GEOPHYSICS, VOL. 71, NO. 5 (SEPTEMBER-OCTOBER 2006); P. H45–H53, 6 FIGS.
10.1190/1.2231110

Enhanced interpretation of magnetic survey data from archaeological sites using artificial neural networks

David. J. Bescoby[1], Gavin C. Cawley[2], and P. Neil Chroston[1]

ABSTRACT

The use of magnetic surveys for archaeological prospecting is a well-established and versatile technique, and a wide range of data processing routines are often applied to further enhance acquired data or derive source parameters. Of particular interest in this respect is the application of artificial neural networks (ANNs) to predict source parameters such as the burial depths of detected features of interest. Within this study, ANNs based upon a multilayer perceptron architecture are used to perform the nonlinear mapping between buried wall features detected within the magnetic data and their corresponding burial depth for surveys in the ancient city of Butrint in southern Albania, achieving a greater level of information from the survey data.

Suitable network training examples and test data were generated using forward models based upon ground-truth observations. The training procedure adopts a supervised learning routine that is optimized using a conjugate gradient method, while the learning algorithm also prunes network elements to prevent overregularization by reducing model complexity. Data processing was further enhanced by introducing rotational invariance using Zernike moments and by utilizing the combined output of a number, or committee, of networks. When applied to a section of survey data from Butrint, the ANN routine successfully predicted the burial depth of a number of detected wall features, with an rms error on the order of 0.20 m, and provided a coherent map of the buried building foundations. The neural network approach offered advantages in terms of efficiency and flexibility over more conventional data-inversion techniques within the context of the study, giving fast solutions for large, complex data sets while having high noise tolerance.

INTRODUCTION

Geophysical prospecting using magnetic surveys plays a routine part in the investigation of archaeological sites and has become an important tool where the potential for archaeological excavation is limited. A large portion of archaeological geophysical surveys are concerned primarily with identifying the location and spatial extent of buried remains, although the data collected are likely to contain further information relating to the depth and geometry of anomalous features. The current study seeks to derive shape and depth estimates for wall foundation features recorded within magnetic survey data, providing an enhanced interpretation of the results.

The inversion of magnetic data to derive source parameters can be tackled in a number of ways, depending upon the application and the desired results. Often the inversion is achieved by constructing a corresponding magnetic model of the subsurface, containing a number of adjustable parameters optimized to find a satisfactory approximation of the data being modeled. This method of successful estimation and reconstruction of subsurface features is demonstrated for archaeological data by Herwanger et al. (2000) through modeling a series of medieval pit dwellings and associated ditches from Unterstammheim, northern Switzerland, and by Eder-Hinterleitner et al. (1996) in the reconstruction of a Neolithic ring ditch system at Puch in lower elevations in Austria.

However, processing large, complex, and noisy data sets presents certain optimization problems likely to prevent a consistent solution using these inversion techniques. This has led to the consideration of neural computing techniques and the use of artificial neural networks (ANNs) for inverting magnetic survey data. The neural network paradigm has been implemented successfully within many areas of geophysics involving problems for which only very complicated solutions exist (Raiche, 1991; Van der Baan and Jutten, 2000; Poulton, 2001, 2002), including the inversion and processing of seismic data (Calderón-Macías et al., 2000; Poulton, 2002) and parameter estimation from well logs (Helle et al., 2001). Closer to the current study, ANNs have been applied successfully to the inversion of electromagnetic and gravity geophysical data (Poulton et al., 1992a,

Manuscript received by the Editor October 26, 2004; revised manuscript received January 30, 2006; published online August 28, 2006.
[1]University of East Anglia, School of Environmental Science, Norwich, Norfolk NR4 7TJ, United Kingdom. E-mail: d.bescoby@uea.ac.uk; p.chroston@uea.ac.uk.
[2]University of East Anglia, School of Information Systems, Norwich, Norfolk, NR4 7TJ, United Kingdom. E-mail: gcc@cmp.uea.ac.uk.

b; Spichak and Popova, 2000; El-Qady and Ushijima, 2001; Ucan et al., 2002), while the inversion and analysis of magnetic data has been demonstrated by Fossati et al. (1992), Guo et al. (1992), and Ucan et al. (2002).

The remainder of this paper describes the development and implementation of ANNs for solving the inverse problem for magnetic survey data. A technical description of the neural network procedure is presented, with particular attention given to the implementation of a suitable learning algorithm. The development of synthetic training data is then described, and the technique is demonstrated using synthetically generated test data before being applied to a section of magnetic survey data from the Roman city of Butrint in southern Albania.

USE OF ARTIFICIAL NEURAL NETWORKS

ANNs can be a powerful tool for performing nonlinear functional mapping between a set of input variables (geophysical data) and a set of output or source parameters, together with particular procedures for optimizing the mapping (Bishop, 1995). The importance of ANNs in this context is that they can function as universal approximators and are able to map any continuous function to arbitrary accuracy (Yarger et al., 1978; Jang et al., 1997; Jain and Martin, 1999). This is achieved through adopting a massively parallel connectionist architecture (Jang et al., 1997) of simple processing units (perceptrons), the basic functioning of which was inspired originally by the biological neuron. The processing unit produces an output or is activated at a certain threshold determined by the value of its weighted input. The neural network optimizes the mapping by using a data set of training data, which contains examples of the functional mapping that the network is to learn.

Neural networks are proven tools for processing geophysical data because of their inherent ability to deal with noisy and incomplete data sets (Raiche, 1991; Spichak and Popova, 2000). However, the main advantage of using neural networks to solve the inversion problem is that they effectively provide nonlinear mapping of the geophysical phenomenon without assuming an explicit physical model of the process (Williams, 1993). Therefore, the total time necessary for a neural network solution depends on the dimensions of the space of unknown parameters rather than the physical dimensions of the modelled area (Spichak and Popova, 2000). This makes ANNs very computationally efficient tools if multiple inversions are required, because once a network has been optimized or trained, it effectively remembers the inversion solution (Spichak and Popova, 2000). It can therefore be applied easily to new or spatially extensive survey data with almost instantaneous results. However, for the effective processing of new data, it is important that the parameters of the training data used to train the original network be comparable with the new data. In this respect, the training data set must be designed carefully for maximum flexibility.

