

Deep learning as a tool for inverse problems resolution: a case study

Sami Barmada

Department of Energy, Systems, Territory and Constructions Engineering (DESTEC), University of Pisa, Pisa, Italy

Alessandro Formisano

Department of Industrial and Information Engineering, Università degli Studi della Campania "Luigi Vanvitelli", Aversa, Italy, and

Dimitri Thomopoulos and Mauro Tucci

Department of Energy, Systems, Territory and Constructions Engineering (DESTEC), University of Pisa, Pisa, Italy

Abstract

Purpose – This study aims to investigate the possible use of a deep neural network (DNN) as an inverse solver.

Design/methodology/approach – Different models based on DNNs are designed and proposed for the resolution of inverse electromagnetic problems either as fast solvers for the direct problem or as straightforward inverse problem solvers, with reference to the TEAM 25 benchmark problem for the sake of exemplification.

Findings – Using DNNs as straightforward inverse problem solvers has relevant advantages in terms of promptness but requires a careful treatment of the underlying problem ill-posedness.

Originality/value – This work is one of the first attempts to exploit DNNs for inverse problem resolution in low-frequency electromagnetism. Results on the TEAM 25 test problem show the potential effectiveness of the approach but also highlight the need for a careful choice of the training data set.

Keywords Deep learning, Topology optimization, Electromagnetic inverse problems, Optimization, Surrogate optimization

Paper type Research paper

1. Introduction

In recent years, various deep learning (DL) approaches have been proposed for solving computationally demanding or difficult-to-model problems in several fields. A typical example of application is the problems inherent to electromagnetism (EM), in which it is often not possible to analytically describe the output (Barmada *et al.*, 2020; Barmada *et al.*, 2021; Khan *et al.*, 2019; Sasaki and Igarashi, 2018). The main advantage of using DL approaches is, in fact, the ability to obtain, in almost negligible time, efficient solutions without the necessity of a specific mathematical formulation, but only an appropriate data



set for the training of the neural networks (NNs). Note that the accuracy of the solution is not always granted; for this reason, the combined use of DL with standard numerical models in optimization procedures is highly recommended to improve effectiveness. Anyway, an additional advantage of using DL is the possibility to exploit the inherent bidirectionality of this approach. In other words, NNs can be trained either to promptly solve the direct electromagnetic problem for given input geometry and sources, to look for the geometry giving the assigned electromagnetic output with assigned sources or, finally, to look for the sources from assigned fields and geometry. In fact, DL approaches have been gaining a constantly increasing interest as a tool for inverse problem resolution in recent years.

A possible definition of inverse problems, relevant for our discussion, can be cast as “the reconstruction of system characteristics, e.g. its inner structure, from observed or desired data”. These problems appear in various applications, such as medical imaging with X-rays (Jin *et al.*, 2017) or other electromagnetic sources (Liang *et al.*, 2020). Inverse problems are often found also in engineering applications, such as the detection of specific materials within other structures by analyzing only the surface, thus, avoiding invasive analysis (Snieder and Trampert, 1999). In the literature, we can observe a constant and continuous improvement of DL approaches applied to image processing or related problems. Several works have proven the effectiveness of classical algorithms (Antun *et al.*, 2020; Genzel *et al.*, 2020; Amjad *et al.*, 2018).

Since the inverse problems we are dealing with are classically formulated as the minimization of a reconstruction error, notable characteristics of such classical approaches are the need for regularization (as raw observed data are frequently compatible with multiple solutions) and the adoption of iterative processes to achieve the minimum error. Although DNNs, when properly trained, can provide a solution in a single step, much care must be given to the regularization of the problem inherently produced by the DL approach. As a matter of fact, the DNN proposes the solution most closely corresponding to the observed data among those considered in the training step. Consequently, DNN does provide an inherent regularization, ruled by the construction of the learning set and by the teaching algorithm: this point needs further investigation from the viewpoint of the authors.

Some notable works collect the literature inherent in DL approaches applied to inverse problems; however, they are mostly focused on image processing problems. Among the possible approaches, we can enumerate a classical deep neural network (DNN), i.e. an NN with more than two layers of neurons, for instance, in Amjad *et al.* (2019), where some regression generalization bounds, precisely obtained via DNN, are compared to classical sparse reconstruction algorithms. Instead, (McCann *et al.*, 2017) collect several convolutional neural networks (CNNs), which are feedforward NNs, to solve imaging problems such as denoising, reconstruction and super resolution. Liang *et al.* (2020) focus on medical applications of magnetic resonance imaging. Other models used in addition to CNN include recurrent neural networks, i.e. NNs where connections between nodes form a directed graph, and generative adversarial networks, i.e. two or more NNs that contest with each other into a game. The efficiency of these approaches is then verified from the point of view of computation in comparison to more classical algorithms. Lucas *et al.* (2018) add multilayer perceptron and autoencoders to the previously listed approaches, whereas (Bai *et al.*, 2020) focus on fully connected neural networks. Finally, Ongie *et al.* (2020) present a taxonomy of inverse problems depending on the type of supervision and knowledge of the corresponding “direct” problem. Although numerous works introduce DL approaches to solve direct problems in EM, contributions dealing with the inverse case are still rare. Recently, Pollok *et al.* (2021) have dealt with the inverse design of magnetic fields via standard CNNs.

In this contribution, following previous work (Barmada *et al.*, 2021) in which some of the authors proposed a topological optimization for the TEAM 25 problem using DNNs as a prompt direct solver, a DL approach is exploited as an inverse solver: starting from the field distribution, the geometrical characteristics of the system are obtained. The aim is to investigate how techniques that are performing well for the direct case are equally performing for the inverse case. In particular, we propose a DL model where the input geometry is represented as a bitmap, and we compare it with a classical machine learning (ML) approach, where the geometry is represented with a small number of parameters.

