SOLUTION OF MAGNETIC INVERSE PROBLEMS USING ARTIFICIAL NEURAL NETWORKS

Minghui Xu Nathan Ida
Department of Electrical Engineering, The University of Akron
Akron, OH 44325-3904, U.S.A.
Tel: +1-330-972-6525, Fax: +1-330-972-6487, e-mail: ida@uakron.edu

***Abstract:** This paper presents an application of artificial neural networks to magnetic inverse problems. Artificial neural networks derive their own computing power through massively parallel distributed structures. They have the ability to learn and therefore generalize. Based on these capabilities, artificial neural networks can be used to solve complex problems which currently are intractable. The application is demonstrated here by two simple identification examples. Its calculated results agree with the analytic result indicating the method provides good accuracy. Furthermore, novel stability tests are also provided. This paper gives better results compare to* [1] *and* [2], *since the rms error is used which is more accurate than full-scale percent rms error.*

***Key words:** Inverse problems, neural networks*

1. Introduction

Recently, artificial neural networks have been introduced to solve electromagnetic inverse problems [1] and [2]. Inverse problems arise in a number of areas. Examples include industrial nondestructive testing, medical diagnostics, geophysical prospecting for petroleum and minerals, and detection of earthquakes. Electrical inverse problems can sometimes be stated as simply as the following: If there is an electric charge in space, it is easy to calculate the electric field around it. What about taking some samples of the electric field to predict the position of the electric charge? Since the inverse problem is highly nonlinear and without formulations to follow, it is very difficult to construct an effective inversion algorithms. An artificial neural network, however, has the following properties: nonlinearity, input-output mapping, fault tolerance and most important, learning from examples. The need for learning from examples is closely related to the difficulty of formulating explicit rules. Artificial neural networks are based on abstracting from the complex details of human thought and building a simple model using a network of simple processors. Artificial neural networks consist of a large number of simple processing elements called neurons or nodes. Each neuron is connected to other neurons by means of directed links, each with an associated weight. The weights represent information being used by the network to solve a problem. The artificial neural network essentially determines the relationship between input and output by looking at examples of many input-output pairs. In learning processes, the actual output of the artificial neural network is compared to the desired output. Changes are made by modifying the connection weights of the artificial neural network to produce a closer match. The procedure iterates until the error is small enough. Some work has been done in this area [1] and [2]. The prediction errors are called the full-scale percent root mean squared error which means dividing the root mean squared error by the range.

This paper presents an artificial neural network approach to magnetic inverse problems. The artificial neural network is used to learn cases of the samples and then generate its own predictions. This paper shows that by the method of back-propagation and properly changing the learning rate, the neural network can give good results in root mean squared error. Also, stability tests reassure that the results are correct and stable. We use two straightforward magnetic problems which have analytic solution to verify the effectiveness of this method. The first problem is to identify two end points of a current segment lying on a 0-3 line segment by sampling the magnetic field intensity at 8 points around a 3-3 region. The length of the current segment ranges from 0 to 3. The second problem is to predict the center of a 1-1 square current loop in a 3-3 region by measuring 10 points of the magnetic field intensity on the boundary. This type of application has particular appeal in nondestructive testing of materials.

2. Structure of artificial neural networks

Artificial neural networks, also called artificial neural systems, neurocomputers, parallel distributed processors or connectionist models are an attempt to mimic the structure and functions of brains and nervous systems of living creatures. Generally speaking, an artificial neural network is an information processing systems composed of a large number of simple processing elements, called artificial neurons or simply nodes. Neurons are interconnected by direct links called connections with an associated weight, which cooperate to perform parallel distributed processing in order to solve a desired computational task. One of the attractive features of artificial neural networks is their capability to adapt themselves to special environmental conditions by changing their connection strengths or structure i.e. changing their weights. Years of studies have shown that artificial neural networks exhibit a surprising number of the brain's characteristics. For example, they learn from experience, generalize from previous examples, and abstract essential characteristics from inputs containing irrelevant data. In this paper we choose the back-propagation method to demonstrate the potential of artificial neural networks to solve magnetic inverse problems.