Neural network procedure

Functional mapping can be illustrated in terms of a mathematical function that contains a number of adjustable parameters whose values are determined using the training data (Bishop, 1995). Such a representative function could be written in the form

$$y = y(\mathbf{x};\mathbf{w}),$$

where $\mathbf{x}$ is a vector of input variables (magnetic field strength values), y is the network output (source depth), and $\mathbf{w}$ denotes the vector of adjustable weights (Bishop, 1995, p. 5). The value of individual weights effectively controls the strength of the interconnections between processing units within the network, determining the overall flow of data. The training procedure is applied to find the optimum values for $\mathbf{w}$, which can be seen as a process of minimizing the error $E_{\mathcal{D}}$ between the desired network output given a particular input vector and the actual value predicted by the network. This process of optimization is often referred to as supervised learning (Bishop, 1995) and utilizes a set of training examples, $\mathcal{D} = \{(\mathbf{x}_i, t_i)\}_{i=1}^{N}$, where t_i is the desired network output or target for the ith training pattern.

Many different types of ANN architectures and training methods have been developed [see Haykin (1994) and Raiche (1991)]. For this study, we adopt the multilayer perceptron (MLP) network architecture and associated learning procedures, which are widely applied to nonlinear functional mapping problems (see Bishop, 1995). The network architecture consists of two layers of adaptive weights with a hidden layer of processing units between the inputs and the output processing unit. The outputs from one layer become the inputs to the next layer, with connections initially running from every processing unit in one layer to every unit in the next layer, forming a feed-forward architecture with no feedback loops. An explicit expression for the complete function represented by the network can therefore be written as

$$y = \sum_{j=1}^{M} w_{kj} z_j g\left(\sum_{i=1}^{d} w_{ji} x_i \right),$$

where w_{kj} denotes second-layer weights, M is the number of hidden layer units, and z_j is their outputs. Similarly, w_{ji} denotes first-layer weights, where d is the number of inputs. The units in the hidden layer employ nonlinear sigmoid activation functions g, while the output unit has a linear activation function, so that the range of possible output values is not limited.

Network training

The minimization of the error $E_{\mathcal{D}}$ between the predicted network output and desired output represented in the training data is essentially a nonlinear regression that can be represented initially by the sum-of-squares error metric, given by

$$E = \frac{1}{2} \sum_{n=1}^{N} \| y(\mathbf{x}^n;\mathbf{w}) - \mathbf{t}^n \|^2, \qquad (1)$$

where y is the output, t is the training data target, and N is the number of patterns in the training set. The derivatives of the error with respect to the network weights $\partial E^n/\partial w_{kj}$ and $\partial E^n/\partial w_{ji}$ are then evaluated using a process known as error back-propagation, corresponding to a propagation of errors backward through the network (Rumelhart et al., 1995; Kecman, 2001).

Optimization

The error function is minimized by applying a conjugate gradient optimization method (Williams, 1991), found to be more efficient than the standard gradient descent method for large numbers of inter-

connecting weights within the network. The search through weight space using the conjugate gradient method is based upon the update rule, given by Williams (1991) as

$$\mathbf{w}_{j+1} = \mathbf{w}_j + \alpha_j \mathbf{d}_j,$$

where $\mathbf{w_j}$ is the weight vector at the jth iteration. The next iteration involves choosing a suitable search direction $\mathbf{d_j}$ and step length $\alpha_\mathbf{j}$ along that direction. For any given search direction in weight space, the minimum of the error function along that direction is found, from which point a new search direction is then computed (Bishop, 1995, p. 279).

DEVELOPING A LEARNING ALGORITHM

To avoid overfitting to the training data, a regularized error function (Tikhonov and Arsenin, 1977) is adopted by adding a term $E_\mathcal{W}$, which has the effect of penalizing overly complex models, i.e.,

$$E = \alpha E_\mathcal{W} + \beta E_\mathcal{D}, \tag{2}$$

where α and β are regularization parameters controlling the bias variance trade-off (Geman et al., 1992). Minimizing a regularized error function of this nature is equivalent to the Bayesian approach, which seeks to maximize the posterior density of the weights (Mackay, 1992a; Neal, 1996), given by

$$P(\mathbf{w}|\mathcal{D}) \propto P(\mathcal{D}|\mathbf{w})P(\mathbf{w}),$$

where $P(\mathcal{D}|\mathbf{w})$ is the likelihood of the data and $P(\mathbf{w})$ is a prior distribution over $\mathbf{w}$. The form of the functions $E_\mathcal{D}$ and $E_\mathcal{W}$ correspond to distributional assumptions regarding the data likelihood and prior distribution over network parameters, respectively.

The usual sum-of-squares metric given in equation 1 corresponds to a Gaussian likelihood,

$$P(\mathcal{D}|\mathbf{w}) = \frac{1}{\sqrt{2\pi\beta^{-1}}} \exp\left\{-\frac{[\mathbf{t}^n - \mathbf{y}(\mathbf{x}^n)]^2}{\mathbf{2}\beta^{-1}}\right\},$$

with fixed variance $\sigma^2 = 1/\beta$. For this study, the Laplace prior propounded by Williams (1995) is adopted, which corresponds to an ℓ_1 norm regularization term,

$$E_\mathcal{W} = \sum_{j=1}^{W} |w_j| \Leftrightarrow P(w) = \frac{1}{2\beta} \exp\left\{-\frac{|w|}{\beta}\right\},$$

where W is the number of model parameters. An interesting feature of the Laplace regularizer is that it leads to pruning of redundant model parameters. From equation 2 at a minimum of E, we have

$$\left|\frac{\partial E_\mathcal{W}}{\partial w_j}\right| = \frac{\alpha}{\beta} \quad w_j > 0, \qquad \left|\frac{\partial E_\mathcal{W}}{\partial w_j}\right| < \frac{\alpha}{\beta} \quad w_j = 0.$$

As a result, any connection weight within the network not obtaining the data misfit sensitivity of α/β is set exactly to zero and can be removed or pruned from the network.

