts model) do not lie inside the transition zone. Therefore, 20 equidistantly distributed points are evaluated in the plastic zone, after the discontinuous yielding, and one point is evaluated in the elastic zone before discontinuous yielding takes place.

Table 4: Identified parameters for the isotropic elasto-plastic material model to fit the experimentally measured stress-strain curves

Material	E [GPa]	k_0 [MPa]	β_1 [-]	γ_1 [MPa]	β_2 [-]	γ_2 [MPa]
AA6014	62	110	6	254	17	1496
DC04	184	170	12	617	26	2827
DP600	175	306	68	13238	14	1354

3.2.3. Results and discussion

To assess the quality of the parameter estimates, the stress-strain curves reproduced by the material model using the identified parameters are compared to the experimentally measured stress-strain curves; see Fig. 6. The identified parameters are listed in Table 4.

The best result is obtained by identifying parameters for the elasto-plastic material model to represent the AA6014 specimen. The reproduced stress-strain curves lies close to the experimentally measured curve. Also, in case of DC04, the identified parameters satisfyingly fit the material model response to the experimentally measured data. Though, the material models response deviates slightly from the experimental data, when discontinuous yielding takes place. The broad zone of discontinuous yielding in the DP600 measurements is challenging for the grey-box model approach. Fitting the isotropic elasto-plastic material model to this zone is generally not possible, as the material model does not take into account discontinuous yielding at all. Furthermore, due to fluctuation in the experimental measurements, the material models response deviates slightly from the experimental curve across the whole plastic zone. Anyhow, a rather pragmatic estimate of k_0 ensures that the material models response still follows the experimental curve acceptably.

From an engineers sight, the experimentally measured stress-strain curves are approximated sufficiently by the material model using the identified parameters. Furthermore, a general advantage of neural network-based approaches is shown. Once the model is trained, it can be used to directly identify material model parameters given tensile test data of different materials.

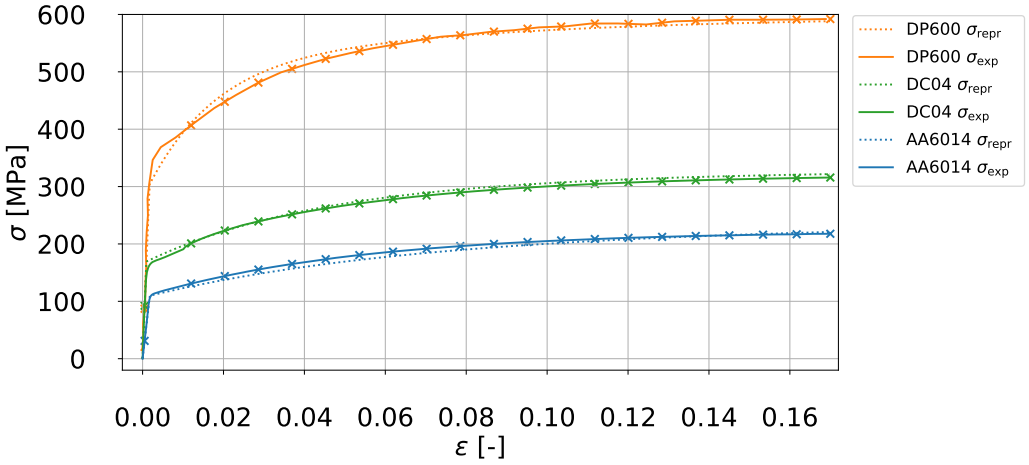


Figure 6: Engineering stress-strain curves of the three metallic materials and the curves obtained by the material model using the identified parameters. The crosses indicate the inputs for the mixture of experts model.

4. Conclusion and outlook

In the present paper, a grey-box model approach for parameter identification of material models is described. An important part of the grey-box model is the integration of a neural network-based model to set up a direct inverse model that estimates optimal material model parameters. For this purpose, typically feedforward neural networks are used, trained with a squared loss function. In this work, it is shown that these algorithms cannot be applied to non-unique parameter identification problems. However, when the problem is non-unique, a dynamically structured model can be applied, such as the proposed mixture of experts model.

