Convolutional neural network for fast prediction of the effective properties of domains with random inclusions

M Vasilyeva$^{1,2}$ and A Tyrylgin$^1$

$^1$ISC, Texas A&M University, College Station, TX, USA
$^2$MMR Laboratory, North-Eastern Federal University, Yakutsk, Russia

E-mail: vasilyevadotmdotv@gmail.com

Abstract. We consider a heterogeneous domain with random inclusions in two-dimensional and three-dimensional formulations. For generation of the train and test datasets, we numerically calculate the effective properties for a given geometry of the heterogeneous domain. We construct a machine learning method to learn a map between local heterogeneous geometries and effective properties. We present numerical results for prediction of the effective properties for 2D and 3D model problems.

1. Introduction

Complex processes in heterogeneous regions occur in many scientific and engineering problems. For example, modeling processes in composite materials or strongly heterogeneous porous media. One of the main difficulties in constructing a mathematical model is the consideration of heterogeneous properties, since the grid resolution of heterogeneity leads to a large number of unknowns and requires large computational resources [5, 9]. To solve such problems, methods such as averaging (finding average, effective characteristics of the medium) and multiscale methods are used. Such methods can significantly reduce the dimension of the problem, by constructing a macro model and calculating the problem on a coarse grid [2, 9]. These methods assume the solution of local problems for taking into account microscale information in approximation on a macro-scale computational grid. However, solution of such problems are computationally expensive for random media. The use of machine learning can significantly reduce the calculation time of effective characteristics in local domains.

In recent years, many highly effective methods for constructing artificial intelligence have appeared. This jump is associated with the development of new methods based on the construction of deep learning methods through convolutional neural networks, and the development of open source libraries. Now, convolutional neural networks have become very popular and are used to solve various problems, including for calculating the properties of the environment, identifying new dependencies, and also for solving partial differential equations without direct calculations of grid problems. Instead of using classical grid algorithms, machine learning methods can be used, where the geometry of the local problem is transferred as input data and instead of solving equations and constructing approximations, it is possible to construct a prediction algorithm based on deep neural networks [7, 8, 3].

In this work, we construct the machine learning methods for a fast calculation of effective characteristic for domains with random inclusions. Convolutional neural network is trained by a set of macroscale characteristics (effective medium properties, local matrices) and geometries (heterogeneity). We present numerical results for prediction of the effective properties for 2D and 3D model problems.
2. Problem formulation

The computational domain is considered as domain with two subdomains

\[ \Omega = \Omega_m \cup \Omega_p, \]

where \( \Omega_m \) is the main domain (matrix) and \( \Omega_p \) are the inclusions subdomain.

**Parabolic problem.** For diffusive heat or mass transfer in heterogeneous domain, we have following unsteady equation

\[ \frac{\partial u}{\partial t} - \nabla \cdot (D \nabla u) = f, \quad x \in \Omega, \quad t > 0, \tag{1} \]

where \( f \) is the source term, \( c = c(x) \) and \( D = D(x) \) is the heterogeneous properties.

We set following initial and boundary conditions

\[ u(x, 0) = u_0, \quad x \in \Omega, \quad t = 0, \quad \text{and} \quad u(x, t) = g(x), \quad x \in \partial \Omega, \quad t > 0. \]

We use finite element method for approximation of the elasticity problem and have following variational formulation: find \( u \in V \) such that

\[ m \left( \frac{\partial u}{\partial t}, v \right) + a(u, v) = l(v), \quad \forall v \in V \tag{2} \]

where

\[ m(u, v) = \int_{\Omega} c(x) \frac{\partial u}{\partial t} v \, dx, \quad a(u, v) = \int_{\Omega} D(x) \nabla u \cdot \nabla v \, dx, \quad l(v) = \int_{\Omega} f \, v \, dx, \]

and

\[ V = \{ v \in H^1(\Omega); v = g, x \in \partial \Omega \}, \quad \tilde{V} = \{ v \in H^1(\Omega); v = 0, x \in \partial \Omega \}. \]

**Elasticity problem.** The elasticity equation with source term \( f \) is given by

\[ -\text{div} \sigma(u) = f, \quad x \in \Omega, \tag{3} \]

where \( \sigma(u) = C : \varepsilon(u) \), \( \varepsilon(u) = (\nabla u + \nabla^T u) / 2 \) and \( C \) is the elasticity tensor.