Eliminating regularization parameters

The hyperparameters α and β can be estimated by maximizing the evidence (MacKay, 1992a) or alternatively may be integrated analytically (Buntine and Weigend, 1991; Williams, 1995). Here the latter approach is followed; the posterior distribution of the parameters is given by

$$p(\mathbf{w}) = \int p(\mathbf{w}|\alpha)p(\alpha)\, \mathbf{d}\alpha. \tag{3}$$

Assuming the Laplace prior, the prior distribution over the weights of the network, conditioned on the regularization parameter α, is given by

$$p(\mathbf{w}|\alpha) = \mathbf{Z}_\mathcal{W}(\alpha)^{-1}\exp\{-\alpha\mathbf{E}_\mathcal{W}\}, \tag{4}$$

where the necessary normalizing constant is given by

$$Z_\mathcal{W}(\alpha) = \left(\frac{2}{\alpha}\right)^W. \tag{5}$$

Substituting equations 4 and 5 into equation 3, adopting the (improper) uninformative Jeffreys prior $p(\alpha) = 1/\alpha$ (Jeffreys, 1939) and noting that α is strictly positive,

$$= \int_0^\infty 2^{-W}\alpha^{W-1}\exp\{-\alpha E_\mathcal{W}\}\mathbf{d}\alpha.$$

Using the gamma integral $\int_0^\infty x^{\nu-1}e^{-\mu x}dx = \Gamma(\nu)/\mu^\nu$ [Gradshteyn and Ryzhic (1994), their equation 3.384], we obtain

$$p(\mathbf{w}) = \frac{\Gamma(W)}{(2E_\mathcal{W})^W}.$$

Taking the negative logarithm and omitting irrelevant constant terms,

$$-\log p(\mathbf{w}) = W \log E_\mathcal{W}. \tag{6}$$

Applying a similar treatment to the data misfit term (assuming a sum-of-squares error), we have

$$E = \frac{1}{2}N \log E_\mathcal{D} + W \log E_\mathcal{W}.$$

It is sensible to assign hidden-layer weights and weights associated with the output unit to different regularization classes so they are regularized separately, having their own adaptively determined scale (MacKay, 1992b; Williams, 1994). This leads to the training criterion used in this study:

$$E = \frac{N}{2} \log E\mathcal{D} + \sum_c W_c \log E_\mathcal{W}^c,$$

where summation is over regularization classes. Here, w_c is the number of weights in class c, and $E_\mathcal{W}^c = \Sigma_{j\in c}|w_j|$ is the sum of absolute values of weights in that class (Williams, 1994).

Choice of data misfit term

While the conventional sum-of-squares misfit term would be appropriate for this study, a data misfit term corresponding to a heteroscedastic (input-dependent variance) Gaussian noise process is adopted, i.e.,

$$E_\mathcal{D} = \sum_{n=1}^{N}\left\{\log \sigma(\mathbf{x}^n) + \frac{[\mu(\mathbf{x}^n) - \mathbf{t}^n]^\mathbf{2}}{\mathbf{2}\sigma^2(\mathbf{x}^n)}\right\}. \tag{7}$$

Using this error function leads to the addition of a second network output unit giving the predicted variance $\sigma(\mathbf{x})$. An exponential acti-

vation function is used for this unit to enforce strictly positive estimates of conditional variance. This approach provides two advantages. First, the estimates of conditional variance provide error bars, indicating the uncertainty of model predictions (Nix and Weigend, 1994, 1995; Williams, 1995). Second, the output of the model now completely specifies the target distribution, so the regularization parameter β is unnecessary.

NETWORK TRAINING AND TEST DATA

Suitable network training data were generated through the 3D forward modeling of magnetic responses for a number of models representing buried wall foundations of a variety of types and burial depths. While training data derived from real-life examples are ideal (i.e., from the subsequent excavation of surveyed areas), the forward-modelled training examples were nonetheless based upon limestone wall features recorded within a limited number of test trench excavations. The modeling was carried out following the method outlined by Sharma (1966) and Bhattacharyya (1978) which considers the magnetization resulting from the magnetic susceptibility distributions of a volume of cuboidal cells. This essentially allows rectangular volumes representing wall structures to be defined within a layered-earth model defined by sampled magnetic susceptibility values.

Six training data set models were created, covering a 10 × 10-m area, with each model incorporating a network of walls at a depth of −0.2 to −0.7 m below the surface in 0.10-m increments and with wall foundations fixed at −1.2 m. Wall thicknesses within each model varied from 1.4–1.7 m. The layout of the training model was designed to simulate a number of building structures, such as building corners, adjoining walls, gaps in walls (entranceways), and palisters. Two wall orientations are featured — north-south and a perpendicular alignment — in accordance with the network test data described below. The problem of representing multiple wall alignments within the training data is addressed later, with the introduction of rotational invariance. Figure 1 shows the basic layout of a training data model and the corresponding forward-modeled magnetic responses at several burial depths. The resulting anomalous magnetic field was calculated over a grid above each model at 0.25-m intervals and the vertical gradient derived, allowing a direct comparison with the survey data (see below).

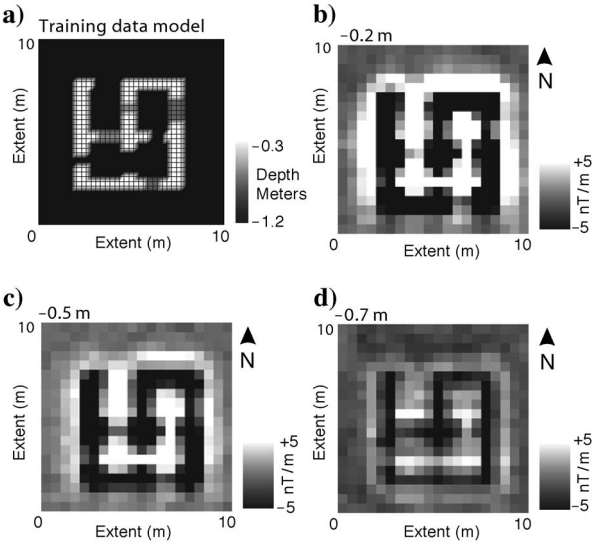

Figure 1. Examples of forward-modelled training data. (a) The training model shows the modelled network of buried limestone walls, including a number of stepped wall heights and entranceways. The resulting forward-modelled responses for model burial depths of (b) 0.2, (c) 0.5, and (d) 0.7 m are also shown.

