NEURAL NETWORKS VS. MULTIPLE LINEAR REGRESSION FOR ESTIMATING PREVIOUS DIAMETER

Susan L. King[1]

Abstract—A neural network is a nonparametric statistical modeling procedure known for its capacity to process nonlinear relationships. For estimating the previous diameter of a tree, the exact functional relationship between the response variable and the independent variables is unknown. The relationship is most likely nonlinear. Multiple linear regression was used to develop a model for estimating the previous diameter of trees in West Virginia. The data were split into a model data set with 8,723 observations and a validation data set with 7,951 observations. The dependent variable was either basal-area increment or diameter increment. Two different sets of independent variables were evaluated. The data were divided into six species groups based on the rank of the average diameter growth of the species. Basal-area increment was a superior dependent variable for the multiple linear regression model. Basal-area increment is a nonlinear transformation of the diameter increment. It was thought that neural networks with its capacity to capture nonlinear relationships might provide an equivalent or superior solution with the diameter increment response variable as opposed to the basal-area increment response variable. All of the basal-area increment models had a higher R^2 than their diameter increment counterparts. Neither technique was superior in all cases. Other issues such as the stopping criteria, initial weight selection, the optimal number of hidden nodes, and the optimal number of hidden layers in a neural network are also discussed.

INTRODUCTION

The goal of a neural network is to mathematically model the brain and to capture its pattern recognition capabilities. Humans are more efficient at processing pattern information such as speech and visual images than any machine, whereas computers are extremely fast at processing information that can be formulated into a sequence of instructions.

A neural network may provide a superior solution over a traditional statistical approach for certain classes of problems (Burke, 1991). These classes include problems in which the distributions are unknown and possibly nonlinear, where outliers may exist, and where noise is present in the data. These are common conditions in forest inventory data. This paper investigates whether neural networks provide improved estimates over the traditional statistical modeling procedure of multiple linear regression for estimating diameter of a tree at an earlier time period.

In multiple linear regression, the relationship between independent and dependent variables is assumed to be linear and interactions among the independent variables must be specified in advance by the user. In neural networks, there is no assumed relationship between the independent and dependent variables. The relationship between independent and dependent variables and the interactions among the independent variables are learned through an iterative process. Neural networks require no assumptions about the distributions, mean, or correlation of the errors.

PROBLEM

The Northeastern Forest Inventory and Analysis (NEFIA) unit of the USDA Forest Service currently uses trees measured at current and previous inventories to develop multiple linear regression equations to estimate previous diameter for those trees not recorded at the previous inventory. The impetus for this study was to develop an improved procedure to estimate a previous diameter for ongrowth and nongrowth trees on plots sampled by variable radius plot sampling. However, the models built can be used whenever a previous diameter is required and the appropriate dependent variables are available. For West Virginia, King and Arner (1998) developed a new procedure to estimate a previous diameter. They used six groupings based on the rank of the average diameter growth for a species. Also in their study, many different independent variables and combinations of independent variables were evaluated. To investigate whether neural networks can provide a better estimate of previous diameter, the best models from the King and Arner study were selected for comparison.

DATA

The data for this project came from 1,965 remeasured forest inventory plots in West Virginia. The trees were initially measured in 1975 and then remeasured in 1988. Only trees larger than 5 inches in diameter at both time periods were included. The data were randomly split into a model building data set with 8,723 observations and a validation data set with 7,951 observations. Before splitting the data, they were grouped into six species groups. There were two sets of independent variables chosen for comparison. The first set of variables are those currently used by NEFIA to estimate the previous diameter. These variables are:

DBH2 = tree diameter at time period 2;

TRCLS2 = tree class at time period 2, a measure of tree quality;

[1] Operations Research Analyst, USDA Forest Service, 100 Matsonford Road, 5 Radnor Corp Ctr., Ste. 200, Radnor, PA 19087-4585.

Citation for proceedings: Stringer, Jeffrey W.; Loftis, David L., eds. 1999. Proceedings, 12th central hardwood forest conference; 1999 February 28-March 1-2; Lexington, KY. Gen. Tech. Rep. SRS-24. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 293 p. [Peer-reviewed paper].

CRNCLS2 = crown class at time 2, a measure of crown position in the canopy;

CRATIO2 = crown ratio at time 2, the proportion of a tree with a live crown;

CRCC2 = CRATIO2 / CRNCLS2;

DCR2 = DBH2 • CRATIO.