One of the most influential developments in artificial neural network was the invention of the back-propagation algorithm, which is a systematic method for training multilayer artificial neural networks. The standard back-propagation learning algorithm for feedforward networks aims to minimize the mean squared error defined over a set of training data. In feedforward artificial neural networks neurons are arranged in a feedforward manner i.e. each neuron may receive an input from the external environment or from the neurons in the former layer, but no feedback is formed. The network architecture for a feed forward network consists of layers of processing nodes. The network always has an input layer, an output layer and at least one hidden layer. There is no theoretical limit on the number of hidden layers but typically there will be one or two. In our case, there is only one hidden layer. Every neuron in each layer of the network is connected to every neuron in the adjacent forward layer. A neuron's activity is modeled as a function of the sum of its weighted inputs, where the function is called the activation function, which is typically nonlinear, thus giving the network nonlinear decision capability. Each layer is fully connected to the succeeding layer. The arrows indicate flow of information (Fig.1). Where n_I is the number of neurons in the input layer, n_H is the number of neurons in the hidden layer, n_O is the number of neurons in the output layer, x_l are the inputs to the input layer where $l=1,...,n_I$, y_k is the value of the hidden layer where $k=1,...,n_H$, z_m is the value of the output layer where $m=1,...,n_O$, where $w_{lk}^{[1]}$ is the weight connecting the lth neuron in the input layer to kth neuron in the hidden layer, and $w_{km}^{[2]}$ is the weight connecting the kth neuron in the hidden layer to the mth neuron in the output layer. The nodes of the hidden and output layer are:

$$y_k = f\left(\sum_{l=1}^{n_l} w_{lk} x l\right) \qquad (k=1,....,n_H) \qquad (1)$$

and

$$z_m = f\left(\sum_{k=1}^{n_k} w_{mk} y_k\right) \qquad (m=1,....,n_o) \qquad (2)$$

where the activation function f is traditionally the Sigmoid function but can be any differentiable function. The Sigmoid function is defined as

$$f(x)= \frac{1}{(1.0+e^{-x})} \qquad (3)$$

The back-propagation method is based on finding the outputs at the last(output) layer of the network and calculating the errors or differences between the desired outputs and the current outputs. When the outputs are different from the desired outputs, corrections are made in the weights, in proportion to the error.

$$\Delta w_{km}^{[2]}= y_k f'(z_m)(z_m - d_m) \qquad (4)$$

where d_m represent the desired output, $k=1,...,n_H$, $m=1,....,n_O$ and

$$f'(x)= \partial f(x)/\partial x \qquad (5)$$

If f is the Sigmoid function, and

$$f'(x)= \frac{e^{-x}}{(1.0+e^{-x})^2} = f(x)(1 - f(x)) \qquad (6)$$

the update rule for the weights from the hidden layer to the output layer is

$$w_{km(new)}^{[2]}= w_{km(old)}^{[2]} + \eta \Delta w_{km}^{[2]} \qquad (7)$$

where $k=1,...,n_H$, $m=1,...,n_O$ and η is the learning rate. The update rule for the weights from the input layer to the hidden layer is

$$\Delta w_{km}^{[1]}= x_l f'(y_k) \sum_{m=1}^{n_O} w_{km}^{[2]} f'(z_m)(z_m - d_m) \qquad (8)$$

$$w_{lk(new)}^{[1]}= w_{lk(old)}^{[1]} + \eta \Delta w_{lk}^{[1]} \qquad (9)$$

where $l=1,...,n_I$, $k=1,...,n_H$. The back-propagation algorithm has been generated based on this structure.

3. Identification of the position of a current segment

Suppose there is a current segment on the y axis between 0 to 3. The end points of the current segment are d_1 and d_2. The length of the current segment is less than 3. The magnetic field intensity at any point p in space can be calculated by the Biot-Savart Law (Fig. 2), which is

$$\overline{H}_p = \hat{a}_\phi \int_{d_2}^{d_1} \frac{Ir}{4\pi(r^2+u^2)^{3/2}} du \qquad (10)$$

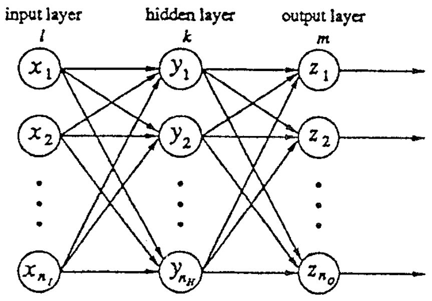

Fig. 1. Multilayer artificial neural network.

