Autoencoder Based Optimization for Electromagnetics Problems

S. Barmada, N. Fontana, D. Thomopulos, and M. Tucci

DESTEC, University of Pisa
56122, Pisa, Italy
sami.barmada@unipi.it, nunzia.fontana@unipi.it, dimitri.thomopulos@unipi.it, mauro.tucci@unipi.it

Abstract **─** In this work a novel approach is presented for topology optimization of electromagnetic devices. In particular a surrogate model based on Deep Neural Networks with encoder-decoder architecture is introduced. A first autoencoder learns to represent the input images that describe the topology, i.e., geometry and materials. The novel idea is to use the low dimensional latent space (i.e., the output space of the encoder) as the search space of the optimization algorithm, instead of using the higher dimensional space represented by the input images. A second neural network learns the relationship between the encoder outputs and the objective function (i.e., an electromagnetic quantity that is crucial for the design of the device) which is calculated by means of a numerical analysis. The calculation time for the optimization is greatly improved by reducing the dimensionality of the search space, and by introducing the surrogate model, whereas the quality of the result is slightly affected.

Index Terms **─** Deep neural networks, surrogate model, topology optimization.

I. INTRODUCTION

Design optimization of electromagnetic (EM) devices based on field computation is nowadays of interest both for research and industries. The conventional approach usually faces the following main challenges:

- It is often difficult to set an adequate design space that includes a solution with satisfactory performance, as the design variables introduced by the user restrict the ability of the optimization models to deal with any arbitrary change in the design of a machine;
- when numerical models are used to calculate the EM fields (i.e., in the majority of the cases since an analytical solution is rarely available), the computational burden resulting from repeated simulations is often excessive.

In some specific cases, when the optimization is not limited to a reduced set of parameters, the first problem can be overcome by topology optimization [1], which does not require the definition of the design variables. In fact, geometries and materials are flexibly represented using a bitmap approach, which describes the device (or the part of the device that needs to be optimized) as a set of pixels. In addition, different materials could be represented by different colors (or grayscale levels). This allows free modification of material boundaries, that could be characterized also by the appearance of holes in the design region, resulting in new shapes which may outperform conventional design. The remarkable drawback is the increased dimensionality of the optimization search space, related to the bitmap resolution and color space.

The second problem has led to the development of several surrogate models to aid the optimization process [2], [3]. Extensive research has been carried out in the field of magnetic equivalent circuits and neural networks, based on curve fitting, to partially or completely bypass computing the field solution using numerical techniques (often Finite Element Analysis, FEA). Most of these methods are usually suitable for specific types of problems and describe systems with very few parameters, i.e., they suffer of the first issue.

Some preliminary studies used deep learning Convolutional Neural Networks (CNNs) as surrogate models for the computation of EM quantities [4]. In fact, CNNs have excellent capability in extracting relevant features from the input image and relating them to a desired output EM quantity. However, evolutionary optimization algorithms are not as well suited as deep neural networks to deal with high dimensional bitmaps as search space.

The motivation of this work is the need to reduce the dimensionality of the search space for topology optimization. In particular, we exploit the feature extraction capabilities of a CNN based autoencoder that learns from the space of input bitmaps, and the encoded space (also called latent space) is used as the search space for the optimization.

The main contributions of this paper are summarized as follows:

- The evolutionary optimization algorithm works in the latent space that represents the original high dimensional bitmap space almost perfectly;
- A new neural network surrogate model approach

Submitted On: September 25, 2019
Accepted On: December 6, 2019

is proposed and applied during optimization, reducing the time cost for calculating the numerical solution;

- The constraints are defined both in the encoded space and in the decoded (original) space.

The proposed method is applied here to a 2D test case, similar to the one shown in [5]: the shape of a "magnetic channel" is optimized, with the aim of maximizing the magnetic energy on a target zone. The 2D simulations are performed with a commercial code, and the results show that the proposed procedure can speed up optimization procedures.

II. AUTOENCODER FOR DIMENSIONALITY REDUCTION AND SURROGATE MODEL

A. Autoencoder for dimensionality reduction

The term autoencoder [6], shown in Fig. 1, is usually referred to an unsupervised neural network composed by an encoder, that maps the input space (usually of large dimension, for instance an image) to a reduced number of features, denoted as code or latent space, and a decoder that maps the latent variables back to the original data.