To the author's knowledge, this work represents one of the first studies of inverse problems in low-frequency EM using DL.

The main contributions of this paper can be identified as follows:

- a real electromagnetic inverse problem was taken into account (electromagnetic press);
- we performed 10000 FEM simulations, obtaining a large data set of accurate solutions for training the NNs, where each solution is given as the magnetic induction field B_x and B_y values in nine points, corresponding to a 25*36 geometry bitmap;
- we performed an accurate model selection of the number of neurons in the hidden layers using fivefold cross-validation (CV) over 8,000 samples;
- we performed the test analysis on 2,000 samples, obtaining very low errors both for direct (6%) and inverse (3.5%) problems, then demonstrating that the NNs can solve these problems very well;
- we show that the inverse problem (reconstruct geometry from magnetic fields) can be solved successfully in two ways using both DL and ML:
 - solving the direct problem and then using an optimization loop;
 - solving the inverse problem as is;
- we compared the two approaches both regarding their accuracy and their computational time and highlighting their advantages and disadvantages;
- we analyzed the behavior of the trained model when dealing with test data coming from a different generative model with respect to training data; and
- all of the results were supported by numerical analysis and statistically significant error calculations.

The remaining of the paper is organized as follows: in Section 2, we briefly introduce and discuss the approaches for inverse problems, and in Section 3, we discuss the used DNN architectures; in Section 4, we present the benchmark problem; in Sections 5 and 6 we discuss the model selection and the test results; finally, Section 7 presents the conclusion.

2. Proposed approaches for inverse problems

As mentioned above, solving an inverse problem means to obtain a system's specific characteristic from a set of available data; often, this data is actually obtained by a forward process. For general shape optimization problems, the inverse problem is the determination of a specific geometry, given the evaluation of an objective function (i.e. field values, mechanical quantities and energy). In typical engineering problems, the relationship between the geometry and the objective function is not easily obtainable and complex evaluations (i.e. numerical solutions of the system) should be performed.

In this contribution, the authors compare two different approaches for solving the inverse problem, which we call the direct problem approach (DPA) and the inverse problem approach (IPA). In both DPA and IPA, the problem consists in determining the geometry corresponding to a given desired field; hence, both approaches aim to solve the inverse problem, where DPA is actually an optimization problem. The schemes of the two approaches using DNNs are shown in Figure 1. In both cases (DPA and IPA), a data set to train the DNNs needs to be generated using finite element analysis (FEA) software. The training data set consists of randomly generated geometry profiles and the corresponding fields.

In DPA, a DNN model is trained to predict the fields given an input geometry, and it is used as a surrogate model inside an optimization loop, seeking the shape that corresponds to a given desired field map. In IPA, on the contrary, a DNN model is trained to predict the geometry given an input field distribution, and it is directly used to obtain the optimal shape corresponding to a given field map. Generally speaking, IPA is faster and straightforward, but it is often badly conditioned and highly under-determined, while DPA suffers less of these issues but requires longer computational times.

3. Deep learning and machine learning models

In this section, we focus on the DNN block of Figure 1. In particular, we propose a DNN architecture for the IPA similar to the one used for the DPA case in a previous work that, for simplicity of exposure, is reported below. Both in this work and in Barmada *et al.* (2021), the input geometry is represented as a binary image subject to some constructive constraints of shape, continuity and feasibility.

The main characteristic of the proposed DNN is the use of an autoencoder, which is indeed an NN composed of an encoder, which maps the input space into a representation with reduced features called latent space, and a decoder which maps the latent space back to the original space generating the reconstructed pattern. In particular, the autoencoder is trained to minimize the reconstructed error or, in general, a loss function $L(X, X') = \|X - X'\|^2$, where X is the input pattern, and X' is the reconstructed pattern.

In Barmada *et al.* (2021), the DPA was implemented as a DL model consisting of two parts, as depicted in Figure 2(a):

- (1) an encoder that reduces the input geometry to the latent space; and
- (2) a feed-forward NN (Goodfellow *et al.*, 2016) that is fed with the latent space variables and yields the corresponding field values as outputs.

In the computer science community, stacking an autoencoder with a NN is considered a DL paradigm that includes the dimensionality reduction step (Goodfellow, 2016). Another

