

Inversion of quasi-3D DC resistivity imaging data using artificial neural networks

AHMAD NEYAMADPOUR^{*,1}, W A T WAN ABDULLAH¹ and SAMSUDIN TAIB²

¹*Department of Physics, University of Malaya, 50603 WP Kuala Lumpur, Malaysia.*

²*Department of Geology, University of Malaya, 50603 WP Kuala Lumpur, Malaysia.*

**e-mail: anpour@gmail.com*

The objective of this paper is to investigate the applicability of artificial neural networks in inverting quasi-3D DC resistivity imaging data. An electrical resistivity imaging survey was carried out along seven parallel lines using a dipole–dipole array to confirm the validation of the results of an inversion using an artificial neural network technique. The model used to produce synthetic data to train the artificial neural network was a homogeneous medium of $100\ \Omega\text{m}$ resistivity with an embedded anomalous body of $1000\ \Omega\text{m}$ resistivity. The network was trained using 21 datasets (comprising 12159 data points) and tested on another 11 synthetic datasets (comprising 6369 data points) and on real field data. Another 24 test datasets (comprising 13896 data points) consisting of different resistivities for the background and the anomalous bodies were used in order to test the interpolation and extrapolation of network properties. Different learning paradigms were tried in the training process of the neural network, with the resilient propagation paradigm being the most efficient. The number of nodes, hidden layers, and efficient values for learning rate and momentum coefficient have been studied. Although a significant correlation between results of the neural network and the conventional robust inversion technique was found, the ANN results show more details of the subsurface structure, and the RMS misfits for the results of the neural network are less than seen with conventional methods. The interpreted results show that the trained network was able to invert quasi-3D electrical resistivity imaging data obtained by dipole–dipole configuration both rapidly and accurately.

1. Introduction

Geophysical surveys represent an efficient way to detect subsurface heterogeneities, including voids, refilled cavities, and so on (Mochales *et al* 2008). In the past few years, the technique of electrical resistivity imaging (ERI) has become one of the most significant geophysical approaches to the investigation of underground near surface structures. However, inversion of DC ERI data is complex, due to its non-linearity (Singh *et al* 2005). This is especially true for high resistivity contrast regions (El-Qady and Ushijima 2001). Because many of the problems associated with geophysical exploration are of a three-dimensional (3D)

nature, several algorithms for treating the 3D ERI problem, based on finite element, finite difference, and integral methods, have been developed (Loke and Barker 1996a; Zhao and Yedlin 1996; Dahlin and Loke 1997; Spitzer 1998; Tsourlos and Ogilvy 1999). Although 3D techniques have evolved considerably in the past few years, in many cases, 2D data acquisition methods remain preferable for logistical reasons. It is often a major advantage to carry out measurements along linear features such as roads, where in many cases, arranging a 3D electrode layout in the field may be impossible. Techniques based on relatively sparse nets of parallel lines may be the most efficient in practical applications. In order to build a 3D resistivity distribution

Keywords. Artificial neural networks; inversion; quasi-3D; resistivity imaging; dipole–dipole; exploration geophysics; geophysics; modelling.

of the subsurface, it is common practice to merge the results from a number of sections (profiles) that are acquired and inverted using a 2D resistivity imaging technique. This may be referred to as a quasi-3D technique. The existing conventional methods for the inversion of geoelectrical resistivity data, however, face a difficult task because of the nonlinear nature of the data (Singh *et al* 2005).

In recent years, artificial neural network (ANN) systems have been developing rapidly (Singh *et al* 2002). Geophysical prospecting mostly includes some problems (e.g., ground penetration radar (Poulton and El-Fouly 1991) and processing of EM sounding data (Poulton *et al* 1992)), which the ANNs have been employing in work. El-Qady and Ushijima (2001) and Singh *et al* (2005) studied the ANNs to solve geophysical inverse problems for 1D and 2D vertical electrical sounding (VES).

In this paper, we investigate the applicability of ANNs in the inversion of quasi-3D ERI data using different learning paradigms. A dipole–dipole array was used in order to delineate horizontal changes in subsurface resistivity. The survey was conducted separately along seven parallel lines. The investigation site consisted of discrete subsurface structures with sharp boundaries between different bodies. Therefore, the 2D least squares algorithm based on the robust inversion technique (RIT) (Claerbout and Muir 1973) was used in the inversion of each apparent resistivity dataset. The inverted results were then combined to construct horizontal depth slices.

Using synthetic data, the ANN was trained and then tested through another set of new synthetic data, as well as real field data. Finally, the inversion results of the ANN were compared with the RIT in both the synthetic and real field data.

2. Robust inversion technique (RIT) methodology

Assume a set of N measured data, $d = (d_1, d_2, d_3, \dots, d_N)^T$ affected by the physical property q of the subsurface. The next step is finding a spatial parameter distribution $q(r)$ that explains the d . Since the data are usually contaminated with noise, we try to fit that part of the data, which is generated by parameter variations (Thomas 2004). The function $q(r)$ is represented using a limited number M of model parameters m_i serving as weighting coefficients for basis functions $W(r)$;

$$q(r) = \sum_{i=1}^M m_i W_i(r), \quad (1)$$

where $m = (m_1, m_2, m_3, \dots, m_N)^T$.

A usual way to define the W_i is to subdivide the region of interest into sub-domains Θ_i which they are convex bodies such as hexahedrons,

$$\text{where } W_i(r) = \begin{cases} 1 & \text{for } r \in \Theta_i \\ 0 & \text{elsewhere} \end{cases}.$$

The main objective of inversion process is to obtain a model m , whose response $q(m)$ fits the data d . For non-linear problems the forward operation depends on the model m itself, which holds for all methods that are based on Maxwell's equations. Beginning from a starting model m_0 , an iterative process is applied to update the model until data fit or convergence is achieved (Thomas 2004). In each iteration of inversion process (k) a new model is calculated by adding a model update Δm_k

$$m_{k+1} = m_k + \Delta m_k. \quad (2)$$

A Taylor approximation of first order yields

$$q(m_k + \Delta m_k) = q(m_k) + \left\{ \frac{\partial q}{\partial m}(m_k) \right\} \Delta m_k + \dots \approx q(m_k) + J \Delta m_k.$$

The second term of the above equation is called Jacobian or sensitivity matrix $J \in \mathfrak{R}^{N \times M}$ with the elements

$$J_{ij}(m_k) = \frac{\partial q_i}{\partial m_j}(m_k). \quad (3)$$

In fact, the Jacobian matrix, J , is the partial derivative of the model response with respect to the model parameters. Setting the response of the new model $q(m + \Delta m)$ equal to the data d we obtain the non-quadratic equation

$$J^T J \Delta m = J^T (d - q(m)) \quad (4)$$

which has to be solved in some sense to minimize the residual vector $\Delta d = d - q(m)$. In practice, the simple least-squares equation (4) is rarely used by itself in geophysical inversion (Loke 2009). In some situations the matrix product $J^T J$ might be singular, and thus the equation (4) does not have a solution for Δm . Furthermore, in some cases, the subsurface geology consists of a number of regions with sharp boundaries between different regions. For such cases, the inversion formulation in (4) can be modified so that it minimizes the absolute changes in the model resistivity values (Claerbout