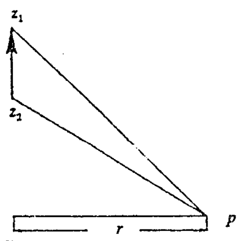

Fig. 2. Magnetic field intensity at point p.

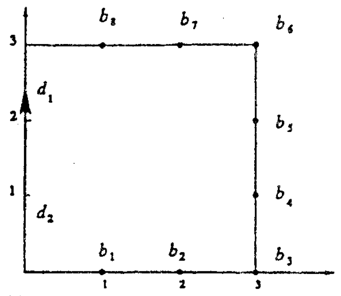

Fig. 3. Identification of the current segment.

For this case, we calculate the magnetic field intensity at $b_1,...,b_8$ (Fig. 3). In a real problem sensors can be placed in the position $b_1,...,b_8$, where the real data can be measured. We calculate 406 cases for different lengths and positions of the current segment between 0 to 3, then use the 406 sets of data as examples to train the back-propagation algorithm to predict the end points of the current segment on the y axis z_1 and z_2 between 0 to 3.

For the back-propagation algorithm the activation function f is the Sigmoid function, the learning rate initially is 0.5 but as the root mean squared (rms) error gets smaller it decreases to 0.3. This is the experience from the training which also matches the idea of learning-rate annealing in [5]. The input layer has 8 neurons, since we have 8 inputs. 10 neurons are chosen in the hidden layer. The output layer has 2 neurons. Assuming the real value of the end points are d_1 and d_2, the neural network calculated result are z_1 and z_2. For the learning process the rms error for z_1 is 0.018. The rms error for z_2 is 0.042. After learning, we use 406 cases to test the system. All the testing cases are different from the learning cases so that the testing cases are independent of the learning cases. For the testing cases the rms error for z_1 is 0.020. The rms error for z_2 is 0.044 (Fig. 4 and 5). To show stability, we add one percent noise to some of the inputs in each of the 406 cases. Still for the tested cases the rms error for z_1 is 0.049, the rms error for z_2 is 0.023 (Table 1).

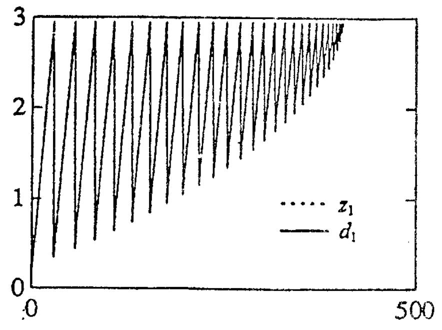

Fig. 4. Testing result for z_1 compared with d_1.

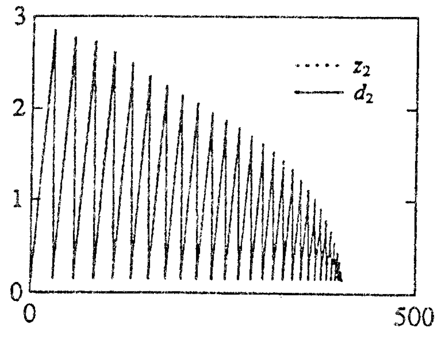

Fig. 5. Testing result for z_2 compared with d_2.

TAable 1
Summaries of the Root Mean Squared Error

rms error	z_1	z_2
Learning process	1.8%	4.2%
Testing process	2.0%	4.4%
Testing process with noisy input	4.9%	2.3%

4. Identification of the position of a rectangular current loop

A 1-1 rectangular current loop is located in a 3-3 squared region. The magnetic field intensity at $b_1,...,b_{10}$ can be calculated by Biot-Savart Law. In a real problem, the magnetic field intensity can be measured by sensors (Fig. 6).

