

Neural Metamodelling of Fields: Towards a New Deal in Computational Electromagnetics

Paolo Di Barba
Dept. of Electrical, Computer and
Biomedical Engineering
University of Pavia
Pavia, Italy
paolo.dibarba@unipv.it

Maria Evelina Mognaschi
Dept. of Electrical, Computer and
Biomedical Engineering
University of Pavia
Pavia, Italy
eve.mognaschi@unipv.it

Slawomir Wiak
Institute of Mechatronics and
Information Systems
Lodz University of Technology
Lodz, Poland
slawomir.wiak@p.lodz.pl

Abstract—In computational electromagnetism there are manyfold advantages when using machine learning methods, because no mathematical formulation is required to solve the direct problem for given input geometry. Moreover, thanks to the inherent bidirectionality of a convolutional neural network, it can be trained to identify the geometry giving rise to the prescribed output field. All this puts the ground for the neural meta-modeling of fields, in spite of different levels of cost and accuracy. In the paper it is shown how CNNs can be trained to solve problems of optimal shape synthesis, with training data sets based on finite-element analyses of electric and magnetic fields. In particular, a concept of multi-fidelity model makes it possible to control both prediction accuracy and computational cost. The shape design of a MEMS design and the TEAM workshop problem 35 are considered as the case studies.

Keywords—Surrogate models, convolutional neural network, field analysis and synthesis.

1. Introduction

In recent years, machine learning has been increasingly considered in engineering science. Electromagnetics is no exception and deep learning techniques begin to be used for field analysis purpose; for instance, a coupled field problem is solved in [1], while the simulation of Maxwell's equations are reported in [2],[3]. In particular, the impressive developments in the area of machine learning and big data have put the ground for pattern recognition in field problems. In this respect, the use of Deep Neural Networks (DNN) [4] might substantially accelerate the solution of field analysis problems. Accordingly, the possibility of applying a DNN models to field analysis and synthesis is here considered: in particular, a bitmap approach, which describes a device geometry as a set of pixels, is proposed. The topology of the selected network resembles that for image segmentation purposes and it is easily trained by means of a set of Finite-Element (FE) analyses: this way, the ground information is amenable to electric and magnetic field distributions over the input domains, simulated from classical FE solvers. Therefore, a physics-informed background makes the neural surrogate reliable in spite of different levels of cost and accuracy. A particularly interesting and promising architecture is the one of a Generative Adversarial Network (GAN) [5], composed of two interconnected Convolutional Neural Networks CNNs, named generator and discriminator, respectively. In fact, GANs have already been successfully applied in many areas, like e.g. image generation and image super-resolution. In electromagnetics, several works show their use in the area of magnetic resonance imaging [6],[7], antenna Q-factor characterization [8], scattering problems [9].

The paper is organized as follows: in Section 2 the general concept of surrogate modelling is presented, while Section 3 shows possible perspectives in field analysis and synthesis open by the use of CNNs; reference is made to the optimal shape synthesis of a Micro Electro Mechanical System MEMS device. In turn, the focus of Sections 4 and 5 is on a GAN architecture with a case study inspired to the TEAM workshop problem 35. Eventually, in Sections 6 and 7 results are presented and discussed.

2. Surrogate Modeling with CNNs

Recently, deep learning techniques have been used to build surrogate models for field analysis of electromagnetic devices [10],[11]. In this respect, deep learning (DL) approaches are particularly suited for developing surrogate models in electromagnetic analysis. In fact, DL models like deep neural networks (DNN), are capable of dealing with high-resolution images as input and output, therefore enabling to directly handle the bitmaps describing geometry and field in a electromagnetically complex scenario. In particular, Convolutional Neural Networks (CNNs) are neural networks which allow to evaluate a field quantity (either a value or a vector of values like e.g. a field distribution), starting from an image [12]; to this end, transposed 2D deconvolution layers are conveniently used. The training of this class of networks is usually based on numerical analyses and in particular the Finite Element (FE) method is used. Once a neural network is trained, it can be used as a surrogate model and placed in an optimization loop. This way, the computational burden is limited to the net training, while the evaluation of the field quantities during the optimization loop is rather inexpensive. Another fascinating and innovative way to use DNNs for solving inverse problems is to invert the network previously trained for solving the associated direct problem: an application in material property identification can be found e.g. in [13].

3. Field analysis and synthesis with CNNs

It appears that more and more Machine Learning (ML) approaches are being exploited as a tool for the resolution of electromagnetic (EM) problems. The advantages of using ML are at least twofold:

i) first, no mathematical formulation is required to implement the problem solver, as Deep Neural Networks (DNN) simply require well suited data sets for their training;

ii) next, once trained, DNNs can solve an EM problem in an almost negligible time with respect to classical numerical methods: this puts the ground for equation-less, and so FEM-less, models of analysis problems.