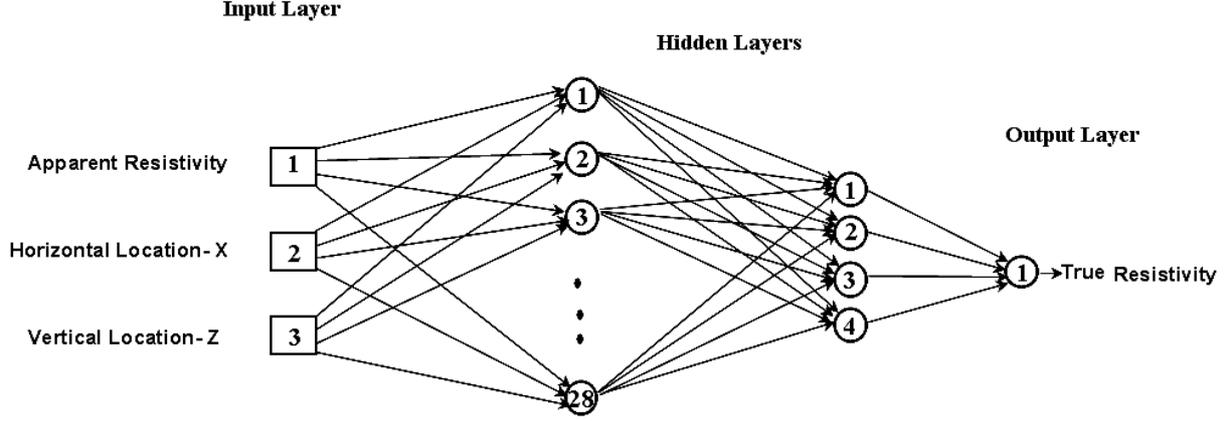


Figure 1. Artificial neural network architecture used in this study.

and Muir 1973). Technically, this is referred to as the robust inversion technique (RIT). One simple method for modification using the standard least-squares formulation is the iteratively reweighted least-squares method (Wolke and Schwetlick 1988). So, the equation (4) can be modified as the following

$$(J^T J + \zeta F_R) \Delta m^k = J^T R_d \Delta d^k - \zeta F_R m^k \quad (5)$$

where

$$F = \alpha_x C_x^T R_m C_x + \alpha_y C_y^T R_m C_y + \alpha_z C_z^T R_m C_z.$$

The factor ζ is known as the damping factor and C_x, C_y and C_z are the smoothing matrices. α_x, α_y and α_z are the relative weights given to the smoothness filters in the x -, y - and z -directions. R_d and R_m are weighting matrices introduced so that different elements of the data misfit and model roughness vectors are given equal weights in the inversion process.

3. Artificial neural network system

The research on artificial neural networks (ANNs) started about 47 years ago, when the theory of perceptron was presented by Rosenblatt (1962). During the mid-1960s, the interest in the ANN decreased, because of the limitations of the theory of perceptron (Minsky and Papert 1969). Since new ANN paradigms have overcome some of these limitations, the ANN has emerged again as an active research area within computer science, engineering, physics and geophysics.

The ANNs are computational systems that simulate the biological neural networks of the human brain. The human brain contains several billion neurons interconnected via synapses in a complex manner, which constitute the network. ANNs are systems made up of a number of simple, highly interconnected processing algorithms (neurons), which process information by their dynamic state response to external inputs. ANNs are also exploration and development tools that allow an easy transformation of input data into desired output parameters. Specific mathematical relationships between inputs and target values need not be known. The ANNs can use knowledge gained from past experiences and apply that knowledge to new problems and situations. Figure 1 shows a schematic diagram of the ANN architecture. The system consists of (1) a set of nodes (artificial neurons) that perform simple computations, (2) a set of interconnections, or synapses, linking pairs of nodes, and (3) a set of labels, known as weights, associated with each interconnection and identifying some property of the interconnection. These weights correspond to the synaptic efficiency of the biological neurons (Aristodemou *et al* 2005). The output is the result of networking between nodes performing specific individual tasks.

Neural network systems are specified by their respective architectures and training rules. These rules specify the initial set of network connection weights and the training indicates how the weights should be adapted to improve performance. In the present study, we applied the most common learning law, i.e., back propagation (BP), as a training law.

Back propagation neural networks (BPNNs), which are widely used in engineering applications, are parallel distributed information processing structures consisting of neurons interconnected via unidirectional signal channels. In a BPNN,

there are at least three successive layers of neurons: an input layer, one or more hidden layers and an output layer. The number of neurons in the input layer is equal to the number of variables of the input data. These neurons do not have input connections but only output ones. The number of neurons in the hidden layer can be varied based on the complexity of the problem and the size of the input information. Neurons in the hidden layer have both input and output connections. The number of neurons in the output layer is the same as the number of output variables. These neurons have only input connections. With the BP learning rules, the goal of learning is to minimize the error between the desired outputs and the calculated outputs of the network. The learning process for the BPNN involves sending the input values forward through the network and then computing the difference between the calculated output and the corresponding desired output from the training dataset. In this algorithm, the error function is minimized using a gradient-descent technique. The necessary corrections to the weights of the network for each moment are obtained by calculating the partial derivative of the error function with respect to each weight. The resulting weight update is then computed.

A signal x_i at the input of the synapse i connected to neuron j is multiplied by the synaptic weight w_{ij} . A summing junction adds the input signals, which are weighted by the respective synapses of the neuron. Each processing unit (neuron) has an activation function that is commonly chosen to be the sigmoid function. We used the logsig function, which is a special type of sigmoid function:

$$f(x) = \frac{1}{1 + \exp(-x)}. \quad (6)$$

The activation function limits the amplitude of a neuron in that it restricts the permissible amplitude range of the output signal to some finite value. The net input to a processing unit j is given by:

$$\text{net}_j = \sum_i w_{ij}x_i + b_j \quad (7)$$

where x_i is the output from the previous layer, w_{ij} is the weight of the link connecting unit i to unit j . The weights associated with each connection indicate the extent to which the conveyed signal is amplified or diminished. The externally applied bias (threshold) b_j increases or decreases the net input of the activation function, depending on whether it is positive or negative, respectively. It also determines the location of the logsig function on the horizontal axis. The activation

value (output) of unit j is given by:

$$a_j = f(\text{net}_j) = \frac{1}{1 + \exp(-\text{net}_j)}. \quad (8)$$

In its simplest form, the weight-update is a scaled step in the opposite direction of the gradient. Hence, the weight-update rule is:

$$\Delta_p w_{ij}(t) = -\varepsilon \cdot \frac{\partial E_p}{\partial w_{ij}}(t) + \alpha \cdot \Delta_p w_{ij}(t-1) \quad (9)$$

where α is the momentum term and determines the influence of the previous iteration on the present one. In this equation, the total error is given by:

$$E = \text{MSE} = \sum_{p=1}^Q \frac{1}{Q} E_p = \frac{1}{Q} \sum_{p=1}^Q \sum_{j=1}^N (d_{pj} - a_{pj})^2. \quad (10)$$

$\varepsilon \in (0, 1)$ is a parameter that determines the step size and is called the learning rate, and d_{pj} and a_{pj} are the target and the actual response value of the output neuron j , which corresponds to each training sample p th. Q is the number of training samples and N represents the number of output units. This error information is propagated backwards through the ANN and the weights are adjusted. After some number of iterations, the training stops when the calculated output values best approximate the desired values.

The BP includes several kinds of paradigms (e.g., batch back propagation, gradient descent, conjugate gradient, Levenberg-Marquardt, and resilient propagation). The main difference between these paradigms is the method of calculating and updating the weights. The mathematical basis of these paradigms is well described in the literature (e.g., Scales 1985; Battiti 1992; Riedmiller 1993; Hagan et al 1996; Powell 1977; El-Qady and Ushijima 2001). It is very difficult to know which training algorithm will be the fastest for a given problem. The speed depends on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal, and whether the network is being used for pattern recognition or function approximation. The next section compares the various training algorithms.

4. Artificial neural network paradigms

It is very difficult to know which ANN training paradigm will be the fastest for a given problem,

as it depends on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal, and whether the network is being used for pattern recognition or function approximation. This section compares some of the most common training algorithms.

