Neural Networks as Statistical Tools for Business Researchers

Kristen Bell Detienne, David H. Detienne and Shirish A. Joshi

Organizational Research Methods 2003; 6; 236
DOI: 10.1177/1094428103251907

Neural Networks as Statistical Tools for Business Researchers

KRISTEN BELL DeTIENNE
Brigham Young University

DAVID H. DeTIENNE
ORINCON-Sygenex

SHIRISH A. JOSHI
Morgan Stanley Dean Witter

Artificial neural networks are rapidly gaining popularity in the hard sciences and in social science. This article discusses neural networks as tools business researchers can use to analyze data. After providing a brief history of neural networks, the article describes limitations of multiple regression. Then, the characteristics and organization of neural networks are presented, and the article shows why they are an attractive alternative to regression. Shortcomings and applications of neural networks are reviewed, and neural network software is discussed.

Keywords: *artificial neural networks; multiple regression; prediction*

Why don't business researchers use neural networks? Artificial neural networks have been successfully employed for analysis and research in the hard sciences for decades. In organizational science, regression and associated techniques have been the widely used tools for testing and analyzing relationships among variables (Landis & Dunlap, 2000). However, in recent years, neural networks have been gaining popularity as statistical and decision tools in applied settings such as business and the social sciences (Al-Deek, 2001; Laguna & Marti, 2002; Neal & Wurst, 2001; Nguyen & Cripps, 2001; Phua, Ming, & Lin, 2001; Racine, 2001). In fact, in some areas of business such as data mining, neural networks have become the most widely used analysis technique (Chen & Sakaguchi, 2000).

There are several reasons for the growing popularity of neural networks. Modern computers are now powerful enough to run large, useful neural networks (Pethokoukis, 2002). Several powerful neural network software packages are now available from companies such as SAS, SPSS, and MATLAB (Gencay & Selcuk, 2001; Sarle, 2002). Furthermore, neural networks are capable of learning using real

Organizational Research Methods, Vol. 6 No. 2, April 2003 236-265
DOI: 10.1177/1094428103251907

data; hence, they do not need the a priori knowledge required by expert systems and regression (Littell, 2000; Mitchell, 2001). Artificial neural networks have also been shown to outperform multiple regression in data analysis in several problem domains (Garson, 1998). Also, neural nets offer capabilities beyond those of regression, such as the ability to deal with nonlinear relationships, missing data, and outliers (Somers, 2001). Finally, neural networks can be used in combination with regression, so researchers can take advantages of the strengths of both techniques.

This article discusses artificial neural networks in the context of statistical analysis and as an alternative to conventional multiple regression. A brief historical review of neural nets is followed by a description of the limitations of multiple linear regression. After discussing the characteristics and organization of neural networks, this article shows why they are an attractive alternative to regression. Then, some of the common applications and shortcomings of neural networks are analyzed. Finally, the major neural network software packages are discussed.

History of Neural Networks

The first major paper on neural networks was published in 1943 by McCulloch and Pitts. Their research showed that human brain functionality could be modeled mathematically and that a network of artificial binary-valued neurons could perform the calculations. In the early 1960s, Rosenblatt (1962) developed a learning algorithm for a model called the perceptron. Research interest in the field of neural networks continued to rise until the publication of a paper by Minsky and Papert (1969) that showed that neural networks possessed the ability to approximate only a limited number of functions. Practically speaking, this has turned out to be not much of a restriction.

A resurgence of interest in the area of artificial neural networks has been fueled by the development of the Hopfield model (Hopfield, 1982), the development of the backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986), and the exponential advances in computational capability. Currently, there is no practical limit on the types of functions a neural net can approximate.

Description of Neural Networks

A neural network may be envisioned as a highly connected structure of processing elements that attempts to mimic the parallel computation ability of the biological brain. When compared to a conventional linear computer, the biological brain may not be able to process serial computations as fast as the computer. However, in certain situations, the biological brain's parallel processing ability can abstract, process, and generalize data instantly, whereas a conventional computer would take massive amounts of time and in most situations be unable to understand the problem. By attempting to emulate the brain, an artificial neural network attempts to overcome the drawbacks of conventional computers.

