Combining neural networks with multiscale techniques for lattice unit cell design

A.I.Pais [1], J.L.Alves [1], J. Belinha [2]

[1]*INEGI Institute of Science and Innovation in Mechanical and Industrial Engineering*
R. Dr. Roberto Frias 400, 4200-465 Porto, Portugal
anapais@fe.up.pt, falves@fe.up.pt
[2]*ISEP Department of Mechanical Engineering, School of Engineering, Polytechnic of Porto*
R. Dr. António Bernardino de Almeida 431, 4200-072 Porto, Portugal
job@isep.ipp.pt

Abstract. Complex non-linear mapping between the input and output data is one of the advantages of neural networks. The aim of this work is to train a neural network to generate the optimum unit cell topology for a given constitutive elastic matrix. In order to reverse homogenization, the neural network maps the correlation between the shape of the unit cell and the constitutive elastic characteristics. With data produced from a collection of various geometries, a dataset of elastic properties and respective geometries was developed. All of these geometries underwent homogenization using periodic boundary conditions. To lessen their impact on the homogenized constitutive matrix, the lattice was modeled as a biphasic material, with the solid phase having the material's properties and the remaining area of the representative volume element (RVE) being treated as the void phase. To make it possible to directly impose the periodic boundary conditions, a uniform mesh of square 2D elements was used. The dataset includes truss-like unit cells based on the FCC and BCC systems, as well as gyroid-like unit-cells, simplified to a 2D representation. In order to increase the dataset, operations between basic unit cell geometries were applied as well as rotations of the unit cell. The neural network is capable of suggesting unit cells. The non-linear mapping between the unit cell elastic properties and geometry reduces the computational cost of running structural optimization to create a unit cell which presents the required properties, for example.

Keywords: Homogenization, Unit cell, Neural networks

1 Introduction

Multiscale modeling analyses the material at one length scale, while the results of such analysis are referent to the properties of the material at another length scale [1]. The use of such numerical homogenization techniques allows for significant savings in computational time as only a representative region of the material is chosen to model all the constituents of the structure [2]. In summary, the porous material is transformed into an equivalent solid material with homogenized properties [3].

Due to their ability to do non-linear mapping between inputs and outputs, neural networks can be used in the design of unit cell structures that abide by the required mechanical properties.

Neural networks can act as surrogate models for the mechanical properties of structures, accelerating the homogenization procedure. Some examples of this application are the works of Ma et al. [4], who use a neural network for the homogenization step in topology optimization, Kim et al. [5] who predict the anisotropic elastic properties of the unit cell using a neural network which takes the geometric parameters of the cell as input and [6] whose work surrogates the homogenization and sensitivity analysis step of the topology optimization procedure with the neural network.

Also, the surrogate model approach can be used in the iterative design of unit cells, combined with optimization algorithms, such as the work of Garland et al. [7] with the design of Pareto optimal unit cells, using a convolutional neural network (CNN) as a surrogate model for the stiffness of the unit cell. Using the trained CNN, a genetic algorithm optimizes the unit cell geometry. Ji et al. [8] calculate the homogenized elastic tensor using a

neural network, while the genetic algorithm optimizes the unit cell geometry.

With respect to the design of unit cells, some other approaches using neural networks can be found in the literature. Zheng et al. [9] present a conditional generative adversarial neural network (cGAN) to create a structure presenting a specific Young's modulus and Poisson ratio and Kollmann et al. [10] present a DL model for the maximization of shear or bulk modulus or the minimization of Poisson's ratio, the latter leading to the design of auxetic structures.

In this work, neural networks are used for the design of a unit cell geometry that matches the given elastic tensor. First, a neural network is trained to take in the unit cell input and output the elastic tensor. This network is used in the objective function of the optimization algorithm. Then, another network is trained to output the unit cell geometry given the required elastic tensor.