An additional advantage of using ML is the possibility to exploit the inherent bidirectionality of this approach. In fact, DNN's can be trained in view of a threefold scope:

- to promptly solve the direct electromagnetic problem for given input geometry, sources and material properties;
- or, alternatively, to look for the geometry giving the prescribed output field with assigned sources and material properties;
- or, finally, to look for field sources or material properties in an assigned geometry.

This, in turn, puts the ground for an optimization-less approach to the automated synthesis of electromagnetic devices. The whole scenario is tempting and surely deserves serious consideration in the community; at the time being, however, the training cost of a network is still a bottleneck in many EM applications, and future work should be addressed towards techniques mitigating the computational burden.

For the sake of an example, let an electrostatic micromotor exhibiting 18 stator electrodes, 6 rotor teeth and 60 μm outer rotor radius, be considered. The stator electrodes are supplied by a three-phase system of square voltages while the rotor potential is floating. A convolutional neural network (CNN) composed of 18 layers is used for implementing the surrogate model [14]; in fact, the architecture of the selected network resembles that of an image decoder/decompressor. In order to train a net, a 2D Finite Element Model (FEM) of the motor is implemented; the driving torque T at no-load condition is computed over a rotation angle of 60 degrees (pole pitch), and its maximum value T_m is considered.

The rotor geometry can vary depending on three parametric variables $x=[x_1, x_2, x_3]=[R_1, \alpha, \beta]$ which are the inner rotor radius and two angles defining the geometry of the rotor tooth, respectively. Two classes of problems have been solved: in terms of forward modelling, given a device geometry, identify the field pattern (CNN-1) or compute the torque profile (CNN-2) as shown in Fig.1. In contrast, in terms of inverse modelling, given the prescribed torque profile, find the relevant device geometry (CNN 3) as depicted in Fig. 2. In particular, CNN-3 is the conjugate network of CNN-2.

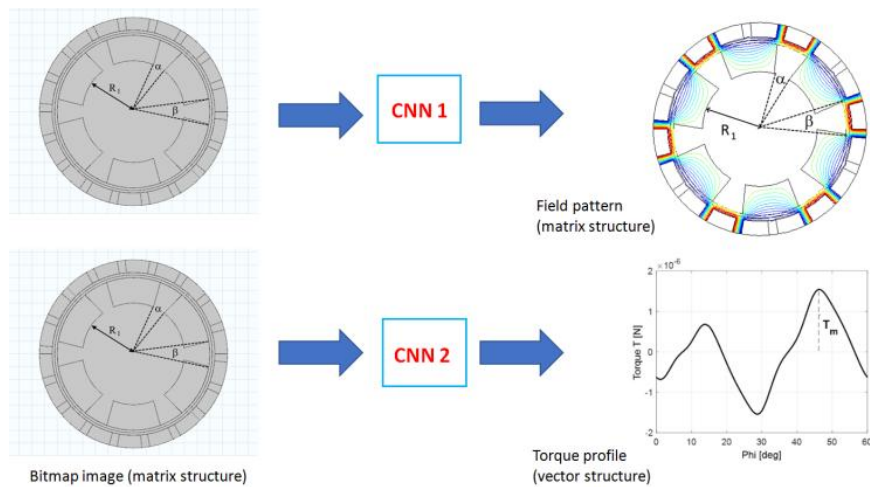


Fig. 1 – CNN-based surrogate model of forward problems: given a micromotor geometry, find the relevant field pattern (up) or, alternatively, find the torque profile (down).

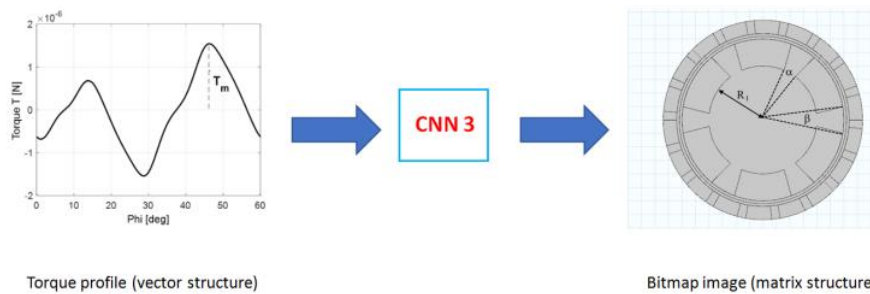


Fig. 2 – CNN-based surrogate model of an inverse problem: given a prescribed torque profile, find the relevant micromotor geometry.