In the back-propagation algorithm the same activation function and changing the learning rate had been used with 10 input neurons 10 hidden neurons and 2 outputs. After learning 400 cases of the positions of the current loop in the 3-3 region, the center of the current loop has been predicted at the output. If (x,y) represent the output of the artificial neural network and (z_d, y_d) represent the real position, the rms error for x in the learning process is 0.030. The rms error for y is 0.033. For the testing cases the rms error is 0.038 for x and 0.035 for y, where all testing cases are different from the learning cases (Fig. 7 and 8). To verify stability, we add one percent noise to some of the inputs in each of the 400 sets. For the tested cases the rms error for x is 0.046, the rms error for y is 0.036 (Table 2).

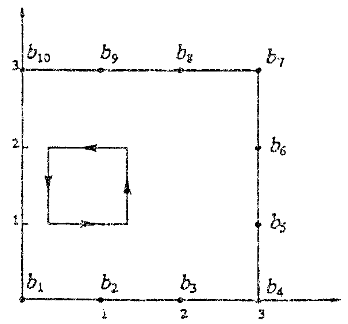

Fig. 6. Identification the center of the current loop.

Table 2
Summaries of the Root Mean Squared Error

rms error	z_1	z_2
Learning process	3.0%	3.3%
Testing process	3.8%	3.5%
Testing process with noisy input	4.6%	3.6%

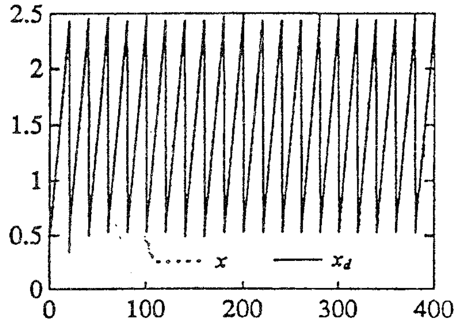

Fig. 7. Testing result for x compared with x_d

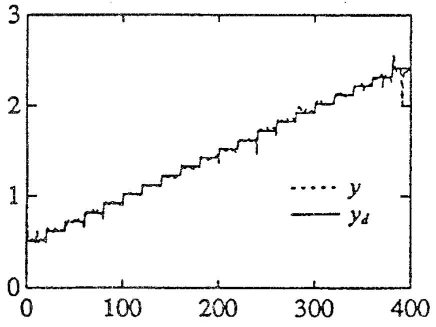

Fig. 8. Testing result for y compared with y_d

5. Conclusions

We have presented an artificial neural network approach to solve magnetic inverse problems. The artificial neural network method is very general. It can be used in different cases, and the solution is quite good and stable. Usually the learning process is done off line. So when the network have been trained, the neural network system will give an instant prediction. This is promising in practical identification problems such as nondestructive testing. The back-propagation method has its limitation in finding the global minimum. In future work evolutionary methods will be used based on the structure of the artificial neural networks.

References

1 E. Coccorese, R. Martone, and F. C. Morabito, "A neural network approach for the solution of electric and magnetic inverse problems", *IEEE Trans. Magnetics,* Vol.30, No.5, pp. 2829-2839, 1994.

2 F. C. Morabito and M. Campolo, "Location of plural defects in conductive plates via neural networks", *IEEE Trans. Magnetics,* Vol. 31, No.3, pp. 1765-1768, 1995.

3 A. Cichocki and R. Unbehauen, *Neural networks for optimization and signal processing,* John Wiley & Sons Ltd., England, 1993.

4 R. Bharath and J, Drosen, *Neural network computing,* Windcrest, 1994.

5 S. Haykin, *Neural networks: A comprehensive foundation,* New York, Macmillan College Publishing Company, 1994.

6 L. Fausett, *Fundamentals of neural networks,* Prentice Hall, New Jersey, 1994.

7 J.A. Edminister, *Theory and Problems of Electromagnetics*, McGraw-Hill, NY, 1979.

8 R.S. Elliott, *Electromagnetics*, McGraw-Hill, NY, 1966.