Limitations of Regression

One attraction of neural networks arises from the many limitations of conventional multiple regression. Multiple regression has numerous disadvantages that hamper its effectiveness as a statistical tool. Many of these disadvantages are overcome by artifi-

cial neural networks. Some of the common faults of linear multiple regression analysis are detailed here.

Linear in Nature

The inability to deal effectively with nonlinearity is a critical drawback of multiple linear regression (Marquez et al., 1991). Compounding the problem is that multiple linear regression returns no direct indicator as to whether the data is best presented linearly. The nature of the phenomenon in management science makes linear statistical analysis inappropriate in many cases. If prior knowledge of nonlinearity exists, then it is easier to treat. However, in many cases, researchers are unaware of the nonlinear relationship between variables and unsure of which statistical tools to use with nonlinear data.

Model Specification Required in Advance

In conventional least squares linear regression, the model must be specified in advance. This a priori model specification makes the problem easy to solve but entails significant guesswork. Assumptions have to be made concerning the underlying relationship between independent and dependent variables. By default, it is often assumed to be linear. In case the model specification is incorrect, the regression line can also be erroneous, even though the error of the fit can be small. Therefore, regression results are highly dependent on the practitioner's choice of the model. In addition, two models are extremely difficult to combine in the same regression equation. Each model is generally specified and analyzed separately (Duliba, 1991).

Assumptions of Regression

The error term ε in a conventional multiple regression model is subject to the following assumptions: the Gaussian distribution of e, zero mean of ε, independence of ε, and the constant variance of ε. Furthermore, the aforementioned assumptions are required conditions for deriving the least squares regression function (Wang, 1998). These assumptions might not hold true in many cases, maligning the regression function.

Not Adaptive

Multiple regression is not adaptive, in the sense that model components cannot be modified as suggested by the data being analyzed. Although individual coefficients can be eliminated from the model (a coefficient of zero nullifies the contribution of that variable to the model), the form of the equation cannot be changed. Even if it is obvious a linear model is not adequate, regression does not directly suggest how to improve the model or segment the data. Similarly, computer regression programs cannot learn or become smarter. The regression equation is typically applied to the whole data set. Furthermore, the regression equation does not grant relative importance to parts of the data set. Significant underlying patterns in the data may be lost due to this macro approach.

Knowledge of Underlying Distribution

Multiple regression requires knowledge of the underlying distribution of the data to specify a model. One requirement of regression is that data be well distributed. However, we cannot expect past operational data or human performance and interaction data to possess an ideal distribution, as is possible with data from well-planned laboratory experiments. Generally, data tend to be clustered around certain regions and skewed with respect to certain variables (Joseph, Wang, & Shieh, 1992). Similarly, most regression models assume that the underlying population follows a Gaussian distribution. This is an unrealistic expectation for many data sets used by organizational scholars.

Multicollinearity

Multicollinearity is a high degree of correlation among two or more variables. Not recognizing correlations among supposedly independent variables is a major defect in regression practice. If variables are strongly collinear, more data are not better. In fact, in an extreme case, collinearity can make the regression unstable; small changes to the input can lead to wild swings in the output. Dealing with collinearity involves trimming the variables out of the data or combining collinear variables into a single score. However, in doing so, correlation with the dependent variable can also be weakened (Walter & Levy, 1979).

Other Limitations

Some additional limitations of regression are as follows. The tendency to regress too many variables at the same time increases error. Generating an unwieldy number of candidate regression equations is another pitfall of multiple regression. Ignoring the sign, magnitude, and statistical significance of individual regression coefficients can also be a drawback of regression (Johnson, 1989). Furthermore, linear regression is especially inept at dealing with outliers in the data set (Denton, 1995; Marquez et al., 1991). Finally, regression equations may be distorted when data are missing and in noisy environments.

Thus, there are several major drawbacks and several minor drawbacks that make regression less attractive for analyzing organizational data. A few of the drawbacks can be overcome by using nonlinear regression. The following section shows how neural networks can handle both linear and nonlinear relationships and how they can overcome almost all of the drawbacks discussed above.