The dimensionality reduction (i.e., compression) made by the encoder is learned in order to minimize the error between the decoder output and original input, i.e., the reconstruction error. Then, the latent representation can be considered as a reduced feature space that fully describes the original high dimensional input space.

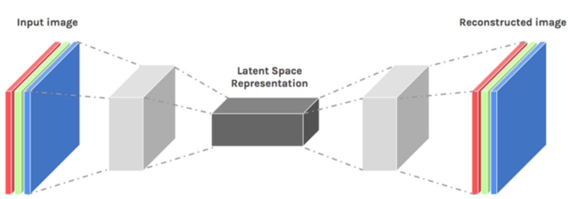


Fig. 1. Typical structure of an autoencoder.

The main idea of this proposal is to train the autoencoder with a proper set of bitmaps describing different geometries of the system to be optimized (i.e., different design solutions). At the end of the training period, the autoencoder has created a consistent representation in the latent space of the different geometries.

For the readers that might not be familiar with the structure of an autoencoder, it can be described, in its simplest form, by a set of equations; given one hidden layer, the encoder stage takes the input $\boldsymbol{x} \in \mathbb{R}^n$ and maps it to $\boldsymbol{h} \in \mathbb{R}^p$ according to:

$$\boldsymbol{h} = \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}), \quad (1)$$

where the image $\boldsymbol{h}$ is usually referred to as code, and the space of $\boldsymbol{h}$ is the latent space. With the same terminology usually used for neural networks, $\sigma, \boldsymbol{W}, \boldsymbol{b}$ respectively are the sigmoidal activation function, the weight matrix and a bias vector that will be learned during the training process.

The decode stage of the autoencoder maps $\boldsymbol{h}$ to the reconstruction $\boldsymbol{x}'$:

$$\boldsymbol{x}' = \sigma'(\boldsymbol{W}'\boldsymbol{h} + \boldsymbol{b}'), \quad (2)$$

in which $\sigma', \boldsymbol{W}', \boldsymbol{b}'$ are not necessarily related to the corresponding quantities of equation (2).

During the training phase, the autoencoder is trained to minimise the reconstruction error, explained as follows:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|^2 =$$
$$= \left\|\boldsymbol{x} - \sigma'\big(\boldsymbol{W}'\big(\sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})\big) + \boldsymbol{b}'\big)\right\|^2. \quad (3)$$

Once this is done, the optimization is performed in the latent space, hence working with a lower number of parameters to be optimized.

The main issue here is the lack of physical meaning of the latent space entries. For this reason, an additional surrogate neural network model is needed.

In order to well represent the input space the autoencoder needs to be trained with a large variety of geometries, including shapes that correspond to low performance designs. The latent space corresponding to the training data is then analyzed by means of determining the upper and lower bounds of each latent variable. These bounds are used in the following as constraints for the optimization, which will be performed in the space of the latent variables. It is important to note that decoder and encoder networks are tightly interconnected, and cannot be adopted separately, and that a properly trained autoencoder ensures univocity of mapping of training data.

B. Surrogate neural network model

For each image (design solution) of the training set we pre-calculated the corresponding EM quantity to be optimized by means of a FEA (but any other computational technique could be used). A second neural network based surrogate model is also trained using the latent representation of the corresponding geometry as input, and the desired quantity as output. This approach allows the surrogate model to benefit from the dimensionality reduction provided by the autoencoder.

The role of the surrogate model is to provide a fast prediction of the objective function, bypassing the expensive numerical computation. The surrogate model is trained offline before optimization, and it is also updated online using the new input-output pairs generated during the optimization process.

Figure 2 shows two neural networks architectures, in the typical graphical representation showing the inputs, the weights and the activation functions. In particular, the top part of Fig. 2 shows the autoencoder having dimensionality from 100 to 20 (these numbers are the ones used in the test case), while the bottom of Fig. 2 shows the surrogate model having as input the

20 variables of the decoder's latent space and as output the desired EM quantity T_{em} (a standard feed forward NN with one hidden layer, [7]). The whole structure represented in Fig. 2, is a Deep Neural Network architecture.

