Mathematical Geology, Vol. 38, No. 2, February 2006 (© 2006)
DOI: 10.1007/s11004-005-9010-z

Comparative Evaluation of Neural Network Learning Algorithms for Ore Grade Estimation[1]

B. Samanta,[2] S. Bandopadhyay,[2] and R. Ganguli[2]

In this paper, comparative evaluation of various local and global learning algorithms in neural network modeling was performed for ore grade estimation in three deposits: gold, bauxite, and iron ore. Four local learning algorithms, standard back-propagation, back-propagation with momentum, quickprop back-propagation, and Levenberg–Marquardt back-propagation, along with two global learning algorithms, NOVEL and simulated annealing, were investigated for this purpose. The study results revealed that no benefit was achieved using global learning algorithms over local learning algorithms. The reasons for showing equivalent performance of global and local learning algorithms was the smooth error surface of neural network training for these specific case studies. However, a separate exercise involving local and global learning algorithms on a nonlinear multimodal optimization of a Rastrigin function, containing many local minima, clearly demonstrated the superior performance of global learning algorithms over local learning algorithms. Although no benefit was found by using global learning algorithms of neural network training for these specific case studies, as a safeguard against getting trapped in local minima, it is better to apply global learning algorithms in neural network training since many real-life applications of neural network modeling show local minima problems in error surface.

KEY WORDS: neural network optimization; local learning algorithm; global learning algorithm; multimodal; ward-net network; activation functions.

INTRODUCTION

Ore grade estimation and control remains one of the most difficult problems to mining engineers and geologists. Capital-intensive mining operations invariably require an accurate knowledge of tonnage and grade of a deposit for mineral appraisal and grade control. Complex formation of an ore deposit, in many cases, makes it more difficult to resolve problems in ore body modeling. Researchers have made frequent improvements over the years for accurately predicting the ore grades using various advance grade estimation techniques. Among these

[1] Received 17 November 2003; accepted 27 May 2005; Published online: 25 May 2006.

[2] University of Alaska Fairbanks, PO Box 755800, Fairbanks, Alaska, 99775; e-mail: ffbs1@uaf.edu; ffs0b@uaf.edu; ffrg@uaf.edu.

techniques, geostatistics perhaps remains the most prevalent technique used today. In recent times, neural network has emerged as an alternative to geostatistics in grade estimation purpose (Ke, 2002; Yama and Lineberry, 1999). Several studies reported successful implementations of the neural network technique for the estimates of spatial attributes. For example, Wu and Zhou (1993) applied neural network for copper reserve estimation. Rizzo and Dougherty (1994) used this techniques for characterization of aquifer properties. Koike and others (2002) investigated neural network for determining the principal metal contents of Hokuroku district in Northern Japan. Koike and Matsuda (2003) also used this technique for estimating content impurities of a limestone mine namely, SiO_2, Fe_2O_3, MnO, and P_2O_5. The authors have also used neural networks (along with geostatistics) for grade estimation in a bauxite deposit and a gold deposit (Samanta, Bandopadhyay, and Ganguli, 2004; Samanta, Ganguli, and Bandopadhyay, 2003). In the bauxite deposit, neural networks and geostatistics showed almost equivalent performance, while for the gold deposit the neural network performed better than the geostatistics (ordinary kriging). Based on earlier experiences, neural networks are further investigated here for ore grade estimation, albeit in a different perspective of optimization of neural network learning.

Proper training can be a problem in neural network modeling. A neural network model working on a grade estimation problem performs mappings from an input space to an output space. For example, given the spatial coordinates as input and grade attribute as output, neural network will be able to generate a mapping function through a set of connection weights between input and output. Hence, output O of a neural network can be defined as a function of inputs X and connection weights W, i.e., $O = \varphi\ (X, W)$, where φ is a mapping function. Training of a neural network, implemented to finding a good mapping function, can be done by adjusting the connection weights between the neurons of a network using a suitable learning algorithm while fixing the network architecture and activation function. In essence, given a set of training patterns consisting of input–output data pairs of spatial coordinates and grade attribute $\{(I_1, D_1), (I_2, D_2), \ldots, (I_n, D_n)\}$, the learning algorithm strives to minimize the training error. One popular error function is squared error function in which error, $e\ (W, I_i, D_i) = (\phi\ (I_i, W) - D_i)^2$. Using a suitable learning rule, a set of connection weights, W, is found so that the squared error function gets minimized.

In a multilayer feedforward neural network, supervised learning algorithm is applied to train a network. Supervised learning used in neural network training can be considered as an unconstrained nonlinear optimization problem in which the objective function (squared error function) is minimized in the search of weight space. The error function spanned by weight space in a neural network might have a single minimum as a global minimum. On the other hand, the error function might generate a very complicated error surface with many local minima in weight space, one of which is a global minimum. For example, Gallagher (1999) argued that a local minimum in neural network is not a major problem, whereas, Shang and Wah (1996) showed that an error surface could be very rugged and might have many local minima. For the first case, local learning algorithms will be adequate. Obviously in the presence of many local minima, local learning algorithms will have difficulty in finding optimal solution and will get trapped in a local minimum point.

It is frequently observed that modelers tend to use local learning algorithms for neural network training without paying much attention to the problem of local minima. Also, many geoscience neural network studies (conducted for estimation purposes) have reported using local learning algorithms for neural network training. For example, quickprop algorithm (Wu and Zhou, 1993), standard back-propagation algorithm (Yama and Lineberry, 1999), and back-propagation algorithm with momentum (Koike, Matsuda, and Gu, 2001; Koike and others, 2002) are all local learning algorithms; although Singer and Kouda (1996) used simulated annealing along with conjugate gradient method for network training to escape from local minima. However, none of the above studies paid attention to the problem of local minima in neural network training. Therefore, the present study was conducted to observe the performance of the local and global learning algorithms for neural network modeling in ore grade estimation and contribute to the literature. The relevance of this study is particularly noteworthy to the geostatisticians since neural network is emerging as a very promising technique in research, and the use of a suitable learning algorithm will play a major role in successful implementation of neural network.

Four local optimization techniques: (i) standard gradient descent back-propagation with a fixed learning rate, (ii) back-propagation with momentum learning, (iii) quickprop back-propagation learning, and (iv) Levenberg–Marquardt back-propagation learning were investigated. Additionally, two global learning algorithms: (i) a trace-based method called NOVEL and (ii) a simulated annealing method were also explored. Comparative evaluation of these techniques in neural network learning optimization has been carried out on three ore grade estimation problems. Furthermore, for a better understanding of the local and global optimization algorithms in neural network modeling, these techniques were first explored on a two-dimensional nonlinear optimization problem containing many local minima in search trajectory. It can also be noted that although the behavior of various local learning algorithms and the simulated annealing has been studied in other neural network applications (particularly in non-mining data), the efficiency of the NOVEL algorithm has not been tested extensively in neural network training.

LOCAL LEARNING ALGORITHMS

Learning the weights of a neural network can be considered as an unconstrained continuous nonlinear minimization problem. In the past, many techniques have been developed for solving nonlinear optimization problems in other disciplines; these methods can be classified into local optimization and global optimization. Methods of local optimization include gradient descent algorithm, Newton's method, and conjugate gradient method. These techniques are also applicable to neural network learning, however, in a different format.

