



HAL
open science

A data-driven identification method based on neural networks for solving statistical inverse problems in computational mechanics

Florent Pled, Christophe Desceliers

► To cite this version:

Florent Pled, Christophe Desceliers. A data-driven identification method based on neural networks for solving statistical inverse problems in computational mechanics. 13th International Conference on Structural Safety & Reliability (ICOSSAR 2021-2022), Tongji University, Sep 2022, Shanghai (virtual), China. hal-03774358

HAL Id: hal-03774358

<https://hal.science/hal-03774358v1>

Submitted on 10 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A data-driven identification method based on neural networks for solving statistical inverse problems in computational mechanics

F. Pled^a and C. Desceliers^a

^aUniv Gustave Eiffel, MSME UMR 8208, F-77454, Marne-la-Vallée, France, E-mail: florent.pled@univ-eiffel.fr,
christophe.desceliers@univ-eiffel.fr

ABSTRACT: This work presents a data-driven machine learning framework for the solution of statistical inverse problems in multiscale computational solid mechanics. The proposed identification method is based on the design of an artificial neural network [Haykin, 1994, Demuth et al., 2014] in order to learn the nonlinear mapping between the hyperparameters of a prior stochastic model of the random compliance field [Soize, 2006] and dedicated quantities of interest of an *ad hoc* multiscale computational model. An initial database containing input and target data is first generated using the multiscale computational model. A processed database is then constructed by conditioning the input data with respect to the target data using classical kernel smoothing techniques in nonparametric statistics [Bowman and Azzalini, 1997] in order to derive an efficient trained neural network for identification purposes. Multilayer feedforward neural networks are then trained from each of the two databases and optimized by considering different network configurations in order to construct a fine-tuned surrogate model of the nonlinear relationship between the hyperparameters (network outputs) and the quantities of interest (network inputs). The performances of the trained neural networks are evaluated in terms of mean squared error, linear regression fit and probability distribution between network outputs and targets for both databases. The (best) trained neural network can then directly be used to identify the output hyperparameters given input observed quantities of interest with no call to the computational model. The efficiency of the neural network-based identification method is illustrated through two numerical examples developed within the framework of 2D plane stress linear elasticity. The proposed method is first validated on synthetic data obtained through numerical simulations and then applied to real experimental data obtained through mechanical tests monitored by digital image correlation on a real heterogeneous biological material (beef cortical bone).

1 INTRODUCTION

This paper addresses the inverse identification of the apparent elastic properties for heterogeneous materials with complex microstructures that can be considered as random linear elastic media, such as some engineered/synthetic and biological/natural composite materials (*e.g.* rock- and wood-like materials, concretes and cementitious

materials, living biological tissues). For such random heterogeneous materials, the uncertainties on the mechanical properties at a given mesoscale are modeled by a non-Gaussian random elasticity (or compliance) field [Soize, 2006, Soize, 2008] whose stochastic model is constructed within the framework of probability theory and information theory. The statistical inverse prob-



lem related to the identification of stochastic models using experimental or numerical data is a very challenging problem with respect to the current available computational resources. Indeed, in classical inverse identification methods, such a statistical inverse problem is typically formulated as a stochastic optimization problem whose solution requires many calls to the underlying forward stochastic computational models, resulting in expensive computational costs even using parallel and high-performance computing. In this work, we propose an innovative data-driven identification method based on neural networks for solving the statistical inverse problem related to the identification of an *ad hoc* stochastic model of the random compliance field within the framework of 2D plane stress linear elasticity theory. The statistical inverse problem is formulated as a function approximation (nonlinear regression) problem and solved using a neural network trained from a numerical database constructed from the stochastic computational model. The proposed neural network-based identification method consists in (i) generating a numerical database using the stochastic computational model to collect a dataset of input vectors of hyperparameters and output vectors of quantities of interest of the stochastic computational model, (ii) postprocessing this initial database by conditioning the quantities of interest with respect to the hyperparameters using the nonparametric statistics to construct a processed database adapted to the identification through a neural network, (iii) constructing a surrogate model based on neural network to learn the nonlinear mapping between the network input vectors of quantities of interest and the network output vectors of hyperparameters, (iv) assessing the performances of the trained neural network, and (v) using the best trained neural network to accurately predict the hyperparameters from given observed quantities of interest. This paper is organized as follows. The construction and analysis of the initial and processed databases are first presented in Section 2. The design

of the neural network is next addressed in Section 3. The performances of the neural networks trained from each of the initial and processed databases are compared and analyzed in terms of normalized mean squared error, linear regression fit and marginal probability density functions between network outputs and targets. Numerical identification results are finally provided in Section 4. The proposed neural network-based identification method is first validated on synthetic data and then applied to real experimental data.