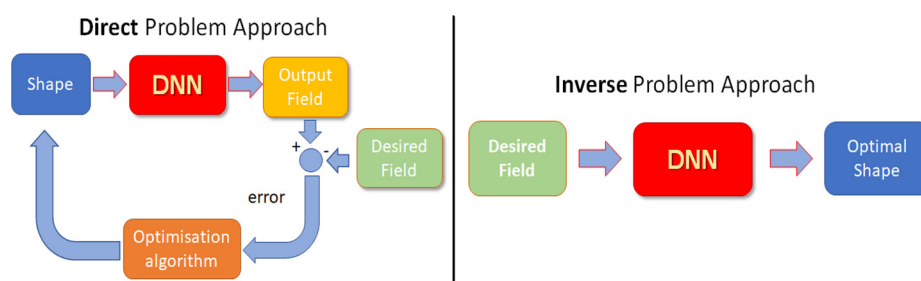


Figure 1. A representation of the approaches considered in the paper

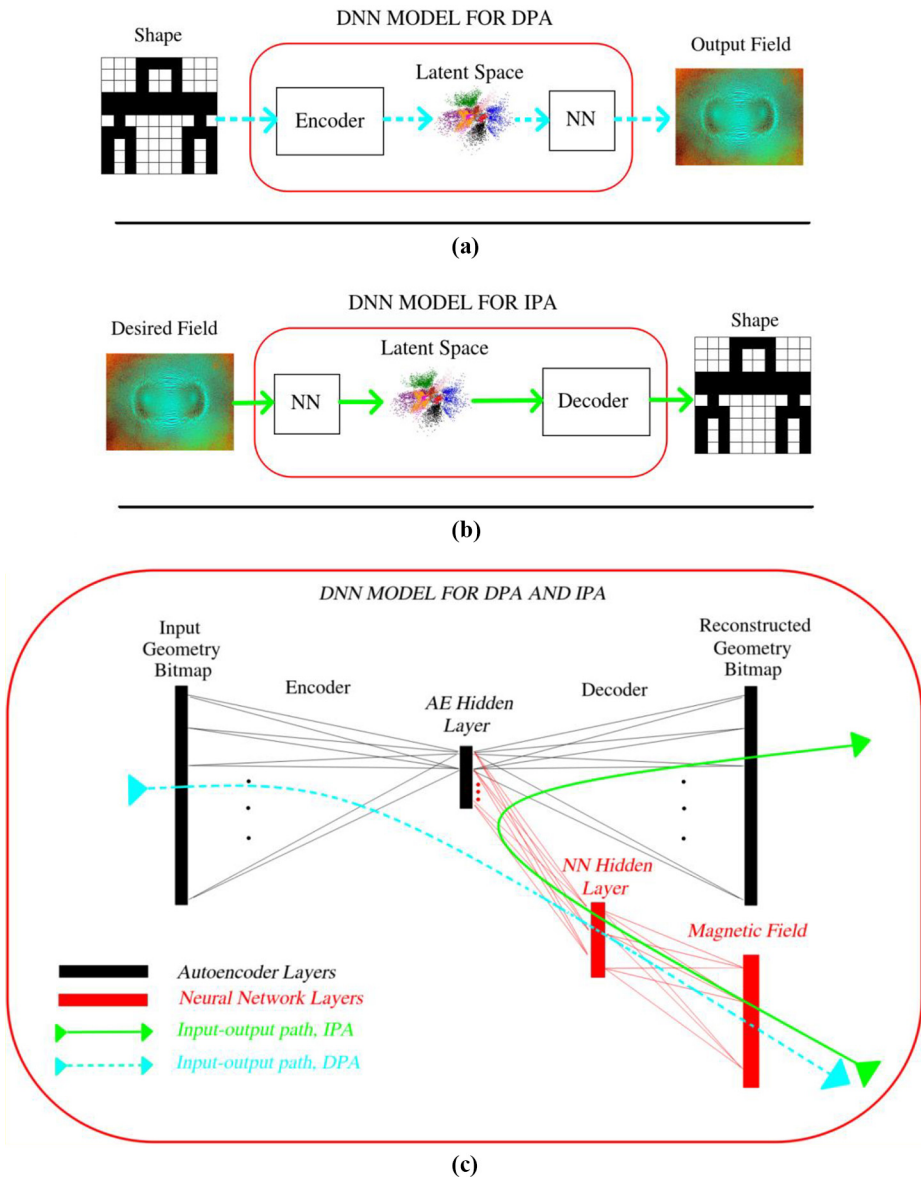


Figure 2.
(a) DL model for DPA, (b) DL model for IPA, (c) DNN model: NN stacked on autoencoder latent space

commonly accepted characteristic for a neural model to be defined as deep is the presence of at least two hidden layers, and the proposed model includes exactly two hidden layers, as shown in [Figure 2\(c\)](#).

The same DL approach is exploited in this work to solve the IPA, as shown in [Figure 2\(b\)](#) and (c). In particular, starting from the magnetic field values, it is possible to train an NN to obtain the latent variables as output (this NN is, of course, different with respect to the DPA case). Hence, we

use the second part of the autoencoder, the decoder, to obtain a final binary image representing the geometry [IPA input–output path in Figure 2(c)]. It is worth to note that the DPA and the IPA in the proposed DL model share the same autoencoder but require different NNs.

As an alternative, and for the sake of comparison, we also consider a case where the input image is nonrepresented with a bitmap but using a small set of geometry parameters. Using such a compact representation of the geometry limits the degrees of freedom but drastically reduces the computational complexity. We can thus avoid using DNN approaches and directly exploit standard ML models such as one hidden layer feed-forward NNs, which can be directly trained to map the relationship between geometry and fields (and vice-versa). With this approach, the intermediate dimensionality reduction to the latent space is not needed because the input geometry is now represented with a lower dimension. We denote this approach as ML model and, as in the previous case, this implementation can be applied to both problems IPA and DPA, as shown in Figure 3.

4. Benchmark problem description

The testing electromagnetic analysis methods (TEAM) represents an open international working group aiming to compare electromagnetic analysis computer codes (www.compumag.org/wp/team/). The TEAM problem 25 (Takahashi, 1996) is specifically designed to test shape optimization methods. This problem copes with a die press used to orient magnetic powders. An electromagnet is employed to generate a field in an air cavity, and iron die molds help get the required flux distribution. The problem statement asks to optimize the shape of the inner and outer die molds to obtain a uniform radial flux density distribution in a cavity (Figure 4 for a schematic drawing). Using the variables defined in Figure 4(c), the flux density along the line e-f must equal (large Ampere-Turns case):

$$B_x = 1.5\cos\vartheta(T); B_y = 1.5\sin\vartheta(T) \quad (1)$$

In the original problem statement, the degrees of freedom are represented by R1 (the inner mold radius) and by the size of L2, L3 and L4 segments in Figure 4(b). The authors focus