The aforementioned example of the MEMS motor is intentionally a conceptual one in order to give the reader an idea how to utilize machine learning methods to solve problems of optimal shape synthesis.

A general strategy in order to reduce the computational burden and still keep a good accuracy in the predicted values, could be the use of a multi-fidelity approach: a network is trained first with a low-fidelity model; then, it is treated as a pre-trained network and subsequently specialized with the high-fidelity model. This way, the pre-trained network has a low, or moderate, computational cost because of the low-fidelity model, while the subsequent training with high-fidelity model is not very expensive because a reduced set of samples is used for training the network.

Formally, a multi-fidelity surrogate fuses information from different models based on a space mapping σ which relates the high- (x_H) and the low-fidelity (x_L) variable spaces, namely

$$x_L = \sigma(x_H) \quad (1)$$

such that

$$f_L(\sigma(x_H)) \cong f_H(x_H) \quad (2)$$

where f_L and f_H are the low and the high fidelity models, respectively.

For the sake of an example, again referring to the previous case study, the high-fidelity field model consists of $n_H=101$ FE analyses, where a step of $\Delta_H=0.6$ degrees is considered for the rotation over 60 degrees; in contrast, $n_L=26$ FE analyses are performed for the low-fidelity field model, where a step of $\Delta_L=2.4$ degrees is considered for rotation. The multi-fidelity approach may consist of training a CNN with low-fidelity model and then refining the network with a set of high-fidelity samples. This new approach allows to obtain an accuracy close to the one obtained with the high-fidelity model, but the computational cost is far lower: about 30% of FE analyses less than using the high-fidelity approach. More details are given in [14].

This way, in general, a physics-informed background makes the neural surrogate reliable in spite of different levels of cost and accuracy. The main practical goal of using this kind of surrogate models is their speed which makes them suitable for optimization loops.

4. Generative Adversarial Networks

A generative adversarial network (GAN) is composed of two networks, i.e. a generator (G) and a discriminator (D). Given a training data set characterized by a probability density function PDF_t , network D is trained in such a way to maximize the probability of correctly classifying real data and artificial data; in turn, network G is trained in such a way to maximize the probability of D to classify artificial data as real data.

The final goal is to synthesise G as a good estimator of PDF_t : as a remarkable result, it can be mathematically proven that the goal is achieved when the probability density function featuring the generator PDF_g is equal to the one of the data set, i.e. $PDF_g = PDF_t$. At this point in time, G is able to generate an optimal solution in an inexpensive way.

In this paper, a GAN is trained exploiting the data derived from a multi-objective optimization problem solved with NSGA-II algorithm; in other words, a subset of the solutions originated during the optimization procedure form the data set. This way, both dominated and non-dominated solutions are supplied to network D: the former are labelled as *real* while the latter are labelled as *fake*. As far as the G network is concerned, it has to be triggered by means of an appropriate signal: here it has been decided to utilize a pulse of Gaussian white noise as the trigger which is a usual choice in the literature; in turn, the output of network G is an (ideally) non-dominated solution in the design space. This puts the ground for densifying the Pareto optimal set, originally approximated by means of the preliminary optimization procedure, at a zero marginal cost. Eventually, a feedforward neural network (multilayer perceptron) is trained using the same data set of the GAN, in order to approximate the forward problem solution. This allows to estimate in an inexpensive way the objectives function values corresponding to the non-dominated solutions generated by network G, without the need of additional field analyses. In Fig. 3 the implemented architecture is shown.

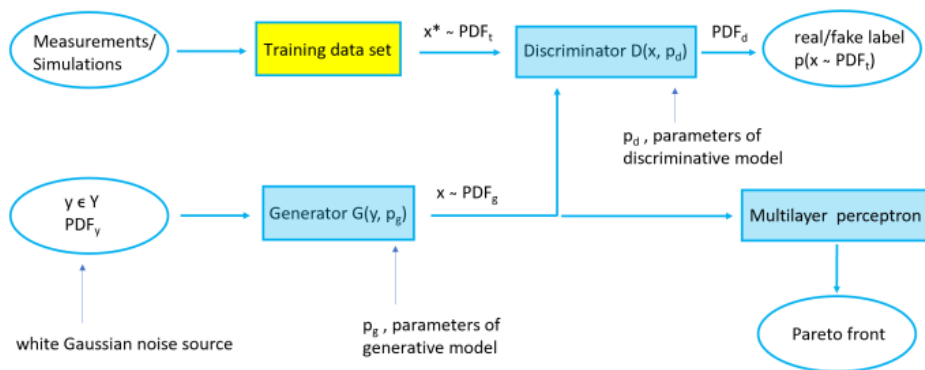


Fig. 3 – Architecture of the implemented GAN.

