An Iterative Algorithm for Electrical Impedance Imaging Using Neural Networks

A. Nejatali and I. R. Ciric
Department of Electrical and Computer Engineering, The University of Manitoba
Winnipeg, Manitoba R3T 5V6, Canada

Abstract — In the electrical impedance imaging algorithms developed so far, the inverse problems involved are treated by using either iterative methods, which generate more accurate results but require large amounts of computation time, or non-iterative methods that are faster but produce less accurate results. In this paper, a new iterative procedure for electrical impedance imaging is presented. At each iteration step, the updated conductivity distribution is used to solve a forward problem and two-layer backpropagation neural networks with nonlinear activation functions are employed for solving an inverse problem. This allows for a smaller computation time with respect to other iterative methods and, at the same time, yields accurate images. Comparison with results obtained by applying an existent impedance tomography algorithm illustrates the efficiency of the proposed iterative method.

Index terms — Impedance tomography, neural networks.

I. INTRODUCTION

Electrical impedance imaging methods can be applied in numerous engineering fields, e.g., in quality control and fault detection for various materials [1], in multicomponent fluid flow analysis [2], medical imaging [3], and geological exploration [4]. These imaging methods are based on measuring the voltage-current relationships for a set of electrodes applied on the object surface, in order to reconstruct the distribution of conductivity and/or permittivity within the object. One has to employ an optimal electrode configuration, with appropriately chosen injected current frequencies and current patterns. Undoubtedly, the solution of the inverse problem, i.e., the computation of the conductivity distribution when the electrode voltages and the injected currents are known, is the most challenging part in an electrical impedance imaging system. This problem can be solved by non-iterative methods, such as those implementing various backprojection techniques [5]-[7] or employing sensitivity coefficients methods [8], and also by iterative methods, such as the equipotential lines method [9], the perturbation method [10], the double constraint method [11], the Newton-Raphson method [12], and the compensation theorem method [13]. The non-iterative methods are fast methods, less sensitive to noise and to electrode displacement, but they produce less accurate results. The iterative methods generate more accurate results, in general, but are computationally expensive, very sensitive to noise and measurement errors, as well as to electrode displacement. Since 1991, several research groups have reported impedance imaging results obtained by non-iterative methods employing neural networks. These networks perform well in a noisy environment when appropriately trained with noisy patterns. One-layer backpropagation neural networks (BPNNs) with linear activation functions have been proposed in [14]-[16] for electrical impedance tomography, and also one-layer BPNNs with nonlinear activation functions [2] and RAM-based neural networks [17]-[18] have been used for pattern association. From the results obtained by applying such non-iterative methods, it has been realized that the neural networks required to solve the inverse problem in its general form and with accurate results would be too big to be realistic, from the point of view of the number of units in the input and in the hidden layers, as well as from that of the number of necessary training patterns. On the other hand, iterative neural networks, such as the recurrent neural networks [19], have successfully been used for pattern recognition. In this paper, we present an impedance imaging procedure based on employing BPNNs in an iterative scheme that enables us to use relatively small neural networks for solving accurately the inverse problem, with several times smaller CPU time compared with existent iterative methods. The proposed method enjoys the advantageous features of the neural networks, including their performance in the presence of noise and the capability of parallel processing, although these issues are not addressed here.

II. THE IMPEDANCE IMAGING SYSTEM

The proposed electrical impedance imaging system employs independent Walsh function patterns [20] for the currents injected into the set of electrodes. These patterns exploit the advantage of operating with a constant current amplitude, allowing at the same time an easy programming of the current sequences. Fig. 1 shows the Walsh function current patterns for a 16-electrode system used to generate the test results presented.

A forward problem solving module is used in this imaging system, whose operation is based on a network approximation numerical method [21], easy to implement and producing fast and sufficiently accurate results. Moreover, it allows to simulate accurately wide area electrodes, which could be more difficult to achieve when using finite element methods [22]. The forward problem solving module is used twice, first for producing the patterns for training the neural networks and again during the iterative procedure itself.
Neural networks have parallel and distributed structures capable to perform complex mappings in hyperdimensional vector spaces [23]. Their basic computational components are simple processing units. The computational neurons realize a weighted sum of input signals, followed by a nonlinear static transformation that shows a saturation behaviour. The neurons are arranged into layers (indicated in Fig. 2) and their number and their dimensions in the respective layers determine the capabilities of the resulting network. The network learns a predefined set of input-output example pairs by using a training cycle. An input pattern is applied as a stimulus to the first layer of neurons and, then, it is propagated through each upper layer until an output is generated. This output pattern is then compared to the desired output and an error is computed for each output unit. The error is then transmitted backward from the output layer to each node in the intermediate layer that directly contributes to the output. Each unit in the intermediate layer receives only a portion of the total error value, based on the relative contribution the unit made to the original output. This process continues from layer to layer until each node in the network receives an error value that describes its relative contribution to the total error. The corresponding connection weights are updated based on their unit relative error value, such that a final state is reached, where all the training patterns are encoded.

In this paper, the inverse problem is solved iteratively by using a set of two-layer BPNNs appropriately trained as described below. Consider for illustration a 16-electrode system, where 15 independent voltages can be identified for each current pattern. Since the number of necessary Walsh function current patterns is 15, there are 225 measured voltages, which constitute the input vector for the neural networks. The output vectors of the neural networks have a number of entries equal to the number of cells in which the object is discretized. In the illustration considered, the object is discretized in 16 cells (as shown in Fig. 4). We use linear activation functions for the units in the output layer and hyperbolic tangent activation functions for the units in the hidden layer. By using linear activation functions in the output layer, the scaling of the outputs of the neural networks can be avoided and also the training episode is much shortened, for instance about twenty times in the tests performed, as compared to the case when using nonlinear activation functions for the units in the output layer.