2 Materials and methods

2.1 Numerical homogenization

The dataset was labeled with the effective elasticity tensor. This tensor is calculated through homogenization theory and can be computed from

$$\overline{C}_{ijkl} = \frac{1}{A}\int_A C_{pqrs}(\varepsilon^{0(ij)}_{pq} - \varepsilon^{(ij)}_{pq})(\varepsilon^{0(kl)}_{rs} - \varepsilon^{(kl)}_{rs})dA \tag{1}$$

where C_{pqrs} is the locally varying elastic tensor and A is the RVE area. The macroscopic strain tensor is represented by $\varepsilon^{0(ij)}$ and $\varepsilon^{(ij)}$ is the locally varying strain tensor. For a plane stress state, the constitutive C matrix will look like

$$C = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_{44} \end{bmatrix} \tag{2}$$

and in order to calculate all the elastic constants, it is necessary to apply to the RVE three different macroscopic strains (for the two normal components and for the shear component). The components of the strain tensor should be zero except for the component corresponding to the elastic constants that are to be determined. To enforce the macroscopic strain field, a deformation gradient $F(x)$ was applied to the RVE with periodic boundary conditions.

Considering a periodic RVE , the boundary of the RVE can be decomposed into two parts $^+$ and . Each point $\mathbf{x}^+$ on $^+$ is connected to just one unique point $\mathbf{x}$ on and the normal vectors to each point are symmetrical so that $\mathbf{n}^+ = \mathbf{n}$ [2].

The local displacement field $\mathbf{u}$ can be decomposed as

$$\mathbf{u}(\mathbf{x}) = \hat{\mathbf{u}}(\mathbf{x}) + \mathfrak{u}(\mathbf{x}) = \mathbf{u}_0 + \mathbf{H}\ \mathbf{x} + \mathfrak{u}(\mathbf{x}) \tag{3}$$

where the macroscopic displacement gradient $\mathbf{H}$ is equal to the macroscopic strain ε up to a rotation, $\hat{\mathbf{u}} = \mathbf{u}_0 + \mathbf{H}\ \mathbf{x}$ is the macrodisplacement that corresponds to the applied strain, and $\mathfrak{u}(\mathbf{x})$ is a micro-displacement. The unit cell simulations assume that in the bulk of the material

$$\mathfrak{u}(\mathbf{x} + a\ _i\ \mathbf{e}_i) = \mathfrak{u}(\mathbf{x}) \tag{4}$$

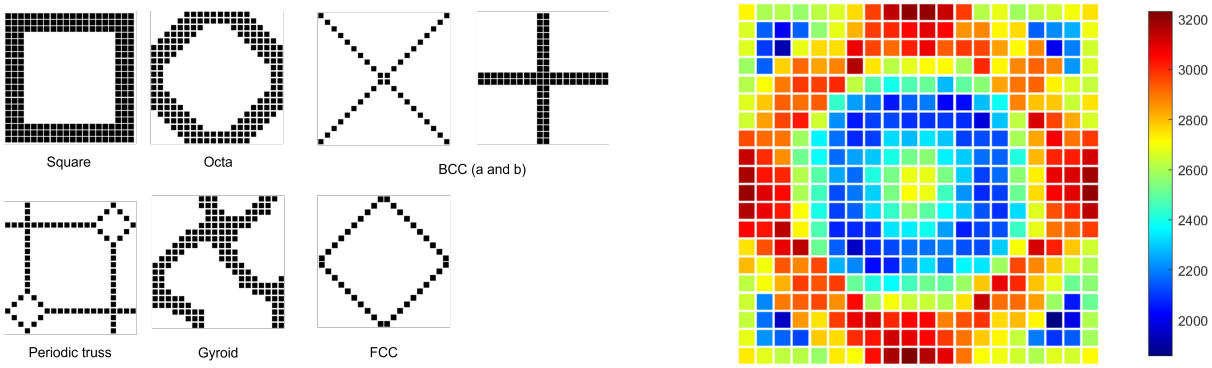
for any integer $_i$ and all positions of $\mathbf{x}$, and where a denotes the unit-cell length and $\mathbf{e}_i$ denotes the principal directions. Thus, it is assumed that the micro-displacement field shares the periodicity of the lattice [11].

The periodicity of micro-displacements is enforced by kinematically constraining the difference in the displacements of paired nodes, and setting this difference as equal to the displacement deduced from the macroscopic strain [11] so that

$$\mathbf{u}\ \mathbf{x}^+ \quad \mathbf{u}\ \mathbf{x} \quad = a\ \mathbf{H}\ \mathbf{n} \tag{5}$$

2.2 Dataset

A dataset of several unit cells was created. In order to create a large enough data set 7 basic cells were created, those being body centered cubic, (BCCa and BCCb),face centered cubic (FCC), square (SQ), octa (OC), periodic truss (PT) and gyroid (G). These basic units are schematized in Figure 1a. The creation of the whole



(a) (b)

Figure 1. Characterization of the dataset (a) basic unit cells; (b) final distribution of unit cell counts for each element of the 20 20 grid

data set consisted of operations of sums and subtractions between the several cells, with different wall thickness combinations, as well as inverting the cells. Due to the chosen resolution, consisting of a 20x20 element grid the operations between the unit cells lead to duplicates which were removed, with a final count of 6265 unit cells. Figure 1b shows a density map of the sum of all geometries following the removal of duplicates.