In any case, a fundamental understanding of the material model is essential and the source of non-uniqueness should be well understood. For a successful use of mixture of experts, it is crucial to define the right number of subtasks to partition the problem. This has to be done in advance and is rather difficult for complex material models. This justifies the need for data-driven approaches to efficiently determine the number of possible solutions of a non-unique problem. Furthermore, the development and analysis of alternative machine learning models to represent one-to-many relations is needed. Generally, the prediction accuracy of neural network-based models and the computational costs of generating data depends highly on the sampling strat-

egy. Unfortunately, for parameter identification problems sampling is done in parameter space and is therefore not goal-directed. In the present paper this problem is tackled by integrating domain knowledge in the sampling process. Thereby, physically meaningful training data is generated and the space of material model responses is covered sufficiently. However, further research is needed to develop data-driven sampling strategies that enable a goal-directed sampling for inverse problems.

It should be mentioned here that the described grey-box model approach is not restricted to the field of material modeling. This holds for the integrated neural network-based models and especially for the described mixture of experts approach to solve non-unique parameter identification problems.

Acknowledgment

This work is part of Fraunhofer project *"Grey-Box-Modelle - Integration von Anwendungswissen in Lernverfahren"*. We thank the Fraunhofer-Gesellschaft for funding and the project members for their support and the helpful discussions. We also thank Johannes Dornheim, ISRG at Karlsruhe University of Applied Sciences, for his comments on the mixture of experts model.

Data availability

The raw/processed data required to reproduce these findings cannot be shared at this time due to technical and time limitations.

References

[1] R. Mahnken, Identification of material parameters for constitutive equations, Encyclopedia of Computational Mechanics.

[2] R. Mahnken, E. Stein, A unified approach for parameter identification of inelastic material models in the frame of the finite element method, Computer Methods in Applied Mechanics and Engineering 136 (3-4) (1996) 225–258.

[3] R. Mahnken, E. Stein, Parameter identification for viscoplastic models based on analytical derivatives of a least-squares functional and stability investigations, International Journal of Plasticity 12 (4) (1996) 451–479.

[4] C. M. Bishop, Pattern recognition and machine learning - Information science and statistics, Springer-Verlag, Berlin, Heidelberg, 2006.

[5] S. Haykin, Neural networks - A comprehensive foundation, Prentice Hall, New Jersey, 1998.

[6] M. I. Jordan, D. E. Rumelhart, Forward models: Supervised learning with a distal teacher, Cognitive Science 16 (3) (1992) 307–354.

[7] G. Yagawa, H. Okuda, Neural networks in computational mechanics, Archives of Computational Methods in Engineering 3 (4) (1996) 435–512.

[8] N. Huber, C. Tsakmakis, Determination of constitutive properties fromspherical indentation data using neural networks. Part I: The case of pure kinematic hardening in plasticity laws, Journal of the Mechanics and Physics of Solids 47 (7) (1999) 1569–1588.

[9] N. Huber, C. Tsakmakis, Determination of constitutive properties fromspherical indentation data using neural networks. Part II: Plasticity with nonlinear isotropic and kinematic hardening, Journal of the Mechanics and Physics of Solids 47 (7) (1999) 1589–1607.

[10] M. Lefik, B. Schrefler, Artificial neural network for parameter identifications for an elasto-plastic model of superconducting cable under cyclic loading, Computers & Structures 80 (22) (2002) 1699–1713.

[11] A. Nardin, B. Schrefler, M. Lefik, Application of artificial neural network for identification of parameters of a constitutive law for soils, Developments in Applied Artificial Intelligence (2003) 423–436.

[12] D. Helm, Pseudoelastic behavior of shape memory alloys: Constitutive theory and identification of the material parameters using neural networks, Technische Mechanik 25 (1) (2005) 39.