The analysis by King and Arner showed that the addition of the variable BAL2 improved the results. BAL2 is the sum of the basal areas of the trees on a plot larger than the subject tree. It is a measure of the competition for light. The addition of the variable BAL2 to the first set of independent variables formed the second set of independent variables. Two response variables were investigated: diameter increment (DI) and basal area increment (BAI).

$$\mathrm{DI} = \frac{(\mathrm{DBH2} - \mathrm{DBH1})}{\mathrm{N}} \qquad (1)$$

or

$$\mathrm{BAI} = \frac{\mathrm{K} \cdot (\mathrm{DBH2}^2 - \mathrm{DBH1}^2)}{\mathrm{N}} \qquad (2)$$

where:

N = number of years between measurements on the plot;

DBH1 = tree diameter at time period 1;

DBH2 = tree diameter at time period 2;

K = 0.005454154, a conversion factor from diameter in inches to basal area in square feet.

Annual increment was used to account for variation in the measurement period among the plots. Basal area and dbh are related by a transformation. Logic would suggest that BAI would be a better response variable than DI. There is not a one-to-one mapping between DI and BAI. A poletimber and a sawtimber tree may have the same DI, but different diameters at both measurement periods. BAI captures the differences in the size of the trees. Thus, 24 models were compared for both multiple linear regression and neural networks.

NEURAL NETWORKS

The type of a neural network chosen for this study is a feedforward backward propagation network (Figure 1). The network consists of three layers: the input layer, hidden layer, and output layer. The layers consist of processing units called nodes. Arcs connect the layers. Each arc has a weight which represents the strength of the connection. The goal of a neural network is to find the best estimate of the weights.

In the input layer, the number of nodes corresponds to the number of independent variables. In the hidden layer, not only is the number of nodes variable, but also there may be more than one hidden layer. Only one hidden layer is shown in Figure 1. In general, only one hidden layer is required. The third layer is the output layer. The number of nodes in this layer corresponds to the number of dependent variables. A backpropagation network has no cycles. All of the arcs move from left to right.

Each observation in the data set forms an input pattern, p. An observation is called an exemplar in neural networks. A linear combination of the input patterns and the weights is formed at each hidden node. This defines a plane in N - 1 dimensional space, where N is the number of input nodes. The hyperplane passes through the origin unless a bias weight is added to the hidden node. Mathematically, this process is represented as:

$$\mathrm{net}_j^p = w_j + \sum_{i=1}^{I} w_{ij} x_i^p \qquad \text{for } j=1,\ldots,J \qquad (3)$$

where:

w_j = the bias term for hidden unit j;

w_{ij} = the weight from input node i to hidden node j;

x_i^p = ith component of the pth exemplar;

I = number of input nodes;

J = number of hidden nodes.

By creating a dummy node with a fixed input value of 1 or -1, the bias can be written as a weight, $w_{I+1,j}$. Thus, equation (3) becomes:

$$\mathrm{net}_j^p = \sum_{i=1}^{I+1} w_{ij} x_i^p \qquad \text{for } j=1,\ldots,J. \qquad (4)$$

A squashing or activation function is applied to net_j^p at each hidden node j. This function introduces nonlinearities into the network. Common squashing functions are the logistic and hyperbolic tangent function. Applying the squashing function to net_j^p yields:

$$y_j^p = f(\mathrm{net}_j^p) \qquad \text{for } j=1,\ldots,J. \qquad (5)$$

The output from each of the j nodes at the hidden layer becomes the input to the k output nodes.

A linear combination of the output from the hidden nodes and the weights, v_{jk}, is formed. As before, a bias term is added. Mathematically, this may be expressed as:

$$\mathrm{net}_k^p = \sum_{j=1}^{J+1} v_{jk} y_j^p \qquad \text{for } k=1,\ldots,K \qquad (6)$$

where:

v_{jk} = the weight from hidden node j to output node k.

A squashing function is applied to net_k^p to obtain the predicted output:

$$o_k^p = f(\mathrm{net}_k^p) \qquad \text{for } k=1,\ldots,K. \qquad (7)$$

Backpropagation is a supervised procedure. That is, it requires an observed dependent variable, t_k^p. The estimated value, o_k^p, is compared with t_k^p, to determine if they are close. One measure of closeness is the sum of the