2 CONSTRUCTION AND ANALYSIS OF THE DATABASE

Within the framework of linear elasticity theory, we consider the statistical inverse problem related to the identification of an *ad hoc* prior stochastic model of the random compliance field $[\mathbf{S}]$ of a heterogeneous elastic medium at a given mesoscale parameterized by a four-dimensional vector-valued hyperparameter $\mathbf{h} = (h_1, h_2, h_3, h_4)$, where $h_1 = \delta$ is a dispersion parameter controlling the level of statistical fluctuations exhibited by $[\mathbf{S}]$ around its mean function $[\underline{S}]$ (assumed to be independent of spatial position and isotropic here), $h_2 = \ell$ is a spatial correlation length characterizing the spatial correlation structure of the Gaussian random fields that are the stochastic germs of $[\mathbf{S}]$, $h_3 = \underline{\kappa}$ and $h_4 = \underline{\mu}$ are the mean bulk and shear moduli of the isotropic mean compliance field $[\underline{S}]$. The interested reader can refer to [Soize, 2006, Soize, 2008, Soize, 2017] for the explicit construction of this prior stochastic model including the algebraic representation of $[\mathbf{S}]$ and the random generator of independent realizations of $[\mathbf{S}]$.

We introduce the random vector $\mathbf{Q} = (Q_1, \dots, Q_9)$ of quantities of interest with values in \mathbb{R}^9 , corresponding to relevant random outputs of a multiscale stochastic forward computational model which are adapted to the inverse identification of the uncertain vector-valued hyperparameter \mathbf{h} . A complete description of the multiscale stochastic forward computational model and the definition of random vector \mathbf{Q} can be found in

2.1 Constructing the initial database by using the stochastic computational model

The statistical inverse identification of the prior stochastic model of $[\mathbf{S}]$ can be performed through the use of a neural network trained from a sufficiently large database. For the construction of such a database, the unknown vector-valued hyperparameter $\mathbf{h} = (h_1, h_2, h_3, h_4)$ is modeled as a random vector $\mathbf{H} = (H_1, H_2, H_3, H_4)$ uniformly distributed in a given compact subset $\mathcal{H} = \mathcal{H}_1 \times \mathcal{H}_2 \times \mathcal{H}_3 \times \mathcal{H}_4$ of \mathbb{R}^4 with mutually statistically independent random components H_1, H_2, H_3 and H_4 . In the following, we consider the following admissible set $\mathcal{H} = [0.25, 0.65] \times [20, 250] \times [8.5, 17] \times [2.15, 5.00]$ in $[-] \times [\mu\text{m}] \times [\text{GPa}] \times [\text{GPa}]$, which is adapted to the biological application presented in Section 4.2 and corresponding to a random heterogeneous microstructure made up of a biological tissue (bovine cortical bone).

The numerical database is constructed by using the multiscale stochastic forward computational model with the prior stochastic model of random compliance field $[\mathbf{S}]$. First, $N_d = 200,000$ independent realizations $\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(N_d)}$ of random hyperparameters vector \mathbf{H} are uniformly drawn in its admissible set \mathcal{H} . Then, for each realization $\mathbf{h}^{(i)}$ of random vector \mathbf{H} , one realization of random compliance field $[\mathbf{S}]$ is generated according to its prior stochastic model and associated random generator, then one realization $\mathbf{q}^{(i)}$ of random vector \mathbf{Q} is computed using the stochastic forward computational model. The complete initial database then contains the N_d independent realizations $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N_d)}$ of the random vector $\mathbf{X} = (\mathbf{Q}, \mathbf{H})$ with $\mathbf{x}^{(i)} = (\mathbf{q}^{(i)}, \mathbf{h}^{(i)})$ for $i = 1, \dots, N_d$. The N_d independent realizations $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(N_d)}$ of random vector \mathbf{Q} correspond to the network input data, while the N_d independent realizations $\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(N_d)}$ of random vector \mathbf{H} correspond to the network target (desired output) data. It should be noted that such an initial database cannot be directly used to train an efficient

neural network for identification purposes, since the nonlinear mapping between \mathbf{Q} and \mathbf{H} is stochastic by nature. As a consequence, a post-processing of the initial database is proposed in the next section and allows the construction of an *ad hoc* processed database.