Non-uniqueness of solutions and viability of the dataset: Running homogenization analysis on the structures indicated that some structures lead to very similar constitutive elastic matrices. For every pair of two configurations leading to the same matrix, considering a tolerance, one of them is removed from the dataset. Therefore, the final dataset consisted of a total of 3404 unit cell geometries.

2.3 Neural networks

Two neural networks were created. The first network aims at serving as a surrogate model to the numerical homogenization procedure and the second to act as a non-linear mapping tool between the constitutive matrix and the lattice geometry. The second task is anticipated to present a higher degree of complexity, due to the non-uniqueness nature of the problem. Both approaches used the same dataset, and operated in different ways. In addition, both networks presented the same training function which is the Levenberg-Marquadt function. The error metric for both networks was the mean square error (MSE). The activation function for all hidden layers is the hyperbolic tangent function.

The implementation of the neural networks was done in MATLAB using the *deep learning toolbox* and the *feedforwardnet* function. An implementation example is shown next:

```
inputs = data(1:size_input,:);
targets = data(target_index,:);

net = feedforwardnet([size_layer1, size_layer2_size_layer_n]);

[net, training_results] = train(net, inputs, targets);
```

Direct neural network

The neural network acting as a surrogate model has 400 input neurons, one for each element of the grid. The geometry is characterized by a uniform pixel grid where each element is classified as 0 (no material = void pixel) or 1 (material pixel). The output of this network is size 4, corresponding to the non-zero constants of the constitutive matrix. Figure 2 shows a scheme of a neural network used to create the homogenization surrogate model where the height of the hidden layers was 50 and the width consisted of 5 hidden layers. The hidden layers presented hyperbolic tangent activation functions and the output layer presented a linear activation function.

Reverse neural network

The neural network to run reverse homogenization has 7 input neurons, 4 for the non-zero constants of the constitutive matrix, an additional one for the volume fraction of the desired structure and 2 more for the coordinates

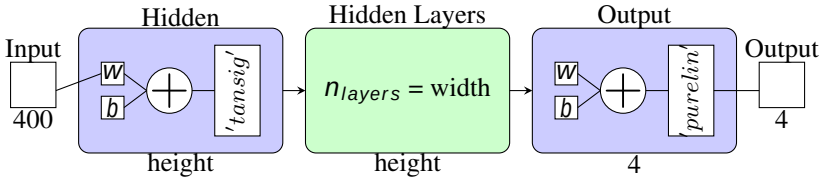


Figure 2. Network architecture scheme for the surrogate model to run direct homogenization

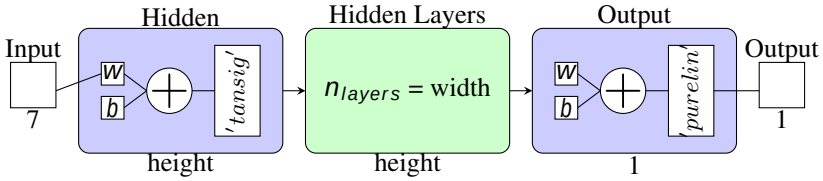


Figure 3. Network architecture scheme for the surrogate model to run reverse homogenization

of the pixel being evaluated. The output consists of the density of the pixel being calculated. Thus, this network must be called 400 times in order to obtain a full grid. This network presents 5 hidden layers, each with 50 neurons and hyperbolic tangent activation functions, as shown in Figure 3.