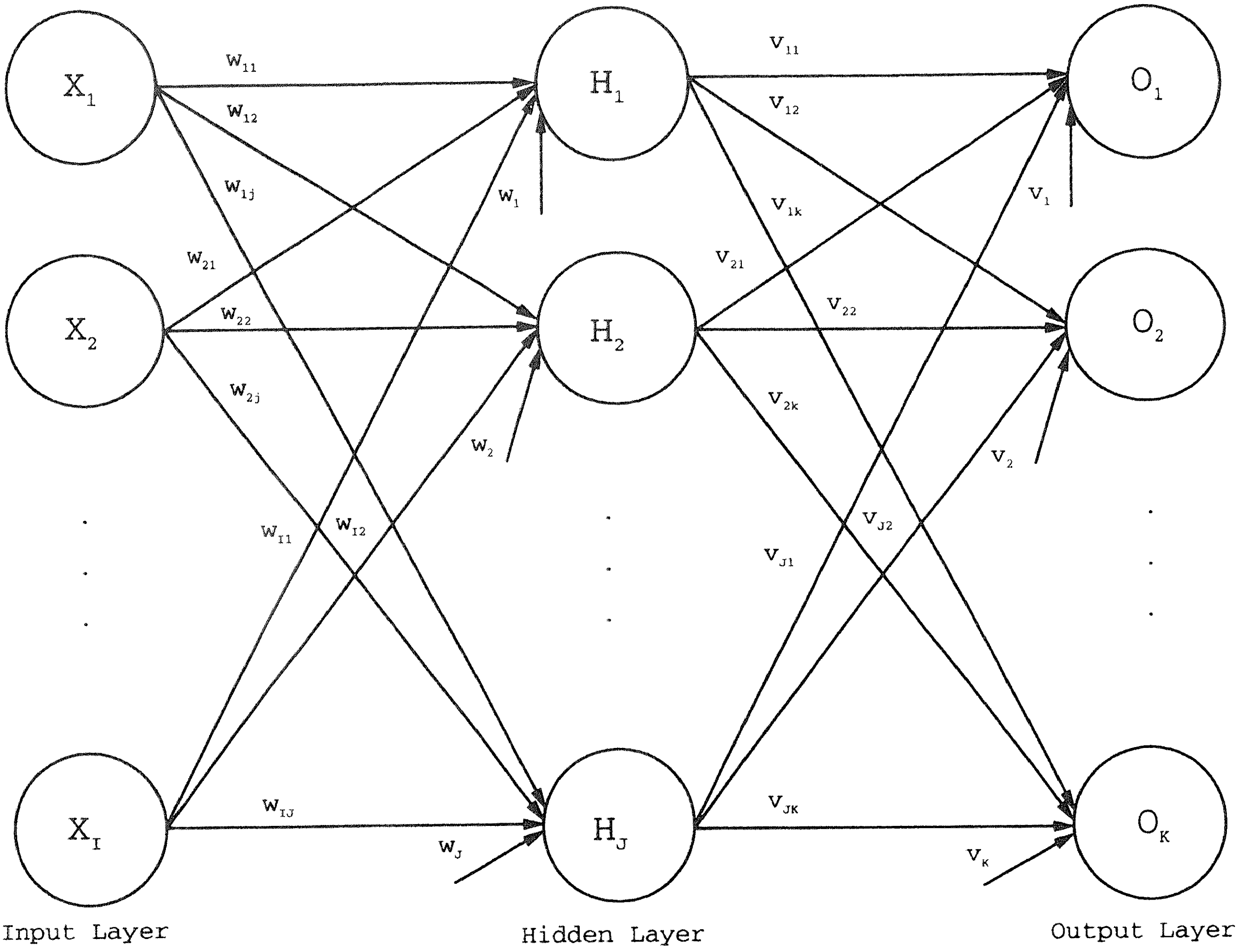

Figure 1—Backpropagation network. The network is fully connected. There is an arc from every node to a node in the next layer.

squared differences between t_k^p and o_k^p. It is used frequently because the derivatives are easy to compute. Thus, the objective function is:

$$\mathrm{Min} \sum_{p} \sum_{k=1}^{K} (o_k^p - t_k^p)^2 . \qquad (8)$$

The objective function for neural networks in equation (8) appears to be similar to that in multiple linear regression. Both techniques minimize the sum of the squared differences between the observed and the predicted values. The variables in both procedures are the weights. However, the two procedures are quite different. The predicted values are different functions of the weights. The number of weights in multiple linear regression depends on the number of input variables, whereas the number of weights in neural networks depends on the number of: input variables, hidden nodes, hidden layers, and output nodes. The objective function in both the neural network and multiple linear regression is an unconstrained minimization problem. The special structure in the multiple linear regression problem allows for the optimal set of weights to be found through solving a system of normal equations. Iterative techniques are used to find the optimal set of weights for a neural network. Each iteration is considered a training period. By updating the weights, the neural network is said to be learning.