Local optimization techniques use some form of gradient information in search strategy, and require calculation of gradient of error with respect to weight vector. Because of hidden layers topology in a neural network, it is not possible for direct calculation of an error gradient with respect to hidden layers connection weights. Instead, an algorithm called back-propagation is used to calculate the gradient. Back-propagation algorithm applies a chain rule for calculating gradient, and which is done by back-propagating the sensitivities (change of error function with respect to net input to a neuron) from output layers to previous layers step by step in the backward direction (output-hidden-input); hence the name "*Back-propagation*." The basic mechanisms and mathematical foundation of the four local optimization techniques studied here can be found in a number of textbooks (Bishop, 1995; Hagan, Demuth, and Beale, 1996; Haykins, 1999). For the convenience of readers and for understanding the outcome of this study, only a comprehensive overview is presented.

Standard Back-Propagation with Gradient Descent (SBP)

The gradient descent algorithm finds a locally optimal solution by iteratively taking small steps in the gradient descent direction. The search procedure starts with a random initial guess of parameters in the weight space. Then weight is updated in each iteration according to following equation:

$$w(n+1) = w(n) - \eta \times \nabla e(w). \tag{1}$$

where η is learning rate parameter and $\nabla e(w)$ is the gradient of error.

The learning rate parameter η plays a major role in convergence of the algorithm. For small values, it causes small changes in weight along the gradient descent direction, which results in very slow progress along search trajectory. On the other hand, for large values, it may result in faster convergence though in some cases, it may overshoot the optimal solution.

Back-Propagation with Momentum (MBP)

A large learning rate causes gradient descent algorithms to oscillate along search trajectory; sometimes it may even cause divergence. However, in order to get the full benefit of faster convergence with large learning rate, oscillation along the search path must be reduced. Use of momentum facilitates to dampen oscillation and renders fast convergence. Momentum algorithm introduces a momentum factor and makes the new weight changes as:

$$\Delta w(n) = \gamma \, \Delta w(n-1) - (1-\gamma) \times \eta \times \nabla e(w) \tag{2}$$

where $\Delta w\ (n) = w\ (n) - w\ (n-1)$, γ = momentum coefficient, $0 \leq \gamma < 1$, $\nabla e\ (w)$ = the gradient of error.

Quickprop Back-Propagation Algorithm

Quickprop algorithm is a variation of momentum algorithm, however, instead of using user supplied momentum factor, the algorithm itself determines the momentum factor. The algorithm estimates the momentum factor by using gradient information on the current and previous steps with the assumption that error surface can be approximated by a parabola. According to Wu and Zhou (1993), the weight update formula of quickprop algorithm is

$$\Delta w(n) = -\eta \nabla(w(n)) + \frac{\nabla(w(n))}{\nabla(w(n-1)) - \nabla(w(n))} \Delta w(n-1) \tag{3}$$

The notations are the same as the MBP algorithm described above.

Levenberg–Marquardt Algorithms (LMBP)

Levenberg–Marquardt algorithm is a modification of Newton's method for nonlinear optimization. The Levenberg–Marquardt algorithm does not utilize second derivatives unlike Newton's method (in the Hessian computation, the second derivative component is ignored assuming it is small). This method is based on the concept of quadratic approximation of error function in a local region. Note that if the error function is truly quadratic in nature, Newton's method finds the minimum solution in a single iteration. Therefore, the success of this technique depends upon how the error function resembles a quadratic function. If the quadratic approximation is not appropriate, the algorithm may diverge. Searching of an optimal solution using this method requires calculation of the inverse of the Hessian matrix, which should be positive definite. Newton's method does not always guarantee the positive definiteness of Hessian matrix. Levenberg–Marquardt introduces a regularization term into the Hessian matrix so that the positive definiteness of the Hessian matrix is guaranteed.

GLOBAL LEARNING ALGORITHMS

Optimization problems in neural network may be unimodal or multimodal depending upon the number of local minima in the space of error surface. Figure 1 presents general features of unimodal and multimodal cases. In a multimodal case with a number of local minima, the following observations can be made: (a) flat regions may mislead gradient-based methods, (b) there may be many local minima that trap gradient-based methods, and (c) gradients may differ by many orders of magnitude, making it difficult for gradient-based algorithms to work efficiently. Therefore, a good search strategy should have properties to escape from local minima once it gets there. The basic mechanisms of two global learning algorithms studied here are described in the following sections.

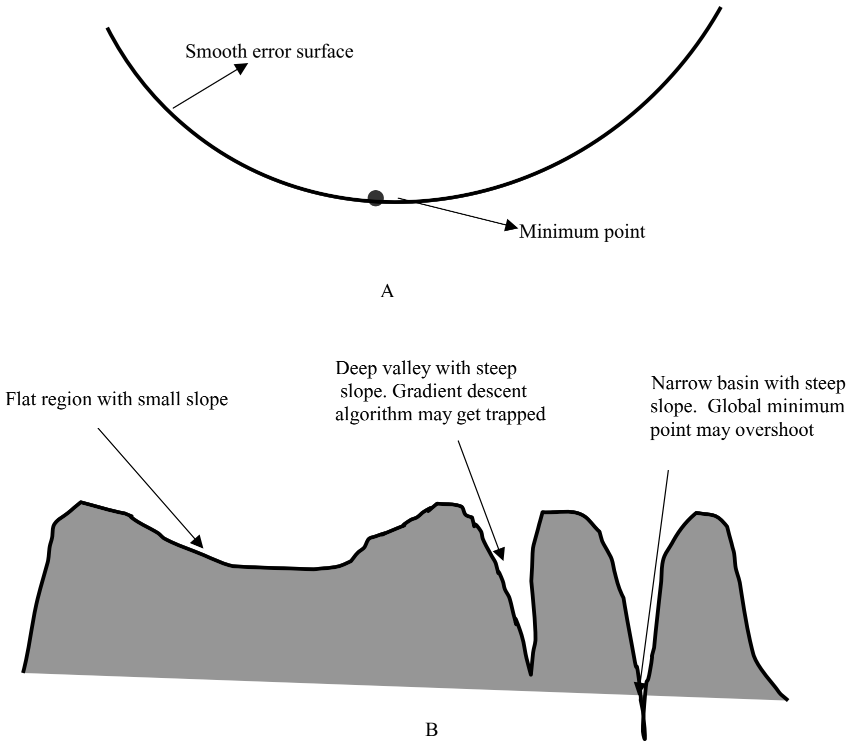

Smooth error surface
Minimum point
A
Flat region with small slope
Deep valley with steep slope. Gradient descent algorithm may get trapped
Narrow basin with steep slope. Global minimum point may overshoot
B

Figure 1. General profile of error surface A, for unimodal and B, multimodal.