Characteristics of Neural Networks

A neural network is a computational structure consisting of several highly interconnected computational elements, known as neurons, perceptrons, or nodes (see Figure 1). Each neuron carries out a very simple operation on its inputs and transfers the output to a subsequent node or nodes in the network topology (Mavrovouniotis & Chang, 1992). Neural networks exhibit polymorphism in structure and parallelism in computation (Wu & Yen, 1992).

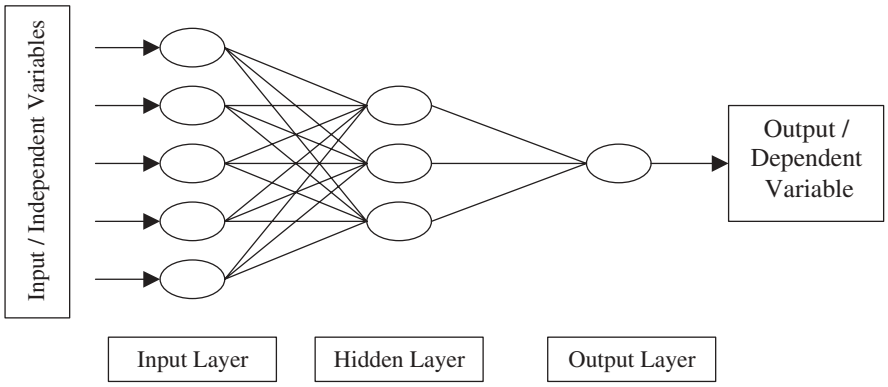

Figure 1: Neural Network Computational Structure

In general, a neural network is characterized by the following three major parts:

- the network's topology,
- the computational characteristics of its elements, and
- the training of the network (Hernandez & Arkun, 1992; Stern, 1996).

Network Computation

The neuron is the basic computational unit of a neural network. The concept of a neuron was developed by McCulloch and Pitts in the 1940s in an attempt to mimic the neurons in the biological brain. Each neuron does the following tasks:

1. Signals are received from other neurons (X_0, X_1, X_2).
2. Signals are multiplied by their corresponding weights (W_0X_0, W_1X_1, W_2X_2).
3. Weighted signals are summed (Sum = $W_0X_0 + W_1X_1 + W_2X_2$).
4. The calculated sum is transformed by a transfer function [F(Sum)].
5. The transformed sum is sent to other neurons (repeats 1-4 above) (Mukesh, 1997, p. 96) (see Figure 2).

The input into a neuron or node is the weighted sum of the outputs from nodes connected to it. Therefore, the net input into a node is

$$\text{Netinput}_i = \sum_j W_{ij} \times \text{output}_j + \mu_i,$$

where W_{ij} are the weights connecting neuron j to neuron i. A negative weight W_{ij} generally means that output from the neuron will decrease and a positive weight will excite the neuron (Stern, 1996). Output_j is the output from node j, and μ_i is a threshold for neuron i. The threshold term is the input to a neuron in the absence of any other input (Warner & Misra, 1996). The threshold term is also called the bias term. Figure 2 illustrates this point.

Artificial neural networks receive inputs and generate outputs in the form of vectors. Netinput_i is simply the dot product of the input and weight vectors.