Optimization problem

The direct neural network can be used as a surrogate function allowing to achieve an optimal geometry through optimization algorithms. The objective function is the difference between the target array and the output array of the network. Therefore, a substantial amount of computational cost is saved because the FEM analysis required to calculate the constitutive elastic matrix is bypassed. In this work, a genetic algorithm is used to minimize the objective function. The design variables are each element in the material map. The optimization problem to be solved is shown next

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = jC_{target} \quad N(x)j \\ \text{subject to} \quad & v_f \quad tol <= \frac{\sum x}{n} <= v_f + tol \\ & x_i >= 0 \\ & x_i <= 1 \\ & x_i \; 2 \; \mathbb{Z} \\ & x_{\Gamma^-} = x_{\Gamma^+} \end{aligned} \tag{6}$$

where n is the length of x and $N(x)$ is the output from the neural network surrogate model. A tolerance is added to the volume fraction constraint. Additionally, a periodicity constraint is added where $x_{\Gamma^-} = x_{\Gamma^+}$ meaning that the elements on the $^+$ side of a boundary, horizontal or vertical, must be the same as the elements on the side of a boundary.

3 Results and discussion

3.1 Direct neural network

The performance of the network for the direct tast is shown in Figure 4 which shows the regression between the targets and the outputs of the separate test set (50 instances) indicating good generalization capability of the network.

3.2 Genetic optimization with surrogate model

Using the surrogate model approach it was possible to achieve solutions with equivalent properties to the target solution. Figure 5 shows an example taking as target values the homogenization results for the unit cell labeled as ”Database” in the Figure.

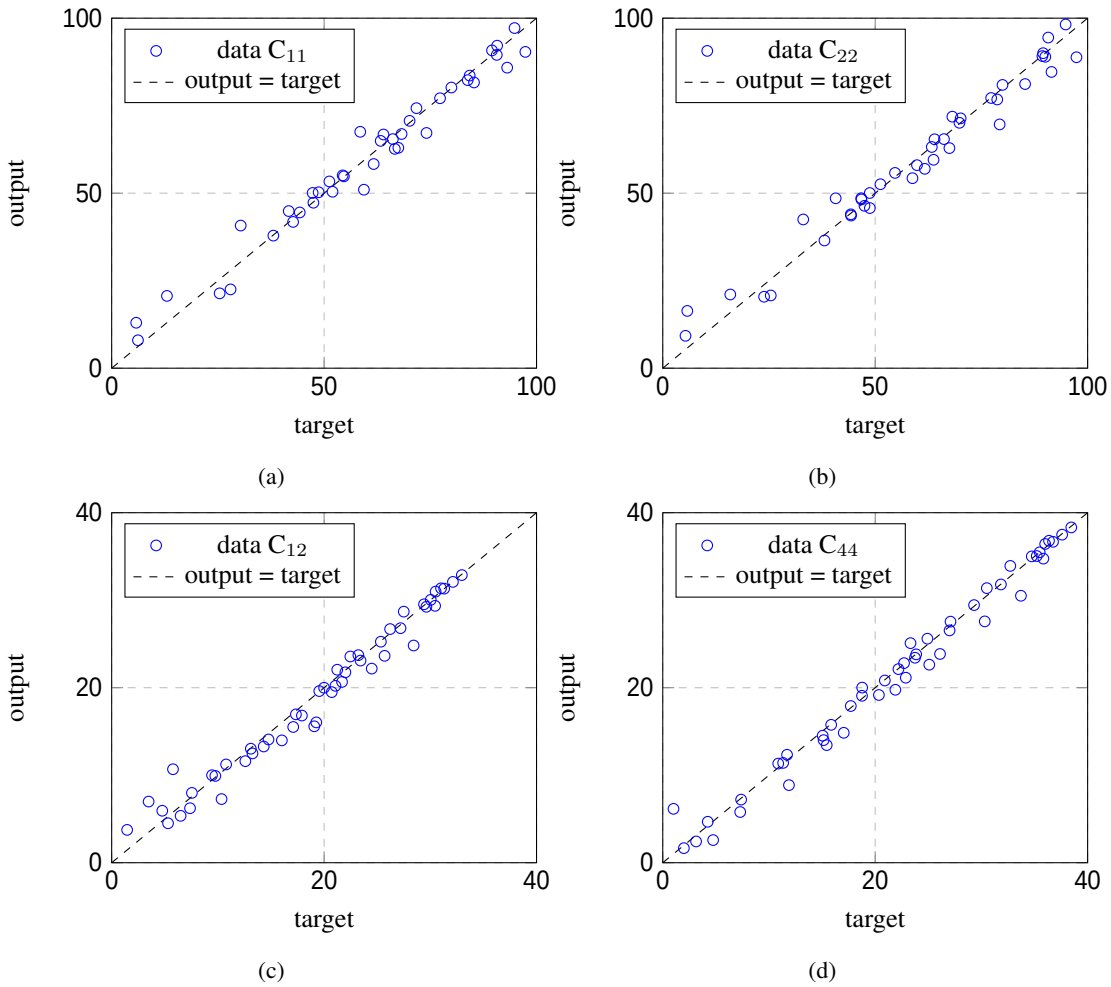
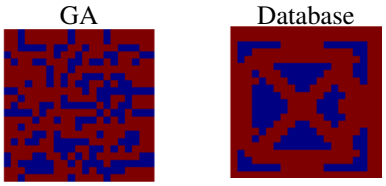
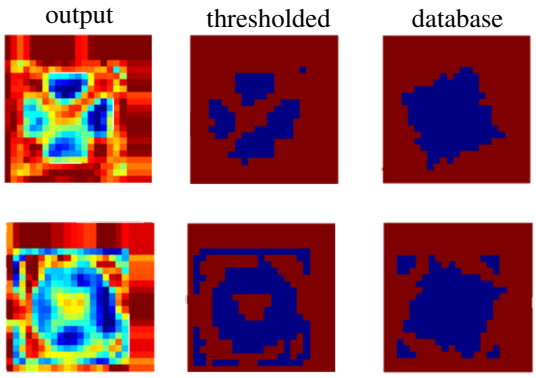