As far as the architecture of the two networks is concerned, the Generator is implemented with a CNN characterized by 13 layers, in which 3 blocks can be highlighted. Each block is composed of a transposed convolutional 2D layer [15], a batch normalization layer [16] and a Rectified Linear Unit (ReLU) function [4], as shown in Table I.

Table I - Generator architecture

Input	Feature input layer (size 10)
	Project and Reshape Layer (size $4 \times 4 \times 512$)
Block 1	Transposed Convolution 2D (size $3 \times 3 \times 128$)
	Batch Normalization
	ReLU activation function
Block 2	Transposed Convolution 2D (size $3 \times 3 \times 64$)
	Batch Normalization
	ReLU activation function
Block 3	Transposed Convolution 2D (size $3 \times 3 \times 32$)
	Batch Normalization
	ReLU activation function
	Transposed Convolution 2D (size $14 \times 32 \times 1$), output size $10 \times 1 \times 1$
	Tanh layer

Specifically, the Generator is triggered by a vector of random numbers, with a Gaussian distribution, and then a reshape layer originating a $4 \times 4 \times 512$ matrix is used. Three blocks, composed of Transposed Convolution 2D layer, Batch Normalization layer and the Rectified Linear Unit activation function, with a number of filters decreasing from block to block, follow. Finally, a Transposed Convolution 2D layer with a suitable filter size allows to obtain a $10 \times 1 \times 1$ matrix, which is in input to the Hyperbolic Tangent activation layer. As output of the Generator a $10 \times 1 \times 1$ matrix with values in the range $[-1, 1]$ is obtained.

In turn, the Discriminator is implemented with a CNN characterized by 14 layers, in which 3 blocks of the same kind described above can be highlighted, as shown in Table II.

Table II - Discriminator architecture

Input	Image input layer ($10 \times 1 \times 1$)
	Dropout (20% probability)
	Convolution 2D (size $3 \times 3 \times 64$)
	Leaky ReLU activation function
Block 1	Convolution 2D (size $3 \times 3 \times 128$)
	Batch Normalization
	Leaky ReLU activation function
Block 2	Convolution 2D (size $3 \times 3 \times 256$)
	Batch Normalization
	Leaky ReLU activation function
Block 3	Convolution 2D (size $3 \times 3 \times 512$)
	Batch Normalization
	Leaky ReLU activation function
	Convolution 2D (size $1 \times 1 \times 1$)

The size of the mini-batch processed by the GAN is set to 16. A mini-batch is a subset of the training set that is used to evaluate the gradient of the loss function and update the weights; one different mini-batch subset is used at each iteration.

The learning rate, i.e. the amount that the weights are updated during training, of the Generator and Discriminator is set to 10^{-4} and $2 \cdot 10^{-3}$, respectively.

5. TEAM Workshop Problem 35

The multi-objective Pareto optimization of a solenoid winding is considered as the case study [17-19]. The winding is composed of twenty series-connected circular turns, with the width of each turn $w = 1$ mm and the height $h = 1.5$ mm, carrying a current of 3 A (corresponding to a current density of 2 A mm^{-2}). Assuming a symmetric distribution with respect to the plane $z=0$, only ten turns are considered in the model (Fig. 4a).

A magnetic field problem is numerically solved using a finite-element axisymmetric model subject to the following boundary conditions: tangential flux lines at $r=0$ and normal flux lines at $z=0$. The magnetic field pattern corresponding to a given geometry of the winding is shown in Fig. 4a, while a detail of a typical mesh is shown in Fig. 4b.