The layered-earth model used within each of the six depth models is defined as four distinct layers of varying magnetic susceptibility, as recorded within excavated trenches. Samples of 1 cm^3 were collected along a vertical column at 2-cm intervals, and their volume magnetic susceptibility was determined using a Bartington Instruments MS1 susceptibility bridge. Magnetic susceptibility values were found to be uniform across the study area and fell between 18 and 82 × 10^{-8} SI. An infilling layer of building rubble abutting the wall structures was also included within the model, with a magnetic susceptibility value of 200 × 10^{-8} SI in accordance with observed values.

Data input

Data were input into the ANN (during both the training and data processing phases) as a series of vectors, each containing 25 magnetic field values (giving 25 inputs into the ANN). Each vector represented a grid of magnetic values over a 2 × 2-m horizontal area, i.e., at a resolution of 0.5 m. This input window was then parsed sequentially over the area of input data in 0.5-m steps to generate a sequence of input vectors $\mathbf{x^n}$. This input window configuration was found to be an optimal trade-off between input window size, the number of input values (which dictates the complexity of the network), and the horizontal resolution of the input data. During network training, the corresponding target data t^n consisted of a single value representing the depth below the surface to the modelled wall features within the training data models described above, falling below the center of the data input window. The target values were randomized slightly by adding random height values in the range ±0–0.05 m to express the uneven nature of recorded wall structures. For areas of input/target data space with no underlying wall feature present, the target depth was set to an arbitrary −1.2 m (corresponding to the approximate depth of recorded Roman occupation levels). The network was therefore trained to produce a null or default output when no magnetic responses relating to wall sections were detected. This effectively produced a network output from which predicted wall features could be visualized as discrete upstanding structures. This is important in the visual presentation of the results as a 3D subsurface representation, but it also has a more subtle filtering function in which any anomalous magnetic responses unrelated to those represented within the training data are assigned a uniform arbitrary depth. The resulting training set had 2730 input pairs (a magnetic vector and corresponding depth value): 455 data pairs from each of the six training data set depth models. The computational time to generate the training data set through forward modeling was four hours.

The network architecture, having 25 inputs, initially was assigned 16 hidden-layer units, giving a total of 450 weighted connections (i.e., between the inputs, the hidden-layer processing units, and the output processing unit). Using the training set described, we found that the network converged to an error minimum after 4000 iterations, by which time only very negligible decreases in E were seen. The trained network retained 196 live connections, i.e., over half of

the network weights were pruned. The number of units in the hidden layer was also reduced to 15, indicating the initial network complexity was adequate.

Test data

The trained ANN was evaluated initially using forward-modelled synthetic test data, based upon the plan of buried wall elements from an excavated Roman house recorded at Butrint. Figure 2a shows the test data model and the corresponding forward-modelled response. The model covers an area of 20 × 20 m; the principal burial depth of modelled wall sections is −0.3 m. The model also contains features that are not incorporated or represented within the training set, such as thicker walls and discrete pillar sections. This effectively forces the network to make generalized predictions over these features, providing insight into the way gaps in the training data are dealt with.

RESULTS

The predicted results obtained from the trained ANN when applied to the synthetic test data are shown in Figure 2b. The rms error between the network output and the test data set is 0.206 m. One can see that the network fairly accurately predicts wall depths and geometries represented in the test data set. The estimates of the conditional variance $\sigma(\mathbf{x})$ associated with the wall predictions is generally small (Figure 2b), although it is most noticeable in regions where a low density of training data is likely, i.e., for features not explicitly represented within the training set data, such as the pillar-like structures (Figure 2a).

Adding noise to the data

The ANN described has some potentially interesting properties when considering the inversion of noisy data. The required mapping between the magnetic field vector $\mathbf{x}$ and the required wall section depth can be viewed as a deterministic function with added noise, which within the survey data has been observed to closely follow a Gaussian distribution. Because the output of the trained network is essentially an estimate of the true mean of the noisy data distributed around $f(\mathbf{x})$, the network output is given by the conditional average of the target data and can be written as

$$y(\mathbf{x}) = \langle f(\mathbf{x}) + \epsilon | \mathbf{x} \rangle = f(\mathbf{x}).$$

Since $\langle \epsilon \rangle$ is approximately zero, the trained network effectively averages over the noise on the data to reveal the underlying deterministic function (Bishop, 1995, p. 205). This can be demonstrated practically by using the network to process noisy test data.

A randomized sample of noise was taken from the magnetic survey data and added to the synthetic data, following the method outlined by Scollar (1970). Here, the Fourier transform of the noise sample was taken and the modulus and angle were calculated. The angular component was then randomized, effectively retaining the power spectrum but altering the position and shapes of anomalous noise features. The randomized noise sample was then added to the modelled test data. The result is shown in Figure 3a, while the predicted results from the network are shown in Figure 3b. The network still predicts the depths and position of the wall features relatively accurately, showing good overall tolerance to noise. The rms error is 0.38 m. However, a number of very large prediction errors (Figure 3b) reflects the level of divergence from the training data.

To improve noise performance, the network was retrained using a modified training set created by adding a similar level of noise to the training data shown in Figure 1, using the above procedure. The predicted output of the retrained network when applied to the noisy test data is shown in Figure 4a. Predicted wall features are more clearly defined, while many of the smaller noise responses within the test data are ignored by the network. The rms error of 0.271 m indicates that training with noisy data is an effective way of building a high tolerance to noise into the network.