Trace-Based NOVEL Method

The NOVEL method is a hybrid of global and local search methods. This method was explored primarily because of its reported success in neural network training on a two-spiral problem (Shang and Wah, 1996). Trace-based global learning is a trajectory-based method that relies on an external force to pull out the search from local minima, and employs local descent algorithms to locate minima. It has three features: exploring solution space, locating promising regions, and finding local minima. In exploring solution space, the search is guided by a continuous terrain-independent trace that does not get trapped into local minima. In locating promising regions, NOVEL uses local gradient to attract the search to a local minimum but relies on the trace to pull it out of the local region once little improvement can be found. Finally, NOVEL selects an initial point for each promising region and uses them as initial points for local algorithm to find local minima.

In the global search phase, there are a number of bootstrapping stages. Figure 2 shows a conceptual diagram of the NOVEL method. The figure shows three stages but number of stages can be varied. Prior stage is coupled to the next stage by feeding its output trajectory as the trace function of the next stage, with a user-defined trace function as the input trace function of the first stage.

Each stage in the global search phase defines a trajectory of weight space, $w(t)$, which is governed by the following equation.

$$w_1(t + \delta t) = w_1(t) - \delta t \cdot [\mu_g \cdot \nabla e(w_1(t)) + \mu_t \cdot (w_1(t) - T(t)] \qquad (4)$$

where t is the autonomous variable, T is a trace function, μ_g and μ_t are constant coefficients. $\nabla e(\cdot)$ is the gradient of the error function. Note that the error is function of $w_1(t)$.

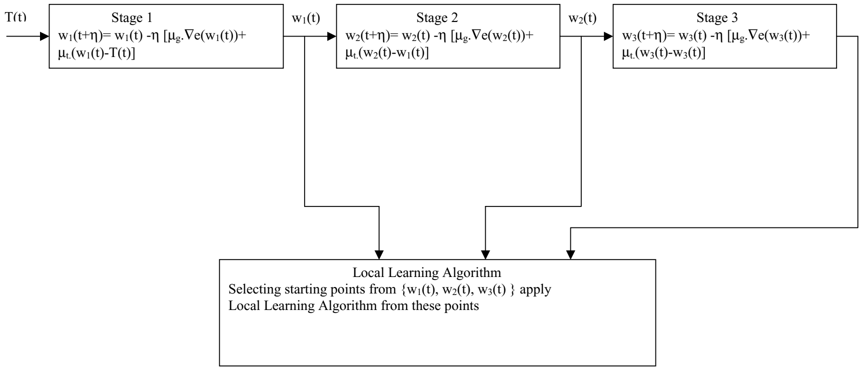

T(t)
Stage 1
w1(t+η)= w1(t) -η [μg.∇e(w1(t))+ μt.(w1(t)-T(t)]
w1(t)
Stage 2
w2(t+η)= w2(t) -η [μg.∇e(w2(t))+ μt.(w2(t)-w1(t)]
w2(t)
Stage 3
w3(t+η)= w3(t) -η [μg.∇e(w3(t))+ μt.(w3(t)-w3(t)]
Local Learning Algorithm
Selecting starting points from {w1(t), w2(t), w3(t) } apply Local Learning Algorithm from these points

Figure 2. Conceptual diagram of NOVEL method.

There are two main components in Equation (4), $\mu_g \cdot \nabla e(w_1(t))$ enables the gradient to attract the trajectory to a local minimum and $\mu_t \cdot (w_1(t) - T(t))$ allows the trace function to lead the trajectory out of the local minimum. The coefficients μ_g and μ_t in the equations may be different for each stage. In earlier stages, more weight can be placed on the trace function, thus allowing the resulting trajectory to explore more regions. In later stages, more weight can be placed on local descents, allowing the trajectory to descend deeper into local basins. To find global minima efficiently, Equation (4) requires a trace function that traverses the search space uniformly. Based on Shang and Wah (1996), the trace function chosen in this study is as follows:

$$T_i(t) = \rho \sin\left[2\pi \left(\frac{t}{2}\right)^{1-(0.05+0.45(i-1)/n)} + \frac{2\pi(i-1)}{n}\right] \tag{5}$$

where i is the ith dimension, ρ is a coefficient specifying the range, and n is the number of dimensions.

After exploring the promising regions in global search phase, local search phase would be initiated. In the local search phase, a traditional descent method, such as the gradient descent method or the Levenberg–Marquardt method can be used.

Simulated Annealing

Simulated annealing (SA) exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system. The algorithm is based on Metropolis and others (1958) who originally proposed it as a means of finding equilibrium configuration of a collection of atoms at a given temperature. The connection between this algorithm and mathematical minimization was first noted by Pincus (1970), but it was Kirkpatrick, Gerlatt, and Vecchi (1983) who proposed that it forms the basis of an optimization technique for combinatorial and other problems.

The algorithm employs a random search around the neighborhood of a current solution. The change in the solution is not only accepted when the objective function decreases, but some changes are also accepted when the objective function increases. This means that the algorithm not only allows downhill movement but also accepts uphill movements. Uphill movements are accepted with a probability:

$$P = \exp\left(-\frac{\partial f}{T}\right) \tag{6}$$

where ∂f is the increase in objective function, T is a control parameter, which by analogy with the original application is known as the system temperature.

Equation (6) reveals that the probability of acceptance of a new solution that increases objective function depends upon two parameters: (a) magnitude of increase in objective function and (b) control parameter/system temperature, T. The probability of uphill movement increases at high temperature and decreases at low temperature. Initially, the temperature values are set at high so that solution space can move freely, thereby, exploring promising local regions. The temperature goes down step by step so that acceptance of uphill movement cuts down slowly. This helps locate the minimum point in the solution space.

UNDERSTANDING OPTIMIZATION ALGORITHMS FOR A NON-LINEAR MINIMIZATION PROBLEM

Prior to the application of optimization techniques in a neural network, the properties of global and local optimization algorithms were studied in a simple two-dimensional nonlinear optimization problem. The optimization problem involves finding the minimum of a function consisting of two variables. This function, called a Rastrigin function, is used in optimal control application. The basic form of Rastrigin function is $f(x_1, x_2) = x_1^2 + x_2^2 - \cos 18x_1 - \cos 18x_2$. The properties of this function are demonstrated by a three-dimensional surface plot in Figure 3A and its corresponding contour plot in Figure 3B. Figure 3 reveals that there are many local minima of this function in the range of $-1 \leq x_i \leq 1$; however, there is one global minimum point. The global minima point occurs at $x_1 = 0$ and $x_2 = 0$, and the functional value at this point is -2.0.

The superiority of the two global learning algorithms over a local gradient descent algorithm was verified in this nonlinear minimization of a multimodal problem. For this part of the study, the same initial random starting point was chosen for all the algorithms. The behavior of the algorithms in finding the global minima is presented in 4–6 for the random starting point $x_1 = 0.95$ and $x_2 = 0.23$. Figure 4 shows the trajectory of the local gradient descent algorithm, which reveals that this local algorithm moves only a little from the starting point before getting trapped in a local minimum ($x_1 = 1, x_2 = 0.35$, and $f(x_1, x_2) = -0.54$).