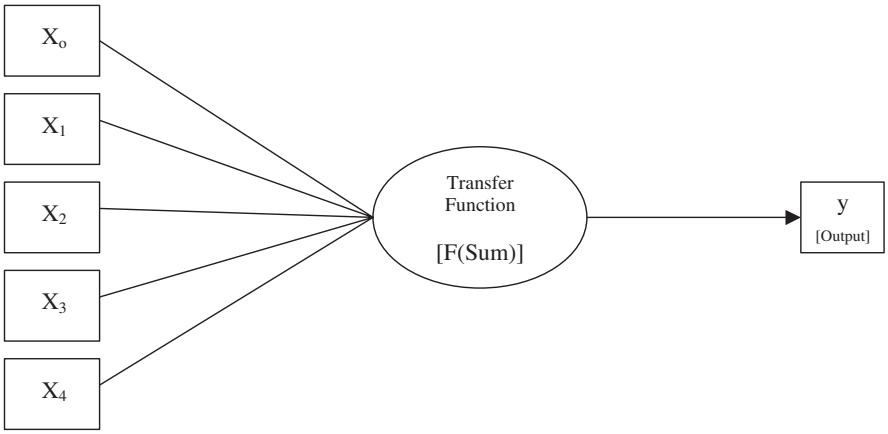

Figure 2: Example of a Simple Neural Network

The prior equation can be expressed by matrix notation as,

$$\text{Netinput}_i = X_j^T W_j + \mu_i,$$

where *X* indicates the vector of inputs, *T* indicates the transpose of the column vector, and *W* indicates the vector of weights. In the vector notation, an additional dot product is sometimes used to give μ_i, which is the bias value. The output of a bias *i* is often 1.0, and the biases (μ_is) are treated in the same manner as the W_is. This additional set of bias weights gives the network a higher degree of flexibility than networks without the additional set of weights (Spining, Darsey, Sumpter, & Noid, 1994). An example of this improved equation is

$$\text{Netinput}_i = X_j^T W_j + X_j^T \mu_i.$$

Computational Characteristics of the Elements

Computation of net input is the core of neural network analysis because it determines why selected neurons become associated with given input patterns (Garson, 1998). The calculated sum of inputs is transformed into output by a neuron using an activation function. An activation function maps any real input into a usually bounded range, often 0 to 1 or –1 to 1. Bounded activation functions are known as squashing functions.[1] If a network has a linear transfer function, then a network with multiple layers can be represented as a network of a single layer that is the product of the weight matrices of each layer. Researchers have also used many nonlinear activation functions. Nonlinear transfer functions between layers allow multiple layers to furnish new modeling capabilities (Harrington, 1993). The specified activation function is arbitrary but often chosen to be a monotonic function (Stern, 1996). Some of the common activation functions are the following:

- linear or identity: $F(x) = x$
- hyperbolic tangent: $F(x) = \tanh(x)$

- threshold: $F(x) = 0$ of $x < 0$, 1 otherwise
- Gaussian: $F(x) = x \exp(-x^2/2)$
- logistic: $F(x) = (1 + e^{-x})^{-1}$ (Sarle, 1994)

The most common activation functions are of the sigmoid or logistic form, such as

$$F(x) = \frac{1}{1 + e^{-x}}.$$

A differentiable activation function is desirable. A minor disadvantage of the sigmoid function is that it does not efficiently interpolate or extrapolate efficiently (Hwang, Lay, Maechler, Martin, & Schimert, 1994).

To briefly summarize, the activation function limits the neuron's response. No matter how weak or strong the inputs, the activation function limits the response to –1 to +1. This is a crucial feature of neural nets and part of the reason a neural net deals with outliers so well. A seven sigma input data point can destroy a regression prediction. For example, if a researcher has an *N* of 50, and the typical range of a variable is 1 to 5, a response of 75 from one of the participants can skew the regression results significantly. However, because the activation function limits the neural network's response, the effect of an outlying rogue data point is less likely to significantly skew the output result.

Network Topology

Traditional neural networks have neurons arranged in a series of layers. The first layer is termed the *input layer.* The last layer is known as the *output layer.* The number of neurons in the input layer corresponds to the number of independent variables used as inputs, and the number of neurons in the output layer corresponds to the number of variables or output data points to be predicted. The layers of neurons in between the input and output layers are called the *hidden layers* (see hidden layers in Figure 1).

When referring to neural network layers, the input layer is generally ignored because the input layer performs no computations. In a feed-forward neural network, the only permissible connections among nodes are between themselves and are directed downward. Connections are not allowed among the neurons of the same layer or the preceding layer. Networks where neurons are interconnected with neurons in the same layer or neurons from a preceding layer are termed *feedback* or *recurrent networks*.

If the neural network model includes estimated weights between the inputs and the hidden layer, and the hidden layer uses nonlinear activation functions, such as the sigmoid or logistic form, the model becomes genuinely nonlinear in its parameters. The resulting model is known as a multilayer perceptron (Sarle, 1994).

The number of nodes in the hidden layer may be as large as required and is related to the complexity of the task the network is designed to perform. More layers may be added by increasing the number of intermediate hidden layers while maintaining full connectivity between successive layers. Each node of a particular hidden layer is usually connected to all nodes in the subsequent layer.

