Multilayer Perceptron Inversion Algorithm in Electrical Engineering Optimization Problems

D. Cherubini, A. Fanni, A. Montisci, P. Testoni

Department of Electrical and Electronic Engineering, University of Cagliari
Piazza d’Armi, 09123 Cagliari, ITALY
e-mail: (davide.cherubini; fanni; augusto.montisci; ptestoni)@diee.unica.it

ABSTRACT

In this work, a neural network based approach for the optimisation of the structure of electromagnetic devices is presented. In particular, a MLP neural network is firstly trained to solve the analysis problem of the studied system. A design problem can be formulated as an inverse problem, i.e., starting from the design requirements the optimal values of the design parameters have to be obtained. The aim of our procedure is to determine the input of the trained neural network, which corresponds to the prefixed target output, given the domain of the input generally imposed by design constraints. An iterative procedure has been developed whose solution will be affected only by an error due to the approximation error of the neural mapping. In order to test the performances of the new proposed algorithm, two optimisation problems were considered: an analytical function with known solution and several local minima, and an electromagnetic application, namely the Loney’s solenoids system.

I. INTRODUCTION

Several optimisation algorithms have been presented in the literature, most of them consist of an iterative procedure where in each iteration a direct problem is solved and a criterion is adopted to modify the trial values of the design parameters [1][2]. The major drawback of these approaches is the computational cost, especially when the direct problem is solved by means of a numerical analysis as for example finite elements analysis (FEM). If a neural network model is available that captures the functional relationship between the design parameters and the objective function of the optimisation problem, the algebraic structure of the neural network itself can be used to find the optimum of the objective function.

The proposed approach consists on inverting a MultiLayer Perceptron (MLP) network, reported in Fig.1, previously trained to approximate the objective function of the optimisation problem [3][4][5]. The inputs correspond to the design parameters, while the output corresponds to the objective function. The aim of our procedure is to determine the inputs which give a prefixed output. The inversion of the neural network consists of finding a solution of the network equations system [3]:

\[ \begin{align*}
    a) \quad W_2 \cdot h + b_2 &= u \\
    b) \quad h &= \sigma(y) \\
    c) \quad W_1 \cdot x + b_1 &= y
\end{align*} \] (1)

where \( x \) is the searched input of the neural network, \( u \) is the desired output, \( y \) and \( h \) are, respectively, the input and the output of the hidden layer. The coefficients of the equations (1.a) and (1.c) are equal to the connections weights, as shown in Fig. 1. In order to apply the proposed method, the domains of the vectors \( u, h, y \) and \( x \) have to be defined. Let \( D_u, D_h, D_y \) and \( D_x \) be such domains. \( D_x \) is defined on the basis of the desired output \( u_0 \).

Fig.1 MLP neural network architecture
The equation (2) together with (1.a) defines the domain $D_h$:

$$u_d - \delta \leq u \leq u_d + \delta$$

which can be represented in the following compact form:

$$L_{\subseteq h} \cdot h \leq d_h$$

The domain $D_z$ is the admissible range for the design parameter of the system. Normally it can be expressed as a convex combination of its vertices. $D_z$ linearly depends on $D_x$. Corresponding points in $D_z$ and $D_y$ have the same convex combination $\alpha$ of the vertices, and this is a practical way to calculate corresponding points in the two domains.

The two domains $D_x$ and $D_y$ are defined into the two spaces which have a bi-univocal correspondence, that permits a projection from the definition space to the other space. By indicating with the apex the projected domain, we will write:

$$D_y^* = \sigma(D_x)$$

The inversion procedure consists of finding a whatever point that belongs to the intersection between the two domains: $y \in \{D_y \cap D_h\}$ or $h \in \{D_y \cap D_h\}$.

## II. INVERSION ALGORITHM

In order to find a point which belongs to the intersection, an iterative procedure is adopted. The points $y$ are described by means of the convex combination $y = V^y \cdot \alpha$ where $V^y$ are the vertices of the domain $D_x$. That ensures that $y \in D_x$ and $\sigma(y) \in D_y^*$, while, in general, $\sigma(y) \notin D_h$. This implies that some inequalities in (4) are violated:

$$g_i(\alpha) = \alpha \cdot \sigma \left( V^y \cdot \alpha \right) > d_{hi}$$

for one or more $i$. In order to reduce this error, the function $g_i(\alpha)$ in (5) has to be reduced and to this end, the gradient of $g_i(\alpha)$ is calculated, to apply a first-order minimization method. At each step, the coefficients $\alpha$ of the convex combination are updated according to:

$$\alpha(n+1) = \alpha(n) - \eta \cdot \nabla g_i$$

In order to avoid to be trapped into a local minimum, a criterion to escape is needed. If a point is found such that only one constraint is violated and the gradient is zero, an escape direction is chosen towards a point named “Ideal Minimum”. This point corresponds to the minimum of $g_i$ in the domain $D_h^*$ and it can be easily found by means of a standard linear programming procedure.