Trace-based NOVEL algorithm starting from same initial point, however, finds the global minimum point efficiently at $x_1 = 0$, $x_2 = 0$ with the global minimum value of -2.0. In order to demonstrate the behavior of the NOVEL algorithm, the trajectories of 10 initial steps of global search phase of the three stages of NOVEL are presented in Figure 5. The first stage in Figure 5A shows that many of the local basins is touched by the 10 initial steps of the global search phase. Note that though all local basins are touched in the global search phase, only 10 steps are shown in the figure for easy visualization. In the first boosting

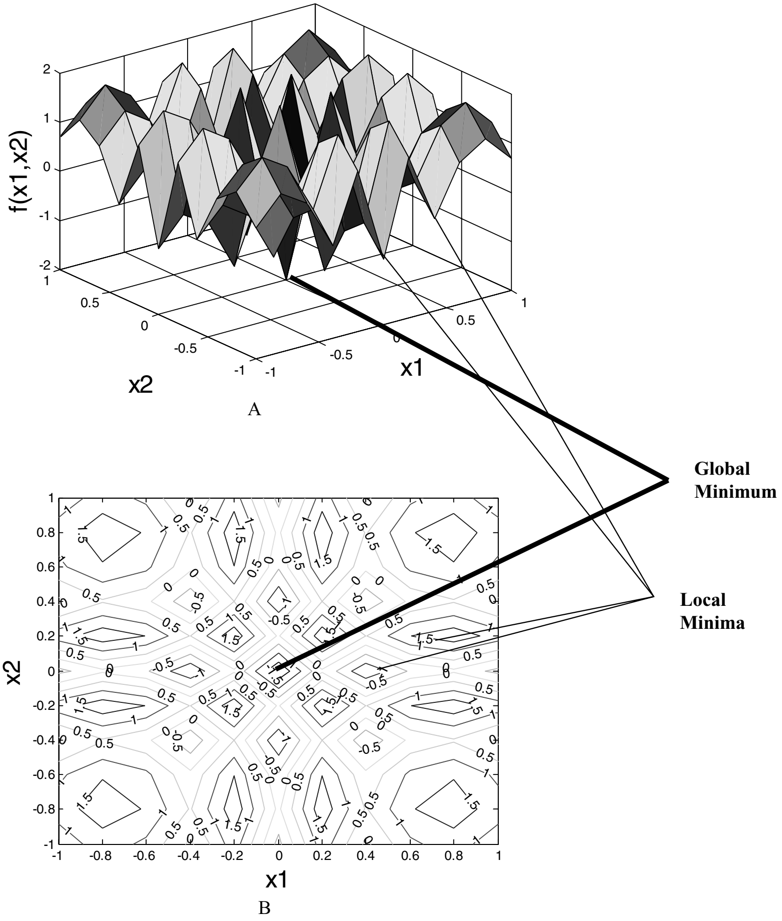
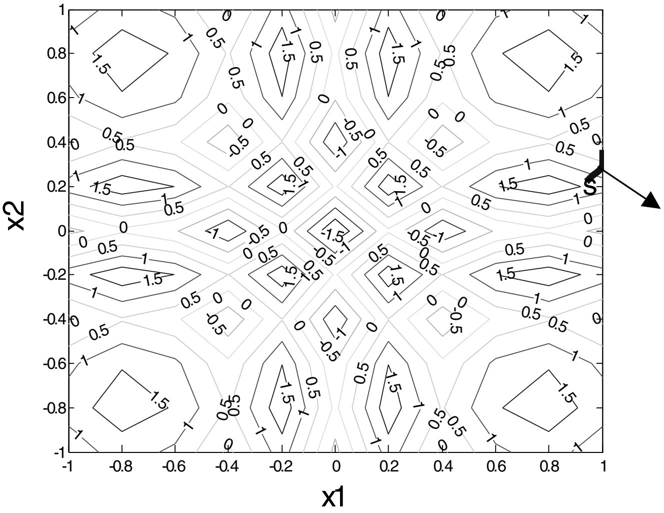


Figure 3. Rastrigin function, A, for surface plot and B, contour plot.

stage (Figure 5B), the points are more attracted toward local basins as it uses the trajectory of the previous stage as the input function, which supplies some gradient information of the trajectory to the boosting stage. In the second boosting stage (Fig. 5C), the gradient information is further enhanced and the trajectory is more attracted toward local gradient direction.

Simulated annealing also finds the global minima efficiently though it sticks to a minimum point of -1.996 at $x_1 = 0.002$ and $x_2 = -0.004$. The behavior of simulated annealing algorithm is presented in Figure 6. Initially, a large number of uphill movements of the solutions are accepted as the control parameter, temperature, is at a high number ($T = 100$). However, it cool downs exponentially at a rate



Trapped in local minimum point

Figure 4. Trajectory of local optimization algorithm (gradient descent) for Rastrigin function.

of $T_i = 0.9\, T_{i-1}$, while allowing a certain number of iterations (either 100 or less than 35 number of accepted transition points whichever it reaches first) at each particular temperature setup. At high temperature, where a lot of uphill movements are accepted, the solution escapes from local minima. When system cools down sufficiently and virtually no uphill movement is accepted, search focuses on the local region, and the system converges to the minimum solution.

The above experiment was repeated for many starting values. The results of which were generally similar to what is described. Therefore, in multimodal cases the gradient-based local learning algorithm is inappropriate.

CASE STUDY APPLICATIONS

Neural network was applied to three case studies of ore grade estimation in a gold deposit, a bauxite deposit, and an iron ore deposit. The performance of the neural network learning algorithms was extensively studied. The goal of this study was to learn the role of various learning algorithms in neural network training optimization. Therefore, generalization properties of these networks were not taken into consideration.

The gold deposit studied here is an offshore placer deposit at Nome district in Alaska. The Nome district is located on the south shore of Seward Peninsula at about latitude 64°30′ N and longitude 165°30′ W. It is 840 km west of Fairbanks and 860 km northwest of Anchorage. For grade evaluation purpose, the lease boundary

