Natural Resources Research, Vol. 14, No. 2, June 2005 (© 2005)
DOI: 10.1007/s11053-005-6955-z

Mapping Mineralization Probabilities using Multilayer Perceptrons

Andrew A. Skabar[1]

Received 21 October 2004; accepted 29 March 2005

Mineral-potential mapping is the process of combining a set of input maps, each representing a distinct geo-scientific variable, to produce a single map which ranks areas according to their potential to host mineral deposits of a particular type. The maps are combined using a mapping function that must be either provided by an expert (knowledge-driven approach), or induced from sample data (data-driven approach). Current data-driven approaches using multilayer perceptrons (MLPs) to represent the mapping function have several inherent problems: they are highly sensitive to the selection of training data; they do not utilize the contextual information provided by nondeposit data; and there is no objective interpretation of the values output by the MLP. This paper presents a new approach by which MLPs can be trained to output values that can be interpreted strictly as representing posterior probabilities. Other advantages of the approach are that it utilizes all data in the construction of the model, and thus eliminates any dependence on a particular selection of training data. The technique is applied to mapping gold mineralization potential in the Castlemaine region of Victoria, Australia, and results are compared with a method based on estimating probability density functions.

KEY WORDS: Mineral exploration, mineral-potential mapping, neural networks.

INTRODUCTION

Mineral-potential mapping can be seen as a process whereby a set of input maps, each representing a distinct geo-scientific variable, are combined to produce a single map which ranks areas according to their potential to host deposits of a particular type. Although the traditional approach is to derive the mapping function on the basis of expert knowledge of mineral causative factors, data driven approaches attempt to discover, or *learn*, the function by measuring in some way the association between mapped predictor variables and a response map that indicates the locations of known occurrences of the sought-after mineral (Bonham-Carter, 1994). The signatures discovered for these known deposits can then be used to highlight other regions of high mineral potential.

[1] Department of Computer Science and Computer Engineering, La Trobe University, Victoria 3086, Australia; e-mail: a.skabar@latrobe.edu.au.

More formally, the data-driven mapping problem can be expressed as follows:

Given:

(1) Background information provided by m layers of data, each of which represents the value of a distinct geoscientific variable x_i at each pixel p;
(2) A subset of pixels, each of which is known from historical data to contain one or more deposits of the sought after mineral;

Find:

A function $f(\mathbf{x})$ that assigns to each pixel p in the study area a value that represents the favorability that pixel p contains one or more of the known deposits, given the evidence supplied by the background information.

The meaning of *favorability* in this problem definition is ambiguous and can refer to any qualitative or

quantitative measure that describes in some general way the likelihood that some area will contain a mineral deposit. Measures that have been used to describe favorability include the *probability, possibility, certainty, belief*, or *plausibility* that a deposit occurs in some given area (Bonham-Carter, 1994). In this paper, the *favorability* is interpreted as a *probability*. Thus, assuming that the evidence for a pixel p is described by a vector $\mathbf{x} = (x_1, \ldots, x_m)$, the objective is to learn a function $f: \mathbf{X} \rightarrow [0, 1]$, where $f(\mathbf{x})$ represents the posterior probability (i.e., conditional probability) that p contains one or more of the known deposits, given the evidence provided by $\mathbf{x}$. Once the function $f(\mathbf{x})$ has been learned, the mineral-potential map can be produced by mapping $f(\mathbf{x})$ for each pixel in the study area. The process is depicted in Figure 1.

There are two general approaches to discovering such a mapping function: (i) density estimation approaches based on estimating probability density functions (pdfs), and (ii) function optimization approaches.

Density Estimation Based Approach

Density estimation based approaches involve estimating the class-conditional pdf, $p(\mathbf{x}|D)$ (i.e., the pdf for pixels that contain at least one of the known deposits), and the class-unconditional pdf, $p(\mathbf{x})$ (i.e., the pdf for all pixels). The densities $p(\boldsymbol{x}|D)$ and $p(\mathbf{x})$ then can be combined with $p(D)$ (i.e., the probability that a randomly selected pixel, in the absence of any evidence for that pixel, contains a known deposit) using

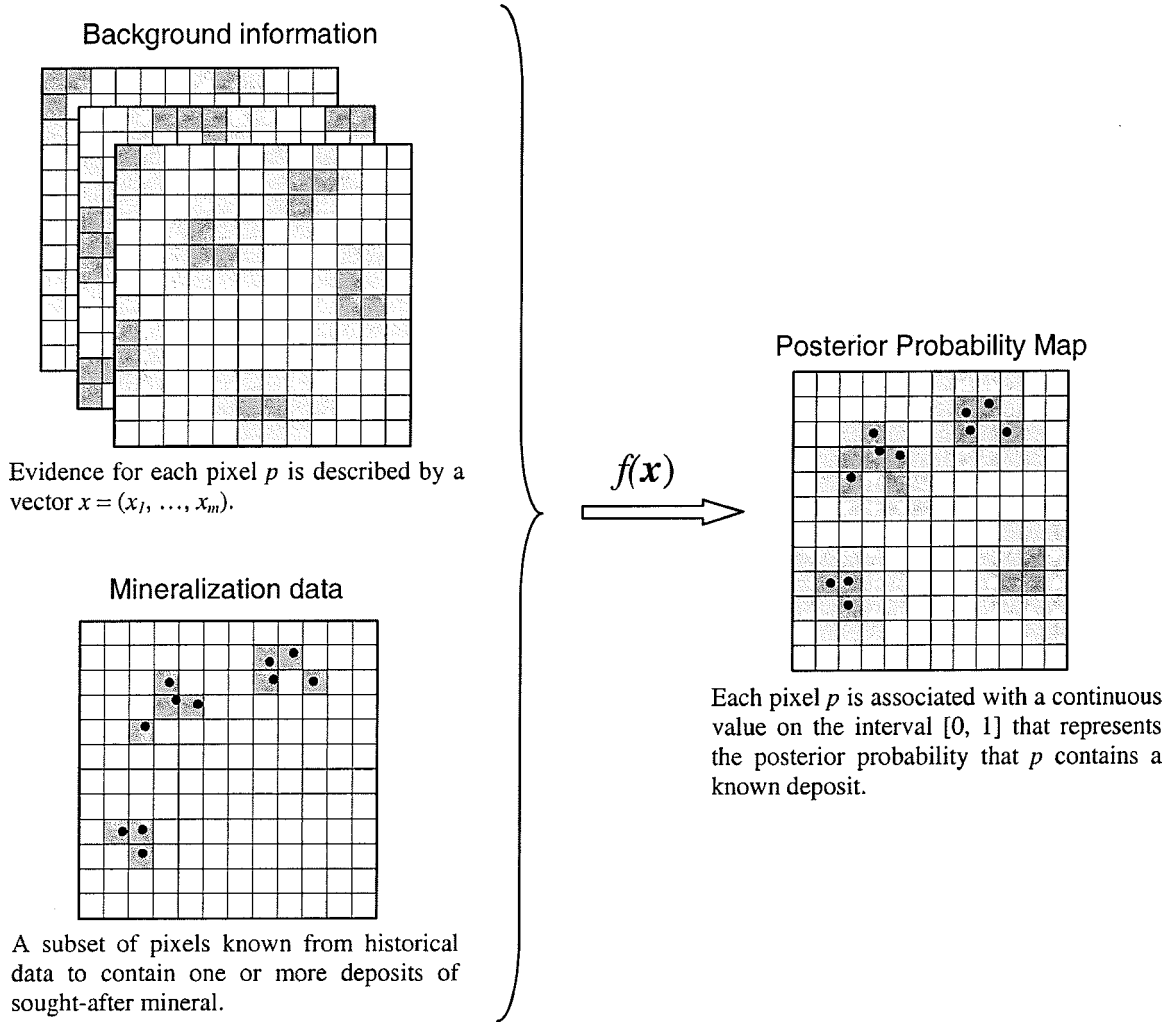

Figure 1. The mineral-potential mapping process. Function $f(\mathbf{x})$ assigns to each pixel a value indicating posterior probability that pixel contains one or more of known target deposits.