- Batch training with weight and bias learning rules (BTWB)

In a BTWB algorithm, the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated for each training example are added together to determine the change in the weights and biases. Momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface (Hagan *et al* 1996). Acting like a low-pass filter, the momentum allows the network to ignore small features in the error surface. This is important because without momentum, a network may get stuck in a shallow local minimum.

- Conjugate gradient with Fletcher reverse updates (CGFR)

The CGFR method is an algorithm that numerically solves particular systems of linear equations, namely those with symmetric and positive-definite matrices. This method is iterative, so it can be applied to sparse systems that are too large to be handled by direction. Such systems arise regularly when numerically solving partial differential equations (Hagan *et al* 1996).

- Resilient propagation (RPROP)

Resilient propagation (RPROP) algorithms are local adaptive learning schemes, performing supervised batch learning in feed-forward neural networks. The basic principle of an RPROP is to eliminate the harmful influence of the size of the partial derivative on the weight step. Only the sign of the derivative can determine the direction of the weight update; the magnitude of the derivative has no effect. The size of the weight change is determined by a separate update value. The update value for each weight and bias is increased whenever the derivative of the performance function, with respect to that weight, has the same sign for two successive iterations. The update value is decreased whenever the derivative, with respect to that weight, changes sign from the previous iteration. If the derivative is zero, the update value remains the same. Whenever the weights are oscillating, the weight change is reduced. If the weight continues to change in the same direction for

several iterations, the magnitude of the weight change increases. The RPROP algorithm is the fastest algorithm for pattern recognition problems and the memory requirements for this algorithm are relatively small in comparison to the other algorithms considered.

- Gradient descent with momentum and adaptive learning rate (GDMA)

The GDMA algorithm adjusts the weights in the direction of steepest descent (negative of the gradient), which is the direction in which the performance function is decreasing most rapidly. It turns out that although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. The GDMA is usually much slower than the other methods, and it has about the same storage requirements as the RPROP. It can, though, still be useful for some problems (Battiti 1992).

- Levenberg-Marquardt with weight and bias learning rules (LMWB)

The LMWB algorithm (Hagan and Menhaj 1994) was designed to approach second-order training speeds without having to compute the Hessian matrix. This optimization technique is more sophisticated and powerful than gradient descent (Singh *et al* 2005). In general, for function approximation problems and for networks that contain up to a few hundred weights, the Levenberg-Marquardt algorithm will have the fastest convergence. In many cases, the LMWB obtains lower mean square errors than any of the other algorithms tested. However, as the number of weights in the network increases, the advantage of LMWB decreases. In addition, the LMWB performance is relatively poor for pattern recognition problems. The storage requirements of the LMWB are larger than the other algorithms tested (Hagan *et al* 1996).

5. Training and testing data

The creation of the synthetic data required to train and test the ANN is explained below. The model used to produce synthetic data consists of a homogeneous medium of $100\Omega\text{m}$ resistivity with an embedded anomalous body of $1000\Omega\text{m}$ (figure 2). High-resistivity contrast regions are expected in the investigation site used to collect the real field data. In the synthetic data, these high-resistivity values were selected in order to train the ANN. Different sizes were selected for the anomalous body, and its location was moved to different positions within the homogeneous model

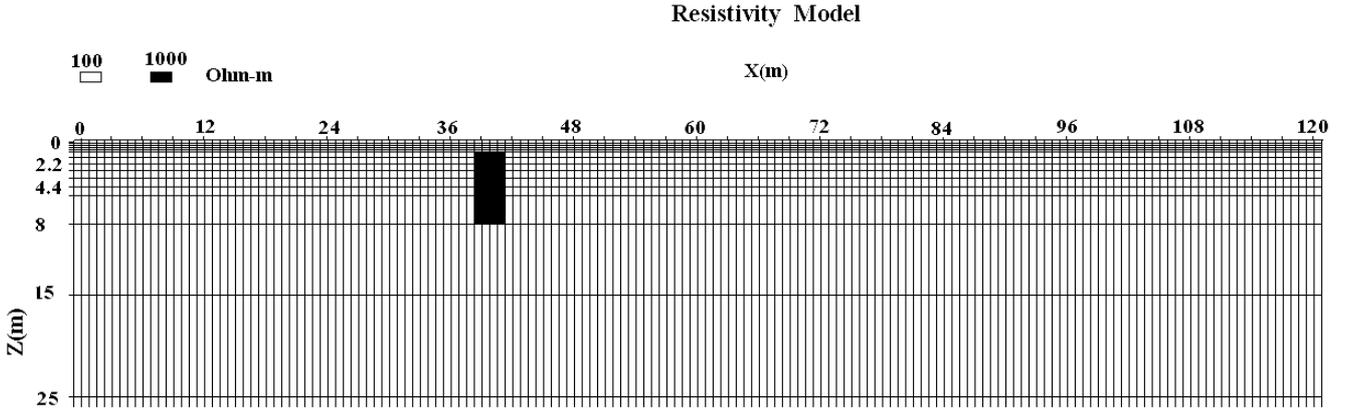


Figure 2. The forward model used to generate the synthetic resistivity dataset. The figure shows one of the positions of the anomalous body.

mesh elements. Each element in the mesh was permitted to be either resistive or conductive. The 2D datasets were generated using a finite element forward modeling code adopted from Loke and Barker (1996b). Each synthetic profile consisted of 101 electrodes and 579 datum points ($n = 1-6$) and the dipole-dipole array was used to generate synthetic data.

In order to reduce computation time (without loss of generality) the data base used in these experiments was smaller than the total data base: 21 randomly selected synthetic datasets (12,159 data point) were used for teaching, while 11 were randomly selected for testing (6369 data point). In order to study the interpolation and extrapolation properties of the ANN, another 24 synthetic datasets (13,896 data point) were generated with different resistivities for the background and anomalous body, ranging from 100–1000 Ωm . The total database contains 56 synthetic datasets (32,424 data point). The synthetic datasets were normalized in a range of $[0, 1]$ for training and testing the ANN; this range allows the logistic sigmoid activation function to restrict the size of the input data (El-Qady and Ushijima 2001).

5.1 Training and testing data pool formation

In this section, we explain how the horizontal and vertical locations of the synthetic resistivity data were determined. These locations are required to train and test the ANN. The horizontal location of the data point is placed at the mid-point of the set of electrodes used to make the measurement (Loke 2009), while the vertical position of the plotting point (depth of investigation) can be placed at the median depth of investigation (Edwards 1977) of the electrode array used. One quantitative means to put a numerical value on the depth of investigation involves using the sensitivity function, or

Table 1. K -value for each data level ($n = 1-6$).

n	1	2	3	4	5	6
k	0.416	0.697	0.962	1.220	1.476	1.730

Frechet derivative, of the array. The sensitivity function essentially tells the degree to which a change in the resistivity of a subsurface section will influence the potential measured by the array. The higher the value of the sensitivity function, the greater is the influence of the subsurface region on the measurement. Mathematically, the sensitivity function is given by the Frechet derivative (McGillivray and Oldenburg 1990). Following Edwards (1977), the median depth of investigation for the dipole-dipole configuration is the electrode spacing (a) multiplied by a coefficient K . Table 1 shows K -values for each data level ($n = 1-6$). Consider the simplest possible array configuration shown in figure 3. The current electrode (C2) is located at $x = 0$, i.e., with all electrodes on the ground surface and “ $a = 2$ ” meters apart (between C2–C1). The measured apparent resistivity corresponding to these electrodes belongs to data level $n = 4$ and is denoted by the letter A. The horizontal location of this point is $x = 4$ (the mid-point of the C2 and P2) and its vertical location, according to table 1, is 2.440 meters ($= 2 \times 1.220$). We can therefore find the apparent resistivity and corresponding true resistivity of this point (4, 2.44) from apparent pseudosection and resistivity model. The same procedure can be done for all data points. As a result, in our synthetic data for each data point, there are four characteristics: the x -, and z -locations and the apparent and true resistivity.