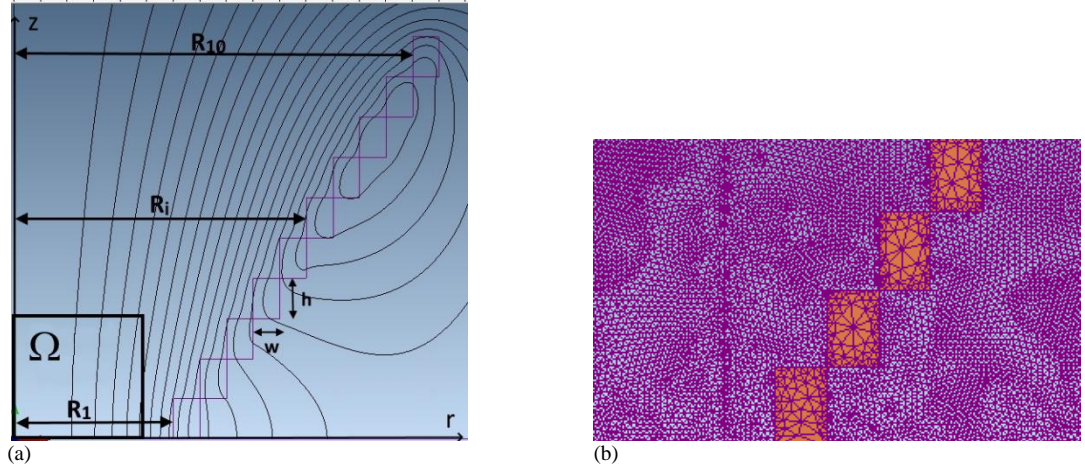


Fig. 4 - Geometry of the winding, control region and flux lines (a), a detail of the mesh (b).

The optimization problem consists of identifying the distribution of turn radii such that the field uniformity in a controlled subregion Ω around the solenoid axis is maximum and, simultaneously, the power loss in the winding is minimum. To this end, the radii $R_1, \dots, R_i, \dots, R_{10}$ of the ten turns (Fig. 4) are assumed to be the design variables, while the two objective functions can be cast as follows

$$f_1 = \sup_{j=1, n_p} \left| B_j - B_0 \right| \quad (3)$$

with $n_p=66$ sample points uniformly spaced inside Ω and $B_0=2$ mT, and

$$f_2 = \sum_{i=1}^{n_t} R_i \quad \text{with } n_t=10. \quad (4)$$

In (3), B_j is the z component of B field at the j -th sample point, while B_0 is the relevant prescribed value. In turn, function f_2 , which computes the sum of the turn radii, is amenable to the total ohmic resistance - and so power losses - of the winding. Both objective functions must be simultaneously minimised.

The subject of maximising the field uniformity in controlled subregion Ω by means of bi-objective formulation (3)-(4) is deeply discussed in [18]-[19] where the benchmark is defined and solved. Here, the attention is focused on a GAN-inspired approach to the solution of the benchmark with a specific scope, i.e. densifying a subregion of the objective space spanned by (f_1, f_2) when design vector $[R_1, \dots, R_k, \dots, R_{10}]$ varies in a feasible region.

In this respect, as far as objective functions (3)-(4) are concerned, the following remark can be put forward: function f_2 is inexpensive to calculate because it can be analytically computed based on the shape of the turn; in contrast, function f_1 needs a field analysis for the given winding geometry. Hence, in order to evaluate the function f_1 in a fast way, a classical feed-forward fitting neural network (FFNN) is implemented. It is composed of 6 layers with [20, 15, 5, 2, 2] neurons, respectively, and one output layer. The activation function used for all layers is the logistic sigmoid activation function, which sets the output in the range [0,1]. The whole set of solutions obtained with NSGA-II method (black dots in Fig. 5), based on FE analyses, is used for training the net, considering the 70% of samples for the training set, 15% for the validation set and 15% for the test set. For training the network the Levenberg-Marquardt method is used. For evaluating the neural network results, true and predicted points are plotted in Fig. 5

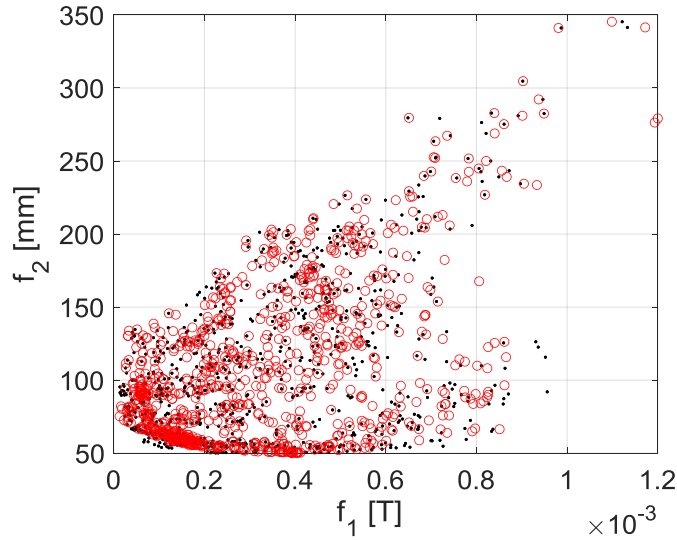


Fig. 5 – True points (NSGA-II results): black dots, predicted points (FFNN): red circles.