Bayes' Theorem,

$$P(D|\mathbf{x}) = \frac{p(\mathbf{x}|D) \times P(D)}{p(\mathbf{x})}, \quad (1)$$

to obtain $P(D|\mathbf{x})$ (i.e., the probability that a cell is mineralized given the feature vector describing that cell).

The simplest approach to estimating pdfs is to assume the form of the distribution (e.g., Gaussian) and to estimate the values of the parameters for that distribution (e.g., mean and covariance in the situation of a Gaussian distribution); however, many data sets do not follow a Gaussian distribution, and attempts to model them in this way will lead to poor estimates of the density functions. A second approach—the *kernel*, or *Parzen*, method—is a nonparametric approach that involves modeling the distribution using a series of probability windows (usually Gaussian) centered at each sample (Parzen, 1962). The overall pdf is the average of all of the individual distributions centered at each point, and the main decision to be made is the choice of the smoothing parameter σ, which defines the width of the windowing function. A third approach, which can be seen as lying somewhere between the two methods, is the *Mixture of Gaussians* approach (Titterington, Smith, and Makov, 1985). In this situation K Gaussian distributions are used to model the data, where K is smaller than the number of sample points. The problem in this situations is to determine the means, covariances, and priors for these K distributions. One method for determining these parameters is the Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977).

Function Optimization Approach

The second general approach to discovering a mapping function involves estimating a function which directly provides a mapping from the input space to a probability. This paper is concerned with functions of the following form, referred to as *multilayer perceptrons* (MLPs):

$$f(\mathbf{x}^n) = h(u) \text{ where } u = \sum_{j=0}^{N_1} w_{kj}\, g\left(\sum_{i=0}^{N_0} w_{ji} x_i^n\right) \quad (2)$$

where N_0 is the number of inputs (i.e., the dimensionality of the input feature vector), N_1 is the number of units in a hidden layer, w_{ji} is a numerical weight connecting input unit i with hidden unit j, w_{kj} is the weight connecting hidden unit j with output unit k, $h(u) = \sigma(u) \equiv (1 + \exp(-u))^{-1}$ (i.e., a *logistic* function), and g(u) is either a logistic function, or some other continuous, differentiable, nonlinear function. MLPs are capable of representing highly nonlinear relationships to an arbitrary degree of accuracy (Cybenko, 1989; Hornick, Stinchcombe, and White, 1989), and the issues in using this approach are selecting a suitable number of hidden layer units, and optimizing the weights.

The generic approach in applying MLPs to most classification and regression problems is to select a set of training examples, and to iteratively adjust the parameters (i.e., weights) of the model such that the overall error between network output and target output is decreased after each iteration. This is referred to as *error back-propagation training* (Rumelhart and McClelland, 1986), and the default error reduction function used in most approaches is the sum-of-squared (i.e., quadratic) error. However, there are several significant problems that arise when applying MLPs to mineral potential mapping tasks. These include a high degree of sensitivity to the selection of training data; dimensionality problems arising from training use a small number of training examples in high dimensional input spaces; nonutilization of contextual information provided by nondeposit data; and difficulty in giving a physical interpretation to the output values. This paper contributes a new method for network training which avoids these problems.

The paper is structured as follows. A critique is provided first of current approaches to applying neural networks to mineral-potential mapping tasks. The critique highlights important inherent problems in these approaches. The issue of training neural networks to represent the probability of mineralization then is addressed: the learning task is specified formally, and maximum likelihood considerations are used to derive an error reduction function appropriate to this task. Empirical results then are provided. These results compare the performance of the proposed technique with an alternative approach, which is based on estimating probability density functions. A detailed description also is provided of a special cross-validation procedure used to optimize the respective model-specific parameters in each situation. The advantages that the proposed approach has to current approaches are discussed.

CRITIQUE OF CURRENT APPROACHES

Seminal papers which have addressed the use of neural networks in mineral exploration include Singer and Kouda (1996), which reports on the application of

neural networks to estimating distance to ore in the Hokoruko District of Japan, and Singer and Kouda (1997), which describes the use of *Probabilistic Neural Networks* to classify deposits into types. However, neither of these works addresses the mineral-potential mapping task as it has been defined in this paper: Singer and Kouda (1997) do not produce an output map, and the output value assigned to pixels in Singer and Kouda (1996) represents distance to ore, and not a value for mineralization favorability per se.

The main interest in this paper is to examine the capability of MLPs to model accurately posterior probabilities on mineral-potential mapping tasks, and the most directly relevant papers in this area are by Brown and others (2000) and Porwal, Carranza, and Hale (2003), who apply MLPs and radial basis functions respectively. Also of relevance are Harris and Pan (1999), Singer and Kouda (1999), and Harris and others (2003), who apply probabilistic neural networks to the prediction of mineralization potential. The remainder of the section outlines the method used by Brown and others (2000) and Porwal, Carranza, and Hale (2003), and highlights some significant problems inherent in their approach (which will henceforth be referred to as the *conventional* approach). Probabilistic neural networks are discussed later.

The general approach used by both Brown and others (2000) and Porwal, Carranza, and Hale (2003) can be summarized as follows:

(1) Represent each pixel in the study area as an input feature vector.
(2) For each feature vector, assign a binary target value to indicate the presence or absence of a known deposit in that pixel. (For convenience we assume a target of 1 for deposit cells, and a 0 for nondeposit cells).
(3) Divide the feature vectors that have a target label of 1 randomly into three sets—a training set, a validation set, and a test set—each of which contains an approximately equal number of examples.
(4) Select nondeposit cells and place these in the training, validation, and test sets such that the ratio of mineralized to nonmineralized cells in each set is approximately 1:1.
(5) Train the network using a gradient-descent algorithm that minimizes the sum-of-squared error on the training examples, monitoring the error on the validation set. Stop training when the network begins to overfit the training data, which is indicated by a decline in performance on validation data.
(6) Apply the trained network to each pixel in the study region and map the results by choosing suitable thresholds to define favorability classes.