5.2 Number of neurons in hidden layers

There is no general theory on the dependence of the recognition errors on the number of neurons

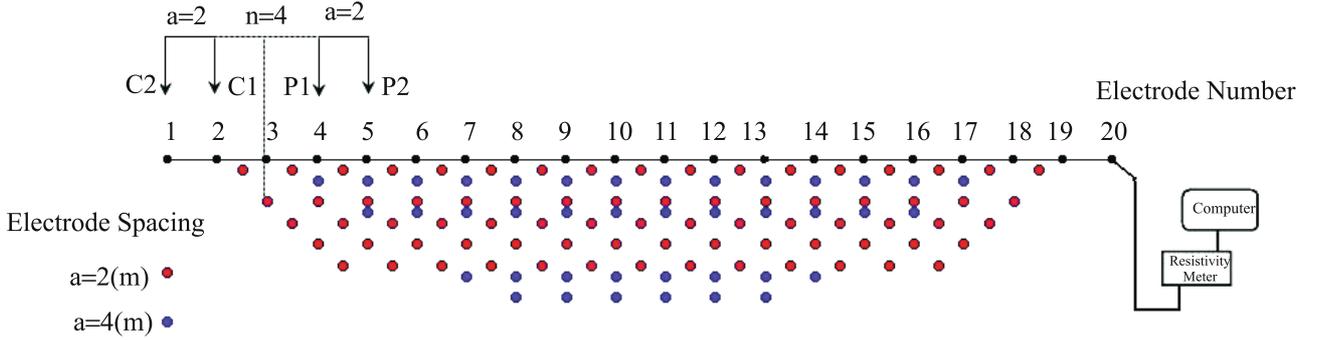


Figure 3. The typical setup for a dipole-dipole configuration with a given number of electrodes along a straight line attached to a multi-core cable.

Table 2. The dependence of the MSE error on the number of neurons in hidden layers.

Nh1	Nh2	MSE error (%)
4	0	0.1649
	4	0.1608
	12	0.0251
	20	0.0204
	28	0.0499
12	32	0.0225
	0	0.0989
	4	0.0793
	12	0.0052
	20	0.0097
20	28	0.0088
	32	0.0016
	0	0.0027
	4	0.0050
	12	0.0007
28	20	0.0003
	28	0.0008
	32	0.0006
	0	0.0006
	4	0.0001
32	12	0.0004
	20	0.0002
	28	0.0006
	32	0.0004
	0	0.006
32	4	0.0009
	12	0.0003
	20	0.0007
	28	0.0004
	32	0.0002

in hidden layers. However, the approximation properties of an ANN are improved when the number of hidden neurons increases (Spichak and Popova 2000). Therefore, we studied the effect

of the number of neurons in hidden layers on the performance of the ANN. The ANN architecture was 3-Nh1-Nh2-1, where Nh1 and Nh2 are the number of neurons in the first and second hidden layer, respectively. The apparent resistivity location (horizontal location x and vertical location z) and apparent resistivity values were used as input data. The true resistivity was selected as the ANN output. This means that there are 3 neurons in the input and 1 in the output layer. The values of Nh1 were assigned as follows: 4, 12, 20, 28, and 32. The values of Nh2 were also assigned according to table 2. The teaching precision was equal to 0.0001, and the number of epochs was selected as infinity in order to attain minimum errors. Table 2 also shows the dependence of the MSE error on the number of neurons in hidden layers.

It has been proved theoretically that a three-layer structure network can estimate any kind of logic function, provided that enough neurons are set in the hidden layer (Irie and Miyake 1988). However, the results of our study shown in table 2 suggest that an ANN with two hidden layers of 28 and 4 neurons in the first and second hidden layers, respectively, has the lowest MSE error.

5.3 Setting the learning rate and momentum

Efficient selection of the training parameters and the network learning paradigm are very important to achieve good performance with the ANN (Baum and Hausler 1989). In BP algorithms, the learning rate, ε , is a small number ($0.1 < \varepsilon < 1.0$) (Aristodemou *et al* 2005) that controls the amount of error that will be negatively added to the interconnection weights for the next iteration (Cranganu 2007). If the learning rate is large, then large changes are allowed in the weight changes and no learning occurs. Conversely, if the learning rate is small, only small changes are allowed, which can increase the learning time. The momentum, α , is a term that dampens the amount of weight change by adding in a portion of the weight change

Table 3. *Effect of learning rate and momentum coefficient in this study.*

Learning rate	Momentum	Epoch	Time(s)	Learning speed
0.001	0	18521	9123.65	2.03
	0.2	8659	6097.89	1.42
	0.4	14887	9482.17	1.57
	0.6	10094	5072.36	1.99
	0.8	9553	8027.73	1.19
0.01	0	3269	2867.54	1.14
	0.2	2282	2402.10	0.95
	0.4	5834	3241.11	1.80
	0.6	8159	4458.47	1.83
	0.8	7598	6279.34	1.21
0.1	0	3908	2605.33	1.50
	0.2	5635	2981.48	1.89
	0.4	9601	5106.91	1.88
	0.6	10214	4621.72	2.21
	0.8	12497	8501.36	1.47
1	0	16097	10385.16	1.55
	0.2	9024	8129.73	1.11
	0.4	24511	12832.98	1.91
	0.6	29005	14648.99	1.98
	0.8	42849	26450.00	1.62

from the previous iteration. The momentum is credited with smoothing out large changes in the weights and with helping the network converge faster when the error is changing in the correct direction. Typical values for the momentum fall between 0 and 1.0 (Aristodemou *et al* 2005). Momentum and learning rate are obviously related,

but their mathematical relation is still not clear (Singh *et al* 2005). We studied the effect of these two parameters, and the results are shown in table 3. The results suggest that a learning rate of 0.01, and a momentum coefficient 0.2, are the most appropriate (table 3).

5.4 Comparison of the ANN paradigms

We tested the following networks learning paradigms: BTWB, GDMA, CGFR, LMWB, and RPROP. These paradigms are mostly based on back-propagation, and the weights are updated after each epoch. The mean square error (MSE) is shown in figure 4 as a function of the number of iterations during the training of different ANN paradigms. According to the variation of the errors, we conclude that the RPROP paradigm was the most efficient for training the dataset. The initial error for the RPROP algorithm is about 0.4 and decreases as the iteration proceeds, until it attains a value of about 0.0001 after 2282 iterations and the network converges. The other algorithms could not achieve the performance goal (0.0001) after 8000 iterations. The comparative results for each paradigm in terms of training speed and epochs (iterations) are summarized in table 4. The training procedure was carried out on a 1.73-GHz Dual-Core PC where it took approximately 40 minutes for the algorithm to converge to a satisfactory performance. Once the ANN converged, the weights were adapted and stored. The ANN then performed the inversion of the field data in a few seconds, using the updated weights, without any more training.