We set following boundary conditions

\[ u = g, \quad x \in \partial \Omega, \tag{4} \]

We use the finite element method for approximation of the elasticity problem and have following variational formulation: find \( u \in W \) such that

\[ a(u, v) = l(v), \quad v \in \bar{W}, \tag{5} \]

where

\[ a(u, v) = \int_{\Omega} (\sigma(u), \varepsilon(v)) \, dx, \quad l(v) = \int_{\Omega} f v \, dx, \]

and

\[ W = \{ v \in [H^1(\Omega)]^d; v = g, x \in \partial \Omega \}, \quad \bar{W} = \{ v \in [H^1(\Omega)]^d; v = 0, x \in \partial \Omega \}. \]

and \( d = 2, 3 \).

3. Coarse grid approximation

In this section, we consider a numerical homogenization method for construction of the coarse grid approximation. In these methods, we solve local problems on a fine grid for calculation of the effective characteristics.

Let \( T^h = \bigcup_i K_i \) be a structured partition of the computational domain \( \Omega \) into elements \( K_i \) \( (i = 1, N_c, N_c \) is the number of coarse cells in \( T^h \)). For numerical solution on the coarse grid, we use numerical homogenization technique. The main idea of numerical homogenization is to identify homogenized coefficients in each coarse grid block and construct the upscaled coefficients.

**Parabolic problem.** To define an upscaled coefficient for each coarse cell \( K_i \), we solve fine-scale problem in local domain \( K_i \)

\[ -\nabla \cdot (D(x) \nabla p^{(l)}) = 0, \quad x \in K_i, \tag{6} \]

with Dirichlet boundary conditions

\[ p^{(l)} = x_l, \quad l = \overline{1,d}, \quad x \in \partial K_i \tag{7} \]
and \( d = 2,3 \) is the dimension of the problem.

The upscaled coefficient \( D^{*K_i} \) is defined by averaging the fluxes
\[
D^{*K_i}_d = \frac{1}{|K_i|} \int_{K_i} D(x) \nabla p^{(l)} \, dx.
\]

For calculation of the upscaled coefficient \( c^{*K_i} \), we use volume averaging
\[
c^{*K_i} = \frac{1}{|K_i|} \int_{K_i} c(x) \, dx.
\]

After the calculation of the upscaled coefficients \( D^{*K_i}(x) \) and \( c^{*K_i}(x) \), we can solve following equation on the coarse grid
\[
c^{*K_i}(x) \frac{\partial u}{\partial t} - \nabla \cdot (D^{*K_i}(x) \nabla u) = f , \quad x \in \Omega , \quad t > 0 ,
\]
where \( D^{*K_i} \) is the anisotropic upscaled coefficient.

**Elasticity problem.** The heterogeneous properties are captured in the effective elasticity tensor \((C^{*K_i})\) and calculated by solution of the local problem in the local domains \( K_i \)
\[
- \text{div} \sigma(u^{(rs)}) = 0 , \quad x \in K_i ,
\]
with Dirichlet boundary conditions
\[
u^{(rs)} = \Lambda^{(rs)} x , \quad x \in \partial K_i ,
\]
where
\[
\Lambda_{ij}^{(rs)} = \frac{1}{2} ( \delta_{ij} \delta_{ls} + \delta_{is} \delta_{jr} ) , \quad r,s = \overline{1,d} .
\]

The components of the effective elastic modulus \( C^{*K_i} \) are defined by averaging the local strains
\[
C_{rspq}^{*K_i} = \frac{1}{|K_i|} \int_{K_i} C_{ijkl} \varepsilon_{ij}^{(rs)} \varepsilon_{kl}^{(pq)} \, dx ,
\]
where \( r,s,p,q = \overline{1,d} \) and \( \varepsilon^{(rs)} = \varepsilon(u^{(rs)}) \) is the deformation field.