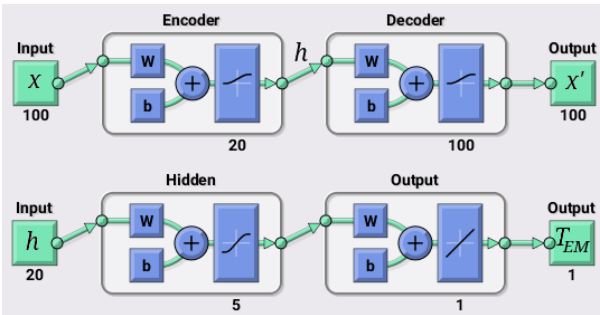


Fig. 2. Top: autoencoder; bottom: surrogate model structure.

III. OPTIMIZATION

As explained before, the optimization is carried out in the latent space, working on the reduced set of parameters. As in any optimization process, a fitness function has to be calculated at each step. The main advantage in using the proposed approach, is that at each iteration the calculation of the fitness function is not performed through a time-consuming numerical solution, but by the surrogate neural network model. This procedure might lead, though, to meaningless solution, because working in the latent space of the autoencoder does not allow the imposition of constraints to the physical variables.

For this reason, the authors propose an approach that will be explained later in this section.

In the literature, when dealing with topology optimization, Genetic Algorithms (GA) are often used; in this formulation, an evolutionary optimization algorithm, previously proposed by the authors is used [8], which is based on self-organizing maps, SOM, and denoted as self-organizing centroids optimization, SOC-opt. It was shown that SOC-opt outperforms many standard evolutionary optimization algorithms in a number of benchmarks. The algorithm uses a population of fixed size, and implements selection and mutation operators.

In the following descriptions, we will refer to a FEA, since this is the numerical method used in our test case; the set of the pre-calculated FEA solutions (that are used to train the autoencoder), are included in a set that we call *FEAdata*.

The optimization is carried out as follows:

1. Initialize the population in the latent space randomly within the bounds of the latent variables;
2. Provide each solution to the surrogate model in order to calculate the corresponding torque;
3. Calculate the new population of feasible solutions;
4. Divide the population in two subsets: *subset1* (eventually empty) contains individuals to be simulated with FEA, *subset2* the remaining population;
5. Decompress each individual of *subset1* to the corresponding bitmap using the decoder section of the autoencoder, provide the bitmap to the FEA software to calculate the fitness function, update the surrogate model with such solutions, add the solutions to *FEAdata*;
6. Provide each individual of *subset2* to the surrogate model in order to calculate the corresponding fitness function;
7. Iterate steps 3 to 6 until a stop criterion is verified.

The classifications of the individuals either in *subset1* or in *subset2* is carried out by means of the following heuristic strategy: a point is included in *subset2* if it is located in the *FEAdata* convex hull; on the contrary, if the individual it is outside the convex hull then the fitness function is evaluated through a regular FEA analysis (*subset1*). The criterion is graphically shown in Fig. 3.

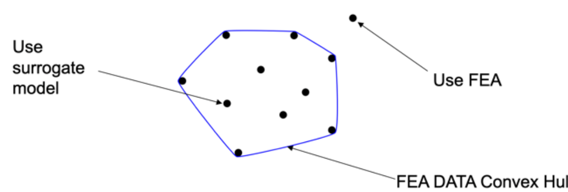


Fig. 3. Fitness function calculation criterion.

IV. TEST CASE

A. Description of the magnetostatics problem

The performance of the method has been tested on a simple magnetostatics test case, very similar to the test case shown in [5].

In particular, we perform the optimization of the distribution of the magnetic material $\mu_r = 1000$ in the design domain: practically speaking the objective of this optimization is maximizing the energy in the target domain finding the best feasible shape for the magnetic circuits. The source of the magnetic field is a permanent magnet characterized by a remnant flux density $B_r = 1T$, while the target domain is characterized by a rectangular shape of $\mu_r = 1$ above a rectangle of ferromagnetic material $\mu_r = 5000$. At first sight it is evident that the solution has to be in the form of a ferromagnetic channel connecting the permanent magnet and the target region. The presence of the iron below the target and the presence of the airgap between the design domain and the target make the problem non-trivial. Figure 4 shows a simple outline of the problem.