There are several problems inherent in this approach:

(1) *Use of sum-of-squared error*. Both Brown and others (2000) and Porwal, Carranza, and Hale (2003) use binary target values to represent the presence or absence of a (known) deposit. The use of sum-of-squared error in back-propagation training is based on the statistical assumption that noise in the training data is distributed with zero mean and constant variance around the target function (Bishop, 1995). Although this assumption is appropriate on most regression tasks (i.e., function approximation tasks in which the target outputs are continuous), it is not always appropriate when the target values are binary, especially when there is a gross imbalance in the number of training examples between classes.
(2) *Dimensionality problems resulting from sparsely populated input space*. Mineralization is a rare event, and it can be assumed that the proportion of cells containing known deposits of the sought after mineral is small. Dividing the mineralized cells among training, validation and test sets means that the number of mineralized examples used for training will be small indeed. In high dimensional input spaces this will introduce dimensionality problems, with the resulting networks displaying high variance; that is, the function represented by the trained network will be heavily dependent on factors such as the initial weights, and the maps resulting from different random restarts will display significant variation.
(3) *Identification of nondeposit training cells*. Whereas cells containing a known deposit are undoubtedly mineralized, cells that do not contain a known deposit may or may not be mineralized. This is a ground truth problem, and generally we cannot assume that the absence of a known deposit indicates that a cell is barren. Because a small number of nondeposit training examples must be

selected from the large corpus of nondeposit cells, and because the presence of misclassified cells in the training set can present the model with contradictory or ambiguous information (Harris and others, 2003), the resulting map can be highly sensitive to the particular choice of nondeposit training examples. Porwal, Carranza, and Hale (2003) address this problem by selecting nondeposit examples randomly from those in which the probability of containing a deposit is small, as determined by using maps modeled previously using a weights-of-evidence analysis. Brown and others (2000) select nondeposit cells randomly from each of 12 main rock units. Either way, the resulting map will depend to some degree on the particular set of nondeposit examples used for training.

(4) *Rapid convergence*. The small training set also indicates that network training will converge rapidly. Consequently, it will be difficult to stop training before overfitting begins to occur.

(5) *Interpretation of network outputs*. There is no standard objective interpretation which can be assigned to the values output by an MLP trained using the method as described. It cannot be assumed, for example, that the outputs represent probabilities.

(6) *Nonoptimal use of available data*. There is a convention in most applications of MLPs (and most areas of supervised machine learning for that matter) to test the generalization performance of a network by applying the (trained) network to novel examples; that is, examples to which the network had not been exposed during training. In most situations this is a reasonable approach because the class value of the training examples is known already, and we are interested primarily in the ability of the network to correctly predict the class of new examples. However, in the situation of mineral-potential mapping, we are trying to predict the likelihood of mineralization, and we do not know in advance what this likelihood is for any of the pixels in the study region. Although we may know that a particular pixel is mineralized, mineralization is the realization of a stochastic process, and we do not know with what *probability* that pixel was mineralized. Thus, the objective is to estimate the value of this probability for all pixels, including those already known to be mineralized. This suggests the following question: Why should we not use all of the available pixels for training? The answer suggested by both Brown and others (2000) and Porwal, Carranza, and Hale (2003) is that the gross imbalance between deposit cells and nondeposit cells will result in poor recognition of deposits. Brown and others (2000) make this explicit:

> "If deposit patterns were represented in the training set in the same proportion as they appear in the total data population, the learning algorithm would optimise the performance for non-deposit patterns ... (it) would not learn to recognise the rare deposit patterns at all or would perform very poorly for this class of patterns" (Brown and others, 2000).

The problem that Brown and others (2000) identify stems from the fact that sum-of-squared error reduction is being used on data with binary-valued target outputs. As will be shown in the next section, sum-of-squared error is not the best choice in this context, and, through an alternative choice of error reduction function, it is possible indeed to use all examples for training, despite the gross imbalance between deposit and nondeposit examples. Moreover, the function represented by a network trained using the proposed method can be shown to represent the posterior probability of mineralization. Not only does the proposed method provide a standard objective interpretation for the network outputs, but it also solves the problem of identifying nondeposit cells, and significantly diminishes problems arising from high dimensionality.

REPRESENTING POSTERIOR PROBABILITIES USING MLPS

This section describes a method by which MLPs can be trained to represent the posterior probability that a cell contains one or more of the known deposits in the study area. Although it may at first sight seem odd to be predicting the probability of containing a known deposit (considering that the ultimate aim is to discover *new* deposits, and not simply to determine the probability for *existing* deposits), it

is completely consistent with the assumptions which are always (at least implicitly) made in data-driven mineral-potential mapping tasks; that is, that the known mineral-deposit occurrences constitute an adequate and unbiased sample of the true deposits in the region, and that by discovering signatures for these known deposits, other regions of high mineral potential also will be highlighted. A formal specification of the task is first provided. Maximum likelihood considerations then are used to derive an appropriate error reduction function for back-propagation training. The resulting error reduction function is then compared analytically with sum-of-squared error in the context of application to mineral-potential mapping tasks.

Specification of Task

Assume the existence of a binary function $g(\mathbf{x})$ that represents the presence or absence of a known deposit in a pixel p with feature vector $\mathbf{x} = (x_1, x_2, \ldots, v_m)$, where x_k is the value of the kth variable for pixel p, and m is the number of input variables. Thus, if pixel p contains a known occurrence, then $g(\mathbf{x}) = 1$, otherwise $g(\mathbf{x}) = 0$. Now let $f(\mathbf{x})$ be a probabilistic function whose output is the probability that $g(\mathbf{x}) = 1$. The objective is to learn the function $f : \boldsymbol{X} \rightarrow [0, 1]$, such that $f(\mathbf{x}) = P(g(\mathbf{x}) = 1)$. Thus, pixels containing one or more known deposits are assigned a target value of one, and all other pixels in the study area are assigned a target value of zero.

The function $f(\mathbf{x})$ can be represented by an MLP. Because the network is required to produce only a single value for each input example, only one output unit is required. Because the output at this unit is to represent a probability, the output of the network should be bounded between 0 and 1, and this can be arranged by using a logistic activation function on the output node. Thus, the structure of the perceptron is exactly that which has been described in Equation (2).

Network Training

The network should be trained such that it represents the function which results in the highest probability of observing the given data. This function, $f_{ML}(\mathbf{x})$, is turned the *maximum likelihood function* or *maximum likelihood hypothesis* (Duda and Hart, 1973). Suppose that an example $\mathbf{x}^n$ with target value t^n is drawn randomly from the training set, and that t^n has a value of 1 if $\mathbf{x}^n$ contains a known deposit, and 0 otherwise. By definition of $f(\mathbf{x})$, the probability that t^n equals 1 is $f(\mathbf{x}^n)$, and the probability that t^n equals 0 is $1 - f(\mathbf{x}^n)$. The probability of observing the correct target value, given $f(\mathrm{x})$, therefore can be expressed as

$$P(t^n | f, \mathbf{x}^n) = f(\mathbf{x}^n)^{t_i}(1 - f(\mathbf{x}^n))^{1-t_i} \quad (3)$$

where $\mathbf{x}^n = (x_1^n, x_2^n, \ldots, x_m^n)$ is the feature vector for pixel p^n, $f(\mathbf{x}^n)$ is the value of the function f applied to vector $\mathbf{x}^n$, and $t^n = 1$ if pixel p^n contains a known deposit, and 0 otherwise. Assuming that the examples are independent and identically distributed (i.i.d.), the probability of observing the correct target value for all examples is given by

$$\prod_{n=1}^{\mathrm{N}} \left\{ f(\mathbf{x}^n)^{t^n}(1 - f(\mathbf{x}^n))^{1-t^n} \right\} \quad (4)$$

where N is the number of examples. The maximum likelihood function, f_{ML}, is the function for which the given expression is a maximum:

$$f_{ML}(\mathbf{x}) = \underset{f}{\operatorname{argmax}} \prod_{n=1}^{\mathrm{N}} \left\{ f(\mathbf{x}^n)^{t^n}(1 - f(\mathbf{x}^n))^{1-t^n} \right\} \quad (5)$$

Taking the logarithm of the expression in braces, which is justified because $\ln(f_{ML})$ is a monotonic function of $f_{ML,}$ and converting the maximization to a minimization by multiplying by -1, the maximum likelihood function is the function for which the following error is minimized:

$$E = -\sum_{n=1}^{\mathrm{N}} \{t^n \ln f(\mathbf{x}^n) + (1 - t^n) \ln(1 - f(\mathbf{x}^n))\} \quad (6)$$

This error function usually is referred to as *cross-entropy* (Hopfield, 1987; Baum and Wilczek, 1988). Alternatively, it can be expressed as

$$E = -\sum_{n=1}^{\mathrm{N}} \ln((1 - |t^n - f(\mathbf{x}^n)|) \quad (7)$$

which makes the interpretation of distance between t^n and $f(\mathbf{x}^n)$ more intuitive (Schumacher, Rossner, and Vach, 1996). Therefore, the maximum likelihood function is the function for which the cross entropy error, and not the sum-of-squared error, is a minimum.