In turn, in order to train the GAN, a subset of the solutions originating during the optimization process based on NSGA-II algorithm are used, i.e. the Pareto front solutions along with the neighboring solutions. The relevant Pareto front is highlighted in Fig. 6a, where all the individuals processed during the optimization are represented with black dots, while the 78 individuals approximating the Pareto front are represented with red stars.

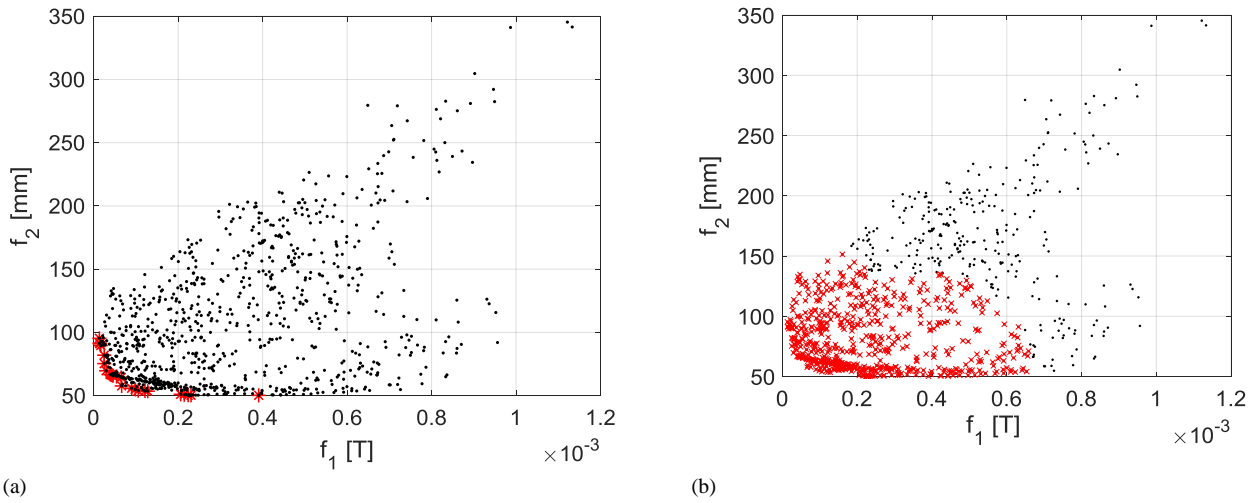


Fig. 6 - NSGA-II results: individuals processed during the optimization, black dot and approximation of the Pareto front, red star (a), the solutions considered for training the GAN are highlighted, red cross (b).

Because the number of solutions in the training set substantially affects the convergence of the GAN, a set of solutions larger than the Pareto front subset is used for training the GAN. In particular, a set of solutions close to the Pareto front has been selected. Specifically, considering a single Pareto optimal solution as the reference, a set of the k solutions closest to the reference one, are taken; subsequently, this procedure is repeated for each Pareto optimal solution. These solutions x^* are considered to be close to the Pareto optimal solutions based on the Minkowski's distance metric d_m of order $p=2$, namely

$$d_m = \left(\sum_{i=1}^n \left| \frac{f_i}{f_{\max_i}} - \frac{\bar{f}_i}{\bar{f}_{\max_i}} \right|^p \right)^{1/p} \quad (5)$$

with $n=2$ number of objectives and $p=2$. According to (5), taking $k=400$ and discarding the duplicate solutions, a cloud of 624 points is obtained. It is worth noting that the 624 points were originated by the FE solver during the NSGA-II optimisation. As can be noted in (5), each component (f_1 and f_2 , respectively) of the distance between two points in the objective function space is normalized by the maximum value f_{\max_i} of the relevant objective function. The set of solutions so found is represented in Fig. 6b with red cross.

6. Results

Because the discriminator learns to classify input vectors as either "real" i.e. belonging to the training set of solutions or "generated" by the Generator, the output of the Discriminator corresponds to the probability \hat{Y} that the input vectors belong to the class "real". The Generator score s_G is the mean of the probabilities corresponding to the Discriminator output for the generated images:

$$s_G = \text{mean}(\hat{Y}_G), \quad (6)$$

where \hat{Y}_G array contains the probabilities for the generated images to be classified as "real".