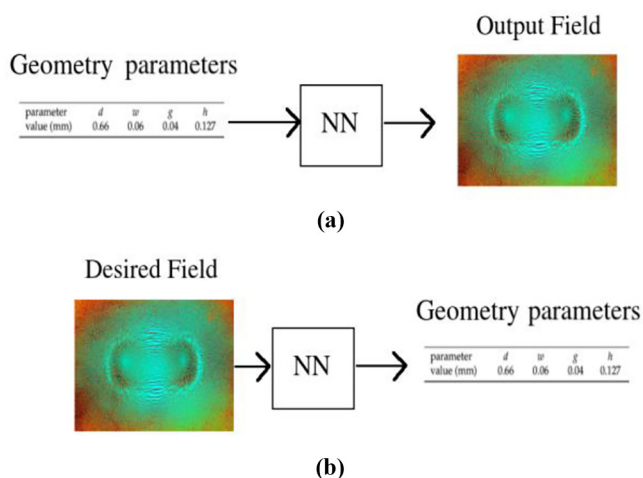


Figure 3.
(a) ML model for DPA, (b) ML model for IPA

their attention in this paper on the large Ampere Turns case because the i-j-k-m curve can be described (as stated in the TEAM problem description) as free curves.

To obtain a topology optimization problem, we fixed R1 in the model used to train the DNN to $R1 = 5 \text{ mm}$, whereas we used a “staircase” representation of the i-j-k-m line, with 25 steps of size 0.5 mm in the x direction and 36 steps of size 0.5 mm in the y direction.

Therefore, both in this work and in previous works, the right mold geometry is described using an image of resolution 36×25 pixels, i.e. 900 binary values subject to constructive constraints. The main constraint is that no iron is allowed inside the cavity.

The parametric geometry representation to be used in the ML case consists of 25 parameters of discrete values $x_n \in \{0, \dots, 36\}$, $n = 1$. Note that TEAM 25 is actually an optimization problem; we turned it into an inverse problem to test the DNN capabilities by simply considering as “observed” data the field components in the nine test points instead of considering the overall root medium squared error on the field in the nine points.

5. Model selection

Table 1 reports the dimensionality of input and output patterns in both DL and ML approaches.

To train the NNs, we generated a data set of 10,000 random profiles of size 36×25 fulfilling the problem constraints (no iron inside the cavity). The fields have been solved in terms of scalar magnetic potential by a FEA, and the 18 components of the magnetic flux (i.e. the values of the magnetic field distribution B_x and B_y in the nine target points) have been calculated straightforwardly.

To perform model selection, we used fivefold CV to determine the number of neurons of each NN, using only 8,000 out of 10,000 samples from the data set. We focused on the three models of Figures 2(b), 3(a) and (b) because the model of Figure 2(a) was analyzed in the previous work (Barmada *et al.*, 2021). All NNs, including the encoder and decoder networks, are implemented as one hidden layer, feed-forward NNs, with sigmoidal activation functions.

Figure 4.
(a) The press geometry, (b) a more detailed view of the die molds, (c) the iron BH curve

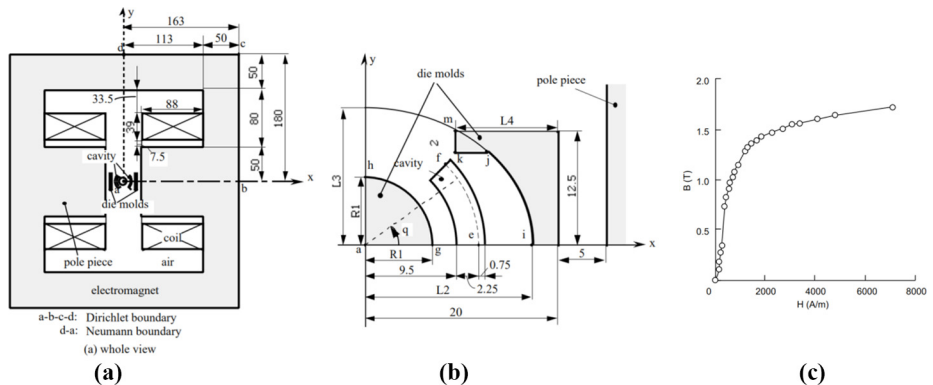


Table 1.
Dimension and type of input and output patterns

Pattern	Model	Dimensionality	Type
Fields in the cavity	DL and ML	18	Real
Geometry bitmap	DL	900	binary
Geometric parameters	ML	25	integer

As an error metric, we use mean absolute percentage error (MAPE), which is defined as the mean value over a sample set of the following absolute percentage error:

$$100 \times \left| \frac{\text{output} - \text{target}}{\text{target}} \right|,$$

where *out* is the output of the model for a given input pattern, and the *target* is the known output value.

In the ML model for DPA (Model 3a), the optimal size of the hidden layer is 22 neurons with a fivefold CV MAPE (calculated on output fields) of 6%, whereas in the ML model for IPA, the optimal solution is 21 neurons (knee of the curve, where the error is within 5% of minimum value) with a fivefold CV MAPE (calculated on output geometric parameters) of 3.3%. A similar result is obtained by the geometric pyramid empirical rule proposed in [Masters \(1993\)](#), i.e. $\sqrt{n_{out}n_{inp}} = 21.21$, where $n_{out} = 18$ and $n_{inp} = 25$ are the sizes of the input and the output, respectively. In [Figure 5\(a\)](#) and (b), we show the fivefold CV MAPE for a different number of neurons for the ML DPA and IPA cases, respectively.