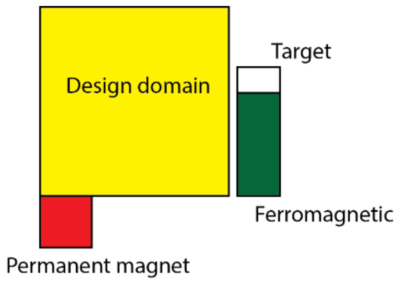


Fig. 4. Description of the test case.

B. Autoencoder training

The proper training set has been obtained by randomly generate 10^4 geometries, represented by 10x10 matrices in which each single entry can be either 1 or 0 (ferromagnetic material or vacuum). In order to have physically reasonable geometries, the following constraints have been imposed: a) the ferromagnetic material be characterized by a connected shape, b) 40% of the design space must be filled by the ferromagnetic material.

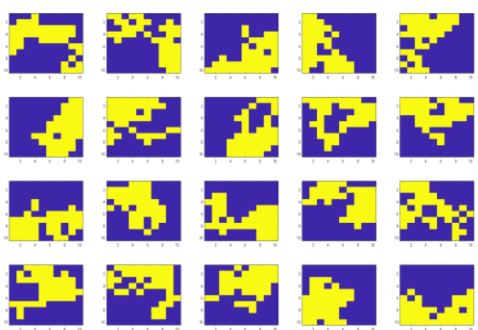


Fig. 5. Example of randomly generated geometries with the above mentioned constraints.

Starting from a dimension of 100 inputs (the number of “pixels” of each image), the latent space is characterized by a dimension of 20 variables and a 5-fold cross validation on the reconstruction error has been evaluated. Figure 5 shows few of the randomly generated geometries, while Fig. 6 shows the relative reconstructed images. In particular Fig. 6 is related to a continuous output (between 0 and 1): a proper threshold is then needed to move back to the materials discrete space (1 or 0). In this case the chosen threshold value is 0.5 and used in Figs. 8 and 9.

The ability of the autoencoder to well represent the original information can be easily verified.

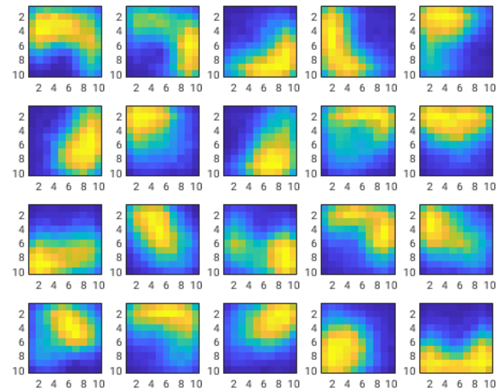


Fig. 6. Reconstructed geometries.

C. Neural network surrogate model training

The neural network surrogate model should be able to estimate the energy in the target zone from the geometry as represented in the latent space: for this reason, the input to the neural network has dimension 20 (latent space variables), while the output has dimension 1 (energy in the target area).

The same randomly generated 10^4 geometries have been simulated with Comsol, a commercial FEM software: each simulation (that includes magnetic field calculation and the evaluation of the magnetic energy in the target area) takes about 2s on an Intel I7 – 6 cores 4.0 GHz CPU.

Figure 7 shows the accuracy of the surrogate model with respect to the results obtained by the FEM model for 1100 geometries that have not been used for the neural network training: it is evident that the output of the surrogate model is accurate, and it can be used in the optimization algorithm.

Each evaluation of the target energy by the use of the surrogate model costs about $40\mu s$, $5 \cdot 10^5$ times faster than the corresponding FEM solution.

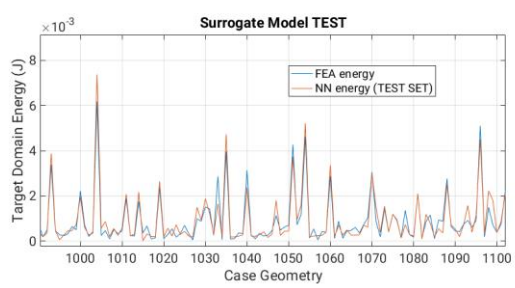


Fig. 7. Comparison between target energy calculated by FEM and by neural network model.