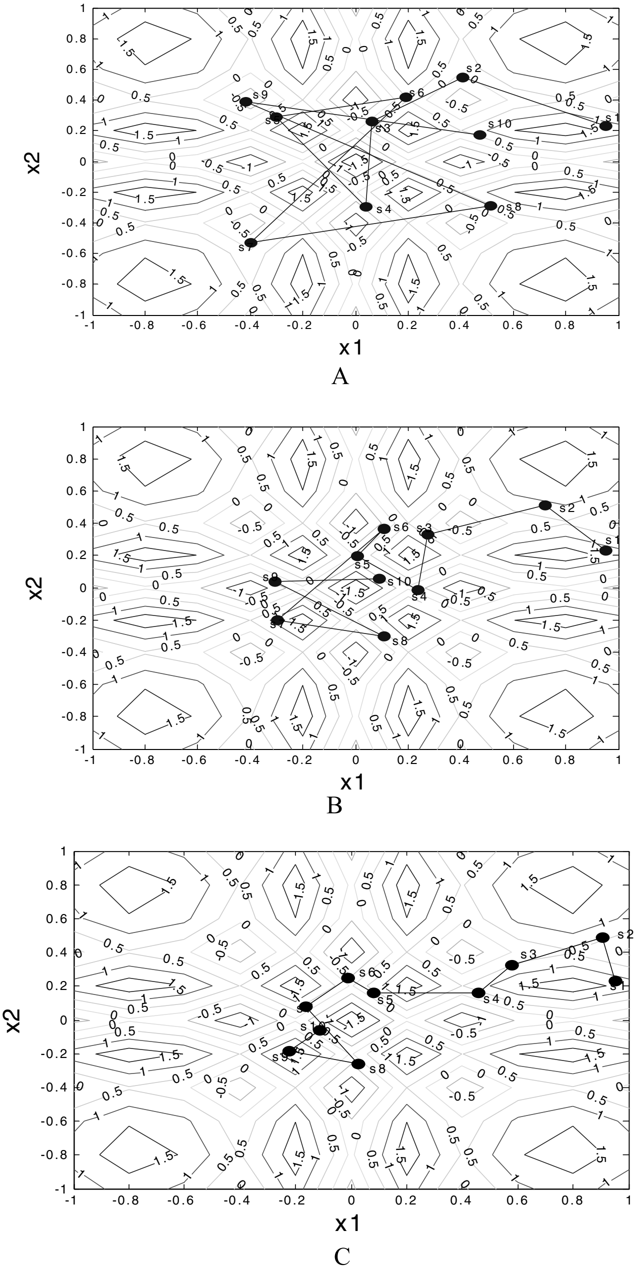


Figure 5. Global search phases of NOVEL (first 10 steps) for Rastrigin function minimization, A for stage 1, B, stage 2, and C, stage 3.

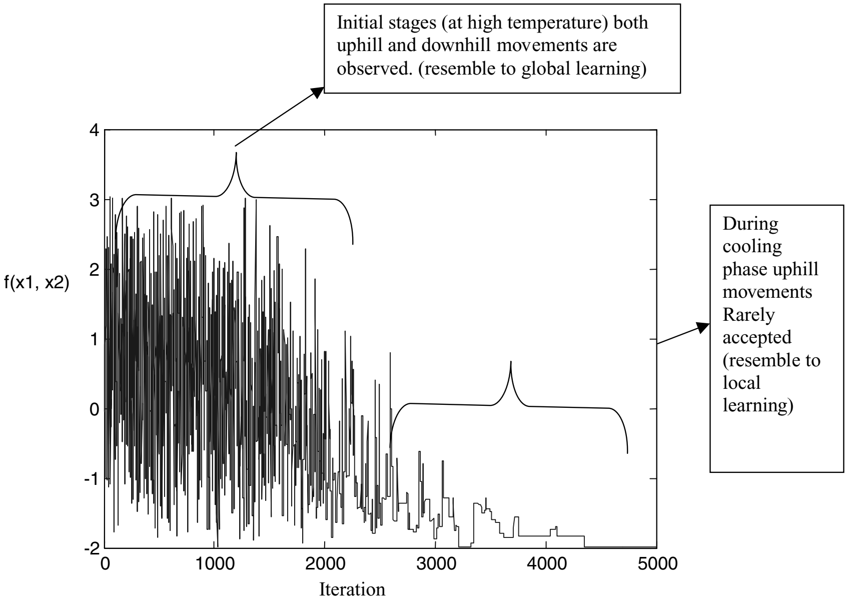


Figure 6. Simulated annealing algorithm for Rastrigin function minimization.

is divided into nine blocks—Coho, Halibut, Herring, Humpy, King, Pink, Red, Silver, and Tomcod. The present study was focused on the Coho block using gold assay values (mg/m^3) derived from 134 exploratory borehole samples located in an irregular grid. Figure 7 presents the spatial plot as well as omnidirectional variogram of the gold concentration.

The bauxite deposit studied here is situated at the Koraput district of Orissa in India. The deposit extends over an area of about 16 km^2 and is the single largest bauxite deposit in India, and one of the largest in the world. For operational convenience, the deposit has been divided into north, central, and south blocks. The central block of the deposit is an integrated part of the lateritic profile, and is derived by the *in situ* chemical weathering of khondalite in tropical surroundings. The central block has been divided into two sectors namely, sector I and sector II. Data used in this study included 126 exploratory boreholes in sector II. The boreholes were drilled mostly in a square grid pattern with 25 m spacing. The critical variable of the bauxite ore is $Al_2O_3\%$ and was considered for grade estimation. Figure 8 shows the spatial plot along with the omnidirectional variogram of $Al_2O_3\%$.

The iron ore deposit studied here is situated at the Keonjhar district of Orissa in India. Ore bearing area covers about 5 km on the eastern slope of the famous iron ore (BONAI) range. Iron ore occurs on the hill slopes in association with its parent rock. The hill in this part rises to about 900 m above the mean sea level

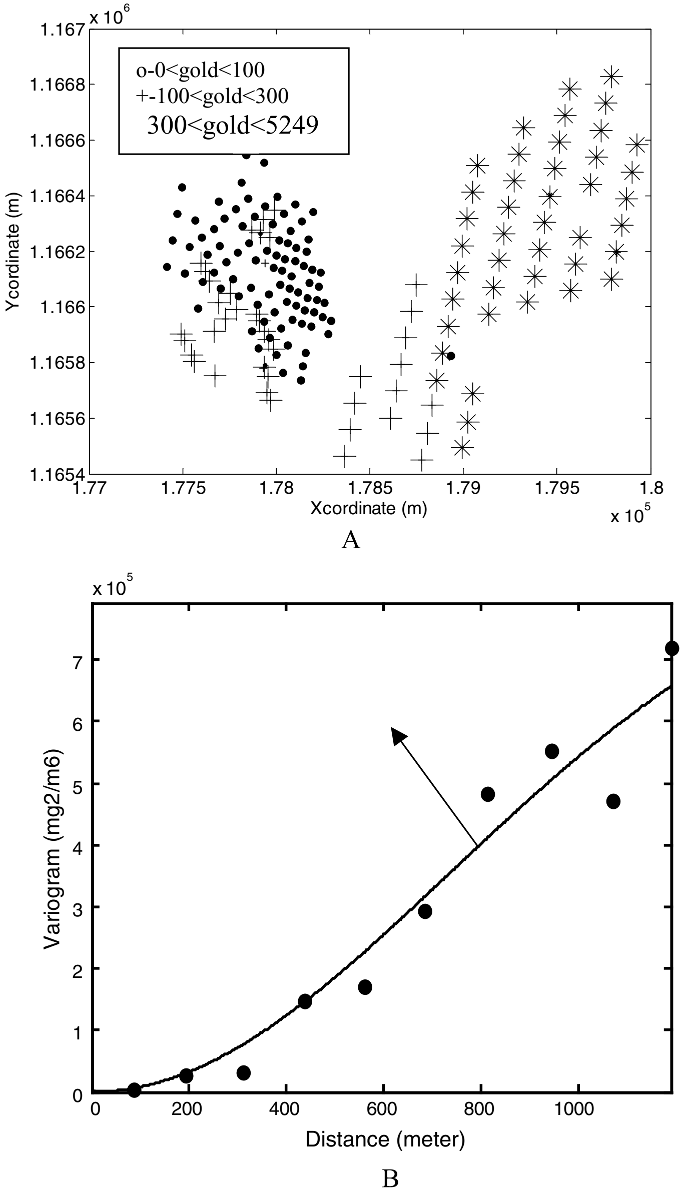
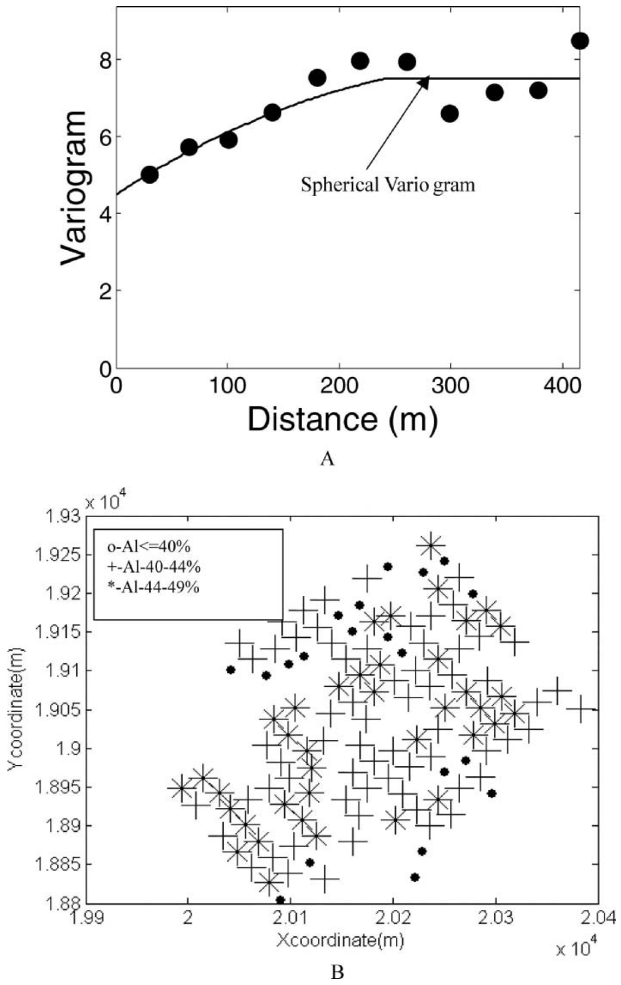