It is important to point out that the MAPE figures given above and in [Table 2](#) for DPA and IPA are not comparable, as they are calculated in different quantities (the fields for DPA and the profiles for IPA) to assess the performance of the corresponding NNs. A comparison between DPA and IPA is performed in the following.

Regarding the DL model for IPA, i.e. the case of [Figure 2\(b\)](#), the optimal size of the latent space is equal to 10, and the best NN size, according to fivefold CV, is 14 neurons.

6. Test results and analysis

In this section, we present the results of the test of the selected IPA models [cases in [Figures 2\(b\)](#) and [3\(b\)](#)], training them with over 8,000 samples and using a test set of 2,000

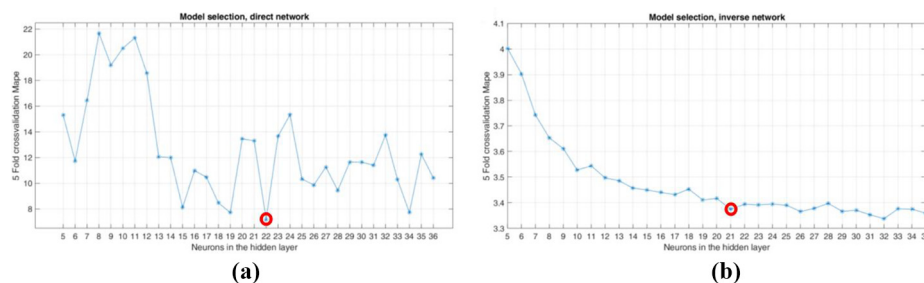


Figure 5.
(a) Fivefold cross-validation for ML-DPA, (b) Fivefold cross-validation for ML-IPA

Model	Neurons in hidden layers	Fivefold CV MAPE (%)
ML and DPA	22	6
ML and IPA	21	3.3
DL and DPA	Autoencoder: 10 Neural Network: 8	5.9
DL and IPA	Autoencoder: 10 Neural Network: 14	3.2

Table 2.
Results of model selection over 8,000 points

samples. As reported in Table 3, the DL model for the IPA case [Figure 2(b)] achieved a test MAPE value of 4%.

In the case of the ML model for IPA [Figure 3(b)], where we represent the profile using 25 parameters of discrete values $y_n \in \{0, \dots, 36\}$, $n = 1, \dots, 25$. the MAPE on the test set is 3.5%. Note that in both DL and ML cases for IPA, the MAPE is calculated on the output profiles; hence, they can be compared. The ML approach shows slightly better results than the DNN approach, and this can be explained by the fact that the more compact representation of the images allows a more efficient operation of the NN in the ML approach and also by the fact that the tuning of the DNN model was carried out as a preliminary test and allows improvements.

All simulations, training and testing have been performed on an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60 GHz.

The low values of MAPE for the IPA cases demonstrate that the proposed approach turns out to be valid for solving inverse problems. Figure 6 shows the comparisons between real (blue line) and predicted (red line) profiles through the proposed approach in some sample test cases. In particular, in Figure 6, we show the i-j-k-m curve representing the right mold of Figure 4(b) as obtained by IPA with a red dashed line.

Table 3.
Results of IPA test
analysis over 2,000
points

Model	Test MAPE (%)
DL and IPA	4
ML and IPA	3.5

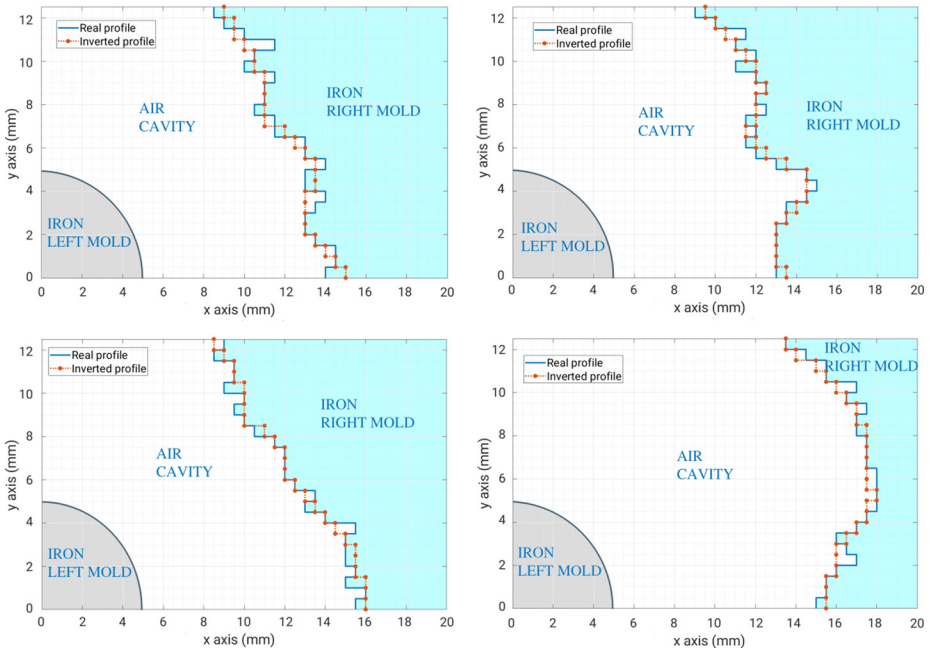


Figure 6.
Comparison between
predicted by DL for
IPA (red dashed line)
and real geometry
profiles (blue line) in 4
cases belonging to the
test data set

Regarding the DPA approaches, the calculation of the test error over 2,000 patterns is not straightforward, as the inversion of each field pattern requires a time of several minutes because an entire optimization needs to be performed. For this reason, we use DPA only for the inversion of a special case, i.e. the optimal field of the TEAM 25 problem, to compare the results with IPA, as discussed in the next section.