Figure 4. Regression for each variable considering the best network



	target	predicted (surrogate model)	FEM
C_{11}	27.9832	23.1122	28.6080
C_{22}	23.8461	21.8166	29.5534
C_{12}	7.3778	16.6494	8.9281
C_{44}	11.8806	12.3598	6.8513

Figure 5. Results obtained with a genetic algorithm using the neural network as a surrogate model to calculate the objective function



	target	predicted (surrogate model)	FEM
C_{11}	42.38795	42.33645	50.00771
C_{22}	43.16202	47.62848	51.07503
C_{12}	14.83079	15.37112	12.84709
C_{44}	13.46677	14.48823	14.16557

	target	predicted (surrogate model)	FEM
C_{11}	12.95064	11.60152	12.52517
C_{22}	13.30584	9.054743	11.48039
C_{12}	1.350524	3.653295	0.954938
C_{44}	0.445987	4.690858	0.466798

Figure 6. Results obtained with the reverse network

Table 1. Time performance comparison, all times are shown in seconds (s)

homogenization	prediction (surrogate NN)	GA	reverse NN
60	0.017796	114.109689	1.890407

3.3 Reverse neural network

Figure 6 shows the results obtained with the neural network trained to do the reverse task. The network output is not discrete as therefore the a threshold is applied to the output in order to create a binary image corresponding to the unit cell.

3.4 Discussion

With respect to the surrogate model, the neural network was able to output the properties of some given structure. Thus, it was feasible to employ this network as a surrogate model for optimization. However it should be noted that the solution by the algorithm is very different in terms of geometry although the mechanical properties are very similar. This is likely due to the fact that the properties of cellular materials are mostly dependent on their density [12]. Also, if the density is higher, the material distribution can be more homogeneous leading to higher isotropy, so the values of C_{44} will be closer to the target value. Further validation of the optimization is required with lower density geometries.

Regarding the reverse network, since this network learns from the dataset, the solutions are much closer to the target value both in terms of the elastic tensor as well as the mechanical properties. Thus, the reverse approach is likely to lead to more manufacturable results than the surrogate model optimization approach.

Finally, it is relevant to mention the significant savings in computational cost provided by neural networks. Table 1 shows a time comparison of the several steps in this work. Taking as reference the homogenization step, it will take on average 60s while a prediction with the surrogate model neural network takes approximately 0.02s. Then, in order to calculate a new geometry fitted to some elastic tensor, the genetic algorithm takes approximately 114s. The reverse neural network to predict a new geometry takes 1.89s which is significantly less than the genetic algorithm.

4 Conclusions

This work suggested a methodology to create 2D unit cells which fit some given elastic properties. In the field of tissue engineering, the presented methodology should aid in the minimization of the difference in mechanical properties between the scaffold and the bone in its surroundings. This difference leads to stress shielding and consequently bone decay. By taking as input the desired elastic properties as shown in Figure 7, neural networks can aid in the design of patient-specific implants.

Acknowledgements. The author truly acknowledges the funding provided by LAETA, under project UIDB