Instead of using a big, single neural network with 225 units in the input layer, 15 neural networks have been employed, each of them for one Walsh function current pattern and with only 15 units in the input layer. This architecture decreases the overall size of the neural networks by a factor of about 15. Experience shows that different Walsh function current patterns require different number of units in the hidden layer of their corresponding BPNNs due to different relationships between the electrode voltages and the cell conductivities. Consequently, the number of units in the hidden layer has been selected to be different for different neural conductivities, namely, between 1500 to 2500.

These neural networks have been trained using a chosen set of patterns for the conductivity distribution within the object. Three conductivity values, 1, 1.2 and 1.4 S/m, have been selected as background conductivities. Three patterns correspond to homogeneous distributions of conductivity within the object, with these three conductivity values. In the other patterns, one cell has a conductivity which is different from that of the background, with values equal to 1/6, 5/6, 7/6 and 11/6 of that of the background. For a number of cells equal to 16, this gives 3x4x16 patterns. The input vectors are given by the voltages associated with these patterns and have been generated by using the forward problem solving module [21]. The electrode voltages and the cell conductivities of the second homogeneous pattern have been subtracted, respectively, from...
those corresponding to the other patterns. The resulted sets of electrode voltages and of conductivities constitute, respectively, the input and the output training vectors for the input and the output layers of the neural networks. Thus, the neural networks are trained with relative values of electrode voltages and of conductivities. This arrangement is necessary for the proposed iterative procedure, described in the next section.

IV. ITERATIVE ALGORITHM

The trained neural networks are employed iteratively as indicated in Fig. 3. The procedure consists of the following main steps.

1) A homogeneous distribution of conductivity equal to 1.2 S/m is used as initial estimate.
2) The electrode voltages corresponding to the respective current patterns are computed in the forward problem solving module.
3) These computed electrode voltages are subtracted from the measured ones and the result is forwarded to the neural networks.
4) The outputs of all the neural networks are correspondingly added and then multiplied by an updating factor in the range from $10^{-4}$ to 0.4 in terms of the rate of convergence of the iterative process.
5) The resulting conductivity distribution is superposed on the previous one and the updated conductivity distribution is again used in the forward problem solving module to compute the new electrode voltages.
6) The error, i.e., the difference between the measured electrode voltages and the computed ones is examined. If the error is smaller than an imposed threshold, then the procedure is terminated. If the error has decreased, the updating factor is increased and the process is repeated from step 4. If the error has increased, then the electrode voltages and the conductivity distribution corresponding to the previous updating factor are selected and the iteration process continues from step 3.

Although each network is trained with patterns having only one cell with a conductivity different from that of the background, this iterative procedure allows the reconstruction of conductivity distribution for objects with all cell conductivities different from that of the background and with values between 1 and 1.4 S/m.

V. RECONSTRUCTION TESTS

Three examples have been chosen to illustrate the operation of the proposed method. In Figs. 4(a) and 4(b) two random conductivity distributions, and in Fig. 4(c) a continuous distribution, with values between 1 and 1.4 S/m, are presented. These conductivity values have been used in the forward problem solving module, with Walsh function current patterns, to obtain the electrode voltages which are taken to be the measured voltages. The reconstructed conductivity distributions are shown in Figs. 4(a'), 4(b') and 4(c'), respectively. The error has been calculated as

$$\text{error} = \left[ \sum_{k} (\sigma_k' - \sigma_k)^2 / \sum_{k} \sigma_k^2 \right]^{1/2}$$

where the summation is performed over all the cells, and $\sigma_k'$ and $\sigma_k$ are the reconstructed and the actual conductivities, respectively.

A Sun Ultra 1 workstation and MATLAB-based software were employed. The training episode for the networks took about 150 hours of CPU time. Although this CPU time seems to be large, it should be noted that the training episode has to be completed only once, in advance, and then, subsequently, it is used for any image reconstruction in the interval considered. To illustrate the efficiency of the proposed method, a Newton-Raphson technique was also implemented on the same machine, using the same MATLAB software environment, for the same number of cells. At least four iterations are needed for the Newton-Raphson method and each iteration requires 71.35 seconds of CPU time. The CPU time required in the presented method is about 5 times smaller.

VI. CONCLUSION

A new iterative method, utilising neural networks, has been proposed for the electrical impedance imaging. It is based on employing at each iteration step appropriately trained neural networks for solving the inverse problem involved, along with a forward problem solving module previously developed by the authors. The method requires a reduced computation time with respect to other iterative methods, such as the Newton-Raphson method [12], and yields more accurate results than fast non-iterative methods, such as those based on backprojection [5]-[7] or on non-iterative neu-
The presented method can also be applied for solving other inverse problems, such as those involving the inverse scattering of electromagnetic or acoustic waves. Preliminary investigations of the behaviour of the proposed imaging algorithm in the presence of noise show, as expected, a good performance of the BPNNs. It should be remarked that the forward problem can be solved by employing a separate BPNN appropriately trained, which will speed up the procedure even further.

**REFERENCES**