2.2 Constructing the processed database by conditioning the initial database

The N_d network input vectors $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(N_d)}$ contained in the initial database are postprocessed and replaced with the N_d new input vectors $\tilde{\mathbf{q}}^{(1)}, \dots, \tilde{\mathbf{q}}^{(N_d)}$ corresponding to independent realizations of a new input random vector $\tilde{\mathbf{Q}}$ defined as the conditional mathematical expectation $\mathbb{E}\{\mathbf{Q}|\mathbf{H}\}$ of random vector \mathbf{Q} given random vector \mathbf{H} evaluated at $\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(N_d)}$, respectively. The i -th realization $\tilde{\mathbf{q}}^{(i)}$ of $\tilde{\mathbf{Q}}$ is then defined by

$$\tilde{\mathbf{q}}^{(i)} = \mathbb{E}\{\mathbf{Q}|\mathbf{H} = \mathbf{h}^{(i)}\} = \int_{\mathbb{R}^n} \mathbf{q} p_{\mathbf{Q}|\mathbf{H}}(\mathbf{q}|\mathbf{h}^{(i)}) d\mathbf{q}, \quad (1)$$

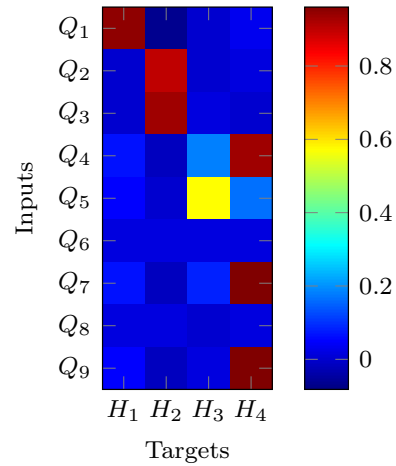
where $\mathbf{q} \mapsto p_{\mathbf{Q}|\mathbf{H}}(\mathbf{q}|\mathbf{h})$ is the conditional pdf of random vector \mathbf{Q} given random vector \mathbf{H} is equal to \mathbf{h} for any $\mathbf{h} \in \mathcal{H}$. The conditional pdf $\mathbf{q} \mapsto p_{\mathbf{Q}|\mathbf{H}}(\mathbf{q}|\mathbf{h})$ has been computed and estimated by using the multivariate kernel density estimation method with a Gaussian kernel function and the usual multidimensional Silverman bandwidth parameters, that is one of the most efficient and popular kernel smoothing techniques in nonparametric statistics [Bowman and Azzalini, 1997, Horová et al., 2012, Givens and Hoeting, 2013, Scott, 2015, Soize, 2017], and the N_d independent realizations $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(N_d)}$ and $\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(N_d)}$ of \mathbf{Q} and \mathbf{H} , respectively. The processed database then contains the N_d independent realizations $\tilde{\mathbf{x}}^{(1)}, \dots, \tilde{\mathbf{x}}^{(N_d)}$ of the random vector $\tilde{\mathbf{X}} = (\tilde{\mathbf{Q}}, \mathbf{H})$ with $\tilde{\mathbf{x}}^{(i)} = (\tilde{\mathbf{q}}^{(i)}, \mathbf{h}^{(i)})$ for $i = 1, \dots, N_d$. The N_d independent realizations $\tilde{\mathbf{q}}^{(1)}, \dots, \tilde{\mathbf{q}}^{(N_d)}$ of random vector $\tilde{\mathbf{Q}}$ contained in the processed database correspond to the new network input data, while the N_d independent realizations

$\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(N_d)}$ of random vector \mathbf{H} contained in the processed database correspond to the network target data that are the same as the ones contained in the initial database.

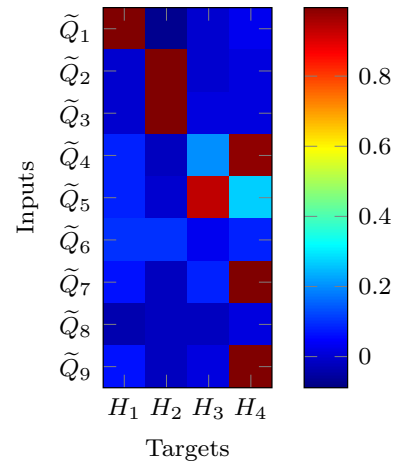
2.3 Analyzing the initial and processed databases