Analytic Comparison Of Cross-Entropy And Sum-of-Squared Error Reduction

In order to appreciate the difference between use of sum-of-squared error and cross-entropy error in the context of mineral potential mapping it is useful to consider the contribution that the error on a single training example makes to the overall error. If

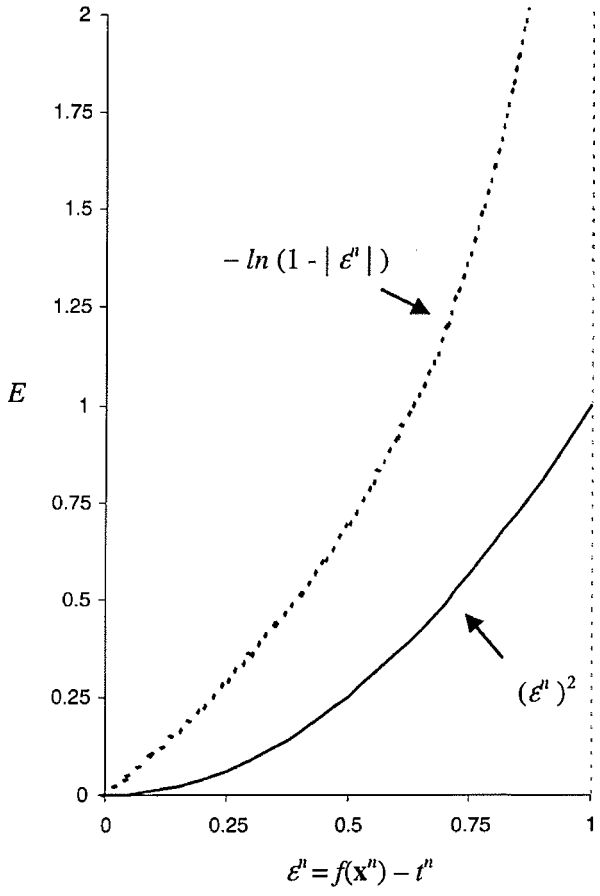

Figure 2. Contribution of individual pattern error ε to total cross-entropy error and total sum-of-squares error.

the network output $f(\mathbf{x}^n)$ for a particular pattern $\mathbf{x}^n$ is expressed as $f(\mathbf{x}^n) = t^n + \varepsilon^n$, then the overall cross-entropy error function, which will henceforth be denoted as E_{CE}, can expressed as

$$E_{CE} = \sum_{n=1}^{N} -\ln(1 - |\varepsilon^n|) \tag{8}$$

and the overall sum-of-squared error function, E_{SS}, as

$$E_{SS} = \sum_{n=1}^{N} ((\varepsilon^n)^2) \tag{9}$$

Figure 2 shows how the contribution of the error ε^n on a single pattern contributes to E_{SS} and E_{CE}. There are several points to note from Figure 2. Firstly, as the absolute value of the error on a single training example approaches 1, the contribution of the cross-entropy error resulting from this example approaches ∞; in contrast, the sum-of-squared error is bounded by a value of 1. Thus if the error of the MLP on one or more examples is maximal (i.e., if the MLP assigns a 1 to an example with a target value of 0, or assigns a 0 to an example with a target value of 1), then the cross-entropy function will assign an infinite error to that hypothesis. This is consistent with our expectations of a function that represents posterior probabilities: it clearly would be contradictory for a hypothesis to assign a probability of 1 to a pattern with target output 0, because if the probability of mineralization is 1, then, by definition, the cell must contain a deposit. Conversely, it would be contradictory for a hypothesis to assign a probability of 0 to an example with a target value of 1, because by definition, if the probability is 0, then the cell cannot contain a deposit. This reasoning cannot be applied to sum-of-squared error reduction, because in this example the maximal error contribution that a single example can make is finite.

A second important difference between the two error functions concerns the relative contributions made by large and small pattern errors, and it can be shown that cross-entropy is more sensitive to small individual pattern errors than is sum-of-squared error. For example, suppose that the two points $(\mathbf{x}^1, 1)$ and $(\mathbf{x}^1, 0)$ each occur in the training set. That is, the two examples have the same feature vector, but different target labels. Suppose further that $f(\mathbf{x}^1)$ is 0.95. Consider E_{CE} first. The contribution to this error function by the first example is $-ln(1 - 0.05) = 0.0513$, and the contribution due to the second example is $-ln(1 - 0.95) = 2.9957$. Now consider E_{SS}. The contribution resulting from the first example is $0.05^2 = 0.0025$, and the contribution from the second example is $0.95^2 = 0.9025$. As expected, the error contribution from the first example is less than the contribution from the second example for both E_{CE} and E_{SS}. But consider now the contribution of the first example *relative to* the contribution from the second. In the situation of E_{CE}, the value of this ratio is $0.0171 (= 0.0513/2.9957)$. In the situation of E_{SS} the value is $0.0028 (= 0.0025/0.9025)$. This indicates that the cross-entropy error function is far more sensitive to small individual errors than is the sum-of-squared error function, and consequently, that cross-entropy is better at estimating small probabilities. In the context of mineral-potential mapping, this is important because mineralization is a rare event, and therefore the expected probabilities will be small.

As was noted in the previous section, both Brown and others (2000) and Porwal, Carranza, and Hale (2003) claim that the deposit cells and nondeposit cells should be represented approximately equally in the training set. It now can be seen that this requirement

arises because of their use of sum-of-squared error; in particular, it results from the fact that the sum-of-squared error contribution for individual examples is bounded. By using all examples for training, and by using cross-entropy and not sum-of-squared error reduction, it is no longer necessary that the numbers of deposit and nondeposit training examples be balanced.

EXPERIMENTAL PROCEDURE AND RESULTS

This section describes the application of the proposed approach to the production of a mineral-potential map showing the favorability for reef gold deposits over a region in the vicinity of the Castlemaine district, Victoria, Australia. In order to test the hypothesis that that the outputs of the MLP represent posterior probabilities, the map produced using the MLP is compared with a map produced using a density estimation-based approach. A description of the study region is first provided. This is followed by a discussion of how model-specific parameters can be optimized for both models, and includes a description of a special cross-validation procedure. Empirical results are then presented.

The Castlemaine Study Area

Castlemaine is located in the southeastern region of Victoria, Australia, and was the site of extensive gold mining in the 19th Century. Almost half of the gold located in Victoria occurred in primary deposits, particularly quartz veins or reefs, in which it was deposited in cracks that opened up during the faulting and folding of Paleozoic sandstone and mudstone beds between 440 and 360 million years ago. The remainder occurs in secondary (alluvial) deposits in soil and creek beds.