D. Optimization procedure results

The SOC-Opt optimization algorithm explained in

Section III has been implemented. The results relative to one optimization procedure (considering a stopping criteria of 10^5 global evaluations) are shown in Figs. 8 and 9. Figure 8 shows the field map and flux lines relative to the best solution among the original random geometries (target energy 0.0116 J), while Fig. 9 shows the same quantities relative the optimization procedure (target energy 0.01306J).

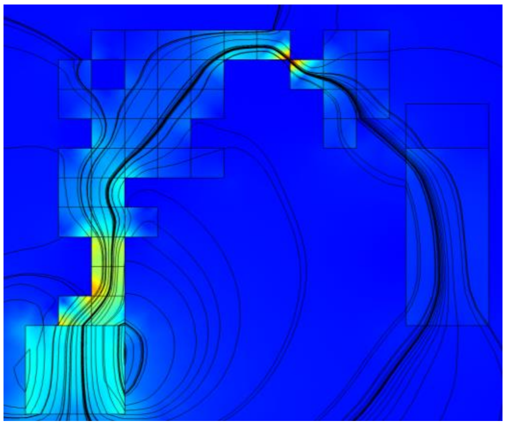


Fig. 8. Field map (B) and flux lines relative to the best solution among the initial pre-calculated random geometries.

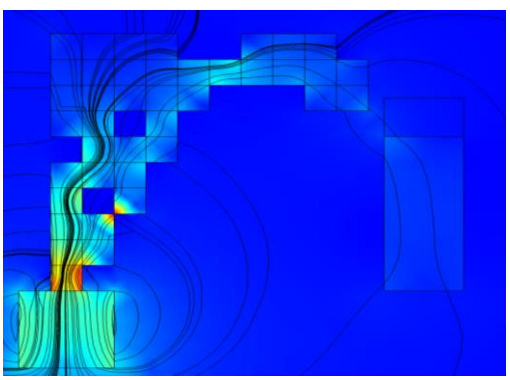


Fig. 9. Field map (B) and flux lines relative to the best solution after the optimization procedure.

The increase of the energy in the target region shows that the optimization procedure has reached its goal. There is no analytical solution to the problem, so we have no guarantee that the reached solution is a local or a global optimal (even though the SOC-Opt algorithm is robust in this point of view), and there might be different geometries giving practically coincident target energies. However, after numerous optimization procedures, the one shown in Fig. 9 is the best result obtained in terms of final energy.

Table 1 shows the CPU time required for training and for optimization, in which only 500 cases over 10^5 were outside the convex hull of the autoencoder (hence needed a FEA evaluation).

It is evident that the cost of the initial pre-calculated solutions is not negligible and it is a price to pay whenever neural networks to be trained are present. In this case, given the specific problem (basically no geometrical shape constraint) the number of iterations of the optimization procedure is one order of magnitude higher than the pre-calculated FEA solution, hence the final CPU time effort is positively affected by the use of proposed technique

Table 1: CPU time for training and optimization

Evaluations	**Time**
10^4 FEA Solutions (pre-calculated)	$2 \cdot 10^4 s$
99500 surrogate model (optimization)	$4s$
500 FEA solutions (optimization)	1000s

V. CONCLUSION

Optimizing an EM device in the latent representation space of an autoencoder has shown to be a promising approach, allowing the flexibility of topology optimization and reducing the dimensionality of both the search space and the surrogate model. Through the decoder it is possible to observe the solutions, and the introduction of a surrogate model approach, which also works in the latent space, reduces the number of required FEA simulations. Further work will be devoted to study the potential application in the case of multiobjective optimization.

REFERENCES

[1] T. Sato, K. Watanabe, and H. Igarashi, “Multimaterial topology optimization of electric machines based on normalized Gaussian network,” *IEEE Transactions on Magnetics*, vol. 51, no. 3, pp. 1-4, Mar. 2015.

[2] B. Xia, Z. Ren, and C. Koh, “Utilizing Kriging surrogate models for multi-objective robust optimization of electromagnetic devices,” in *IEEE Transactions on Magnetics*, vol. 50, no. 2, pp. 693-696, Feb. 2014.