A sensitivity analysis of the network target data with respect to the network input data has been performed for both the initial and processed databases. Figure 1 shows the usual statistical estimate of the matrix of correlation coefficients between each of the nine components Q_1, \dots, Q_9 (resp. $\tilde{Q}_1, \dots, \tilde{Q}_9$) of input random vector \mathbf{Q} (resp. $\tilde{\mathbf{Q}}$) and each of the four components H_1, \dots, H_4 of target random vector \mathbf{H} computed from the N_d independent realizations $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(N_d)}$ of \mathbf{Q} (resp. $\tilde{\mathbf{q}}^{(1)}, \dots, \tilde{\mathbf{q}}^{(N_d)}$ of $\tilde{\mathbf{Q}}$) and the ones $\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(N_d)}$ of \mathbf{H} for the initial (resp. processed) database. We observe that the first component H_1 of \mathbf{H} is highly correlated to the first component Q_1 of \mathbf{Q} (resp. \tilde{Q}_1 of $\tilde{\mathbf{Q}}$) and is almost not correlated with the other components of \mathbf{Q} (resp. $\tilde{\mathbf{Q}}$) for the initial (resp. processed) database. Then, the second component H_2 of \mathbf{H} is strongly correlated to the second and third components Q_2 and Q_3 of \mathbf{Q} (resp. \tilde{Q}_2 and \tilde{Q}_3 of $\tilde{\mathbf{Q}}$) and have very small correlations with the other components of \mathbf{Q} (resp. $\tilde{\mathbf{Q}}$) for the initial (resp. processed) database. Lastly, the third component H_3 of \mathbf{H} is mostly correlated with the fifth component Q_5 of \mathbf{Q} (resp. \tilde{Q}_5 of $\tilde{\mathbf{Q}}$) and to a lesser extent with the fourth component Q_4 of \mathbf{Q} (resp. \tilde{Q}_4 of $\tilde{\mathbf{Q}}$), while the fourth component H_4 of \mathbf{H} is highly correlated with the fourth, seventh and ninth components Q_4, Q_7 and Q_9 of \mathbf{Q} (resp. \tilde{Q}_4, \tilde{Q}_7 and \tilde{Q}_9 of $\tilde{\mathbf{Q}}$) and to a lesser extent with the fifth component Q_5 of \mathbf{Q} (resp. \tilde{Q}_5 of $\tilde{\mathbf{Q}}$) for the initial (resp. processed) database. It should be noted that the values of the significant correlation coefficients are higher for the processed database than for the initial database, and therefore the target random vector \mathbf{H} is more sensitive to the processed input random vector $\tilde{\mathbf{Q}}$ than to

the initial one \mathbf{Q} . Also, let us mention that the sixth and eighth components Q_6 and Q_8 of \mathbf{Q} (resp. \tilde{Q}_6 and \tilde{Q}_8 of $\tilde{\mathbf{Q}}$) are almost uncorrelated with any component of \mathbf{H} for the initial (resp. processed) database, so that the corresponding N_d independent realizations $q_6^{(1)}, \dots, q_6^{(N_d)}$ and $q_8^{(1)}, \dots, q_8^{(N_d)}$ of Q_6 and Q_8 (resp. $\tilde{q}_6^{(1)}, \dots, \tilde{q}_6^{(N_d)}$ and $\tilde{q}_8^{(1)}, \dots, \tilde{q}_8^{(N_d)}$ of \tilde{Q}_6 and \tilde{Q}_8) could have been removed from the initial (resp. processed) database.



(a) Initial database with input random vector $\mathbf{Q} = (Q_1, \dots, Q_9)$



(b) Processed database with input random vector $\tilde{\mathbf{Q}} = (\tilde{Q}_1, \dots, \tilde{Q}_9)$

Figure 1. Matrix of correlation coefficients between each of the components of input random vector \mathbf{Q} (resp. $\tilde{\mathbf{Q}}$) and of target random vector \mathbf{H} for the initial database (a) (resp. the processed database (b)).



3 DESIGN OF THE NEURAL NETWORK

3.1 *Defining the neural network architecture, data division and training algorithm*

In this work, we consider multilayer feed-forward static neural networks composed of an input layer with 9 neurons, an output layer with 4 neurons, and one (or two) hidden layer(s) of neurons in between. A tan-sigmoid transfer function is used in the hidden layer(s) and a linear transfer function in the output layer. Different configurations have been tested for the two-layer (resp. three-layer) neural network with one (resp. two) hidden layer(s) and one output layer, by considering a number of hidden neurons varying from 4 to 500 (resp. from 4 to 75) in the hidden layer (resp. each of the two hidden layers). The complete dataset containing the N_d input and target vectors has been randomly divided into three distinct subsets for training, validation and testing. The first subset (training set) consisting of 70% of the data is used to train the neural network with the backpropagation training algorithm [Hagan et al., 1996] by performing gradient (or Jacobian) computations and updating the network parameters (weights and biases). The second subset (validation set) consisting of 15% of the data is used to monitor the training progress by computing the mean squared error on the validation set during the training iterations in order to stop the training before overfitting the data for a good network generalization. The third subset (test set) consisting of 15% of the data is used to assess the network performance after training and validation. Both the network input and target vectors are preprocessed and normalized in the standard range $[-1, 1]$ before presenting and applying them to the neural network. After training, the network output vectors are then transformed back to the original scale (units) of the network target vectors for future use. The initial weight and bias values in the hidden layer(s) and in the output layer have been generated using the Nguyen-Widrow initial-