The study region used in this report is in the vicinity of Castlemaine, and extends from a Northwest corner with coordinates 251,250 mE, 5,895,250 mN, to a Southeast corner with coordinates 258,250 mE, 5,885,000 mN (all specified coordinates are Northings/Eastings, referenced according to AMG Zone 55 AGD 66). Based on a grid-cell resolution of 50 m by 50 m, the study region was represented by a rectangular grid consisting of 141 cells in the horizontal direction and 206 cells vertically. In total, 16 input layers were used. These included three layers based on magnetics (magnetic field intensity, first derivative of magnetic field intensity, and automatic gain control filtered magnetics); five layers based on radiometrics (Th, U, K, TotalCount, K/Th); seven based on geochemistry (Au, As, Cu, Mo, Pb, W, Zn), and distance to closest fault. The number of documented known reef gold deposits in the study area is 148. Full details on data preprocessing, interpolation of point-based data, etc. is given in Skabar (2000). Information on Victorian geology is in Cochrane, Quick, and Spencer-Jones (1995) and Clark and Cook (1988). The Castlemaine Goldfield was described in Willman (1995).

Parameter Selection and Cross-Validation Procedure

The performance of both MLP and density estimation based approaches depends heavily on selecting suitable values for model specific parameters: the number of hidden layer units for the MLP approach, and the width of the Gaussian window for the Parzen method. For the MLP approach, a procedure is required for determining when to stop training. The value of these parameters should be selected such that the generalization capability of the model is maximized.

A general approach to determining optimal parameter values is to use *cross-validation*. The usual cross-validation technique applied on standard classification tasks involves dividing the training examples into n approximately equal sized groups. A classifier is trained using examples from all but one of these groups, and performance is monitored on the examples from the remaining group (i.e., the holdout set). This procedure then is repeated n-1 times, in each setuation with a different combination of groups used for training. Results on all holdout sets are then combined, providing an overall measure of the generalizational capability of the model.

The method that has been proposed requires that all examples be used for training, and consequently, the standard cross-validation procedure is not applicable. A modified cross-validation procedure can be described as follows, where for simplicity, 4-fold cross-validation is assumed:

(1) Replicate the entire data set four times.
(2) For each of the four replicates, select one quarter of the positive examples (i.e., 1/4 of the examples whose target output value is 1) and change the target value of these examples from 1 to 0. Refer to these examples as the

holdout set (note that the holdout examples for each of the four data sets should be nonoverlapping; that is, they should have no examples in common).

(3) For each of the four data sets, train a network using all examples in the data set, and cross-entropy error reduction. (Note that the examples in the holdout set are being used for training; however, the fact that these cells contain a known deposit is hidden from the training algorithm.)

(4) During training, monitor the value of the product of the likelihood to the holdout examples:

$$l = \prod_{i=1}^{m} f(\mathbf{x}_i) \tag{10}$$

where m is the number of holdout examples, and $f(\mathbf{x}_i)$ is the network output for example $\boldsymbol{x}_i$. As training progresses, the likelihood on the holdout examples will increase until a point is reached where overfilling begins to occur, and this will be signified by a decrease in the likelihood on the holdout examples. Stop training at this point and store the current network.

(5) After a network has been trained for each of the four data sets, calculate the overall likelihood on holdout data by taking the product of the four individual likelihoods calculated according to Equation (10).

Step 4 provides the criterion to use in order to determine when to stop training. However, the optimal number of hidden layer units must be determined, and this can be done by selecting the network configuration that results in the highest overall likelihood on holdout data (i.e., the quantity calculated in Step 5). This is essentially a sequential search problem and can be solved as follows. Train a network with a one hidden unit and calculate the overall likelihood on holdout data. Then increment the number of hidden units by 1 and again apply Steps 1 to 5. Continue incrementing the size of the network and calculating likelihood on holdout data until the likelihood begins to decrease. Select the network structure that gives best likelihood on holdout data.

The procedure for the density estimation based approach using the Parzen window technique for estimating pdfs is analogous to the procedure described for MLPs, except that in this Situation it is the σ value describing the width of the Gaussian window that must be optimized. In this example, we start with a large value for σ, and decrease it until the likelihood on holdout data begins to fall. Note that the holdout examples should not be used in the calculation of the class conditional pdf, $p(\boldsymbol{x}|D)$.

Empirical Results

The MLP was trained using the scaled conjugate gradients algorithm (Møller, 1993). Figure 3 shows the results of applying the cross-validation procedure as described.

Note that the measure on the vertical axis is *geometric mean likelihood*. This is just the geometric mean of the likelihood on all holdout examples, and is calculated as

$$gml = \left(\prod_{i=1}^{m} f(\mathbf{x}_i) \right)^{1/m} \tag{11}$$

This is a convenient measure because it allows likelihood to be compared directly with the prior probability of mineralization, which is 0.0038. Observe that in the example of the MLP, as the number of hidden layer units is increased, the mean likelihood on training examples continues to increase, but the likelihood on the holdout examples reaches a maximum of approximately 0.006, corresponding to a network with eight hidden layer units. The fact that generalization does not decrease significantly as the number of hidden layer units is increased beyond this value is because the regularization term used in training. Regularization is a technique used to restrict weights from becoming too large, thus helping prevent over-fitting (see Bishop, 1995). With the Parzen approach, observe that there is a clear maximum of slightly more than 0.006 corresponding to a Gaussian window width (σ value) of 1.2.

The maps produced using each of the approaches are shown in Figure 4. The darkest gray level represents a posterior mineralization probability of greater than 0.03, and white represents a probability of less than 0.0005. Intermediate levels are scaled in between. Visual inspection of the maps shows that the regions predicted most favorable generally coincide with the location of the known deposits, which are indicated by points. However, in each situation there are some deposits which fall in regions of low favorability.

The similarity between the maps can be measured numerically using the product moment