There are existed several approaches for the numerical homogenization methods based on the two-scale asymptotic analysis with solution of the local problems in representative volume \( K_i \) with different types of boundary condition.

4. **Convolutional neural network**

We construct convolutional neural network (CNN), where as input parameters we set dataset of fine grid properties for local domain and as output, we set effective properties \( (D_{ij}^{*K_i}, C_{lnij}^{*K_i} \text{ and/or } c^{*K_i}) \). The machine learning algorithms are used for the recognition of dependence in data and provide accurate and fast calculation of the effective properties in numerical homogenization.

![Figure 1. Dataset for 2D problems. Random circle inclusions](image1)

![Figure 2. Train set for 3D problems. Random sphere inclusions](image2)
We consider randomly generated geometries. Such random geometries, for example can be used for simulation of the composite materials, where for improving physical characteristics some number of inclusions added in the homogeneous matrix. As dataset for construction of the machine learning algorithm, we use geometries, where in homogeneous matrix we have some number of circle inclusions (see Figures 1 and 2 for illustration). This dataset used to train, validate and test the neural network. We have two datasets: 2D and 3D arrays. Dataset for two-dimensional problems contains 1250 local geometries and 500 geometries for three-dimensional problems. Example of such images are shown in Figures 1 and 2. The images are generated using different number of inclusions, circle radius and center positions. We divide dataset randomly into training, validation and test sets. The input arrays have size $M^d$ with $M = 64$ for $d = 2$ and $M = 32$ for $d = 3$, where each value represent normalized property of the heterogeneous media. For example, it can be heat conductivity, heat capacity, permeability or Young’s modulus. The scale of each element in the dataset are scaled to fall within the range 0 to 1. For representing heterogeneous properties high resolution images can be used for better accuracy of the homogenization or multiscale methods.

Let the input of the network is $d$-dimensional normalized array

$$X_i = K_i, \ i = 1, N,$$

with $K_i$ has size $M^d$ and $N$ is the size of the dataset.

The output for the numerical homogenization is the normalized array of the effective properties, where each property normalized separately

$$Y_i = \{D \times K_i, C \times K_i \text{ and/or } c \times K_i\}, \ i = 1, N.$$

To generate a dataset for machine learning, the samples $X_i$ are generated randomly. To compute $Y_i$, we use numerical homogenization method for effective properties calculation

$$\{X_1, \ldots, X_i, \ldots, X_N\} \rightarrow \text{Numerical homogenization} \rightarrow \{Y_1, \ldots, Y_i, \ldots, Y_N\}.$$

We divide dataset to a train, validation and test sets with size $N_{\text{train}}$, $N_{\text{val}}$ and $N_{\text{test}}$. As test set, we take 20\% of data, another 80\% divided between train and validation set in 80/20 proportion. We train convolutional neural network (CNN) on following data.

$\text{Train set:} \{(X_1, Y_1), \ldots, (X_i, Y_i), \ldots, (X_N, Y_{N_{\text{train}}})\} \rightarrow \text{CNN}$

We use validation set to validate a training process and test data for testing constructed machine learning algorithm. After that use constructed network for fast prediction of the effective properties for a given $d$-dimensional geometry:

$$X \rightarrow \text{CNN} \rightarrow Y.$$

In machine learning algorithm, we compute $\hat{Y}_i = F(X_i, W)$, where $X_i$ is the $i$-th input data and $W$ represent system parameters. Let $E_i$ is the loss function $E_i = G(Y_i, F(X_i, W))$, that measure difference between correct (real) output $Y_i$ for input $X_i$ and output produced by the system. The average loss function over whole train set then calculated using a mean square error (MSE)

$$E_{\text{train}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} ||Y_i - F(X_i, W)||^2.$$

In calculations, we also calculate validation set loss function.

5. Results

We present numerical results for model problem in local domains $K = [0,1]^d$, $d = 2,3$. The model parameters used are as follows:

• Laplace operator: $k_m = 10^{-2}$, $k_p = 1$.

• Elasticity operator: $k_m = 10^{-2}$, $k_p = 1$.