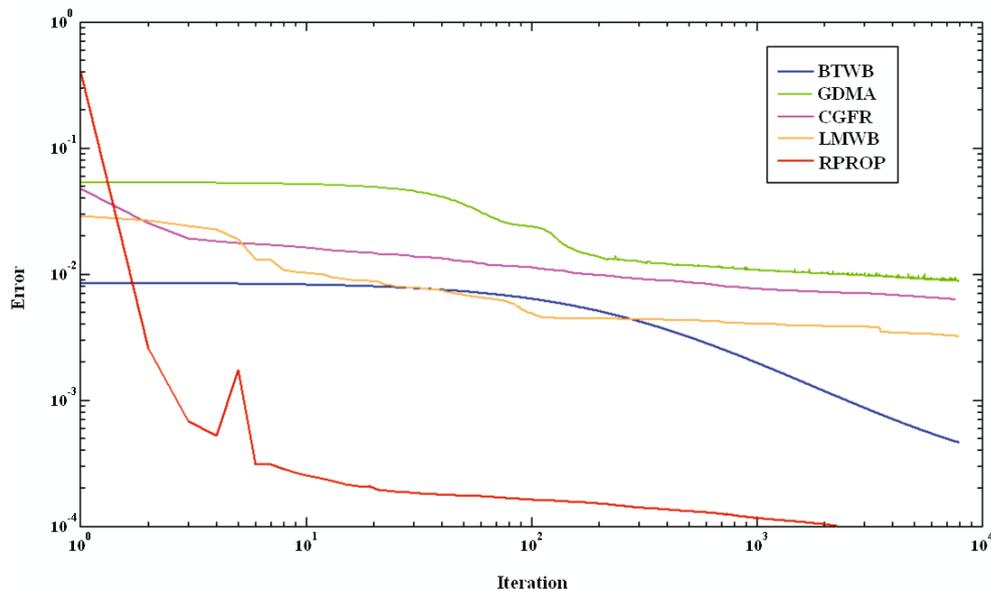


Figure 4. RMS errors as a function of iteration number for different ANN paradigms during the training stage.

Table 4. The comparative results for each paradigm in terms of training speed and epochs.

Paradigm	Time(s)	Epochs	MSE error (%)
Batch training with weight and bias learning rules (BTWB)	11880.7	8000	≈ 0.0004
Conjugate gradient with Fletcher reverse updates (CGFR)	10980.1	8000	≈ 0.0064
Resilient propagation (RPROP)	2402.1	2282	0.0001
Gradient descent with momentum and adaptive learning rate (GDMA)	10080.2	8000	≈ 0.0009
Levenberg-Marquardt with weight and bias learning rules (LMWB)	10341.6	8000	≈ 0.0032

6. Testing the trained ANN performance with synthetic data

The ANN performance was tested using 11 test sets. These test sets were not used during the training stage. The error range is between 0.0001 and 0.0003 for the training data. The error range for the test data is between 0.0001 and 0.0006. It is evident that the error values are higher for the test data compared to the training sets.

Two examples of synthetic test models are shown in figures 5 and 6. These models were generated using a finite elements code (Loke and Barker 1996b). The number of electrodes was 21 with a minimum electrode spacing of 2 m. The resistivity of anomalies is $1000 \Omega\text{m}$ and the background resistivity is $100 \Omega\text{m}$. The resistivity model in figure 5 consists of two rectangular shapes as anomaly at both sides of the grid. Figure 5(a) and (c) show the results of inversion by the RIT and the ANN method respectively. The apparent resistivity data used to obtain these results are free of noise. When the inversion results are compared with the resistivity model, we can see that the ANN result is more realistic than the RIT. Furthermore, we considered 3% noise uniformly in generated synthetic data in order to see the effect of noise in the inversion by both methods. Figure 5(b) and (d) show the inversion results of noisy data for the RIT and the ANN methods respectively. According to these figures, the accuracy of the ANN result is still better than the RIT.

As another example, we considered a more complex anomaly in resistivity model and the inversion results for the RIT and the ANN methods are shown in figure 6. Similar to the previous example, figure 6(a) and (c) are related to the measurements without any noise and figure 6(b) and (d) are related to the data involving 3% uniform noise. As we can see in figure 8, the ANN results

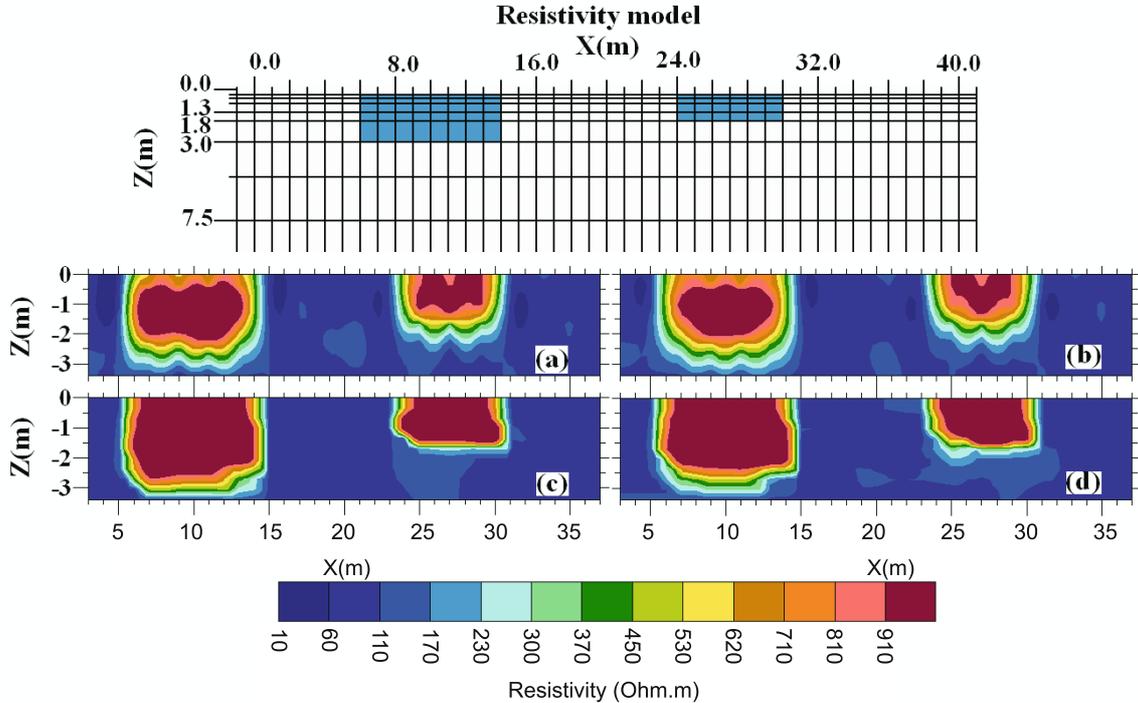


Figure 5. Inversion results of a synthetic test model by the RIT (a) without noise and (b) with 3% uniform noise and the ANN method (c) without noise and (d) with 3% uniform noise. The resistivity of background in resistivity model is $100 \Omega\text{m}$ and the resistivity of anomaly is $1000 \Omega\text{m}$.