Although the size of the network and the number of neurons may be varied, no commonly accepted rules are available for designing the configuration of a neural network.

In function generation and regression applications, a neural network with fewer than the required hidden nodes will be unable to generate complicated functions. However, too many hidden nodes could cause unwanted oscillation of the fitted curve. In terms of dealing with nonlinearity in the data, fewer than the required hidden nodes will be unable to generalize the nonlinearity and too many nodes may result in overfitting (Wang, 1998). This is analogous to choosing the correct order of polynomial (or other function) for a nonlinear regression fit. A very-high-order polynomial will wiggle to fit noise in a way not representative of the actual underlying phenomenon.

Several researchers have found techniques to ascertain the required number of nodes. For instance, Kung and Hwang (1988) found that the number of neurons in the single hidden layer should be equal to the number of distinct training patterns. Masahiko (1989) concluded that N input patterns required $(N - 1)$ neurons in the single hidden layer. Unfortunately, it is difficult to select the optimum size of a network without knowing in advance the rules that the network is going to abstract from the data (C. W. Lee, 1997).

Rumelhart, Widrow, and Lehr (1994) stated that minimal networks provide more efficient generalization performance than more complex networks do. Smaller networks learn more quickly, operate more quickly with less complexity, and are simpler to interpret in terms of rules (Hagiwara, 1993; Looney, 1996). This is because the shortage of units forces the algorithm to develop general rules to discriminate between input patterns, whereas otherwise it would tend to learn each item of data as a special case. Extending training time in the second case to improve discrimination will have only a detrimental effect on generalization (C. W. Lee, 1997).

Neural networks generalize better when succeeding layers of neurons are smaller than the preceding ones (Kruschke, 1989; Looney, 1996). In addition, Hornik, Stinchcombe, and White (1989) showed that continuous functions on compact subsets of R^p can be uniformly approximated by two-layer neural networks with sigmoid activation functions. The magnitude of weights is more important than the size of the network. The error of a neural network increases with the magnitude of the parameters (weights) and not on the number of parameters (Bartlett, 1998).

No method exists to optimally determine what a network topology should be. Some rules of thumb and math principles help guide the process for determining how many layers the neural network should have.

The simplest neural network consists of just an input layer and output layer, with no hidden layers. In the case of identity activation functions, this is exactly a single matrix multiplication, where the input vector is the input layer, the output vector is the output layer, and the matrix is given by the neural network's weights. This is mathematically linear. Even with nonlinear activation functions, the behavior will typically be close to linear as given by a single matrix multiplication, except for outlier inputs. As matrix multiplication covers this case very well and offers analytic methods for solving for the weights, zero hidden-layer neural networks are rarely used.

Single hidden-layer neural networks, although more general than zero hidden-layer networks are, cannot model all continuous functions. Also, they often take more nodes to do the same problem than a two-layer network (Warner & Misra, 1996). For these reasons, single hidden-layer networks are not often used.

The two-layer neural network can model any continuous function (cf. http://hendrix.ei.dtu.dk) to an arbitrarily desired degree of accuracy. Two-layer networks can model both linear and nonlinear functions and offer many benefits over matrix mathe-

matics. This ability to model continuous functions makes neural networks with two hidden layers the optimal choice for many problems.

Three hidden-layer neural networks can be used to model situations with discontinuous behavior. Most analyses look for relationships (preferably smooth and linear) between inputs and outputs. If inputs and outputs map in a discontinuous manner (an infinitesimally small change in input leads to a macroscopic change in the output), such behavior is often difficult to explain. Explanations are often better done linguistically rather than mathematically. As an example, income tends to rise with age, but there can be a sudden discontinuous drop at a forced retirement age.

For the majority of problems of interest, a network with two hidden layers is the common choice. Networks with three or more hidden layers are rarely used. Too many layers can lead to overfitting, where the network tries to follow the noise or swings wildly between training points.