Figure 7. Spatial variability of gold concentration A, for spatial plot and B, omnidirectional variogram.



A

B

Figure 8. Spatial variability of Al_2O_3% A, for spatial plot and B, omnidirectional variogram.

and about 400 m above the local valley. The ore was formed about 3100 million years ago through metasomatism of marine volcanic sediments. The parent rock consists of banded hematite quartzite (BHQ), banded hematite jasper (BHJ), and laterite. There are 39 exploratory borehole information used for this study. The average spacing of the boreholes is 100 m along the strike direction. The spacing is somewhat lesser in the dip direction. The Fe% is the main constituent for the

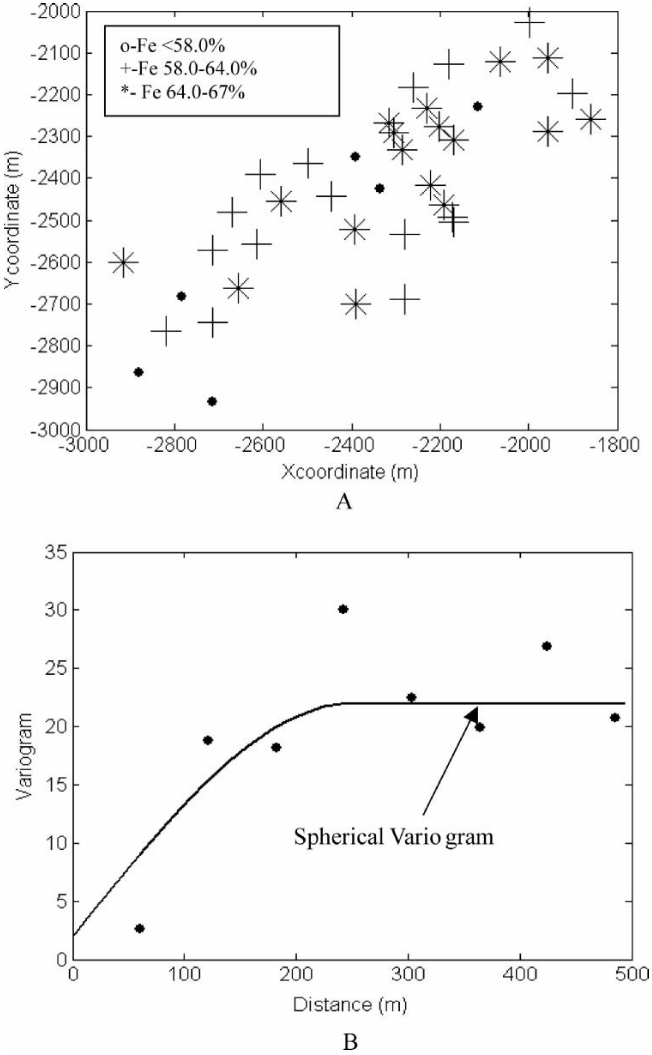


Figure 9. Spatial variability of Fe% A, for spatial plot and B, omnidirectional variogram.

iron ore deposit and was considered in this study. Figure 9 presents the spatial plot and omnidirectional variogram of Fe%.

Table 1 presents the summary statistics of the three data sets. The gold and iron data sets reveal high variance, whereas the bauxite data set shows low variance compared to other two data sets.

Table 1. Summary Statistics of the Data Sets

Data	Mean	Variance	Maximum	Minimum
Gold	507.26	1,067,400	5120	0.60
Bauxite	42.58	6.89	48.9	33.50
Iron	61.46	40.39	66.82	37.52

NEURAL NETWORK FOR GRADE ESTIMATION

For grade estimation using neural network, northing and easting coordinates were used as input variables, and grade attribute was used as an output variable for the respective data sets. For example, output variables for gold, bauxite, and iron data were gold (mg/m^3), Al_2O_3%, and Fe%, respectively. The complex spatial structure between input and output patterns is captured through a network via a set of connection weights, which are adjusted during training of the networks. The network captures an input–output relationship through training and acquires certain prediction capability so that for a given input (northing and easting coordinates) the network produces an output (grade).