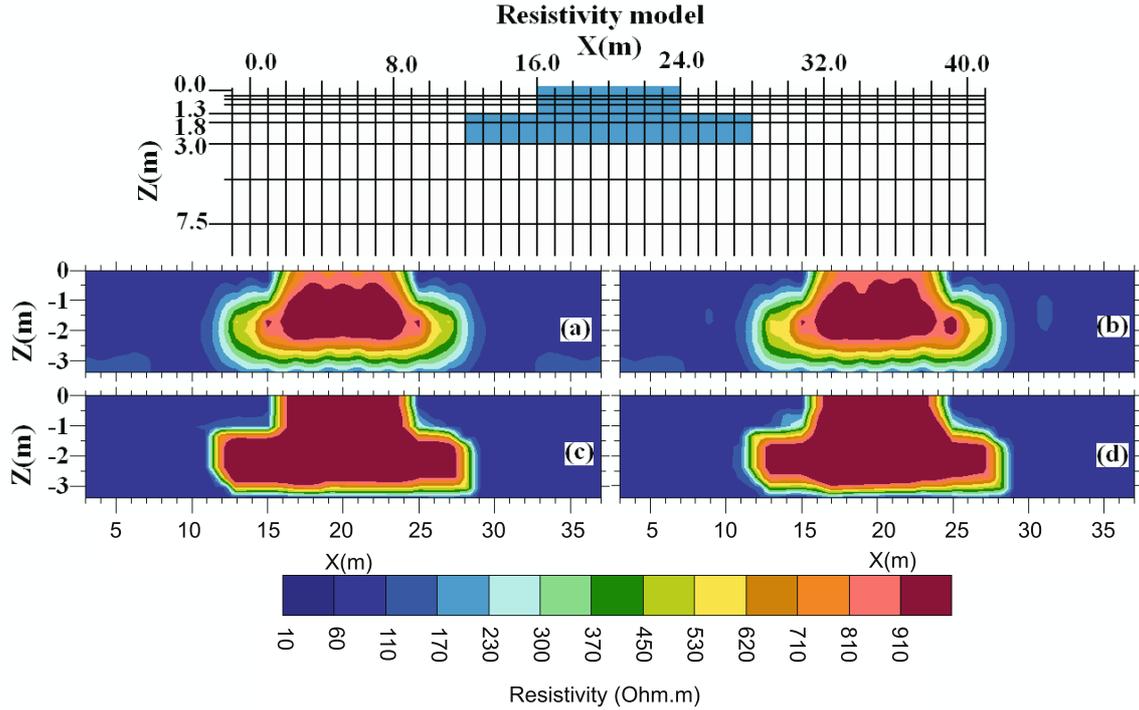


Figure 6. Inversion results of a complex synthetic test model by the RIT (a) without noise and (b) with 3% uniform noise and the ANN method (c) without noise and (d) with 3% uniform noise. The resistivity of background in resistivity model is $100 \Omega\text{m}$ and the resistivity of anomaly is $1000 \Omega\text{m}$.

(figure 6c–d) show the location and dimensions of the anomaly more accurately than RIT (figures 8a and 6b).

7. The ANN interpolation and extrapolation properties

In order to study the interpolation and extrapolation properties of the ANN, another 13896 synthetic data points (24 sets) were generated. The range from $100\text{--}1000 \Omega\text{m}$ was divided into 100, 200, 300, 400, 500, and $600 \Omega\text{m}$ as the background resistivity, and different resistivity values for the anomalous body were considered. For example, when the background resistivity was $100 \Omega\text{m}$, the resistivity values for the anomalous body were 300, 700, 2000, and $4000 \Omega\text{m}$ to cover the interpolation and extrapolation properties of the ANN. The details of the resistivity distribution for the other test sets can be seen in table 5. In table 5, the RMS error between the results of the ANN and the corresponding true resistivity distributions are shown for each test set. For example, the RMS errors for four separate test sets, in which all of them have a background resistivity of $200 \Omega\text{m}$; the anomalous body in these sets has a resistivity of 400, 800, 4000, and $6000 \Omega\text{m}$, respectively. The RMS error for all interpolation and extrapolation test sets are in the

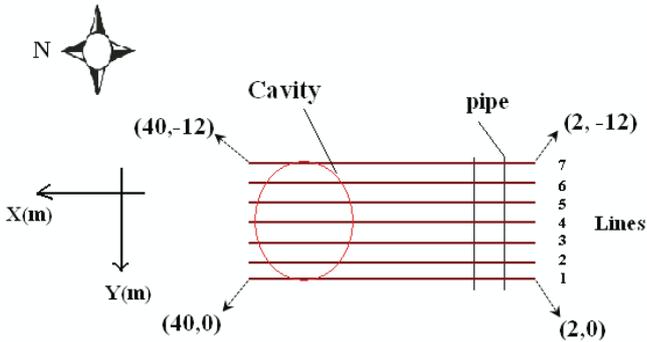
range of $0.0001\text{--}0.008$. Thus, we can conclude that the network is properly designed and trained.

8. ANN interpolation of the field data

To check the ability of ANNs to perform the DC resistivity inversion, the trained ANN was applied to a quasi-3D field dataset. The site is located near the University of Malaya, and the survey area stretches over a flat playground covered with grass. The direct resistivity measurement of the soil at the surface gives values ranging from $150 \Omega\text{m}$ to $600 \Omega\text{m}$. The soil is relatively porous, very sandy, and probably originates from weathered bedrock. There are, however, areas of more resistive materials within the soil. The high resistivity values are caused by construction materials left behind during the construction of a horizontal pipe system. The investigation site consists of a large cavity (outcropping and downward to 7 m) at the north side of the site (figure 7). In addition, there is a horizontal concrete pipe (at a depth of 5.5 m) to the south. The resistivity values of the concrete, measured directly, range from $850 \Omega\text{m}$ to $1500 \Omega\text{m}$. These values were used to determine the boundaries of the concrete structures in the inversion results. Values greater than $1500 \Omega\text{m}$ were assumed to correspond to the empty space within the concrete or the cavity.

Table 5. RMS error between the results of the ANN and the corresponding true resistivity distributions for each test set.

Resistivity of back ground (Ωm)					
100	Resistivity of anomaly (Ωm)	300	700	2000	4000
	RMS error (%)	0.00048	0.00038	0.00041	0.00049
200	Resistivity of anomaly (Ωm)	400	800	4000	6000
	RMS error (%)	0.00054	0.00046	0.00071	0.0008
300	Resistivity of anomaly (Ωm)	500	700	5000	7000
	RMS error (%)	0.00532	0.00762	0.0067	0.00069
400	Resistivity of anomaly (Ωm)	600	800	6000	8000
	RMS error (%)	0.00792	0.00645	0.0007	0.00078
500	Resistivity of anomaly (Ωm)	700	900	9000	11000
	RMS error (%)	0.00179	0.00262	0.00011	0.00013
600	Resistivity of anomaly (Ωm)	800	900	15000	20000
	RMS error (%)	0.00791	0.00696	0.0071	0.00199

Figure 7. Location map. Survey consists of seven north-south lines. The dipole-dipole arrays were used in the x -direction.

Pole-dipole and pole-pole arrays were not suitable in this case because the field conditions did not allow the placement of remote electrodes at a sufficient distance from the survey lines. The dipole-dipole array (figure 3) has better horizontal resolution and better depth coverage at the ends of the lines compared to the Wenner array. At the investigation site, the horizontal extent of subsurface structures is more interesting than the vertical structures. Therefore, the dipole-dipole array was used for this study.

Measurements were collected along seven parallel lines, and the electrode cables were oriented in the x -direction, which is approximately north to south (figure 7). Each cable covered 38 m between the first and the last take-out, with a spacing of 2 m in the x -direction. Roll-along measurements using

a y -spacing of 2 m were carried out until a grid of 20×7 electrodes was covered. Thus, the total investigation area was 38×12 m, and 140 electrode positions were used. Measurements were only taken in the x -direction to limit the time needed for data acquisition.

The median depth of investigation for a dipole-dipole array depends on the ' n ' factor as well as the ' a ' factor, and the voltage is inversely proportional to the cube of the ' n ' factor for the same current electrode spacing. Thus, the signal strength is very small for ' n ' values greater than 6. Therefore, to gain reliable measurements we used electrode spacing of $a = 2 - m$ and ' n ' values not greater than 6, which gave a dataset of 85 data points for each line ($n = 1-6$). The background noise levels were checked before measurement began, and the vertical stack was set to four. The relative standard deviation for each stack was checked during the measurements, as it is a valid indicator of the quality of the measurements (Tohon *et al* 2004). The standard deviation of the measurements was typically less than 4%. Seven separate final sets of apparent resistivity measurements were collected using a Terrameter SAS 4000 instrument (Dahlin 1996).