In Figure 1, we show local domains, where inclusions are depicted with white color for 2D problem. Local domains for 3D case, we depict in Figure 2. For Laplace operator, we present results for 2D and 3D formulations. Elasticity operator results are presented for 2D formulation.
Figure 3. Loss for numerical homogenization. Left: 2D. Right: 3D

Figure 4. Learning performance of CNN for 2D problem. $Y_i = \{D_{11}^1, D_{22}^1, D_{12}^1, c^1, K_i\}$ (from left to right). Parity plots comparing preference property values against predictions made using CNN. First row: train dataset (green color). Second row: test dataset (blue color)

Dataset is divided into 750:250:250 ratio for training, validation and test sets for 2D problem. For 3D problem, we divided into 300:100:100 ratio. Learning performance of CNN for 2D and 3D problems are presented in Figures 4, 5 and 6, where we present a parity plots comparing preference property values against predictions made using CNN. Convergence of the loss function, is depicted in Figure 3.

Figure 5. Learning performance of CNN for 2D problem. $Y_i = \{C_{1111}^1, C_{1122}^1, C_{2222}^1, C_{1212}^1, C_{1211}^1, C_{1212}^1\}$ (from left to right). Parity plots comparing preference property values against predictions made using CNN. First row: train dataset (green color). Second row: test dataset (blue color)
Figure 6. Learning performance of CNN for 3D problem. 

\[ Y_i = \{ D_{11}^{*K_i}, D_{22}^{*K_i}, D_{33}^{*K_i}, D_{12}^{*K_i}, D_{23}^{*K_i}, D_{13}^{*K_i}, c^{*K_i} \} \] (from left to right). Parity plots comparing preference property values against predictions made using CNN. First row: train dataset (green color). Second row: test dataset (blue color).

The network has 4 for 2D and 3 for 3D alternating convolutional and pooling layers with rectified linear units (RELU) activation layer, and 2 fully connected layers. For calculations, we use 50 epoch with a batch size 40 and Adam optimizer with learning rate \( \epsilon = 0.001 \). In order to prevent overfitting, we use dropout with rate 25\%. For 2D problem, we use 2D convolutions with size 3\times3 and 2\times2 max pooling layers. In 3D problem, we use a same idea as 2D with 3\times3\times3 convolutions and 2\times2\times2 max pooling layers. The numerical calculations of the effective properties has been implemented with the open-source finite element software FEniCS [6]. Construction of the machine learning algorithm is based on the library Keras [4].

Acknowledgements
The work is supported by the mega-grant of the Russian Federation Government (N 14.Y26.31.0013).

References
[1] Allaire G 1992 Homogenization and two-scale convergence SIAM Journal on Mathematical Analysis 23 (6) 1482–1518
[2] Bakhvalov N and Panasenko G 1984 *Homogenisation averaging processes in periodic media* (Springer)
[3] Chan S and Elsheikh A H 2018 A machine learning approach for efficient uncertainty quantification using multiscale methods Journal of Computational Physics 354 493–511
[4] Chollet F et al. Keras: Deep learning library for theano and tensorflow URL: https://keras.io/
[5] Chung E T, Efendiev Y, Li G and Vasilyeva M 2016 Generalized multiscale finite element methods for problems in perforated heterogeneous domains Applicable Analysis 95 (10) 2254–2279
[6] Logg A, Mardal K-A and Wells G 2012 *Automated solution of differential equations by the finite element method: The FEniCS book, volume 84* (Springer Science & Business Media)
[7] Srisutthiyakorn N 2016 Deep learning methods for predicting permeability from 2-d/3-d binary segmented images In SEG International Exposition and 86th Annual Meeting, Dallas, Texas, USA 16–21
[8] Sudakov O, Burnaev E and Koroteev D 2018 Driving digital rock towards machine learning: predicting permeability with gradient boosting and deep neural networks. arXiv preprint arXiv:1803.00758
[9] Talonov A and Vasilyeva M 2016 On numerical homogenization of shale gas transport Journal of Computational and Applied Mathematics 301 44–52