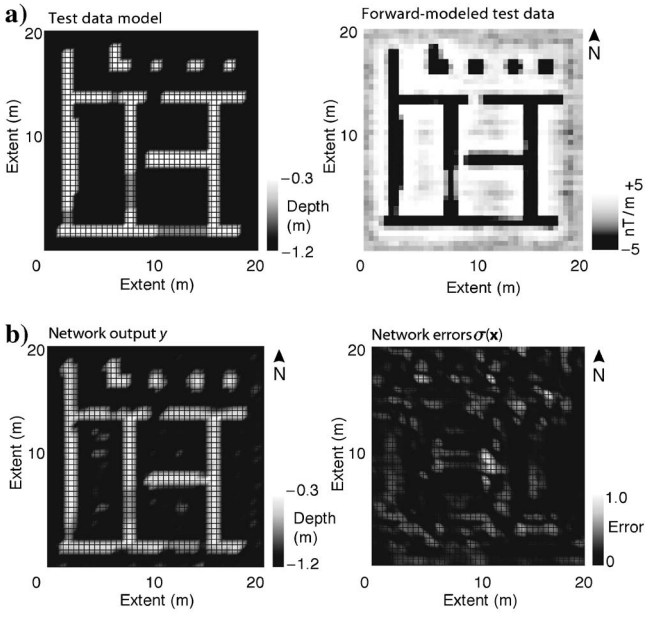

Figure 2. Forward-modelled test data. (a) Plan of the test data model, showing modelled wall foundations and corresponding forward-modelled response. (b) Trained neural network output y and predicted conditional variance or errors $\sigma(\mathbf{x})$.

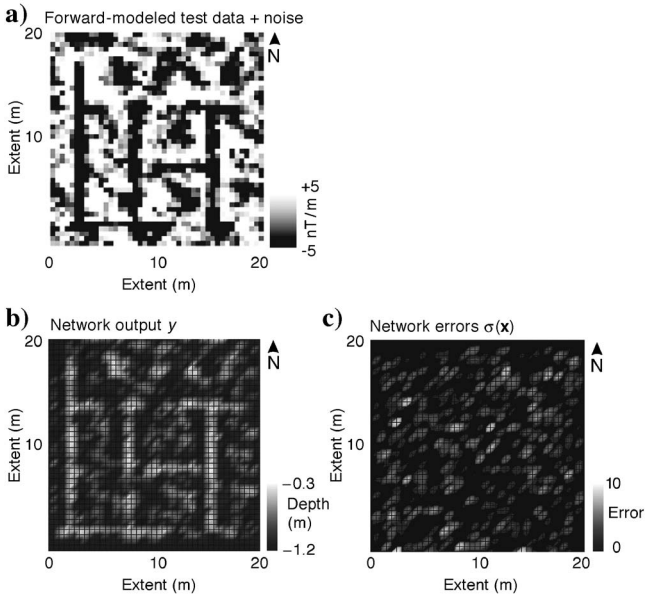

Figure 3. (a) Revised test data with increased level of noise. (b) Network output y and (c) predicted errors $\sigma(\mathbf{x})$ when applied to the revised test data. Note the magnitude of predicted errors.

Enhanced network data processing

The training data shown in Figure 1 represent the magnetic responses of linear wall features in two perpendicular directions only. In practice, wall alignments recorded within archaeological data could occur at any orientation in relation to the survey direction. One method of addressing this problem and also increasing the training routine's efficiency is to introduce rotational invariance into the processed data. This essentially means that the equivalent rotationally invariant form of the magnetic response from a given wall section is the same at any arbitrary orientation. Theoretically, only one orientation of the training data needs to be modelled for all possible orientations to be represented.

Because the magnetic response of buried features changes under rotation as a result of the inclination of the geomagnetic field, a reduction-to-pole operator was first applied to the data, adding overall symmetry to the magnetic anomalies represented (Gunn, 1975). Rotational invariance was then achieved by calculating the corresponding ninth-order Zernike moments of each input window of data, following Prismall et al. (2002) and outlined in Appendix A. Zernike moments are constructed using a set of complex polynomials that form a complete orthogonal basis set invariant to rotation, possessing little information redundancy with a high robustness to noise (Teh and Chin, 1988; Kim and Kim, 2000). Input normalization was applied to the transformed data to encourage the network weights to remain small.

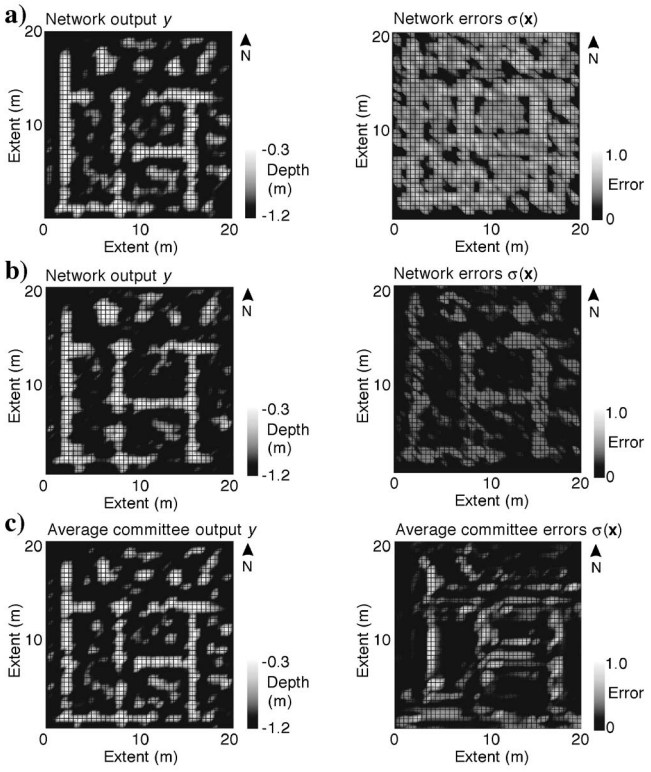

Figure 4. (a) Network output and predicted errors for a network trained using noisy training data. (b) Network output and predicted errors using the Zernike moments preprocessing procedure to introduce rotational invariance into the data. (c) Weighted average committee output and averaged predicted errors for a committee of five networks.

The predicted results from a network trained using rotationally invariant training data (for similarly invariant input data) is shown in Figure 4b. The rms error is 0.268 m, indicating that 20 moments (m = 7) represent the data reasonably accurately. There has been a reduction in the dimensionality of the input data because there are now only 20 network inputs, with no apparent loss in performance. This is because a reduction in dimensionality leads to less sparse population of the input space for a training data set of limited size. Using a higher order of moments and an increased number of network inputs was ineffective.

The implementation of the rotational invariance routine potentially makes a large section of the training data set redundant, since only one orientation of wall now needs to be modeled as opposed to the orthogonal wall sections present in the training data model (Figure 1a). However, because a significant noise element is also being incorporated, there is a high degree of variation between the resulting training data elements. The rotationally invariant training set therefore contains a larger number of noisy examples compared with previous training sets, adding to the overall flexibility of the trained network.