Separate neural networks were used for each of the data sets. The network architecture used for the neural network application was a Ward-net network. Ward-nets are predefined architectures in the neural network software Neuroshell2. The advantage of using the Ward-net architecture is that this type of network is able to employ different activation functions in hidden layers. As a result, complex nonlinear input–output pattern is captured by a combination of multiple hidden units with different activation functions. Although the same network architecture

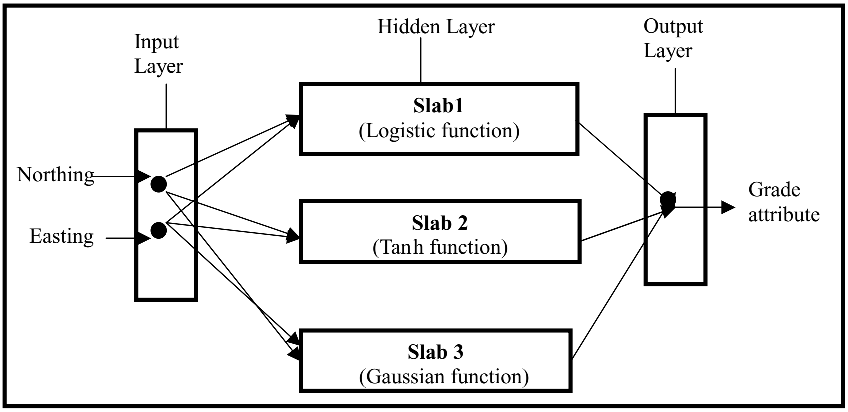


Figure 10. Ward-net architecture of neural network for grade estimation.

was used for all the data sets; the number of hidden units was varied for the different data sets. A combination of Logistic, Tanh and Gaussian activation functions were used in the hidden layers of the network. The Ward-net architecture used in this study has three layers: the input layer, the hidden layer, and the output layer (Fig. 10). The hidden layer consists of three hidden slabs of Logistic, Tanh, and Gaussian activation functions. Number of nodes in the hidden slabs varied for different data sets. For the gold data set, nine hidden neurons were used with three neurons in each slab. For the bauxite data set, 15 hidden neurons were used with 5 neurons in each slab. For the iron ore data set, nine hidden neurons were used with three neurons in each slab.

RESULTS

Prior to applying the neural network model, the input and output data values were normalized in the [0,1] scale. Therefore, the error presented is based on the normalized transformed data. The neural network model was trained using the four local and two global learning algorithms mentioned earlier. The starting point for all the learning algorithms, local and global, was same. The behavior of the local and global learning algorithms is extensively studied. However, since the properties of local learning algorithms are discussed at length in neural network literature, only their performance is reported in this present study to provide a comparitive evaluation of various algorithms. Table 2 presents the best solutions found for the four local algorithms. These solutions were obtained after running each of the learning algorithms for 50,000 epochs. From the table, it can be observed that the LMBP algorithm provided superior performance for all the data sets. The minimum mean squared errors as well as number of iterations to reach them were minimum for the LMBP algorithm. Quickprop learning algorithm shows an improved performance for the bauxite and iron ore data sets when compared to SBP and MBP algorithm, however, for the gold data set its performance is slightly inferior.

In applying the NOVEL algorithm to this study, three stages in the global search phase were used. The algorithm starts with the autonomous variable, $t = 0$, and at each iteration, t is changed to $t + \delta t$, where δt is chosen as 0.8. After trial and error with various combinations of μ_g, and μ_t (Eq. 4), the values for $\mu_g = 0.01$ and $\mu_t = 1$, were selected for all the data sets. These coefficients were kept constant for all the three stages in this analysis. The gradient of error for each hidden unit was calculated using chain rule, as it is a common practice in standard back-propagation algorithm. All the three stages in global search phase were executed for each time unit. The algorithm was run for 100 iterations in global search phase. Initial points for the local search phases were selected from the trajectory of the three stages, because each trajectory identifies new starting points, which may

Table 2. Performance of Local Learning Algorithms

Data set	Learning algorithm	Minimum MSE	Epoch number at minimum MSE
Gold	SBP	0.0041	11,408
	MBP	0.0041	11,637
	Quickprop	0.0061	6275
	LMBP	0.00017	2198
Bauxite	SBP	0.0265	15,370
	MBP	0.0265	15,236
	Quickprop	0.0251	58
	LMBP	0.020	156
Iron	SBP	0.026	19,004
	MBP	0.027	19,253
	Quickprop	0.0132	357
	LMBP	0.012	4034

lead to better local minima. Further, instead of using a single minimum point from each trajectory, the best solutions for periodic time intervals were chosen as initial points. This strategy generated many initial points from which local searches were initiated. In this study, 5 best initial points at the intervals of 20 iterations were selected from each trajectory, which altogether generated 15 initial points. An attempt was also made to capture initial points at intervals of 10 iterations; however, results did not improve. After selecting the initial points using global phase, the local search strategy was started for each of the initial points. LMBP algorithm was used for the local search phase, since earlier results on the data sets revealed improved performance of LMBP algorithm.

In experimenting with simulated annealing algorithm, a cooling scheduling was first verified. After trying with different schedule parameters, the following parameters were selected for the three data sets. The initial temperature T_0 was chosen as 20. This temperature decreased exponentially at the rate of $T_i = 0.9T_{i-1}$. For each temperature setting, the algorithm performed a number of iterations. Changing of one temperature to another temperature value was executed based on two conditions: (i) number of iterations reached 200 or (ii) number of transitions of the solutions exceeded 60. The first condition was imposed since at lower temperature acceptance of new solutions is very low. The algorithm was stopped when the number of temperature changes reached 250.

Table 3 presents the performance of the two global learning algorithms along with the LMBP algorithm (best local learning algorithm) for the three data sets. The results indicate that both global and local learning algorithms performed almost equally well for all the data sets. Therefore, the benefit of using global learning algorithm was not quite evident from the neural network modeling of the three data sets. One of the possible reasons might be that the smooth error surface trajectory

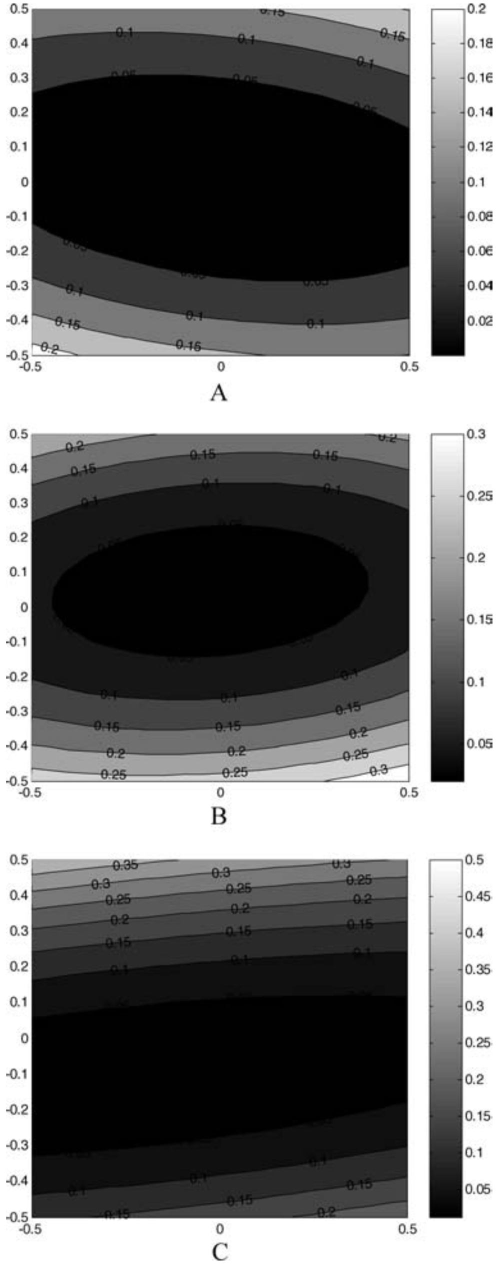
Table 3. Performance of NOVEL, SA, and LMBP Algorithms