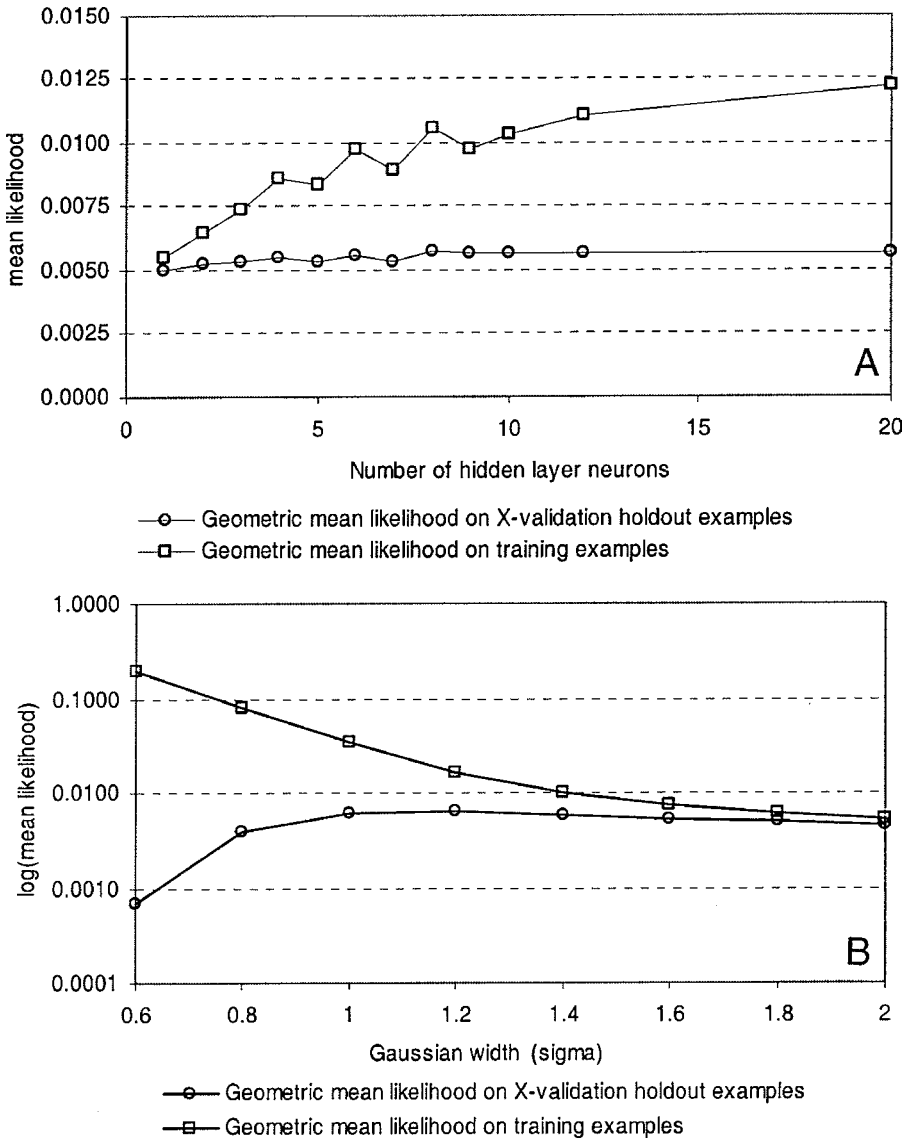

Figure 3. A, Mean geometric likelihood on training and holdout examples versus number of hidden layer units for MLP approach; B, Mean geometric likelihood on training and holdout examples versus width of Gaussian window for Parzen window density estimation based approach.

correlation, which is defined as

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \qquad (12)$$

where x and y are the pixel values (*i.e.*, posterior probabilities) for each of the two maps, $\bar{x}$ and $\bar{y}$ are their respective means, and n is the number of pixels in the map. Note, however, that r compares the maps on the basis of the numeric *values* assigned to corresponding pixels, and it is also useful to compare maps on the basis of the *ranking* assigned to pixels. This can be done using Spearman's rank correlation, r_s, which is defined as

$$r_s = \frac{\sum_{i=1}^{n}(R_x - \bar{R}_x)(R_y - \bar{R}_y)}{\sqrt{\sum_{i=1}^{n}(R_x - \bar{R}_x)^2 \sum_{i=1}^{n}(R_y - \bar{R}_y)^2}} \qquad (13)$$

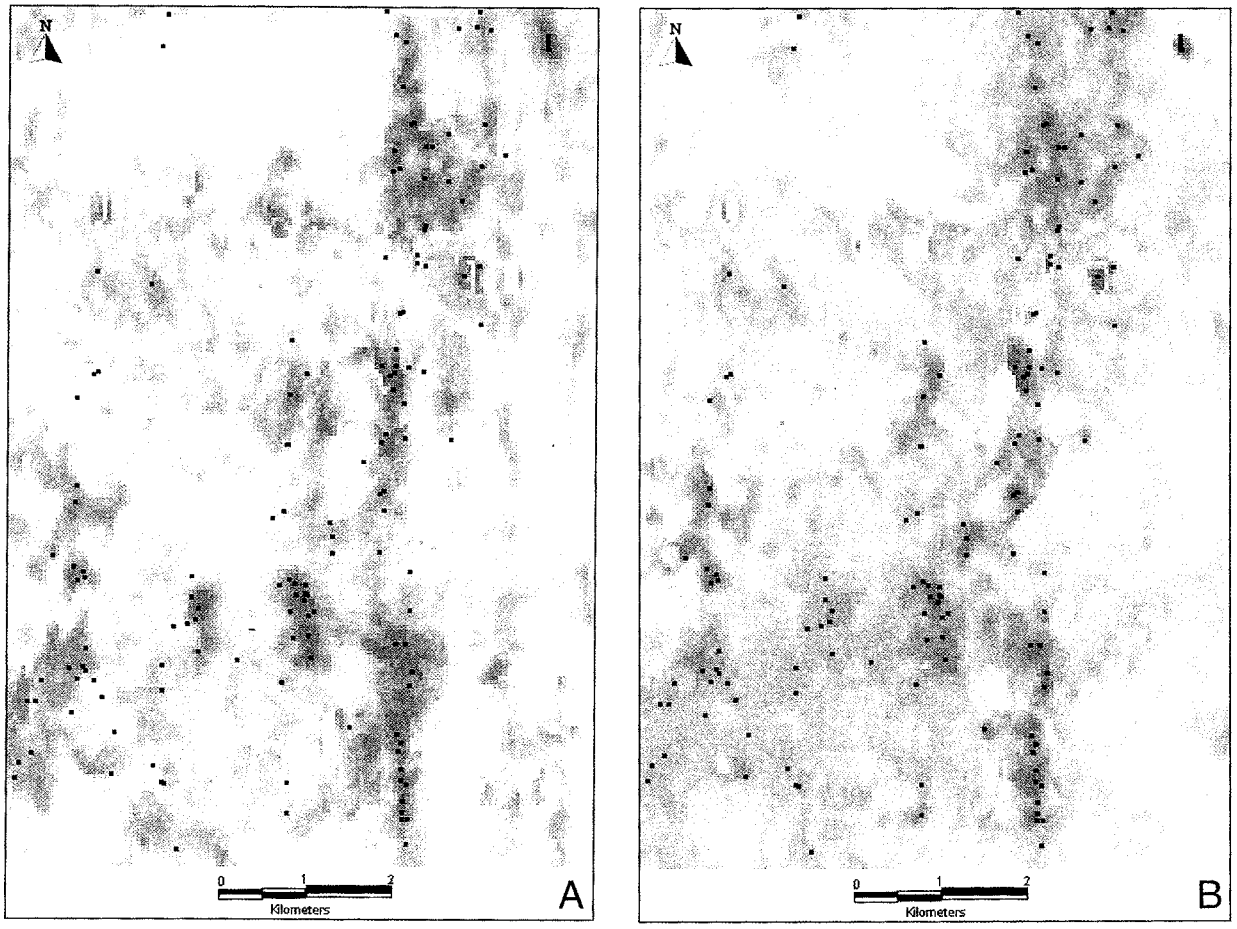

Figure 4. Probability maps. Darkest gray level represents probabilities greater than 0.03; white represents probabilities below 0.0005. Points indicate locations of known deposits: A, MLP approach; B, Parzen window density estimation based approach.

where R_x and R_y are the ranks of x and y respectively and the bar indicates the mean value as before. The r and r_s values for the maps are respectively 0.66 and 0.64, indicating a high degree of similarity.

The maps can be compared by their corresponding cumulative deposits versus cumulative area curves. Such a curve can be constructed by ranking pixels according to their assigned posterior probability value, and plotting the cumulative deposits against cumulative area as the posterior probability is decreased from its maximum to its minimum value. The curves are shown in Figure 5.