[3] A. Khan, V. Ghorbanian, and D. A. Lowther, “Deep Learning for magnetic field estimation,” *Proceedings of 2018 IEEE Conference on Electromagnetic Field Computation (CEFC)*, Hangzhou, China, 2018.

[4] H. Igarashi and H. Sasaki, “Topology Optimization Accelerated by Deep Learning,” *Proceedings of 2018 IEEE Conference on Electromagnetic Field Computation (CEFC)*, Hangzhou, China, 2018.

[5] R. Kawamata, S. Wakao, N. Murata, and Y.

Okamoto, “Development of Encoder-Decoder Predicting Search Process of Level-set Method in Magnetic Circuit Design,” *Proceedings of Compumag. 2019*, Paris, France, July 16-19, 2019.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, vol. 1, Cambridge: MIT Press, 2016.

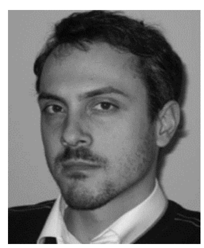
[7] S. Haykin, *Neural Networks*, *A Comprehensive Foundation*, Prentice Hall, 1998

[8] S. Barmada, M. Raugi, and M. Tucci, “An evolutionary algorithm for global optimization based on self-organizing maps,” *Engineering Optimization*, vol. 48, no. 10, pp. 1740-1758, Jan. 2016.

Sami Barmada received the M.S. and Ph.D. degrees in Electrical Engineering from the University of Pisa, Italy, in 1995 and 2001, respectively. He currently is Full Professor with the Department of Energy and System Engineering (DESTEC), University of Pisa. His teaching activity is related to circuit theory and electromagnetics. His research activity is mainly dedicated to applied electromagnetics, EM fields calculation, power line communications and wireless power transfer devices, non destructive testing. He is author and coauthor of approximately 100 papers in international journals and refereed conferences. Prof. Barmada was the recipient of the 2003 J. F. Alcock Memorial Prize, presented by the Institution of Mechanical Engineering, Railway Division, for the Best Paper in Technical Innovation, he is IEEE Senior Member and ACES Fellow. He served as ACES President from 2015 to 2017, he is a member of the International Steering Committee of the CEFC Conference and he has been general chairman and technical program chairman of international conferences.

Nunzia Fontana was born in Agrigento, Italy, in 1984. She received the M.Sc. degree (summa cum laude) in Telecommunications Engineering and the Ph.D. in Remote Sensing from the University of Pisa, Pisa, Italy, in 2008 and in 2012, respectively. In 2011, she carried out an internship with Queen Mary University of London, London, U.K. From 2012 to 2016, she was a Post-Doctoral Researcher with University of Pisa. From 2016 to 2019, she was a Researcher with the Italian Inter-University Consortium for Telecommunications and she was External Lecturer in biomedical engineering at University of Pisa. She is currently an Assistant Professor with the University of Pisa. Her current research interests include: Wireless Power Transfer; antennas, impedance matching networks design, prototyping and RF testing; radio frequency coils design for magnetic resonance and RF testing and bioelectromagnetics. Her research activity has been published in several international scientific journals and in a number of international conference proceedings.

Dimitri Thomopulos was born in Bologna, Italy, in 1987. He received the M.Sc. degree in Business and Industrial Engineering and Ph.D. in Automatic Control Systems and Operational Research from the University of Bologna, Bologna, Italy in 2012 and in 2016, respectively. From 2017 to 2018, he was a Post-Doctoral Researcher at LIX CNRS - École Polytechnique, Palaiseau, France. He is currently Research Fellow at the University of Pisa. His current research interests include: combinatorial optimization, non-linear optimization, mostly applied to energy problems; power-systems, micro-grids, hydro unit commitment, and wind farm optimization; computational intelligence and big data analysis, with applications in electromagnetism.



Mauro Tucci received the Ph.D. degree in Applied Electromagnetism from the University of Pisa, Italy, in 2008. Currently, he is an Associate Professor with the Department of Energy, Systems, Territory and Constructions Engineering, University of Pisa. His research interests include computational intelligence and big data analysis, with applications in electromagnetism, non destructive testing, powerline communications, forecasting and smart grids. He is author and coauthor of approximately 80 papers in international journals and refereed conferences, he is IEEE Senior Member and he served as Technical Program Chair of ACES 2017 conference.