Many techniques are available for solving unconstrained minimization problems. These techniques include gradient descent, the quasi-Newton techniques of conjugate gradients and Davidon-Fletcher-Powell, the modified Newton technique of Levenberg and Marquardt, stiff differential equations, genetic optimization, and simulated annealing. Historically, gradient descent has been applied by the neural network community to solve equation (8). In fact the name backpropagation refers to the process of applying the chain rule of calculus to compute the error gradient for each weight in the network. The error gradient is used in updating the weights in gradient descent. The error is said to be propagated backwards. Gradient descent may be advantageous if the problem is implemented on a parallel computer. However, most problems are implemented on a serial computer, and gradient descent on these machines has been abandoned by the optimization community in favor of more sophisticated techniques. The difficulties of gradient descent are well documented. Sarle (1994), Masters (1995), Bishop (1995), and Bazarra, Sherali and Shetty (1993) all discuss the limitations of gradient descent and are excellent references on the conjugate gradient method, the Davidon-Fletcher-Powell, and the Levenberg-Marquardt algorithm. Kollias and Anastassiou (1988) discuss applying the Levenberg-Marquardt algorithm to neural networks. Owens and Filken (1989) saw the similarity between a system of stiff differential equations and the gradient descent approach. They claim that stiff differential equations provide a more rapid and accurate convergence than either gradient descent or conjugate gradient methods. Hassoun (1995) provides an introduction to simulated annealing and genetic optimization for neural networks. Masters (1995) is another good introductory reference for simulated annealing in neural networks. Another text by Masters (1993) provides introductory material on both topics.

In selecting the technique to solve equation (8), the number of weights must be taken into account. Sarle (1994) recommends using the Levenberg-Marquardt algorithm for networks with tens of weights, the Davidon-Fletcher-Powell algorithm for networks with hundreds of weights, and the conjugate gradient procedure for large problems with thousands of weights. Most of the 24 subproblems were solved using the Levenberg-Marquardt algorithm. These subproblems had only tens of weights. The other two techniques were tried on a few subproblems, but the value of their objective functions was larger. Gradient descent was also tried on several of the subproblems. It proved to be a superior technique for only the response variable DI in the first subgroup. The gradient descent algorithm was user written in SAS/IML(SAS Institute Inc., 1989). The other three algorithms are part of PROC NLP (SAS Institute Inc., 1997) in SAS/OR. SAS had a Beta release macro that was available in Release 6.10. This macro was modified and used in this study. Updated SAS neural network macros with a GUI interface are now part of the SAS Data Mining Solution.

Several other issues relating to the implementation of the neural network need to be discussed. First, the activation function must be selected. The purpose of an activation function is to induce nonlinearity into the network through a nonlinear transformation. With a linear function, the output is a weighted sum of the inputs. A squashing function is an activation function that maps any real input into a bounded range, usually between 0 and 1 or between -1 and 1. The two most common squashing functions are the logistic function and the hyperbolic tangent function. Other functions may be used so long as they are differentiable. Smooth activation functions decrease the training time. Kalman and Kwasny (1992) argue that the hyperbolic tangent function was the best of the sigmoidal functions. Bishop (1995) states that the hyperbolic tangent function often increases algorithm convergence over a logistic function. The choice of an activation function may be different at the output node. The range of the activation function at the output node should correspond to the range of the dependent variable. A categorical dependent variable would have a different activation function than a continuous output variable. The output data for the diameter increment problem were scaled. The hyperbolic tangent function was used at both the hidden and output layers. The logistic function was tried on some of the subproblems and it did not provide a superior solution.

Second, there are two different philosophies concerning the scaling of the independent and dependent variables. Some believe that it is not necessary to scale the independent or dependent variables. The size of the weights will make any necessary adjustments. In the diameter increment problem, the input and output variables were scaled to values that correspond to the range of the squashing function. Because the hyperbolic tangent function is used as the squashing function, the continuous variables are scaled between -0.9 and 0.9. The endpoints, -1 and 1, in the range are not used because they correspond to the inputs $-\infty$ and $+\infty$, respectively. The class variables are first broken into indicator variables and then scaled like the continuous variables. The lowest value of the indicator variable corresponds to -0.9 and the largest value corresponds to 0.9.

Third, the selection of the initial weights is a major issue. None of these techniques guarantee a global minimum. The choice of initial weights can influence the quality of a local solution. The initial weights in this project were selected by random numbers. There are many heuristic procedures for selecting the initial weights. One procedure by Piovoso and Owens (1991) was tried on several subproblems. In this procedure, the weights between the input layer and the hidden layer are found by principal component analysis. The weights between the hidden layer and the output layer are found by multiple linear regression. In this project, after trying several different seeds, a random number generator always provided a set of weights that found a lower value to the objective function than the Piovoso and Owens procedure.