The optimization required by DPA was carried out in a genetic algorithm (Sivanandam and Deepa, 2008) with elitism, with a population of 200 individuals, and defining adequate constraints so that the cavity shape is preserved. The genetic algorithm was stopped after about 50,000 function evaluations, as there was no improvement of the best solution in the last 5,000 function evaluations. Note that the optimization was performed using only the DNN surrogate model, and the time cost for a function evaluation is 12 ms, resulting in a total time for optimization of 600 s. Considering the time cost for generating the 10,000 solutions with FEM and training the model, a total time of 35,940 s was required.

When comparing DPA and IPA, considering that the training time is almost identical, the main difference lies in the time for obtaining one inversion, which is 600 s for DPA and 12 ms for IPA, as reported in Table 4.

6.1 Testing data from a different generative model

Both the DL and ML models showed to be able to solve the inverse problem directly (IPA approach) and using an optimization loop (DPA approach). In particular, considering IPA, the NNs obtain good results when predicting the geometry given a field distribution from a test set that was not used for training. However, both the test and training data come from the same generative model, where the geometry is described using a piecewise function with 25 steps. To completely avoid the so-called inverse crime, we have provided the IPA-trained DNN with an input magnetic field that has been generated by FEA software using a smooth elliptical geometry.

In particular, we calculated the field distribution using FEA corresponding to the smooth geometry depicted on the top left of Figure 7, and we used this field distribution as an input pattern of the trained DNN for IPA, obtaining the predicted geometry, which is shown in Figure 7 on the top right. Then, we calculated the magnetic field using FEA, corresponding to the DNN predicted geometry. At the bottom of Figure 7, we compare the original field of the smooth geometry and the final field of the inverted geometry. It can be observed that, even if the two geometries are quite different (absolute percentage error between geometries is 35%), the fields are similar (absolute percentage error between fields is 5%); in particular, the field of the inverted geometry is very close to the original one in 7 out of 9 control points. In fact, being the inverse problem ill-posed, in general, very similar field distributions can be obtained with very different geometries, and even in this case, the DNN for IPA shows a satisfactory behavior.

6.2 Optimal solution of TEAM 25

As a final test case, we consider the original TEAM 25 scope, that is, obtaining a geometry that gives a desired field in the control points as defined in equation (1), which we denote as

Model	DL and DPA	DL and IPA
Dataset generation	10,000 s	Same
Training and model selection	5,340s	Same
Inversion of 1 sample	600s	12 ms

Table 4. Comparison of computational effort between IPA and DPA

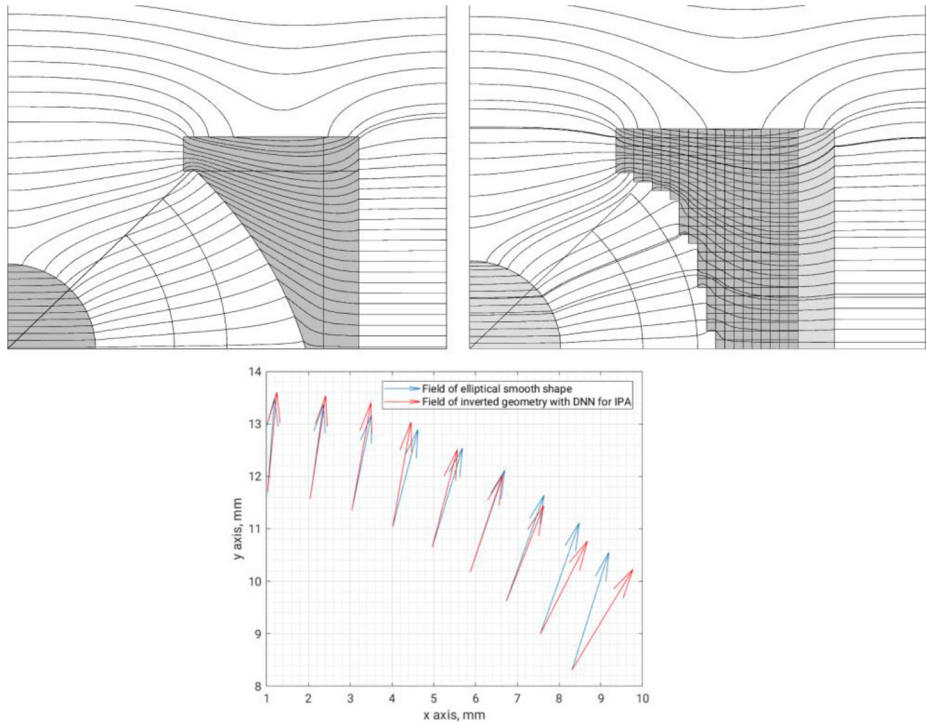


Figure 7.
Comparison between
elliptical geometry
(top left) and output
geometry inverted
using DNN for IPA
(top right)

Note: Bottom: comparison between the field of the elliptical geometry (blue line) and the field of the DNN inverted geometry (red line)

an optimal field. For this problem, we consider both DPA and IPA. It is important to note that the TEAM 25 problem defines the desired field distribution but not the corresponding geometry, which is unknown: the geometry is the desired output of the optimization problem. In the first moment, as in the previous work (Barmada *et al.*, 2021), we solved the optimization problem using the DNN for the DPA model inside an optimization loop, obtaining an optimal geometry shown in Figure 8, on the top left. Then, we approached the problem with the DNN for IPA, directly feeding the model with the optimal field defined by the TEAM 25 problem, and the resulting geometry is shown in Figure 8 on the top right. The solution with the DNN for IPA violates the constraints for the mold shape (the ferromagnetic material enters the cavity), whereas this is not the case for the solution with DPA, where constraints are managed inside the optimization loop and, for this reason, always fulfilled. The reason why the DNN for IPA returns an unfeasible solution can be explained by the fact that the data set does not contain examples sufficiently close to the optimal field. The generation of the data set was performed to obtain random profiles that have no holes inside the right mold and that follow a preferable elliptic shape with a certain probability to resample the shape of the cavity.