[13] J. F. Unger, C. Könke, An inverse parameter identification procedure assessing the quality of the estimates using Bayesian neural networks, Applied Soft Computing 11 (4) (2011) 3357–3367.

[14] D. J. MacKay, A practical Bayesian framework for backpropagation networks, Neural Computation 4 (3) (1992) 448–472.

[15] L. Morand, J. Pagenkopf, D. Helm, Material-based process-chain optimization in metal forming, in: Proceedings of Applied Mathematics and Mechanics, Vol. 17, Society of Applied Mathematics and Mechanics, 2017, pp. 709–710.

[16] J.-L. Chaboche, G. Rousselier, On the plastic and viscoplastic constitutive equations - Part I: Rules developed with internal variable concept, Journal of Pressure Vessel Technology 105 (2) (1983) 153–158.

[17] R. L. Taylor, K. S. Pister, G. L. Goudreau, Thermomechanical analysis of viscoelastic solids, International Journal for Numerical Methods in Engineering 2 (1) (1970) 45–59.

[18] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton, Adaptive mixtures of local experts, Neural Computation 3 (1) (1991) 79–87.

[19] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, `http://www.deeplearningbook.org`.

[20] S. E. Yuksel, J. N. Wilson, P. D. Gader, Twenty years of mixture of experts, IEEE Transactions on Neural Networks and Learning Systems 23 (8) (2012) 1177–1193.

[21] D. Helm, Stress computation in finite thermoviscoplasticity, International Journal of Plasticity 22 (9) (2006) 1699–1727.

[22] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, Design and analysis of computer experiments, Statistical Science 4 (4) (1989) 409–423.

[23] M. D. McKay, R. J. Beckman, W. J. Conover, Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (2) (1979) 239–245.

[24] S. R. Waterhouse, D. MacKay, A. J. Robinson, Bayesian methods for mixtures of experts, in: Advances in Neural Information Processing Systems, 1996, pp. 351–357.

[25] C. M. Bishop, M. Svensen, Bayesian hierarchical mixtures of experts, in: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2002, pp. 57–64.

[26] V. Tresp, Mixtures of Gaussian processes, in: Advances in Neural Information Processing Systems, 2001, pp. 654–660.

[27] C. M. Bishop, Mixture density networks, Technical report NCRG 4288, Neural Computing Research Group, Aston University, Birmingham (1994).

[28] M. I. Jordan, R. A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, Neural Computation 6 (2) (1994) 181–214.

[29] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[30] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning., in: OSDI, Vol. 16, 2016, pp. 265–283.

[31] Z. Ghahramani, Solving inverse problems using an EM approach to density estimation, in: Proceedings of the 1993 Connectionist Models Summer School, Psychology Press, 1994, pp. 316–323.

[32] A. Kanaujia, D. Metaxas, Learning ambiguities using Bayesian mixture of experts, in: Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, IEEE, 2006, pp. 436–440.

[33] A. K. Nandi, F. Klawonn, Detecting ambiguities in regression problems using TSK models, Soft Computing 11 (5) (2007) 467–478.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[35] A. Butz, J. Pagenkopf, M. Baiker, K. Silbermann, D. Landgrebe, Virtuelle Kennwertermittlung für die Umformsimulation von Feinblechen - Teil 2, EFB-Forschungsbericht Nr. 464, Europäische Forschungsgesellschaft für Blechverarbeitung e.V. (EFB) (2017).

[36] M. Baiker, Mikrostruktursimulationen zur Ermittlung von Materialeigenschaften für Blechwerkstoffe und Anwendung in der Umformsimulation, Dissertation, Karlsruher Institut für Technologie (2016).

[37] J. Pagenkopf, Bestimmung der plastischen Anisotropie von Blechwerkstoffen durch ortsaufgelöste Simulationen auf Gefügeebene, Dissertation, Karlsruher Institut für Technologie (2018).

[38] J. Sietsma, R. J. F. Dow, Creating artificial neural networks that generalize, Neural Networks 4 (1) (1991) 67–79.