ization method [Nguyen and Widrow, 1990]. The scaled conjugate gradient (SCG) optimization algorithm has been used as back-propagation algorithm to train both the two- and three-layer neural networks. All the computations have been performed using the MATLAB Neural Network Toolbox™ [Beale et al., 1992] (now part of the Deep Learning Toolbox™), the Parallel Computing Toolbox™, the Statistics and Machine Learning Toolbox™ and the Optimization Toolbox™. The multilayer neural networks have been trained and simulated by using a high-performance GPU on a single computer with three GPUs and a hundred CPUs.

3.2 *Measuring the neural network performances*

The performances of the trained two- and three-layer neural networks have been assessed by (i) computing the normalized mean squared error (mse) between the network outputs and corresponding targets, (ii) performing a linear regression analysis, and (iii) comparing the marginal probability density functions (pdfs) of each component of random vector $\mathbf{H} = (H_1, H_2, H_3, H_4)$ of hyperparameters estimated by using the univariate Gaussian kernel density estimation method [Bowman and Azzalini, 1997] with the network output data on the one hand and with the network target data on the other hand.

As a first measure of the network performance, the normalized mean squared errors obtained for the trained two- and three-layer neural networks have been computed for the training, validation, test and complete datasets. The best trained neural networks obtained with the SCG algorithm have been chosen as the ones with the lowest normalized mse on the test set. For the initial (resp. processed) database, the best trained two-layer neural network contains 50 (resp. 400) hidden neurons in the hidden layer with 704 (resp. 5604) weights and biases, while the best trained three-layer neural network contains 75 and 20 (resp. 75 and 50) hidden neurons in the first and second hidden lay-



ers, respectively, with a total of 2354 (resp. 4754) weights and biases. Table 1 shows the normalized mean squared errors obtained for the best two-layer and three-layer neural networks trained from the initial and processed databases, for the complete dataset and each of the training, validation and test subsets. For both initial and processed databases and both two- and three-layer neural networks, the network performances (normalized mse) are similar for each of the training, validation, test and complete datasets. However, the network performances obtained with the processed database (around 10^{-5}) are significantly better than that obtained with the initial database (around 10^{-2}). Also, the best three-layer neural network shows slightly better performances than the best two-layer neural network for both the initial and processed databases. Finally, the best overall network performance (normalized mse computed on the complete dataset) is obtained with the three-layer neural network trained from the processed database and is equal to 3.48×10^{-5} , and the corresponding training, validation and test performances are equal to 3.47×10^{-5} , 3.53×10^{-5} and 3.48×10^{-5} , respectively.

Table 1. Normalized mean squared errors obtained for the best two-layer and three-layer neural networks trained from the initial and processed databases.

Initial database		
Dataset	Two-layer neural network	Three-layer neural network
Training	1.46×10^{-2}	1.44×10^{-2}
Validation	1.48×10^{-2}	1.46×10^{-2}
Test	1.45×10^{-2}	1.44×10^{-2}
Complete	1.46×10^{-2}	1.45×10^{-2}
Processed database		
Dataset	Two-layer neural network	Three-layer neural network
Training	4.55×10^{-5}	3.47×10^{-5}
Validation	4.66×10^{-5}	3.53×10^{-5}
Test	4.55×10^{-5}	3.48×10^{-5}
Complete	4.57×10^{-5}	3.48×10^{-5}

As a second measure of the network performance, a linear regression analysis is performed between the network outputs and corresponding targets. The regression plots across the complete dataset are shown in Figures 2 and 3 for the best three-layer neural network trained from the initial and processed databases, respectively. In each plot, the network outputs and corresponding targets are represented by open black circles, the perfect fit (outputs exactly equal to targets) is represented by a dashed green line, and the best linear fit (linear regression between outputs and targets) is represented by a solid red line. The regression value (R -value), defined as the usual statistical estimate of the correlation coefficient between each output and the corresponding target, is given at the top of each regression plot. For the initial database, the best linear fit between network outputs and corresponding targets, although not perfect, is fairly good with R -values over 0.95 for dispersion parameter H_1 , 0.96 for spatial correlation length H_2 , 0.70 for mean bulk modulus H_3 , and 0.98 for mean shear modulus H_4 . Nevertheless, the scatter plots of the network outputs and corresponding targets exhibit a large dispersion, showing that some data points in the dataset have poor fits, especially for H_3 . For the processed database, the best linear fit between network outputs and corresponding targets is almost perfect with R -values very close to 1 for all random hyperparameters H_1 , H_2 , H_3 and H_4 . Also, the scatter plots of the network outputs and corresponding targets exhibit a very small dispersion for all random hyperparameters H_1 , H_2 , H_3 and H_4 , showing a significantly better fit for the processed database than for the initial database.