The 2D resistivity imaging data obtained by the dipole-dipole array was inverted separately using the RIT. After seven iterations, the inversion process converged with the data misfit error shown in table 6. Initially, we tried different numbers of iterations for the inversion process, but the relative change in error on iterations beyond seven

Table 6. Data misfit error (%) for seven data field sets after seven iterations of the inversion process.

Line number	1	2	3	4	5	6	7
Robust inversion method	8.35	7.93	4.54	7.20	8.11	5.57	4.70
ANN inversion method	8.00	8.30	4.10	3.90	7.00	4.45	3.86

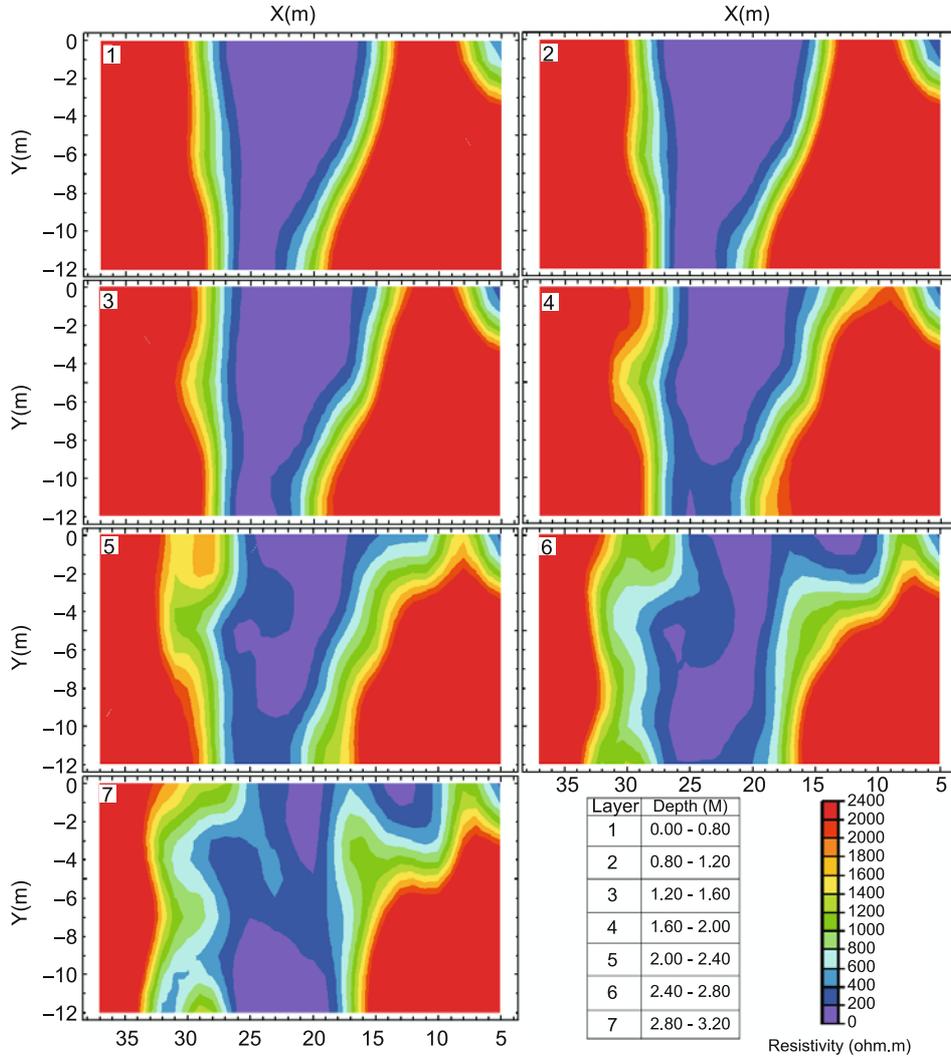


Figure 8. The 3D resistivity model obtained by the ANN method, as horizontal depth slices.

was only about 0.3%. Because this is a negligible error change, the number of iterations was limited to seven in order to avoid over-fitting.

Field datasets are inherently noisier than the synthetic data. Therefore, to attain sufficient performance on the field data, the ANN must be adequately trained to detect the main features of the input data. The error for observed data in the inversion process using the ANN is relatively higher than errors for the synthetic data.

The trained ANN was also applied to field datasets of seven lines (1 to 7) and used for

studying the performance of the adopted network. These data were also used to compare the two methods. Horizontal depth slices were constructed using the inverted data output of the ANN as well as the results of the inversion using the RIT (figures 8 and 9). There was a good correlation between the inversion results for the quasi-3D dipole-dipole data obtained with the RIT and with the ANN approach. It is clear that the depth slices resulting from the ANN show more detailed structures, which may be related to construction material left in the investigation site.

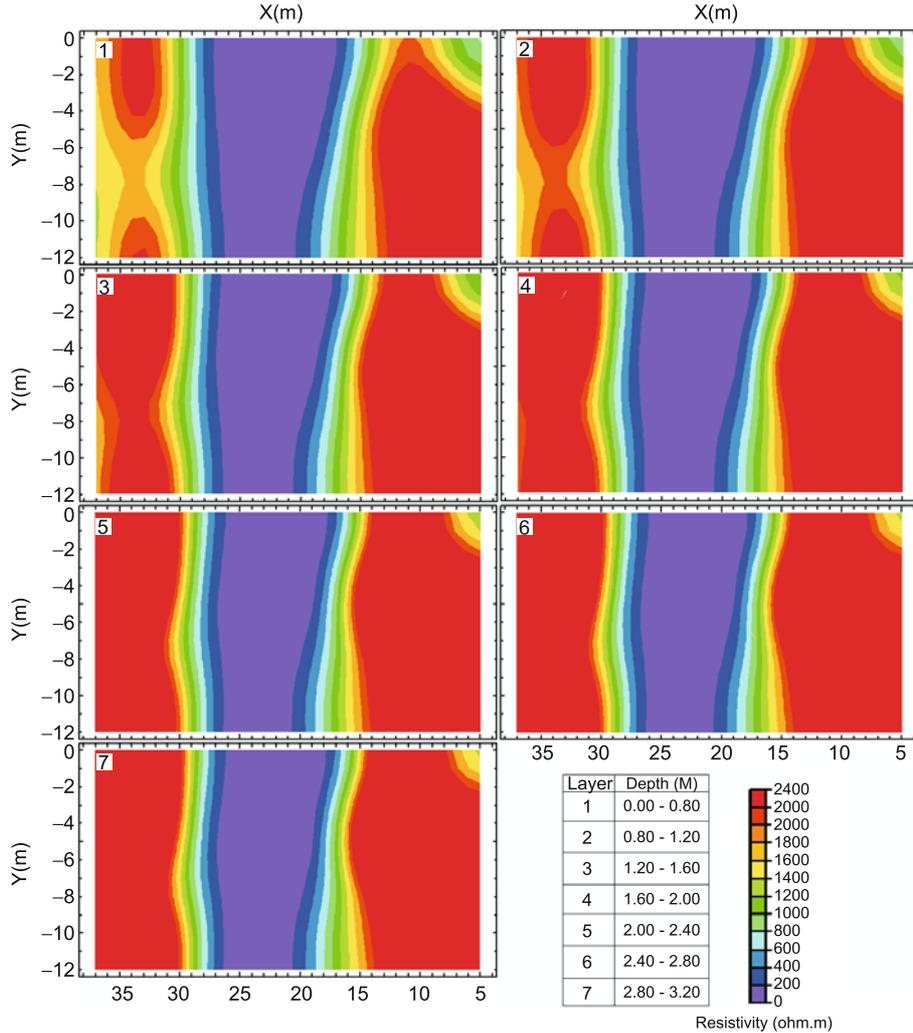


Figure 9. The 3D resistivity model obtained by the RIT, as horizontal depth slices.