This preferable shape of the mold was expected to improve the field distribution inside the cavity to be close to the optimal field. However, being the optimal shape unknown, evidently, the generated geometries included in the dataset were not able to provide output

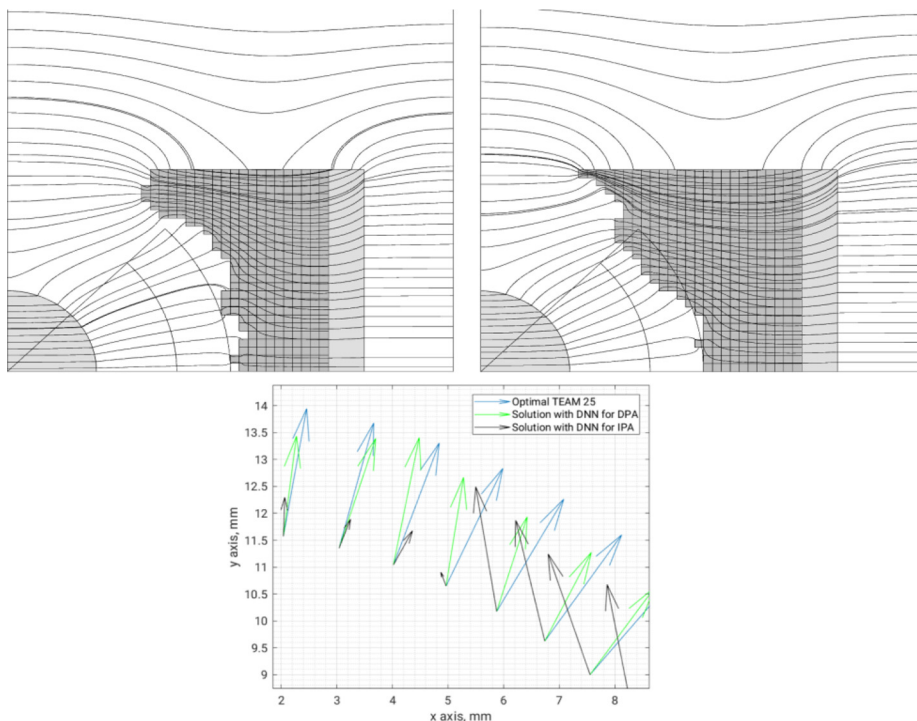


Figure 8.
Comparison between
optimal DPA solution
(top left) and output
geometry inverted
from the optimal field
using DNN for IPA
(top right)

Note: Bottom: comparison between the fields obtained to solve the TEAM 25 problem

fields very close to the optimal field of TEAM 25. For this reason, the IPA fails, and this raises important evidence: the dataset generation is perhaps the most important and delicate part of the process for IPA. At the bottom of [Figure 8](#), we show the field vectors in the nine control points inside the cavity for the optimal case, the DPA case, and the IPA case.

7. Conclusion

The application of ML approaches to the straightforward resolution of shape optimization problems has been considered, with promising results. The performance of a DNN as IPA turned out to be as effective as its use in an optimization loop as DPA. However, the approach has room for improvement and development. It can be observed that in some cases, the DPA that uses optimization is a more effective approach, even if slower, as it allows finding a feasible solution even using a data set with reduced generalization possibilities. On the contrary, the IPA operates well when the patterns to be inverted are well represented in the training dataset. In fact, IPA fails only in one presented case, which is the inversion of the optimal solution of TEAM 25, which was originally presented as a problem to be solved using optimization algorithms and not as an inverse problem. On the other hand, the IPA works perfectly for the 2,000 cases of the test set (never used in training), always giving feasible solutions with errors below 4%. Also, the IPA works very well when the presented field comes from a geometry not generated as those used for training, as illustrated in Section 6.1. As a general conclusion, we shall say that the DNN or ML-based

IPA seems not to be well suited to directly solve optimization problems, but it is perfectly suited to solve inverse problems as far as the input samples are well represented by the training set. On the other hand, the DPA approach with optimization shall be preferred for solving very accurately one-shot critical cases. Relaxing the generative model of the random profiles to include different geometries can in theory, improve the performance of the DNN for IPA, but the space where the optimal solution has to be found is unknown apriori, making this procedure not trivial. As a cue for future work, the authors believe that the most effective approach to overcome the problem of infeasible solutions is to introduce different mechanisms that increase exploration, and thus, lead to the generation of more training examples in the region where the searched solution lies. This can be done, for example, by combining DPA and IPA, thus, performing a preliminary optimization to obtain good examples to train the DNN for IPA. Another approach could be to slightly perturb the desired field that leads to an unfeasible geometry to generate geometries that gradually tend to the feasible region. As a final comment, from preliminary experiments carried out to numerically analyze the sensitivity, it comes out that even small changes in the desired magnetic field values can cause large changes in physical shape, which is a clear indication of the problem's ill-posedness. The authors believe that further regularization is needed, and this aspect will be fundamental in future research to make the DNN perform better in terms of feasibility, even for solutions with high criticality.