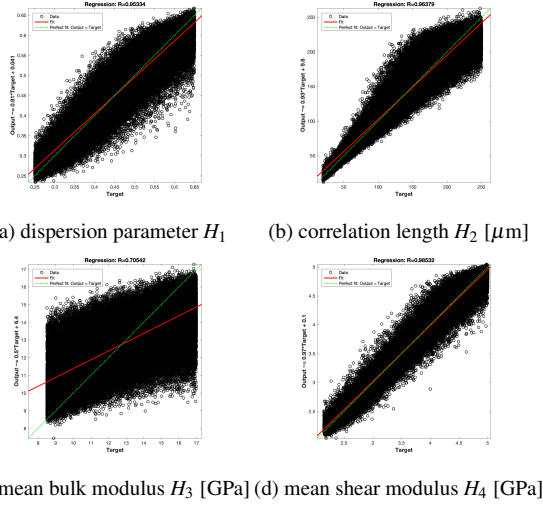


Figure 2. Initial database: linear regression between network outputs and corresponding targets for each random hyperparameter H_1 , H_2 , H_3 and H_4 .

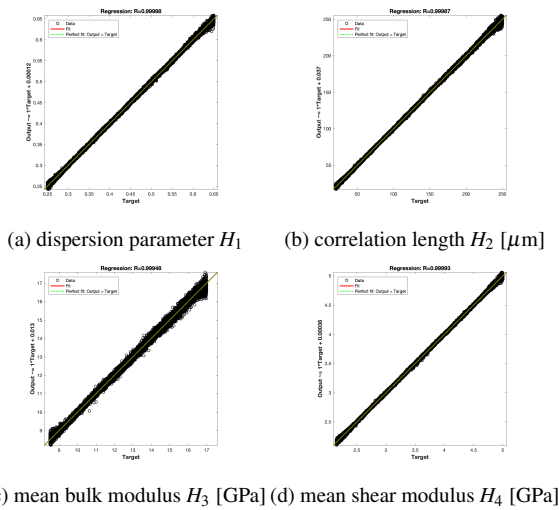


Figure 3. Processed database: linear regression between network outputs and corresponding targets for each random hyperparameter H_1 , H_2 , H_3 and H_4 .

As a third measure of the network performance, the marginal pdfs p_{H_1} , p_{H_2} , p_{H_3} and p_{H_4} of random hyperparameters H_1 , H_2 , H_3 , H_4 , respectively, which are assumed to be uniform random variables, are estimated by using the univariate Gaussian kernel density estimation method [Bowman and Azzalini, 1997] with the N_d network output data obtained using the best three-layer neural network trained from the initial database on the one hand and from the processed database on the other hand. These output pdfs are compared in Figure 4 to the

uniform target pdfs and to the target pdfs estimated by using the univariate Gaussian kernel density estimation method with the N_d associated target data. The output pdfs constructed from the output vectors of the best neural network trained with the processed database perfectly match the associated target pdfs, while the output pdfs constructed from the output vectors of the best neural network trained with the initial database have a worse fit, especially for H_3 .

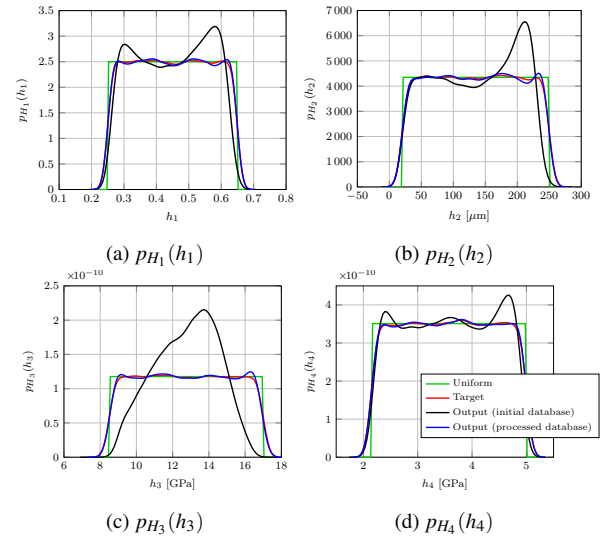


Figure 4. Probability density functions (pdfs) p_{H_1} , p_{H_2} , p_{H_3} and p_{H_4} of random variables H_1 , H_2 , H_3 and H_4 , respectively, with the uniform target pdfs (green), the estimated target pdfs (red), and the estimated output pdfs computed by using the best neural network trained with the initial database (black) and with the processed database (blue)

Finally, the best neural network trained with the processed database (obtained by conditioning the network input vectors contained in the initial database with respect to the network target vectors) can directly be used for identifying the value \mathbf{h}^{out} of random hyperparameters vector \mathbf{H} corresponding to a given observed vector \mathbf{q}^{obs} of quantities of interest. The conditioning of the initial database then appears to be a key step in obtaining an efficient trained neural network for solving the statistical inverse problem.