9. Conclusion

In this study, we applied the ANN approach to invert 3D ERI data. We investigated how a four-layer network with 28 and 4 neurons in the first and second hidden layers can invert quasi-3D ERI data. The ANN was trained on 21 synthetic datasets in which each set consisted of 579 samples. In the training process, the resilient propagation algorithm (RPROP) was the most efficient paradigm for the inversion of the apparent resistivity data. A study on appropriate selections of learning rate and momentum coefficients was done, and the results suggest a learning rate of 0.01 with momentum coefficient 0.2 being the most efficient.

The test on 35 synthetic datasets indicated that the ANN system was able to converge accurately to the target. In 11 test datasets, the background resistivity was $100 \Omega\text{m}$ with an embedded anomalous body of $1000 \Omega\text{m}$. The resistivity of the background and the anomalous body in

another 24 test datasets was varied ranging from $100\text{--}1000 \Omega\text{m}$ in order to check the interpolation and extrapolation properties of trained ANN. The maximum RMS error range was between 0.0001 and 0.008.

Tests on field data were also carried out using a known subsurface structure; the results from the inversion and forward modeling show that the ANN approach was accurate and very rapid. We compared the ANN to the conventional procedure to determine its relative efficiency and accuracy. The ANN system resolved the isolated feature well, and a good correlation between the results of the ANN and the conventional procedure was found. However, the ANN results show more details of the subsurface structure than the conventional procedure. The particular advantage presented by the ANN for resistivity inversion is that once the network is trained, it can perform the inversion of such quasi-3D DC resistivity datasets very rapidly. The ANN's results can be used as an initial

model for inversion in order to decrease the inversion time.

Acknowledgements

The work behind this paper received financial support from the University of Malaya (UM), Malaysia, through a fellowship of reference (PS203/2008B) which is gratefully acknowledged. The authors also thank two anonymous reviewers for their constructive criticism.

References

- Aristodemou E, Pain C, De Oliveira C, Goddard T and Harris C 2005 Inversion of nuclear well-logging data using neural networks; *Geophys. Prospect.* **53** 103–120.
- Battiti R 1992 First and second order methods for learning: Between steepest descent and Newton's method; *Neural Computation* **4**(2) 141–166.
- Baum E and Hausler D 1989 what size net gives valid generalization? In: *Advances in Neural Information Processing Systems I* (ed.) Touretzky D, 80–90. Morgan Kaufman.
- Claerbout J F and Muir F 1973 Robust modeling with erratic data; *Geophysics* **38** 826–844.
- Cranganu C 2007 Using Artificial Neural Networks to predict the presence of overpressured Zones in the Anadarko Basin, Oklahoma; *Pure Appl. Geophys.* **164** 2067–2081.
- Dahlin T 1996 2D resistivity surveying for environmental and engineering applications; *First Break* **14** 275–284.
- Dahlin T and Loke M H 1997 Quasi-3D resistivity imaging-mapping of three dimensional structures using two dimensional DC resistivity techniques; *Proceedings of the 3rd Meeting of the Environmental and Engineering Geophysical Society*, Aarhus, Denmark, 143–146.
- Edwards L S 1977 A modified pseudosection for resistivity and induced-polarization; *Geophysics* **42** 1020–1036.
- El-Qady G and Ushijima K 2001 Inversion of DC resistivity data using neural networks; *Geophys. Prospect.* **49** 417–430.
- Hagan M T and Menhaj M 1994 Training feed forward networks with the Marquardt algorithms; *IEEE Transactions on Neural Networks* **5**(6) 989–993.
- Hagan M T, Demuth H B and Beale M H 1996 *Neural Network Design*, Boston, MA: PWS Publishing, chapter 10–12.
- Irie B and Miyake S 1988 Capabilities of three-layered perceptrons; In: *Proceedings of the 1st IEEE International Conference on Neural Networks*, SOS Printing, San Diego, CA 1 (eds) Caudill M and Butler C, 641–648.
- Loke M H and Barker R D 1996a Practical techniques for 3D resistivity surveys and data inversion; *Geophys. Prospect.* **44** 499–523.
- Loke M H and Barker R D 1996b Rapid least-squares inversion of apparent resistivity Pseudo-sections using quasi-Newton method; *Geophys. Prospect.* **44** 131–152.
- Loke M H 2009 Tutorial: 2-D and 3-D electrical imaging surveys. www.geoelectrical.com
- Practical Guide to 2D and 3D Surveys. <http://www.terrajp.co.jp/lokenote.pdf>.
- McGillivray P R and Oldenburg D W 1990 Methods for calculating Fréchet derivatives and sensitivities for the non-linear inverse problem: A comparative study; *Geophys. Prospect.* **38** 499–524.
- Minsky M L and Papert S A 1969 *Perceptron*: MIT press.
- Mochales T, Casas A M, Pueyo E L, Pueyo O, Roman M T, Pocovi A, Soriano M A and Anson D 2008 Detection of underground cavities by combining gravity, magnetic and ground penetrating radar surveys: A case study from the Zaragoza area, NE Spain; *Environ. Geol.* **53** 1067–1077.
- Poulton M and El-Fouly A 1991 Preprocessing GPR signatures for cascading neural network classification; *61st SEG meeting*, Houston, USA, Expanded Abstracts 507–509.
- Poulton M, Sternberg K and Glass C 1992 Neural network pattern recognition of subsurface EM images; *J. Appl. Geophys.* **29** 21–36.
- Powell M J D 1977 Restart procedures for the conjugate gradient method; *Math. Program.* **12** 241–254.
- Riedmiller M 1993 *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, San Francisco 586–591.
- Rosenblatt F 1962 *Principles of neuro-dynamics*: Spartan books.
- Scales L E 1985 *Introduction to non-linear optimization* (New York: Springer-Verlag).
- Singh U K, Somvanshi V K, Tiwari R K and Singh S B 2002 Inversion of DC resistivity data using neural network approach; In: *Proceedings of the International Groundwater Conference, Dindigul, India, IGC-2002*, 57–64.
- Singh U K, Tiwari R K and Singh S B 2005 One-dimensional inversion of geoelectrical resistivity sounding data using artificial neural networks – a case study; *Comput. Geosci.* **31** 99–108.
- Spichak V V and Popova I V 2000 Artificial neural network inversion of MT – data in terms of 3D earth macro – parameters; *Geoph. J. Int.* **42** 15–26.
- Spitzer K 1998 The three-dimensional DC sensitivity for surface and subsurface sources; *Geophys. J. Int.* **134** 736–746.
- Thomas G 2004 *Inversion Methods and Resolution Analysis for the 2D/3D Reconstruction of Resistivity Structures from DC Measurements*. Ph.D thesis, University of Freiberg, pp. 160
- Tohon D S, Vannesta K, Sintubin M, Muchez P and Waelkens M 2004 Two-dimensional resistivity imaging: a tool in archaeoseismology: An example from Ancient Sagalassos (Southwest Turkey); *Archaeol. Prospect.* **11** 1–18.
- Tsourlos P and Ogilvy R 1999 An algorithm for the 3-D Inversion of Tomographic Resistivity and Induced Polarization data: Preliminary Results; *J. Balkan Geophys. Soc.* **2**(2) 30–45.
- Wolke R and Schwetlick H 1988 Iteratively reweighted least squares algorithms, convergence analysis, and numerical comparisons: *SIAM Journal of Scientific and Statistical Computations* **9** 907–921.
- Zhao S and Yedlin M J 1996 Some refinements on the finite-difference method for 3-D dc resistivity modeling; *Geophysics* **61** 1301–1307.