The black and gray solid curves represent respectively the predictive performance on mineralized training examples and the predictive performance on mineralized holdout examples. The dashed curve shows the cumulative sum of posterior probabilities. (Note that the cumulative deposits are expressed as a fraction of the total number of deposits; thus the maximum of 1).

For the MLP (Fig. 5A), it can be seen that the curve for prediction on training deposits corresponds closely with the curve representing the cumulative sum of posterior probabilities. This provides support for the claim that the outputs of the MLP represent the posterior probability that a pixel contains one or more of the training deposits. Further support for the claim is provided by the fact that the sum of network outputs over all examples is equal, to within approximately 0.5%, to the number of examples containing a target output of 1 (i.e., the number of mineralized examples in the training set). In regard to prediction on holdout mineralized examples, approximately 45% of the holdout deposits occur in the 10% of the region predicted as most favorable.

For the density estimation based approach, the predictive performance on holdout data is

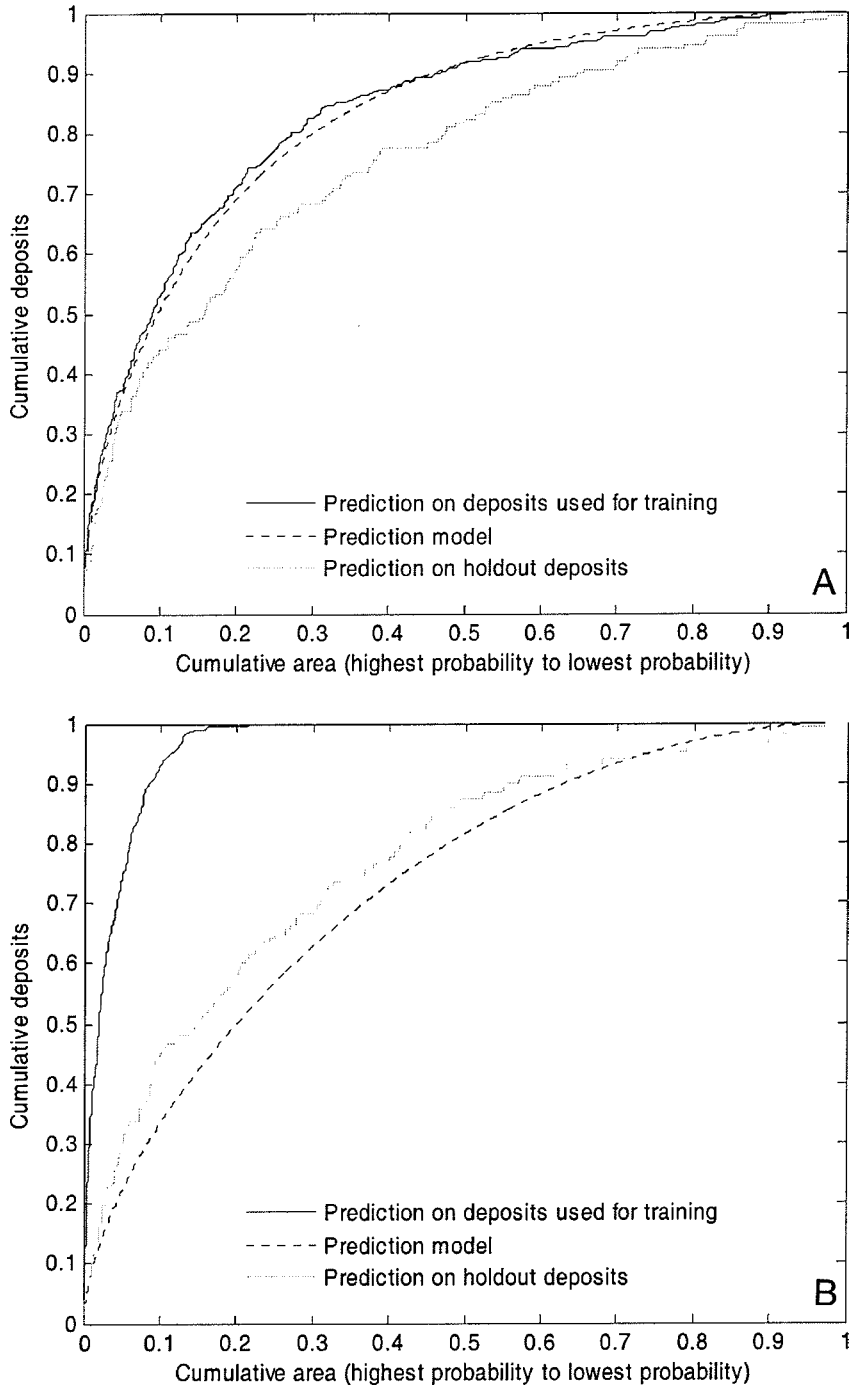

Figure 5. Cumulative deposits versus cumulative area: A, MLP with 8 hidden units; B, Parzen window density estimation based approach with $\sigma = 1.2$.

approximately the same as for the MLP, but the fit between the curve representing prediction on training deposits and the curve representing the cumulative sum of posterior probabilities is poor, indicating that the predicted values do not represent posterior probabilities as accurately as does the MLP approach. The reason for this is that the probability density functions are estimates of the true distributions of the examples,

and the resulting posterior probabilities therefore depend on the quality of these estimates. Of course, the width of the kernel window could have been increased easily until these curves coincided, but then this would not give optimal performance on the holdout mineralized data.

DISCUSSION

Two main features distinguish the method described in this paper from the conventional approach to applying neural networks to mineral potential mapping tasks: (i) the use of *all* cells in the study for training; and (ii) the use of *cross-entropy* error reduction for network weight optimization. The implications of these features for the reliability of the resulting maps now are discussed.

There are several important advantages to using all examples for training. Firstly, because no selection of training examples is required, any dependence of the map on this selection is eliminated. Secondly, using all examples for training significantly reduces the dimensionality problems resulting from a sparsely populated input space. For example, in the case study provided here, approximately 65,000 examples were used for training; if deposit and nondeposit cells were represented equally in the training set, then the number of training examples would be approximately 300. Using all examples for training reduces the variance in the network enormously, and thus the final network will be far less sensitive to factors such as initial weight assignments. Also convergence will be slower, thus allowing better precision in using the special cross-validation procedure to stop training. Finally, using all examples for training ensures that maximal use is made of the context provided by the data. The objective, after all, is to assign to each pixel in the study area a value indicating its likelihood of being mineralized, and, to this end, it makes complete sense to use all of the available data for training. This does not cause problems with cross-validation, because, as described, this can be performed by holding out only the *label* attached to holdout examples; that is, the training algorithm sees the feature vector of holdout mineralized examples, but does know that the holdout example is mineralized.

The second distinguishing feature of the approach is the use of cross-entropy for error reduction. This paper has shown theoretically that if the network is trained using cross-entropy error reduction on a training set consisting of all examples in the study region, then the network output can be interpreted as the posterior probability of mineralization, given the evidence associated with the pixel. The objection could be raised that we do not know the true prior probability of mineralization, and that the output is therefore not a true posterior probability. However, this is not a valid objection because the output represents the posterior probability that an example contains one or more of the *known* deposits. Obviously, if more deposits were discovered, this would affect the prior probability, and in this situation the outputs could be linearly rescaled to account for this, or alternatively, the entire network could be retrained incorporating the newly discovered deposits into the training data. Nevertheless, the outputs of the network have a definite interpretation as probabilities, which is not the situation with the conventional approach.