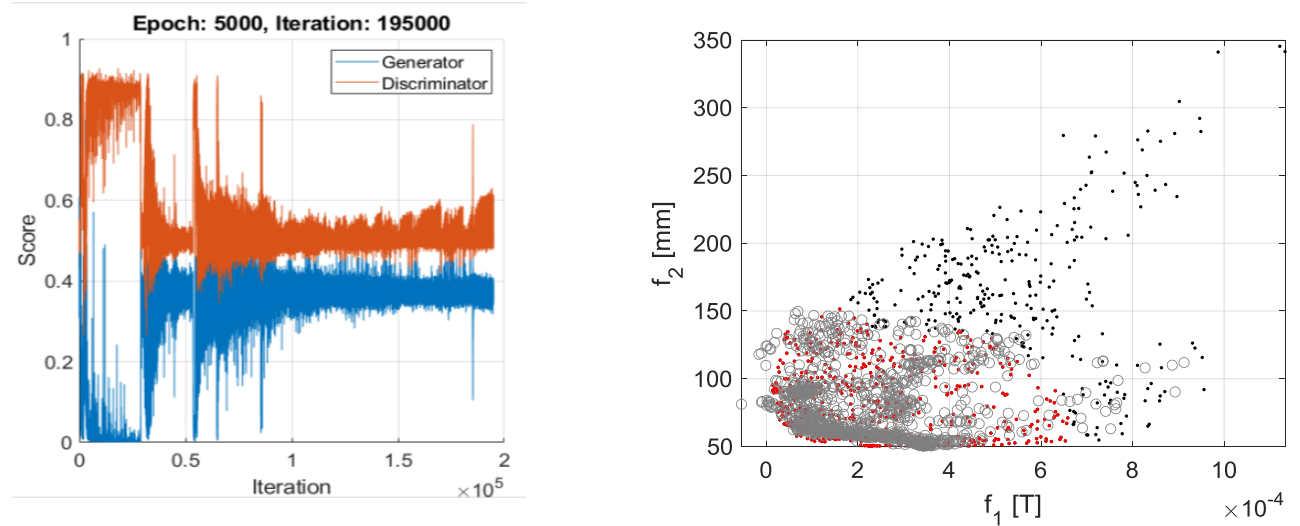
Given that $1 - \hat{Y}$ is the probability of a vector belonging to the class "generated", the Discriminator score is the mean of the probabilities of the input vectors belonging to the correct class:

$$s_D = \frac{1}{2} \text{mean}(\hat{Y}_R) - \frac{1}{2} \text{mean}(1 - \hat{Y}_G) \quad (7)$$

where \hat{Y}_R contains the discriminator output probabilities for the real vectors Eq. (7) is under the assumptions that the numbers of real and generated vectors passed to the discriminator are equal.

The GAN was trained for 5,000 epochs with 624 samples. By definition, one epoch lasts when an entire dataset is passed forward and backward through the neural network once. Since one epoch is too big to feed to the computer at once, the dataset is divided in several smaller batches: 64 samples per batches are here considered.

The history of Generator and Discriminator scores during the training are shown in Fig. 7a.



(a) (b)
Fig. 7 - GAN results: history of scores of Generator and Discriminator (a) and solutions generated by the trained GAN, grey circles, NSGA-II solutions used for GAN training, red dots, whole set of NSGA-II individuals, black dots (b).

After Fig. 7a it can be noted that both Generator and Discriminator explore different configurations - in particular this happens in the first half of the training history - and eventually a trend towards convergence is observed. In particular, the Discriminator score is around 0.5 and it is a good value, because it means that the Discriminator is no longer able to detect the difference between real and generated vectors.

After the GAN converged, the trained Generator is used by itself for generating 1,000 vectors, starting from 1,000 vectors of random numbers. Each generated vector belongs to the design space and contains the set of 10 radii defining the solenoid geometry. These vectors are then evaluated in the objective space with the FFNN previously trained for the computation of the f_1 function, while f_2 function is analytically computed. The results are shown in Fig. 7b; in particular the grey circles are the points generated by the Generator, mapped into the objective space through the FFNN.

The nice feature is that grey circles are inexpensively generated; this way, thousands of points can be generated with no extra cost for discovering new solutions. Consequently, the objective space and in particular the region of interest (red dots) is enriched of points and arbitrarily densified. This makes it possible to obtain thousands of solutions, i.e. different "good" geometries, which all belong to the area of interest at zero marginal cost.