4 NUMERICAL RESULTS

The proposed neural network-based identification method is first validated on synthetic data obtained through numerical simulations and then applied to real experimental data obtained through mechanical tests monitored by digital image correlation on a real heterogeneous biological material (beef cortical bone). Based on the numerical results obtained in Section 3.2, we now consider the best three-layer neural network trained with the processed database for identification purposes.

4.1 Validation on synthetic data

We first consider a given input vector \mathbf{q}^{obs} of quantities of interest contained in the test dataset of the processed database for validating the proposed neural network-based identification method. The network output vector \mathbf{h}^{out} is directly computed by using the trained neural network with \mathbf{q}^{obs} as input vector and compared to the corresponding target vector $\mathbf{h}^{\text{target}}$. The identification results are given in Table 2.

Table 2. Synthetic data: comparison of output vector $\mathbf{h}^{\text{out}} = (h_1^{\text{out}}, h_3^{\text{out}}, h_3^{\text{out}}, h_4^{\text{out}})$ with associated target vector $\mathbf{h}^{\text{target}} = (h_1^{\text{target}}, h_3^{\text{target}}, h_3^{\text{target}}, h_4^{\text{target}})$ obtained for a given input vector \mathbf{q}^{obs} contained in the test dataset.

Hyperparameter	h_1	h_2	h_3	h_4
	[-]	[μm]	[GPa]	[GPa]
Target value	0.5514	172.36	12.398	4.672
Output value	0.5501	172.61	12.322	4.693
Relative error	0.24%	0.14%	0.61%	0.46%

We observe that the values of output vector \mathbf{h}^{out} computed by using the trained neural network with \mathbf{q}^{obs} as input vector are very close to the corresponding values of target vector $\mathbf{h}^{\text{target}}$ with relative errors less than 1% for each of the hyperparameters h_1, h_2, h_3 and h_4 .

4.2 Application to real experimental data

We now consider a given input vector \mathbf{q}^{obs} of observed quantities of interest coming from

experimental measurements of 2D displacement fields obtained from a single static vertical uniaxial compression test performed on a cubic specimen (with dimensions $1 \times 1 \times 1 \text{ cm}^3$) made of a biological tissue (beef femur cortical bone) and monitored by 2D digital image correlation (DIC) on one observed side of the cubic specimen corresponding to a 2D square domain with dimensions $1 \times 1 \text{ cm}^2$. Such experimental kinematic field measurements have been carried out in [Nguyen et al., 2016] and already used in [Nguyen et al., 2015, Zhang et al., 2020] for identifying the apparent elastic properties of bovine cortical bone at mesoscale. As for the previous validation example on synthetic data, the trained neural network is used to compute the output vector \mathbf{h}^{out} for the experimentally observed input vector \mathbf{q}^{obs} . Table 3 reports the values of output vector \mathbf{h}^{out} .

Table 3. Real experimental data: output vector $\mathbf{h}^{\text{out}} = (h_1^{\text{out}}, h_3^{\text{out}}, h_3^{\text{out}}, h_4^{\text{out}})$ obtained for a given input vector \mathbf{q}^{obs} of experimentally observed quantities of interest.

Hyperparameter	h_1	h_2	h_3	h_4
	[-]	[μm]	[GPa]	[GPa]
Output value	0.6106	65.906	10.448	4.598

The values of the output vector $\mathbf{h}^{\text{out}} = (0.6106, 65.906, 10.448, 4.598)$ in $([-], [\mu\text{m}], [\text{GPa}], [\text{GPa}])$ obtained for the input vector \mathbf{q}^{obs} are close to the values of the optimal vector $\mathbf{h}^{\text{opt}} = (0.533, 61.111, 10.500, 4.667)$ in $([-], [\mu\text{m}], [\text{GPa}], [\text{GPa}])$ obtained in the previous work [Zhang et al., 2020] by solving a costly multi-objective optimization problem using a fixed-point iterative algorithm with the same experimental measurements as those used in the present work. The identified values obtained with the previous method in [Zhang et al., 2020] result from a compromise between computational efficiency and numerical accuracy and are therefore less accurate than the ones obtained with the neural network-based identification method proposed in this work. Finally, the network output values are in agreement with the



identified values already published in the literature for this type of biological tissue (bovine cortical bone).