Fourth, the number of layers must be selected. Hornik, Stinchcombe, and White (1989) showed that a neural network with one hidden layer and an arbitrary squashing function can approximate most functions. In practice, the need for a second hidden layer occurs when a piecewise-continuous function must be approximated. This condition is not present in the diameter or basal-area growth problem, so only one hidden layer was used.

Fifth, there are no equations or formulas for selecting the optimal number of hidden nodes. Each situation is different. Too many hidden units cause an inability to generalize and the data are overfitted. Similarly, too few hidden nodes will cause an underfitting of the data. An underfitted or an overfitted model does not generalize well, that is, predict accurate dependent or output variables from a new set of independent or input variables. One way to control generalization is through the selection of the number of hidden units and their connections. This is model selection. The simplest model selection technique is to determine the optimal number of hidden nodes through experimentation. Starting with one or two nodes, a solution is found. Another node is added and the problem is reoptimized. This process is repeated until the objective function value begins to increase. The number of hidden nodes corresponding to the minimal objective function value is optimal. This procedure was used in the diameter increment model. In most of the 24 problems, two hidden nodes were optimal. An alternative is to start with a large number of hidden nodes and gradually remove complete hidden units or only remove selected connections. This is pruning. Reed (1993) describes several pruning procedures.

Sixth, in any iterative algorithm, a major issue is when to stop. The error in the model data set monatonically decreases as a function of the iteration number. The error in the validation data set decreases and then increases as the neural network starts to overfit. The algorithm is terminated when the validation data set reaches its minimum. This procedure is called stopped training. Regularization procedures such as stopped training improve generalization by controlling the size of the weights. Other regularization procedures such as weight decay, training with noise, and Bayesian estimation are described by Bishop (1995). Stopped training was used in this project.

COMPARISON STATISTICS

The statistics used to compare multiple linear regression with neural networks for both the model and the validation data set are R^2, the mean of the squared errors (MSE), the mean of the absolute errors (MAE), and the mean of the arithmetic errors (ME). The ME indicates bias, whereas the MSE and the MAE both indicate precision as well as bias. The MAE is more robust and less sensitive to outliers than the MSE.

RESULTS AND DISCUSSION

A comparison of the results between neural networks and multiple linear regression for the ranked mean species groups is presented in Tables 1 and 2. The overall results were obtained by combining the response values and the predicted response values for each of the six species groups and forming one group. The appropriate statistics are then calculated. Because the goal of a model is generalization, the results for the validation data set are more important than those for the model data set.

For the BAI models, all of the results are expressed as DI using the translation:

$$\widehat{\mathrm{DI}}(\mathrm{BA}) = \frac{\sqrt{\dfrac{(\mathrm{BA1} + \mathrm{N} \cdot \widehat{\mathrm{BAI}})}{\mathrm{K}}} - \mathrm{DBH1}}{\mathrm{N}} \qquad (9)$$

where:

BA1 = tree basal area at time period 1, K • DBH1^2;

BÂI = predicted basal-area increment;

N = number of years between measurements of the tree;

All of the statistics use DÎ(BA) as the predicted diameter increment for the basal-area models.

From the overall results for the response variable DI, neural networks was superior. It had a slightly higher R^2, a slightly lower MAE, and a slightly lower MSE. For the individual species groups for the response variable DI, neural networks was superior for the NEFIA variables, but the results were mixed for addition of BAL2. Still, neural networks predominates. The ME was larger for the neural network model as expected. An assumption of multiple linear regression is that the expected value of the errors is zero. Neural networks, on the other hand, is a nonparametric procedure and the mean of the arithmetic error will not necessarily be zero. The addition of the variable BAL2 improved the results for both neural networks and multiple linear regression. For the response variable BAI, multiple linear regression more frequently provided a superior solution than neural networks as indicated by the R^2, MAE, and MSE.

The software used in this study was the first beta version of SAS's neural network macros. Later beta versions were significantly enhanced. A subset of the 24 subproblems was selected to access the impact of the new software on the diameter increment prediction problem. The subset had BAI as the response variable and NEFIA + BAL2 as the independent variables. Neural networks had the most difficulty with this subset. The results are in Table 3. With the new macros, neural networks is the winner for the model data set. The R^2 is higher for all species groups, and the MAE and MSE are lower for most of the species groups. However, the new macros did not improve the results for the validation data set. Only the three smallest species groups have a higher R^2 and a lower MAE in the validation data set as compared with the results in Table 2. Neural networks performed slightly worse for the third and sixth species groups. The new macros did not significantly alter the results; and so, it was decided not to pursue modeling the remaining subgroups. Also other independent variables from the King and Arner (1998) study were tried on this subset of the 24 problems. They did not improve the results.