Use of a committee of networks

Further improvements in network prediction can be gained by combining the output from a number of identically trained networks known as a committee, since networks with the same architecture and training set are likely to be trained to different local minima of the error function.

The concept of a committee of networks arises naturally within the Bayesian framework. The posterior distribution of a network's weights can be assumed to have an approximately Gaussian distribution centered on the local minimum of E_D (Bishop, 1995, p. 397). A committee of networks therefore effectively models a set of Gaussians, one centered upon each local minimum (Bishop, 1995, p. 422). The posterior distribution of the weights can be represented as

$$p(w|D) = \sum_i p(m_i, w|D) = \sum_i p(w|m_i, D)P(m_i|D),$$

where m_i denotes one of the nonequivalent minima of the error function and all of its symmetric equivalents (Bishop, 1995, p. 423). This distribution can then be used to determine other quantities by integration over the whole of weight space, so that the mean output predicted by the committee is given by

$$\begin{aligned}\bar{y} &= \int y(\mathbf{x};\mathbf{w})p(\mathbf{w}|D)\,\mathrm{d}\mathbf{w} \\ &= \sum_i P(m_i|D)\int_{\Gamma_i} y(\mathbf{x};\mathbf{w})p(\mathbf{w}|m_i,D)\,d\mathbf{w} \\ &= \sum_i P(m_i|D)\bar{y}_i, \end{aligned} \tag{8}$$

where Γ_i denotes the region of weight space surrounding the ith local minimum and $\bar{y}_i$ is the corresponding network prediction averaged over this region (Bishop, 1995, p. 423). Equation 8 shows that the predicted output is a linear combination of the predictions made by each of the networks corresponding to a distinct local minimum, weighted by the posterior probability of that solution (Bishop, 1995, p. 423). The network committee can therefore be expected to improve generalization because the extension from a single Gaussian

to a Gaussian mixture provides a more accurate model for the posterior distribution of weights (Bishop, 1995, p. 424).

The results from using a committee of five networks is shown in Figure 4c. The rms error for the committee output is 0.231 m, giving a marginal reduction in error. The combination of the outputs from individual networks is achieved by taking the conditional variance $\sigma(\mathbf{x})$ of each prediction and converting them into a set of normalized weight values for each network. The output from the committee is then calculated using a simple weighted average that effectively penalizes individual network predictions with a large associated uncertainty by minimizing their contribution to the overall committee output.

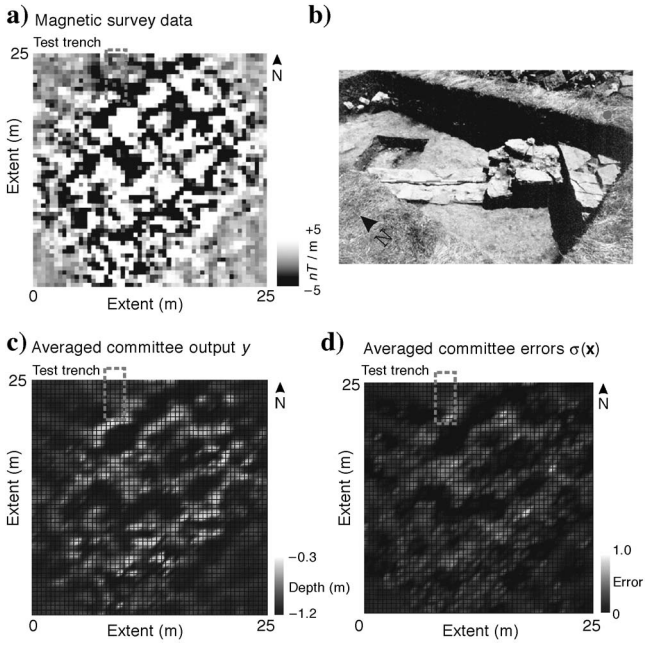

Figure 5. Application of a network committee to real data. (a) Section of magnetic survey data from Butrint over a small building structure. (b) Small test trench excavation associated with building structure. (c) Weighted average committee output and (d) averaged predicted errors for a committee of five networks applied to the magnetic survey data.

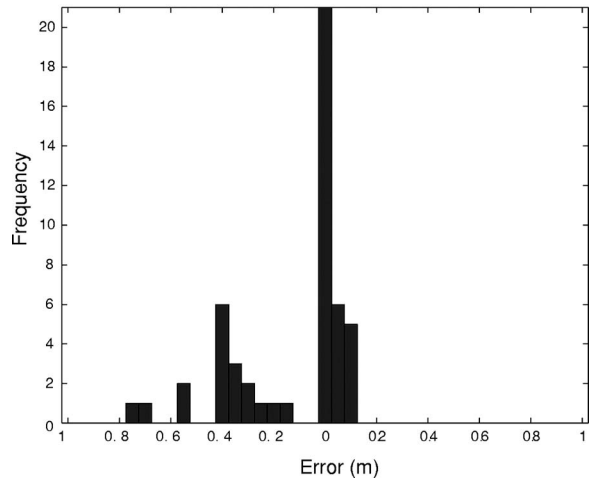

Figure 6. Frequency graph of differences between predicted and recorded wall section depths over the test trench (40 measurements).

Processing magnetic survey data

A committee of networks was used to process magnetic survey data from Butrint. The survey data, derived from a cesium vapor magnetometer in gradiometer mode, revealed a series of former building outlines as linear negative anomalies, often with strongly contrasting positive anomalous responses. Figure 5a shows a section of magnetic survey data (measurement interval, 0.5 m; line spacing, 0.5 m), while Figure 5b shows an associated 5 × 2-m test trench, revealing a surviving section of limestone wall.

The combined output from the committee, when applied to the survey data, is shown in Figure 5c. Over the test trench, the rms error is 0.228 m. The mean average of predicted errors from individual networks (Figure 5c) indicates locations where the predicted errors were highest. A frequency graph showing the residual errors or difference between predicted depth values and recorded wall depths for 40 sample points over the test trench is shown in Figure 6. Note that the committee has a tendency to predict wall foundations at a slightly deeper level than those recorded within the test trench, although it is difficult to draw meaningful conclusions from such a limited amount of ground-truth data.