5 CONCLUSION

A data-driven identification method based on artificial neural networks has been presented in this paper for solving the statistical inverse problem related to the identification of the hyperparameters of a prior stochastic model of the random compliance field characterizing the apparent elastic properties of heterogeneous materials with complex random microstructure. This statistical inverse problem has been formulated as a function approximation problem and solved by using an artificial neural network trained from a large numerical database. An initial database has first been constructed by using forward numerical simulations of a multiscale stochastic forward computational model. A processed database has then been derived by conditioning the input data contained in the initial database with respect to the target data. A sensitivity analysis of the target data with respect to the input data contained in each of the two databases has been performed. Two- and three-layer feedforward neural networks have been trained with each of the two databases and optimized by considering different network configurations. Numerical results show that the neural networks trained with the processed database exhibit much better performances in terms of mean squared error, linear regression fit and marginal probability distributions between network outputs and targets compared to the ones trained with the initial database. The conditioning of the initial database turns out to be an essential step in obtaining an efficient trained neural network for solving the underlying statistical inverse problem. Finally, the proposed neural-network based identification method has been validated on synthetic data coming from numerical simulations and then applied to real experimental data coming from physical experiments on a biological tissue (beef cortical

bone).

REFERENCES

- [Beale et al., 1992] Beale, M. H., Hagan, M. T., and Demuth, H. B. (1992). Neural network toolbox user's guide. *The MathWorks Inc.*
- [Bowman and Azzalini, 1997] Bowman, A. W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis*. Oxford University Press, Oxford.
- [Demuth et al., 2014] Demuth, H. B., Beale, M. H., De Jess, O., and Hagan, M. T. (2014). *Neural Network Design*. Martin Hagan, USA, 2nd edition.
- [Givens and Hoeting, 2013] Givens, G. H. and Hoeting, J. A. (2013). *Computational Statistics*. John Wiley & Sons, Hoboken, New Jersey, 2nd edition.
- [Hagan et al., 1996] Hagan, M. T., Demuth, H. B., and Beale, M. H. (1996). *Neural Network Design*. PWS Publishing Co., Boston, MA, USA.
- [Haykin, 1994] Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.
- [Horová et al., 2012] Horová, I., Koláček, J., and Zelinka, J. (2012). *Kernel Smoothing in MATLAB: Theory and Practice of Kernel Smoothing*. World Scientific Publishing Co. Pte. Ltd., Singapore.
- [Nguyen and Widrow, 1990] Nguyen, D. and Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *1990 IJCNN International Joint Conference on Neural Networks*, volume 3, pages 21–26.
- [Nguyen et al., 2016] Nguyen, M.-T., Allain, J.-M., Gharbi, H., Desceliers, C., and Soize, C. (2016). Experimental multiscale



measurements for the mechanical identification of a cortical bone by digital image correlation. *Journal of the Mechanical Behavior of Biomedical Materials*, 63:125–133.

[Nguyen et al., 2015] Nguyen, M.-T., Desceliers, C., Soize, C., Allain, J.-M., and Gharbi, H. (2015). Multiscale identification of the random elasticity field at mesoscale of a heterogeneous microstructure using multiscale experimental observations. *International Journal for Multiscale Computational Engineering*, 13(4):281–295.

[Pled et al., 2021] Pled, F., Desceliers, C., and Zhang, T. (2021). A robust solution of a statistical inverse problem in multiscale computational mechanics using an artificial neural network. *Computer Methods in Applied Mechanics and Engineering*, 373:113540.

[Scott, 2015] Scott, D. W. (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, Inc., second edition.

[Soize, 2006] Soize, C. (2006). Non-Gaussian positive-definite matrix-valued random fields for elliptic stochastic partial differential operators. *Computer Methods in Applied Mechanics and Engineering*, 195(1–3):26–64.

[Soize, 2008] Soize, C. (2008). Tensor-valued random fields for meso-scale stochastic model of anisotropic elastic microstructure and probabilistic analysis of representative volume element size. *Probabilistic Engineering Mechanics*, 23(2–3):307–323. 5th International Conference on Computational Stochastic Mechanics.

[Soize, 2017] Soize, C. (2017). *Uncertainty Quantification: An Accelerated Course with Advanced Applications in Computational Engineering*, volume 47 of *Interdisciplinary Applied Mathematics*. Springer International Publishing, 1 edition.

[Zhang et al., 2020] Zhang, T., Pled, F., and Desceliers, C. (2020). Robust Multiscale Identification of Apparent Elastic Properties at Mesoscale for Random Heterogeneous Materials with Multiscale Field Measurements. *Materials*, 13(12).