Neural networks does not always outperform multiple linear regression. This conclusion was also reached by Desai and Bharati (1998). They found that for predicting excess returns on large stocks, neural networks outperformed multiple linear regression in periods of high volatility. Otherwise, multiple linear regression was superior. These results parallel a study by Markham and Rakes (1998). Using computer generated data, Markam and Rakes conclude that there is a significant interaction between sample size and variance. Multiple linear regression performs better for low-variance problems and neural networks performs better for high-variance problems. The results are mixed and dependent on sample size for medium-variance problems. Neural networks was superior for large sample sizes, and multiple linear regression was superior for small sample sizes. One explanation of why neural networks did not outperform multiple linear regression in this project is that there is not enough

Table 1—Comparison statistics for neural networks and mutiple linear regression for NEFIA variables

Sub-group	No. of trees	NN	REG	NN	REG	NN	REG	NN	REG
		----------R^2----------		----------ME----------		--------MAE--------		---------MSE---------	
				DI response variable and model data set					
1	257	0.0463	0.0449	-0.0051	0.0000	0.0334	0.0322	0.0017	0.0017
2	947	0.1099	0.0846	0.0004	0.0000	0.0409	0.0415	0.0027	0.0028
3	2489	0.1649	0.1568	0.0016	0.0000	0.0430	0.0434	0.0031	0.0032
4	2371	0.2359	0.2011	0.0019	0.0000	0.0541	0.0556	0.0049	0.0051
5	783	0.2449	0.2193	0.0004	0.0000	0.0557	0.0564	0.0050	0.0052
6	1876	0.2440	0.2116	0.0050	0.0000	0.0664	0.0689	0.0074	0.0077
All	8723	0.3731	0.3512	0.0020	0.0000	0.0517	0.0528	0.0046	0.0048
				DI response variable and validation data set					
1	97	0.0256	0.0011	-0.0020	0.0018	0.0335	0.0336	0.0017	0.0017
2	616	0.1106	0.0787	0.0011	0.0024	0.0419	0.0425	0.0029	0.0030
3	2417	0.1973	0.1832	0.0034	0.0001	0.0436	0.0436	0.0031	0.0031
4	2448	0.2262	0.1606	0.0013	0.0016	0.0544	0.0561	0.0049	0.0054
5	509	0.1184	0.1148	0.0105	0.0043	0.0620	0.0598	0.0062	0.0062
6	1864	0.2334	0.2015	0.0044	-0.0021	0.0686	0.0700	0.0077	0.0081
All	7951	0.3521	0.3216	0.0032	0.0005	0.0537	0.0545	0.0049	0.0051
				BAI response variable and model data set					
1	257	0.2481	0.2349	-0.0011	0.0011	0.0293	0.0288	0.0013	0.0013
2	947	0.2806	0.3155	0.0029	0.0004	0.0366	0.0359	0.0022	0.0021
3	2489	0.4005	0.3704	-0.0003	0.0012	0.0366	0.0374	0.0022	0.0024
4	2371	0.4679	0.4706	0.0007	0.0010	0.0456	0.0452	0.0034	0.0034
5	783	0.4371	0.4734	-0.0044	0.0021	0.0491	0.0464	0.0037	0.0035
6	1876	0.4353	0.5127	-0.0181	0.0022	0.0607	0.0540	0.0055	0.0048
All	8723	0.5418	0.5647	-0.0039	0.0013	0.0451	0.0435	0.0034	0.0032
				BAI response variable and validation data set					
1	97	0.1311	0.2036	0.0097	0.0022	0.0301	0.0300	0.0015	0.0014
2	616	0.3248	0.3153	-0.0043	0.0029	0.0374	0.0368	0.0022	0.0022
3	2417	0.3848	0.3921	0.0036	0.0168	0.0373	0.0373	0.0023	0.0023
4	2448	0.4539	0.4495	0.0018	0.0018	0.0455	0.0455	0.0035	0.0035
5	509	0.3751	0.4008	0.0051	0.0027	0.0508	0.0497	0.0044	0.0042
6	1864	0.4221	0.5143	0.0110	0.0014	0.0594	0.0551	0.0058	0.0049
All	7951	0.5222	0.5518	0.0044	0.0018	0.0458	0.0447	0.0036	0.0034

variability in the data. Another explanation is that the relationship between BAI and the independent variables is linear. In this situation, a neural network can not outperform a linear model.