CONCLUSIONS

The use of ANNs, while utilizing relatively straightforward and well-established neural computing techniques, succeeds in producing useful depth predictions for buried wall features. Importantly, the technique offers a high tolerance to noisy data. While some form of forward modeling is a common element within most inversion techniques, the ANN approach effectively separates the forward-modeling (training) phase from the computation or mapping of the inversion. This allows the practically instantaneous inversion of data by a trained network, where many inversions can be made cost effectively. This speed is related to the fact that no explicit physical model is assumed. However, a significant investment needs to be made in deriving accurate training data and efficient learning procedures, which underpin the successful application of this technique.

The ANN described in our study was designed and trained to process a specific archaeological data set, where forward-modelled responses for buried wall structures at varying depths within a fixed four-layer earth model formed the basis of the training set. A grid of 25 magnetic field strengths formed the network input, while the corresponding target data/network output consisted of a single depth value within the center of the input data grid. The very specific nature of the training data means that the resulting committee of trained networks could only be applied to new magnetic survey data derived from very similar geophysical conditions (including instrument setup) and therefore would not have universal application without being retrained with a newly derived training data set. Within the context of the current project, however, where the processed magnetic survey data represent a small portion of potential data from a large Roman settlement buried beneath a uniform floodplain, future survey data can be inverted cheaply and quickly by the trained networks. In this situation, the ANN approach is far more efficient than applying conventional inversion algorithms, although it would be interesting to compare a single ANN inversion with a conventional algorithm in terms of overall speed and performance. Neural net-

work techniques continue to offer many interesting possibilities for geophysical data processing, particularly in the fields of inversion and signal processing.

ACKNOWLEDGMENTS

The authors acknowledge the Institute of World Archaeology and the Butrint Foundation for help and support throughout this project. The work was funded by the Natural Environment Research Council grant NER/S/A/2000/03321A. A significant level of support was also provided by the Packard Humanities Institute (PHI).

APPENDIX A

ZERNIKE MOMENTS

Complex Zernike moments are constructed using a set of complex polynomials that forms a complete orthogonal basis set defined upon a unit disc in polar coordinates. Here the theory outlined by Prismall et al. (2002) is followed, where the orthogonal moments are given by

$$Z_{mn} = \frac{m+1}{\pi}\int_0^{2\pi}\int_0^1 f(r,\theta)[V_{mn}(r,\theta)]^* dr\, d\theta,$$

where m is the order of the moment ($m = 0,1,2, \ldots, \infty$) and where n is an integer representing the repetition subject to the conditions $m + |n|$ is even and $|n| \le m$. The value $V_{mn}(r,\theta)$ is the complex-valued Zernike polynomial, with * indicating the complex conjugate. For a discrete square image of size $N \times N$, Z_{mn} can be calculated using

$$Z_{mn} = \frac{m+1}{\pi}\frac{1}{(N-1)^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[V_{mn}(r,\theta)]^* f(x,y).$$

To map the image onto a unit disc, Cartesian coordinates were converted to polar coordinates using $r = \sqrt{(x^2+y^2)}/\sqrt{2N}$ and $\theta = \tan^{-1}(y/x)$ so that the center of the input window becomes the origin of the unit disc. The Zernike polynomial $V_{mn}(r,\theta)$ is defined by Prismall et al. (2002) as

$$V_{mn}(r,\theta) = R_{mn}(r)\exp^{-jn\theta},$$

where the radial polynomial $R_{mn}(r)$ is

$$R_{mn}(r) = \sum_{s=0}^{m-|n|/2}\frac{(-1)^s(m-s)!\, r^{m-2s}}{s!\left(\frac{m+|n|}{2}-s\right)!\left(\frac{m-|n|}{2}-s\right)!}.$$

This polynomial is such that over the unit disc, $|R_{mn}(r)| \le 1$ and $R_{mn}(1) = 1$ for any values of m and n. The first six orthogonal radial polynomials are

$$R_{00}(r) = 1, \quad R_{11}(r) = r$$
$$R_{20}(r) = 2r^2 - 1, \quad R_{22}(r) = r^2$$
$$R_{31}(r) = 3r^3 - 2r, \quad R_{33}(r) = r^3, \ldots, \text{etc.}$$

The number of Zernike moments generated, up to and including order m, is $(m/2+1)(m+1)$. The total number of Zernike moments of order $m = 10$ would therefore be 66. However, Prismall et al. (2002) shows that

$$V^*_{mn}(r,\theta) = V_{m,-n}(r,\theta),$$

from which it follows that

$$Z^*_{mn} = Z_{m,-n}.$$

Because of this relationship, only the moments with $n \ge 0$ need to be known, reducing the total number of moments for m from 10 to 36. Theoretically, none of the original information is lost if the number of moments calculated approaches infinity (Teh and Chin, 1988). However, in this application it is necessary to limit the total number of moments to give a realistic number of inputs into the network. For an order of $m = 7$, the total number of moments is 20.

The Zernike moments generated range from approximately 1.3 to 35×10^{-6}. It is therefore necessary to rescale the input variables. This was achieved by taking each set of input data x_i independently and calculating the mean $\bar{x}_i$ and variance σ_i^2 following the method outlined by Bishop (1995, p. 298):

$$\bar{x}_i = \frac{1}{N}\sum_{n=1}^{N} x_i^n$$

$$\sigma_i^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_i^n - \bar{x}_i)^2,$$

where $n = 1,2, \ldots, N$ are the number of patterns in the data set. The data for a particular input were then rescaled using

$$\tilde{x}_i^n = \frac{x_i^n - \bar{x}_i}{\sigma_i}.$$

The transformed input variables given by $\tilde{x}_i^n$ have zero mean and unit standard deviation over the transformed training set (Bishop, 1995, p. 298). The input normalization effectively encourages the network weights to remain small once randomized prior to training.