Data set	Learning algorithm	Minimum MSE
Gold	LMBP	0.00017
	NOVEL	0.00017
	SA	0.00021
Bauxite	LMBP	0.020
	NOVEL	0.018
	SA	0.020
Iron	LMBP	0.011
	NOVEL	0.011
	SA	0.010

spanned by neural network modeling in the weight space. The error surface might not be as rugged as revealed in many applications of neural network modeling cited in the literatures. To delve further into this issue, a study on error surface was conducted. Because of high dimensionality of neural network modeling, it was not possible to display the error surfaces along all the dimensions. Instead, error surface along different pairs of dimensions was studied. An error surface was generated along selected pair of dimensions by clamping other dimensions at some fixed values. For example, Figure 11 presents an error surface along two arbitrary dimensions for the three data sets. The surface is around a solution found by the LMBP algorithm. From the figure, it can be seen that error surface around the solutions for all the cases have very smooth error surface, therefore, local learning algorithm efficiently reaches the bottom of the error surface.

Further an analysis of neural network model fitting is presented in the Table 4. This analysis was performed by back-transforming the normalized output of the data to the original scale. The results show that the mean squared error for the gold data set is relatively high. This is expected because of the high variance of the gold data. In fact, mean squared error for the gold data set is 0.4% of the total variance. The R^2 value for the gold data set is also very high, which shows the ability of neural network model to capture spatial relation between input and output variables. For the bauxite data set, the neural network model shows relatively low mean squared error, probably because of low variance of the bauxite data; however, the R^2 value is poor. This is because of poor spatial correlation of the bauxite data set within the study area. The variogram model presented in Figure 8 shows a high nugget component when compared with the regional component. Our earlier investigation (Samanta, Ganguli, and Bandopadhyay, 2005) on this data set using kriging technique also resulted in poor R^2 value. For the iron ore data set, neural network model was a better fit.

In this study no attempt was made to ensure generalization of the neural network. Therefore, the results shown in Tables 2–4 need not reflect the absolute



A

B

C

Figure 11. Projected two-dimensional error surface of neural network model around a solution found by LMBP algorithm, for A, gold data, B, Bauxite data, and C, Iron data.

Table 4. Summary Statistics of Neural Network Model Performance Using Three Algorithms

	Gold			Bauxite			Iron		
Statistics	LMBP	NOVEL	SA	LMBP	NOVEL	SA	LMBP	NOVEL	SA
Bias	0.13	0.13	1.83	0.12	0.09	0.06	−0.62	−0.62	0.05
Mean absolute error	43.20	43.20	53.69	1.76	1.58	1.66	2.42	2.42	2.19
Mean squared error	4368	4368	5586	4.86	4.40	4.89	11.35	11.35	9.14
R^2	0.9959	0.9959	0.9948	0.29	0.36	0.28	0.72	0.72	0.76

performance of the neural networks. The tables should be used only as a means of comparison across algorithms with the added restriction that the algorithm convergence was not uniform, i.e., convergence was governed by criteria pertinent to the individual methods.

CONCLUSION

In this study, the behavior of various optimization learning algorithms in neural network modeling for ore grade estimation was studied. The local and global learning algorithms performed equally for all the three data sets. Therefore, no significant benefit could be attributed to global learning algorithms in these specific data sets. A reason as to why the local and global learning algorithms perform equally well is that the error surface found in neural network modeling of these data sets was very smooth and possibly unimodal in nature. However, the superiority of global learning algorithm was clearly demonstrated on a multimodal nonlinear minimization problem. Although, neural network modeling of these data sets did not provide any basis for selecting global learning algorithms, it is suggested that one should apply global learning algorithms as a safeguard in order to avoid trapping in local region for neural network modeling. This will at least boost the confidence of a modeler that the network is not trapped in local minima, since many real-life applications of neural network modeling had problems of local minima. On the other hand, use of global learning will require more computational time.

REFERENCES

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

Gallagher, M. R., 1999, Multi-layer perceptron error surfaces: Visualization, structure and modeling: Unpublished PhD dissertation, University of Queensland, 225 p.

Hagan, M. T., Demuth, H. B., and Beale, M., 1996, Neural network design: PWS Publishing Company, Boston, 19 chapters.

Haykins, S., 1999, Neural networks: A comprehensive foundation, 2nd ed.: Prentice Hall, New Jersey, 824 p.

Ke, J., 2002, Neural network modeling of placer ore grade spatial variability: Unpublished Doctoral Dissertation, University of Alaska Fairbanks, 251 p.

Kirkpatrick, S., Gerlatt, C. D., Jr., and Vecchi, M. P., 1983, Optimization by simulated annealing: Science, v. 220, p. 671–680.

Koike, K., and Matsuda, S., 2003, Characterizing content distributions of impurities in A limestone mine using a feed forward neural network: Nat. Resour. Res., v. 12, no. 3, p. 209–223.

Koike, K., Matsuda, S., and Gu, B., 2001, Evaluation of interpolation accuracy of neural kriging with application to temperature-distribution analysis: Math. Geol., v. 33, no. 4, p. 421–448.

Koike, K., Matsuda, S., Suzuki, T., and Ohmi, M., 2002, Neural network-based estimation of principal metal contents in the Hokuroku district, Northern Japan, for exploring Kuroko-type deposits: Nat. Resour. Res., v. 11, no. 2, p. 135–156.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., 1958, Equations of state calculations by fast computing machines: J. Chem. Phys., v. 21, p.1087–1092.

Pincus, M., 1970, A Monte Carlo method for the approximate solution of certain types of constrained optimization problems, Oper. Res., v. 18, p. 1225–1228.

Rizzo, D. M., and Dougherty, D. E., 1994, Characterization of aquifer properties using artificial neural networks: Neural kriging: Water Resour. Res., v. 30, no. 2, p. 483–497.

Samanta, B., Bandopadhyay, S., and Ganguli, R., 2004, Data segmentation and genetic algorithms for sparse data division in Nome placer gold grade estimation using neural network and geostatistics: Mining Exploration Geol., v. 11, nos. 1–4, p. 69–76.

Samanta, B., Ganguli, R., and Bandopadhyay, S., 2005, Comparing the predictive performance of neural networks with ordinary kriging in a bauxite deposit: Transactions of Institute of Mining and Metallury, v. 114, p. 129–139.

Shang, Yi, and Wah, B. W., 1996, Global optimization for neural network training: IEEE Comput., v. 29, no. 3, p. 45–54.

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

Wu, X., and Zhou, Y., 1993, Reserve estimation using neural network techniques: Comput. Geosci., v. 19, no. 4, p. 567–575.

Yama, B. R., and Lineberry, G. T., 1999, Artificial neural network application for a predictive task in Mining: Mining Eng., v. 51, no. 2, p. 59–64.