7. Conclusion

According to the recent literature, CNNs exhibit good capabilities to work as surrogate models in electromagnetics, mimicking both field analysis and field synthesis by means of equation-less models. In fact, deep learning models are capable of dealing with high resolution images as input and output, thus enabling to directly handle the bitmaps describing geometry as well as field distribution of a complex device. On the one hand, in fact, the bitmap like approach allows for a much more flexible description of the design space in contrast to classical parametric approaches which are based on a finite-dimensional space; on the other hand however, the bitmap like approach substantially increases the dimensionality of the design space. To fix the problem, a possible remedy is the use of a variable fidelity model which makes it possible to decrease the computational cost subject to a prescribed accuracy. In turn, a GAN-inspired approach has proven to be an effective method for synthesising an almost perfect twin model of the solution space, which enables to discover new solutions at zero marginal cost. All this puts the ground for a new deal in computational electromagnetics.

References

- [1] R. Gong and Z. Tang, "Training Sample Selection Strategy Applied to CNN in Magneto-Thermal Coupled Analysis," in *IEEE Transactions on Magnetics*, vol. 57(6), 2021.
- [2] P. Zhang, Y. Hu, Y. Jin, S. Deng, X. Wu and J. Chen, "A Maxwell's Equations Based Deep Learning Method for Time Domain Electromagnetic Simulations," in *2020 IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS)*, 2020.
- [3] M. Baldan, G. Baldan and B. Nacke, "Solving 1D non-linear magneto quasi-static Maxwell's equations using neural networks," in *IET Science, Measurement and Technology* vol. 15(2), 2021, pp. 204-217.
- [4] I. Goodfellow, Y. Bengio, A. Courville: *Deep Learning*. The MIT Press, 2016.
- [5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative Adversarial Nets", *Advances in Neural Information Processing Systems*, 2014.
- [6] M. Tian and K. Song "Boosting Magnetic Resonance Image Denoising With GANs," in *IEEE Access*, vol. 9, 2021, pp. 62266-62275.
- [7] M. Jiang, Z. Yuan, X. Yang, J. Zhang, Y. Gong, L. Xia and T. Li "Accelerating CS-MRI Reconstruction With Fine-Tuning Wasserstein Generative Adversarial Network," in *IEEE Access*, vol. 7, 2019, pp. 152347-152357.
- [8] J. Vijayamohan, O. Noakoosteen, A. Gupta, M. Martinez-Ramon and C. G. Christodoulou "On Antenna Q-factor Characterization with Generative Adversarial Networks," in *IEEE Int. Symp. on Antennas and Propagation*, Montreal, Canada, 2020, pp. 1643-1644.
- [9] Z. Ma, K. Xu, R. Song, C. Wang and X. Cheng "Learning-Based Fast Electromagnetic Scattering Solver Through Generative Adversarial Network," in *IEEE Trans. Antennas Prop.*, vol. 69(4), 2021, pp. 2194- 2207.
- [10] Khan A., Ghorbanian V., Lowther D., "Deep Learning for Magnetic Field Estimation", *IEEE Trans. on Magnetics*. 2019;55(6):7202304.
- [11] Khan A., Mohammadi M.H., Ghorbanian V., Lowther D., "Efficiency Map Prediction of Motor Drives Using Deep Learning", *IEEE Trans. on Magnetics*. 2020;56(3):7511504.
- [12] S. Barmada, N. Fontana, A. Formisano, D. Thomopoulos, M. Tucci, "A Machine Learning Surrogate Model for Topology Optimization", *Proc. CEFC-2020*, November 16-18, 2020.
- [13] M. Baldan, P. Di Barba, B. Nacke. Magnetic properties identification by using a bi-objective optimal multi-fidelity neural network. *IEEE Transactions on Magnetics*, vol.57, no.6, 2021.
- [14] P. Di Barba, M.E. Mognaschi, S. Wiak, "CNN-based Surrogate Model of the Electrostatic Field for MEMS: a Multi-fidelity Approach", *Proc. ISEF-2021*, September 20-23, 2021.
- [15] V. Dumoulin and F. Visin, A guide to convolution arithmetic for deep learning, arXiv preprint arXiv:1603.07285, 2016, pp. 1–28.
- [16] S. Ioffe and C. Szegedy. Batch normalization: accelerating deep network training by reducing internal covariant shift. *32nd International Conference on Machine Learning*, ICML, 1, 2015, pp. 448-456.
- [17] <https://www.compumag.org/wp/team>
- [18] Di Barba, P., Mognaschi, M.E., Lowther, D.A., Sykulski, J.K., A benchmark TEAM problem for multi-objective pareto optimization of electromagnetic devices, *IEEE Transactions on Magnetics*, 54 (3), 2018.
- [19] Di Barba, P., Mognaschi, M.E., Lowther, D.A., Sykulski, J.K., Improved solutions to a TEAM problem for multi-objective optimization in magnetics, *IET Science, Measurement and Technology*, 14 (8), 2020, pp. 964-968.