CONCLUSIONS

Neither neural networks nor the multiple linear regression significantly outperformed the other technique. Neural networks followed the same trend as multiple linear regression. All of the statistics indicate significant improvement by using the response variable BAI instead of DI. The addition of the variable BAL2 also improved slightly both models as indicated by the statistics. King and Arner (1998) found that the addition or substitution of other variables had little impact on the model.

There is contradictory information in books and journals on neural networks. There is no consensus on the selection of the initial weights, model selection, regularization, and scaling of input and output variables. An effort has been made to try new architectures on a subset of the 24 models as they become available in SAS.

Table 2—Comparison statistics for neural networks and multiple linear regression for NEFIA variables + BAL2

Sub-group	No. of trees	NN	REG	NN	REG	NN	REG	NN	REG
		----------R^2----------		----------ME----------		--------MAE--------		---------MSE---------	
DI response variable and model data set									
1	257	0.0702	0.0671	-0.0003	0.0000	0.0323	0.0324	0.0016	0.0016
2	947	0.1476	0.1200	0.0002	0.0000	0.0397	0.0405	0.0026	0.0026
3	2489	0.1933	0.1925	0.0044	0.0000	0.0419	0.0425	0.0030	0.0030
4	2371	0.2649	0.2311	0.0011	0.0000	0.0529	0.0548	0.0469	0.0049
5	783	0.2696	0.2628	-0.0026	0.0000	0.0553	0.0550	0.0048	0.0049
6	1876	0.3198	0.2882	0.0017	0.0000	0.0628	0.0652	0.0067	0.0070
All	8723	0.4097	0.3907	0.0017	0.0000	0.0501	0.0513	0.0043	0.0045
DI response variable and validation data set									
1	97	0.0357	0.0515	0.0012	0.0014	0.0334	0.0330	0.0016	0.0016
2	616	0.1355	0.1231	0.0030	0.0010	0.0412	0.0413	0.0028	0.0028
3	2417	0.2076	0.2186	0.0092	-0.0001	0.0436	0.0426	0.0030	0.0030
4	2448	0.2241	0.1964	0.0015	0.0009	0.0547	0.0552	0.0049	0.0051
5	509	0.1578	0.1590	0.0064	0.0023	0.0594	0.0585	0.0059	0.0059
6	1864	0.2895	0.2672	0.0063	-0.0008	0.0664	0.0671	0.0072	0.0074
All	7951	0.3739	0.3612	0.0054	0.0003	0.0531	0.0530	0.0047	0.0048
BAI response variable and model data set									
1	257	0.2426	0.2549	-0.0037	0.0012	0.0300	0.0287	0.0013	0.0013
2	947	0.3350	0.3391	0.0020	0.0005	0.0351	0.0352	0.0020	0.0020
3	2489	0.4005	0.3876	-0.0003	0.0013	0.0366	0.0369	0.0022	0.0023
4	2371	0.4811	0.4782	0.0006	0.0012	0.0451	0.0452	0.0033	0.0033
5	783	0.4895	0.4914	0.0005	0.0023	0.0457	0.0455	0.0034	0.0034
6	1876	0.5282	0.5355	-0.0120	0.0027	0.0548	0.0528	0.0046	0.0046
All	8723	0.5782	0.5782	-0.0023	0.0016	0.0433	0.0429	0.0031	0.0031
BAI response variable and validation data set									
1	97	0.1623	0.2342	0.0093	0.0018	0.0296	0.0296	0.0014	0.0013
2	616	0.3599	0.3449	-0.0012	0.0016	0.0357	0.0359	0.0021	0.0021
3	2417	0.3848	0.4048	0.0036	0.0018	0.0373	0.0372	0.0023	0.0023
4	2448	0.4658	0.4619	0.0003	0.0013	0.0454	0.0454	0.0034	0.0034
5	509	0.4098	0.4161	0.0033	0.0019	0.0495	0.0494	0.0041	0.0041
6	1864	0.4655	0.5294	0.0171	0.0026	0.0566	0.0546	0.0054	0.0048
All	7951	0.5416	0.5634	0.0054	0.0019	0.0449	0.0444	0.0035	0.0033

This work represents another step in evaluating the power of neural networks. Finding a 'good' set of initial weights can be a time consuming process, and some thought must be given to the frequency of use and the required precision of the final model before abandoning traditional techniques.