Vice versa, if a point is found such that two or more constraints are violated, the function $g(\alpha)$ is calculated summing up the $g_i(\alpha)$ functions corresponding to the violated constraints, each one weighted by the error $e_i = g_i(\alpha) - d_{hi}$. In the case of gradient equal to zero, an escape direction is chosen towards the barycentre of the ideal minima corresponding to the violated constraints.

## III. THE TESTS

### III.1 Benchmark analytical problem

The initial testing of the proposed procedure has been performed by using analytical functions. An interesting benchmark is the Schwefel’s function as it presents several local minima, as the global minimum is known, and at the end it has been used as benchmark from other authors [6].
In Fig. 2 the real objective function, the related contour plot, and the inversion procedure trace is reported in the 2-D case. The MLP neural architecture used during the test have two input nodes, one hidden layer of 20 sigmoidal nodes and 1 linear output node. The used training set has 12,500 examples, while 25,000 examples have been used to validate the training. A number of 1,500 epochs has been performed, obtaining a final MSE equal to $10^{-7}$. The absolute minimum of the real function is in $(0.842; 0.842)$ and corresponds to the value $-837.965$. By assuming a tolerance $\delta = 8 \times 10^{-8}$ the inversion algorithm reach a point where the real function value is $-837.963$. Note that the procedure finds the global minimum of the real objective function, crossing several local minima.

### III.2 Optimal electromagnetic devices design

In this subsection, the optimisation of the Loney’s solenoids system. This device is used when a highly uniform magnetic field, along a prefixed region, is required. Its simplest configuration consists of a main coil and two so-called correcting, or shim, coils. In Fig. 3, the cross-section of this model is shown.

The objective of the inverse problem is to determine some geometrical parameters of the correcting coils in order to have uniformity of the field along the axis of the solenoids. Several other researchers have studied this problem [3][7][8][9][10] and results are available to perform comparison of different optimisation methods.

In the problem at hand, we assume that the current density be the same in the coils, and that some geometrical parameters are fixed, as suggested in [7], and reported in Fig. 3.

The design variables are the position $S$ and the length $L$ of the shim coils. The objective function is defined as the dishomogeneity of the induction in the controlled sub-region sampled in 10 points along the solenoids axis. In the literature several heuristic approaches have been presented that attempted to find solution falling in the global minimum region that corresponds to objective function value $D < 3 \times 10^{-8}$ and it is indicated as GMR in Fig. 4. The resulting objective function is a non-analytic, ill-conditioned function exhibiting U-shape, steep and narrow valley, having several local minima and moreover it presents numerous narrow basins of attraction.

Even if the objective function is extremely rough and it is very difficult to be learned by the neural network, it is possible to find other smoothest quantity to be approximated by a neural network. In particular the difference between the induction in one generic sampled point and the value of the induction at centre of the solenoid axis, presents a particularly smooth behaviour, very easy to be learned by a neural network. The MLP neural architecture we used have two input nodes, one hidden layer of 20 sigmoidal nodes and 10 linear output nodes.

$$f(x) = \sum_{i=1}^{n} 500 \cdot x_i \sin \sqrt{500 \cdot x_i} \quad x_i \in [0,1] \forall i$$
The used training set has 625 examples, while 100 examples have been used to validate the training. A number of 3000 epochs has been performed, obtaining a final MSE equal to $3 \cdot 10^{-14}$. A zero value of the objective function is searched with a tolerance $\delta = 5 \cdot 10^{-7}$. The inversion procedure reach the point $(S = 0.12411 \, \text{m}; L = 0.021311 \, \text{m})$ whose value of the approximated objective function is $3.48 \cdot 10^{-7}$, corresponding to a value of the real objective function equal to $2.88 \cdot 10^{-8}$.

In Fig. 4 the contour plot of the real objective function is reported within the search trace of the inversion procedure which follows the steepest valley in the Global Minimum Region.

IV. CONCLUSIONS

An iterative algorithm for the inversion of an MLP neural network trained to learn the functional relationship between design parameters and objective function is presented. The procedure is able to solve any inverse SP electromagnetic problem with a prefixed error degree on the specified target value. The performances of the algorithm have been tested on some benchmark problems and the results have been compared with those presented in the literature. The use of a neural network based function approximation is particularly suited for the design of electromagnetic structures where the objective function is extremely heavy to evaluate, as for example when it has to be defined through a numerical method, as FEM.

REFERENCES