References

- Amjad, J., Sokolić, J. and Rodrigues, M.R.D. (2018), "On deep learning for inverse problems", *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1895-1899, doi: [10.23919/EUSIPCO.2018.8553376](https://doi.org/10.23919/EUSIPCO.2018.8553376).
- Amjad, J., Lyu, Z. and Rodrigues, M. (2019), "Deep learning for inverse problems: bounds and regularizers", ArXiv, abs/1901.11352.
- Antun, V., Renna, F., Poon, C., Adcock, B. and Hansen, A.C. (2020), "On instabilities of deep learning in image reconstruction and the potential costs of AI", *Proceedings of the National Academy of Sciences*, Vol. 117 No. 48, pp. 30088-30095, doi: [10.1073/pnas.1907377117](https://doi.org/10.1073/pnas.1907377117)
- Bai, Y., Chen, W., Chen, J. and Guo, W. (2020), "Deep learning methods for solving linear inverse problems: research directions and paradigms", *Signal Processing*, ISSN 0165-1684, Vol. 177, p. 107729, doi: [10.1016/j.sigpro.2020.107729](https://doi.org/10.1016/j.sigpro.2020.107729).
- Barmada, S., Fontana, N., Sani, L., Thomopoulos, D. and Tucci, M. (2020), "Deep learning and reduced models for fast optimization in electromagnetics", *IEEE Transactions on Magnetics*, Vol. 56 No. 3, pp. 1-4.
- Barmada, S., Fontana, N., Formisano, A., Thomopoulos, D. and Tucci, M. (2021), "A deep learning surrogate model for topology optimization", *IEEE Transactions on Magnetics*, Vol. 57 No. 6, pp. 1-4.
- Genzel, M., MacDonald, J. and März, M. (2020), "Solving inverse problems with deep neural networks - robustness included?", ArXiv, abs/2011.04268.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep Learning*, MIT Press, Cambridge, Massachusetts.
- Jin, K.H., McCann, M.T., Froustey, E. and Unser, M. (2017), "Deep convolutional neural network for inverse problems in imaging", *IEEE Transactions on Image Processing*, Vol. 26 No. 9, pp. 4509-4522, doi: [10.1109/TIP.2017.2713099](https://doi.org/10.1109/TIP.2017.2713099).
- Khan, A., Ghorbanian, V. and Lowther, D. (2019), "Deep learning for magnetic field estimation", *IEEE Transactions on Magnetics*, Vol. 55 No. 6.
- Liang, D., Cheng, J., Ke, Z. and Ying, L. (2020), "Deep magnetic resonance image reconstruction: inverse problems meet neural networks", *IEEE Signal Processing Magazine*, Vol. 37 No. 1, pp. 141-151, doi: [10.1109/MSP.2019.2950557](https://doi.org/10.1109/MSP.2019.2950557).

-
- Lucas, A., Iliadis, M., Molina, R. and Katsaggelos, A.K. (2018), "Using deep neural networks for inverse problems in imaging: beyond analytical methods", *IEEE Signal Processing Magazine*, Vol. 35 No. 1, pp. 20-36, doi: [10.1109/MSP.2017.2760358](https://doi.org/10.1109/MSP.2017.2760358).
- McCann, M.T., Jin, K.H. and Unser, M. (2017), "Convolutional neural networks for inverse problems in imaging: a review", *IEEE Signal Processing Magazine*, Vol. 34 No. 6, pp. 85-95, doi: [10.1109/MSP.2017.2739299](https://doi.org/10.1109/MSP.2017.2739299).
- Masters, T. (1993), *Practical Neural Network Recipes in C++*, Morgan Kaufmann, London, England.
- Ongie, G., Jalal, A., Metzler, C.A., Baraniuk, R.G., Dimakis, A.G. and Willett, R. (2020), "Deep learning techniques for inverse problems in imaging", *IEEE Journal on Selected Areas in Information Theory*, Vol. 1 No. 1, pp. 39-56.
- Pollok, S., Bjørk, R. and Jørgensen, P.S. (2021), "Inverse design of magnetic fields using deep learning", *IEEE Transactions on Magnetics*, Vol. 57 No. 7, pp. 1-4, doi: [10.1109/TMAG.2021.3082431](https://doi.org/10.1109/TMAG.2021.3082431). Art no. 2101604
- Sasaki, H. and Igarashi, H. (2018), "Topology optimization accelerated by Deep learning", *Proceedings of IEEE CEFC 2018*, pp. 28-31 Hangzhou, China.
- Sivanandam, S.N. and Deepa, S.N. (2008), "Genetic algorithms", *Introduction to Genetic Algorithms*, Springer, Berlin, Heidelberg, 15-37.
- Snieder, R. and Trampert, J. (1999), "Inverse problems in geophysics", in A. Wirgin, (Ed.), *Wavefield Inversion*, Springer Verlag, New York, NY, pp. 119-190. [10.1007/978-3-7091-2486-4_3](https://doi.org/10.1007/978-3-7091-2486-4_3)
- Takahashi, N. (1996), "Optimization of die press model", available at: <https://www.compumag.org/wp/team/>

Further reading

- Di Barba, P., Mognaschi, M.E., Lowther, D.A. and Sykulski, J.K. (2018), "A benchmark TEAM problem for multi-objective Pareto optimization of electromagnetic devices", *IEEE Transactions on Magnetics*, Vol. 54 No. 3, pp. 1-4. Art no. 9400604.

Corresponding author

Mauro Tucci can be contacted at: mauro.tucci@unipi.it