Table 3—Comparison statistics for neural networks and multiple linear regression for DEFIA + BAL2 variables using an updated version of SAS macros

Sub-group	No. of trees	NN	REG	NN	REG	NN	REG	NN	REG
		----------R^2----------		----------ME----------		--------MAE--------		---------MSE---------	
				BAI response variable and response data set					
1	257	0.4828	0.2549	0.0010	0.0012	0.0229	0.0287	0.0009	0.0013
2	947	0.4104	0.3391	0.0022	0.0005	0.0330	0.0352	0.0018	0.0020
3	2489	0.4022	0.3876	0.0013	0.0013	0.0363	0.0369	0.0022	0.0023
4	2371	0.5030	0.4782	0.0018	0.0012	0.0436	0.0452	0.0032	0.0033
5	783	0.4932	0.4914	-0.0005	0.0023	0.0459	0.0455	0.0034	0.0034
6	1876	0.5634	0.5355	-0.0012	0.0027	0.0513	0.0528	0.0043	0.0046
All	8723	0.5991	0.5782	0.0008	0.0016	0.0416	0.0429	0.0029	0.0031
				BAI response variable and validation data set					
1	97	0.1964	0.2342	0.0094	0.0018	0.0278	0.0296	0.0014	0.0013
2	616	0.3665	0.3449	0.0030	0.0016	0.0349	0.0359	0.0021	0.0021
3	2417	0.3813	0.4048	0.0013	0.0018	0.0378	0.0372	0.0024	0.0023
4	2448	0.4659	0.4619	-0.0013	0.0013	0.0451	0.0454	0.0034	0.0034
5	509	0.4121	0.4161	0.0007	0.0019	0.0495	0.0494	0.0041	0.0041
6	1864	0.4619	0.5294	0.0247	0.0026	0.0555	0.0546	0.0054	0.0048
All	7951	0.5403	0.5634	0.0062	0.0019	0.0446	0.0444	0.0035	0.0033

REFERENCES

Bazaraa, Mokhtar S.; Sherali, Hanif D.; Shetty, C.M. 1993. Nonlinear programming. New York: John Wiley. 638 p.

Bishop, Christopher M. 1995. Neural networks for pattern recognition. New York: Oxford University Press. 482 p.

Burke, Laura Ignizio. 1991. Introduction to artificial neural systems for pattern recognition. Computers and Operations Research. 18(2): 211-220.

Desai, Vijay S.; Bharati, Rakesh. 1998. Annals of Operations Research. 78(1998): 127-163.

Hassoun, Mohamad H. 1995. Fundamentals of artificial neural networks. Camdridge, MA: The MIT Press. 511 p.

Hornik, Kurt; Stinchcombe, Maxwell; White, Halbert. 1989. Multilayer feedforward networks are universal approximators. Neural Networks. 2: 359-366.

Kalman, Barry L.; Kwasny, Stan C. 1992. Why tanh: choosing a sigmoidal function. Baltimore, MD: International Joint Conference on Neural Networks: 578-581.

King, Susan L.; Arner, Stanford L. [In press]. Estimating previous diameter for ingrowth trees on remeasured horizontal point samples. Proceedings 12th central hardwood forest conference; 1999 February 28 - March 3; Lexington, KY.

Kollias, Stefanos; Anastassiou, Dimitris. 1988. Adaptive training of multilayer neural networks using a least square estimation technique. In: Proceedings of IEEE international conference on neural networks I: 383-390.

Markham, Ina S.; Rakes, Terry R. 1998. The effect of sample size and variability of data on the comparative performance of artificial neural networks and regression. Computers and Operations Research. 25(4): 251-263.

Masters, Timothy. 1993. Pratical neural networks recipes in C++. San Diego: Academic Press. 493 p.

Masters, Timothy. 1995. Advanced algorithms for neural networks: a C++ sourcebook. New York: John Wiley. 431 p.

Owens, A.J.; Filkin, D.L. 1989. Efficient training of the back propagation network by solving a system of stiff ordinary differential equations. Washington, DC: International Joint Conference on Neural Networks II: 381-386.

Piovoso, Michael J.; Owens, Aaron J. 1991. Sensor data analysis using neural networks. San Padre Island, TX. Proceedings of the 4th international conference on chemical process control: 101-118.

Reed, Russell; Pruning Algorithms-A Survey. 1993. IEEE Transactions on Neural Networks. 4(5): 740-747.

Sarle, W.S. 1994. Neural network implementation in SAS software. Proceedings of the 19th annual SAS users group international conference. Cary, NC: SAS Institute Inc.: 1551-1573.

SAS Institute Inc. 1989. SAS/IML® Software: usage and reference. Version 6, 1st ed. Cary, NC: SAS Institute Inc. 501 p.

SAS Institute Inc. 1997. SAS/OR technical report: the NLP procedure. Cary NC: SAS Institute Inc. 146 p.