The problem that the relative scarcity of mineralized cells causes for the conventional approach has been acknowledged by Brown, Gedeon, and Groves (2003), who propose adding noise to the mineralized training patterns, thus creating additional synthetic deposit training data. Adding noise is a valid approach to expanding the training set on many inductive learning tasks, and indeed it may improve the performance of the conventional MLP approach to mineral-potential mapping. However, the use of all examples for training, in conjunction with the use of cross entropy error minimization, eliminates the perceived requirement that deposit and nondeposit training examples be represented in equal proportion in the training data, and thus obviates the need for any such random expansion of the training data.

This paper has considered a density estimation based approach in which class-conditional and class-unconditional pdfs are estimated using the Parzen window technique, and combined using Bayes' Theorem to arrive at an estimate of the posterior probability. It is possible and straightforward to cast this approach into a neural network framework, and the resulting networks are referred to as *Probabilistic Neural Networks* (PNNs) (Specht, 1990). It should be realized, however, that the reformulation of the technique as a neural network is cosmetic only, and does not add anything to the original formulation.

In the context of mineral-potential mapping, it has been claimed that "when the probability that an area is mineralized is the objective of analysis, PNN is the appropriate neural network architecture" (Harris and Pan, 1999). It can be seen that while PNNs may be *an* appropriate architecture, MLPs are also

appropriate, given an appropriate training procedure, which this paper has described.

Moreover, the results presented in this paper have shown that the density estimation based approach and the MLP approach display approximately equal performance in regard to their predictivity on holdout examples. This should be expected, in fact, providing that care is taken in determining the values of the respective model-specific parameters (number of hidden units for the MLP, and kernel width for the density estimation approach). Parameter tuning is an integral part of the training process, and through the use of a special cross-validation procedure, this paper has shown that two fundamentally different techniques can yield similar results, mutually supporting the claim that the resulting maps can be interpreted as representing the conditional probability of mineralization, given the evidence supplied by the background data.

CONCLUSION

A new technique has been presented for applying MLPs to the problem of mineral-potential mapping. The technique uses all pixels in the study region for training, thus eliminating any sensitivity to the particular selection of training examples. Providing that cross-entropy error reduction is used for training, the outputs of the MLP can be interpreted as representing the posterior probability that a pixel contains one or more of the known deposits, given the feature vectors describing the pixel. The use of cross-entropy error reduction, together with the use of all examples for training, makes the technique much less susceptible to the dimensionality problems suffered by current approaches to applying MLPs in this area. It also ensures that use is being made of the contextual information provided by all nondeposit examples.

REFERENCES

Baum, E. B., and Wilczek, F., 1988, Supervised learning of probability distributions by neural networks, *in* Anderson, D. Z., ed., Neural Information Processing Systems: Am. Ins. Physics, New York, p. 52–61.

Bishop, C. M., 1995, Neural networks for pattern recognition: Oxford Univ. Press, Oxford, 482 p.

Bonham-Carter, G. F., 1994, Geographic information systems for geoscientists: modeling with GIS: Pergamom Press, Oxford, 398 p.

Brown, W. M., Gedeon, T. D., and Groves, D. I., 2003, Use of noise to augment training data: a neural network method of mineral–potential mapping in regions of limited known deposit examples: Natural Resources Research, v. 12, no. 2, p. 141–152.

Brown, W. M., Gedeon, T. D., Groves, D. I., and Barnes, R. G., 2000, Artificial neural networks: a new method for mineral prospectivity mapping: Jour. Australian Earth Sciences, v. 47, no. 4, p. 757–770.

Clark, I., and Cook, B., eds., 1988, Victorian geology excursion guide: Australian Acad. Science, Canberra, Australia, 489 p.

Cochrane, G. W., Quick, G., and Spencer-Jones, D., eds., 1995, Introducing Victorian geology (2nd edn,): Geol. Soc. Australia Incorporated (Victorian Division), Melbourne, Australia, 304 p.

Cybenko, G., 1989, Approximation by superpositions of a sigmoidal function: Math. Control, Signals, and Systems, v. 2, p. 303–314.

Dempster, A. P., Laird, N. M., and Rubin, D. B., 1977, Maximum likelihood from incomplete data via the EM algorithm: Jour. Roy. Statist. Soc. Series B, v. 39, p. 1–38.

Duda, R. O., and Hart, P. E., 1973, Pattern recognition and scene analysis: John Wiley & Sons, New York, 482 p.

Harris, D., and Pan, G., 1999, Mineral favorability mapping: a comparison of artificial neural networks, logistic regression, and discriminant analysis: Natural Resources Research, v. 8, no. 2, p. 93–109.

Harris, D., Zurcher, L, Stanley, M., Marlow, J., and Pan, G., 2003, A comparative analysis of favorability mappings by weights of evidence, probabilistic neural networks, discriminant analysis, and logistic regression: Natural Resources Research, v. 12, no. 4, p. 241–256.

Hopfield, J. J., 1987, Learning algorithms and probability distributions in feed-forward and feed-back networks: Proc. Nat. Acad. Sciences, v. 84, p. 8428–8433.

Hornick, K., Stinchcombe, M., and White, H., 1989, Multilayer feed-forward networks are universal approximators: Neural Networks, v. 2, no. 5, p. 359–66.

Møller, M., 1993, A scaled conjugate gradient algorithm for fast supervised learning: Neural Networks, v. 6, no. 4, p. 523–533.

Parzen, E., 1962, On estimation of a probability density function and mode: Ann. Math. Statistics, v. 33, no. 3, p. 1065–1076.

Porwal, A., Carranza, E. J., and Hale, M. 2003, Artificial neural networks for mineral-potential mapping: a case study from the Aravalli Provence' Western India: Natural Resources Research, v. 12, no. 3, p 155–171.

Rumelhart, D. E., and McClelland, J. L., 1986, Parallel distributed processing: exploration in the microstructure of cognition (Vols. 1 & 2), MIT Press, Cambridge, MA, 611 p. and 547 p.

Schumacher, M., Rossner, R., and Vach, W., 1996, Neural networks and logistic regression: part 1 & 2: Computational Statistics & Data Analysis, v. 21, p. 661–701.

Singer, D. A., and Kouda, R., 1996, Application of a feed-forward neural network in the search for Kuroko deposits in the Hokoruko District, Japan: Math. Geology, v. 28, no. 8, p. 1017–1023.

Singer, D. A., and Kouda, R., 1997, Classification of mineral deposits into types using mineralogy with a probabilistic neural network: Nonrenewable Resources, v. 6, no. 1, p. 27–32.

Singer, D. A., and Kouda, R., 1999, A comparison of the weights-of-evidence method and probabilistic neural networks: Natural Resources Research, v. 8, no., 4, p. 287–298.

Skabar, A., 2000, Inductive learning techniques for mineral potential mapping: unpubl. docforal dissertation, School Electrical and Electronic Systems Engineering, Queensland Univ. Technology, Australia, 225 p.

Specht, D. F., 1990, Probabilistic neural networks: Neural Networks, v. 3, no. 1, p. 109–118.

Titterington, D. M., Smith, A. F. M., and Makov, U. E., 1985, Statistical analysis of finite mixture distributions: John Wiley & Sons, New York, 243 p.

Willman, C. E. 1995, Castlemaine Goldfield: Castlemaine-Chewton, Fryers Creek 1:10 000 Maps: Victoria Energy and Minerals, Geol. Survey Rept. 106.