REFERENCES

Bhattacharyya, B. K., 1978, Computer modelling in gravity and magnetic interpretation: Geophysics, **43**, 912–929.
Bishop, C. M., 1995, Neural networks for pattern recognition: Oxford University Press.
Buntine, W. L., and A. S. Weigend, 1991, Bayesian back-propagation: Complex systems, **5**, 603–643.
Calderón-Macías, C., M. K. Sen, and P. L. Stoffa, 2000, Artificial neural networks for parameter estimation in geophysics: Geophysical Prospecting, **48**, 21–47.
Eder-Hinterleitner, A., W. Neubauer, and P. Melichar, 1996, Reconstruction of archaeological structures using magnetic prospection: Analecta Praehistorica Leidensia, **28**, 131–137.
El-Qady, G., and K. Ushijima, 2001, Inversion of DC resistivity data using neural networks: Geophysical Prospecting, **49**, 417–430.
Fossati, M., A. Zerilli, G. Ronchini, and B. Apolloni, 1992, Lineament analysis for potential field data using neural networks: 62nd Annual International Meeting, SEG, Expanded Abstracts, 6–9.
Geman, S., E. Bienenstock, and R. Doursat, 1992, Neural networks and the bias/variance dilemma: Neural Computation, **4**, no. 1, 1–58.
Gradshteyn, I. S., and I. M. Ryzhic, 1994, Table of integrals, series and products, 5th ed.: Academic Press.
Gunn, P. J., 1975, Linear transforms of gravity and magnetic fields: Geophysical Prospection, **23**, 300–321.
Guo, Y., R. Hansen, and N. Harthill, 1992, Feature recognition from potential fields using neural networks: 62nd Annual International Meeting, SEG,

Expanded Abstracts, 1–5.
Haykin, S., 1994, Neural networks: A comprehensive foundation: Macmillan Publ. Co.
Helle, H. B., A. Bhatt, and B. Ursin, 2001, Porosity and permeability prediction from wireline logs using artificial neural networks: A North Sea case study: Geophysical Prospecting, **49**, 431–444.
Herwanger, J., H. Maurer, A. G. Green, and J. Leckebusch, 2000, 3-D inversion of magnetic gradiometer data in archaeological prospecting: Possibilities and limitations: Geophysics, **65**, 849–860.
Jain, L. C., and N. M. Martin, 1999, Fusion of neural networks, fuzzy sets and genetic algorithms: Industrial applications: CRC Press.
Jang, J. S. R., C. T. Sun, and E. Mizutani, 1997, Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence: Prentice-Hall, Inc.
Jeffreys, H. S., 1939, Theory of probability: Oxford University Press.
Kecman, V., 2001, Learning and soft computing: MIT Press.
Kim, W. Y., and Y. S. Kim, 2000, A region-based shape descriptor using Zernike moments: Signal Processing-Image Communication, **16**, no. 1, 95–102.
Mackay, D. J. C., 1992a, A practical Bayesian framework for backprop networks: Neural Computation, **4**, 448–472.
———, 1992b, A practical Bayesian framework for backprop networks: Neural Computation, **4**, 448–472.
Neal, R. M., 1996, Bayesian learning for neural networks: Springer-Verlag New York, Inc.
Nix, D. A., and A. S. Weigend, 1994, Estimating the mean and variance of the target probability distributions: International Conference on Neural Networks, Proceedings, 55–60.
———, 1995, Learning local error bars for nonlinear regression, *in* G. Tesauro, D. Touretzky, and T. Leen, eds., Advances in neural information processing systems 7: MIT Press, 489–496.
Poulton, M. M., 2001, Computational neural networks for geophysical data processing: Pergamon Press, Inc.
———, 2002, Neural networks as an intelligence amplification tool: A review of applications: Geophysics, **67**, 979–993.
Poulton, M. M., B. K. Sternberg, and C. E. Glass, 1992a, Location of subsurface targets in geophysical data using neural networks: Geophysics, **57**, 1534–1544.
———, 1992b, Neural network pattern recognition of subsurface EM images: Journal of Applied Geophysics, **29**, 21–36.
Prismall, S. P., M. S. Nixon, and J. N. Carter, 2002, On moving object reconstruction by moments: 13th Annual Conference, British Machine Vision, Proceedings, 73–82.
Raiche, A., 1991, A pattern recognition approach to geophysical inversion using neural nets: Geophysical Journal International, **105**, 629–648.
Rumelhart, D. E., R. Durbin, R. Golden, and Y. Chauvin, 1995, Backpropagation: The basic theory, *in* Y. Chauvin and D. E. Rumelhart, eds., Backpropagation: Theory, architectures and applications: Lawrence Erlbaum, 1–34.
Scollar, I. G., 1970, Fourier transform methods for the evaluation of magnetic maps: Prospezioni Archeologiche, **5**, 9–41.
Sharma, P. V., 1966, Rapid computation of magnetic anomalies and demagnetization effects caused by bodies or arbitrary shape: Pure and Applied Geophysics, **64**, 89–109.
Spichak, V., and I. Popova, 2000, Artificial neural network inversion of magnetotelluric data in terms of three-dimensional earth macroparameters: Geophysical Journal International, **142**, 15–26.
Teh, C.-H., and R. T. Chin, 1988, On image analysis by methods of moments: IEEE Transactions on Pattern Analysis and Machine Intelligence, **10**, 496–513.
Tikhonov, A. N., and V. Y. Arsenin, 1977, Solutions of ill-posed problems: John Wiley & Sons, Inc.
Ucan, O. N., E. Bilgili, and A. M. Albora, 2002, Magnetic anomaly separation using genetic cellular neural networks: Journal of the Balkan Geophysical Society, **5**, no. 3, 65–70.
Van der Baan, M., and C. Jutten, 2000, Neural networks in geophysical applications: Geophysics, **65**, 1032–1047.
Williams, P. M., 1991, A Marquardt algorithm for choosing the step-size in backpropagation learning with conjugate gradients: School of Cognitive and Computing Science Research Paper 229, University of Sussex.
———, 1993, Aeromagnetic compensation using neural networks: Neural Computing and Applications, **1**, 207–214.
———, 1994, Bayesian regularisation and pruning using a Laplace prior: School of Cognitive and Computing Science Research Paper 312, University of Sussex.
———, 1995, Using neural networks to model conditional multivariate densities: School of Cognitive and Computing Science Research Paper 371, University of Sussex.
Yarger, H. L., R. R. Robertson, and R. L. Wentland, 1978, Diurnal drift removal from aeromagnetic data using least squares: Geophysics, **46**, 1148–1156.