Once the number of hidden layers has been chosen, the next problem is determining how many nodes to use in each layer.[2] The input layer is easy; this is equal to the number of inputs for the system. Likewise, the output layer is determined by the number of desired outputs.[3]

Where the neural network art comes into play is choosing the number of nodes for the hidden layers. Conventional matrix algebra (which is the special case of a neural network with identity activation functions) would dictate matrix dimensions be set equal to either the number of inputs or outputs. Going much larger than this number can lead to overfitting. Going much smaller can lead to underfitting.

Unfortunately, there is not a rigorous mathematical test to determine the optimal number of nodes to put in a hidden layer. Trial and error methods are necessary, particularly if the functional input-output relationships are not known. One common rule of thumb is to use one more node in the hidden layers than in the input layer. This provides a good compromise for many problems and provides the confidence of being close to matrix multiplication dimensions.

Training the Network

In general, the behavior of any training process depends on the following factors:

1. data set,
2. initial weight set,
3. architecture of the neural network,
4. activation functions and their biases (if any), and
5. step gain and adaptive strategy for it (Looney, 1996).

The first four determinants have been discussed in previous sections. This part of the article discusses number five and other aspects of training the network.

Neural networks "learn" in two principal ways: supervised and unsupervised learning. Supervised training is used when a training set consisting of inputs and outputs, or a set of examples with known results, is available. A supervised network must have a training set consisting of an input vector paired with a corresponding output vector. Supervised learning means learning a static mapping from a vector **X** to a vector **Y**, where there is a training set containing data on both vectors. The vector **X** contains a pattern and vector **Y** contains a classification of that pattern (Werbos, 1991). The network uses the training set to learn any general features that may exist in the training set

and also to determine an error and then adjust itself with respect to the error. The error adjustment is termed *backpropagation*, which is discussed shortly. Once adequately trained, the network will be capable of predicting patterns from new data not used in training.

The most common form of a supervised network is the backpropagation network discussed in this article. Some of the other types of networks with supervised training include the following: mean field annealing, recurrent backpropagation, perceptron, Boltzmann machine, linear vector quantization, ARTmap, Cauchy machine, fuzzy cognitive map, backpercolation, and cascade correlation (Spining et al., 1994).

On the other hand, unsupervised networks are used when training sets with known outputs are not available. In vector terms, the output vectors are unavailable. This means that the network is trained with absolutely no directive feedback. Such networks use inputs to adjust themselves so that similar input gives similar output. In effect, the network adjusts itself to correlations in the input data. During training, output nodes are not used; output nodes are used only in interpreting the results (Spining et al., 1994). Unsupervised networks are ideal for real-time applications. Some examples of unsupervised networks are Kohonen self-organizing maps, Kohonen topology preserving maps, fuzzy associative memory, Hopfield nets (discrete, continuous), linear associative memory, optimal linear associative memory, adaptive resonance theory, fuzzy associative memory, temporal associative memory, and counterpropagation (Spining et al., 1994).

Neural networks can be used in classification or in prediction modes. A neural network is trained with known cases. Once the weights are frozen and the network is put into use, repeats of earlier learned input patterns will classify the output as a known output pattern. If the input data are incomplete (e.g., if some of the input nodes are unknown), or if a new input pattern is presented, neural networks are good at prediction. It will make an "educated guess," in contrast to a computer program that halts on unknown input or a regression equation that can be totally wrong if one input is missing.

Backpropagation

The backpropagation algorithm is a method to ascertain the weights in a multilayer feed-forward neural network. One of the major reasons for renewed interest in neural networks has been the development of the backpropagation algorithm. Cybenko (1989) illustrated the power of the backpropagation algorithm by showing that any continuous function could be approximated to arbitrary accuracy by a two-layer feed-forward network with a sufficient number of hidden units.[4] Backpropagation, as formulated by Rumelhart et al. (1986), with acknowledgment of the earlier work by Parker (1985), is the most widely used method of adapting artificial neural networks for pattern classification applications (Werbos, 1988).

Backpropagation is a conceptually simple iterative gradient descent algorithm.[5] As stated above, neural networks learn by changing the weights of interconnection. During the training phase, network outputs are compared to desired output. If the two match, no change is made to the network. If desired inputs and actual inputs do not match, an expression for the error is generated. This error is backpropagated through the network and the weights are accordingly adjusted. Figure 3 shows the backpropagation process.

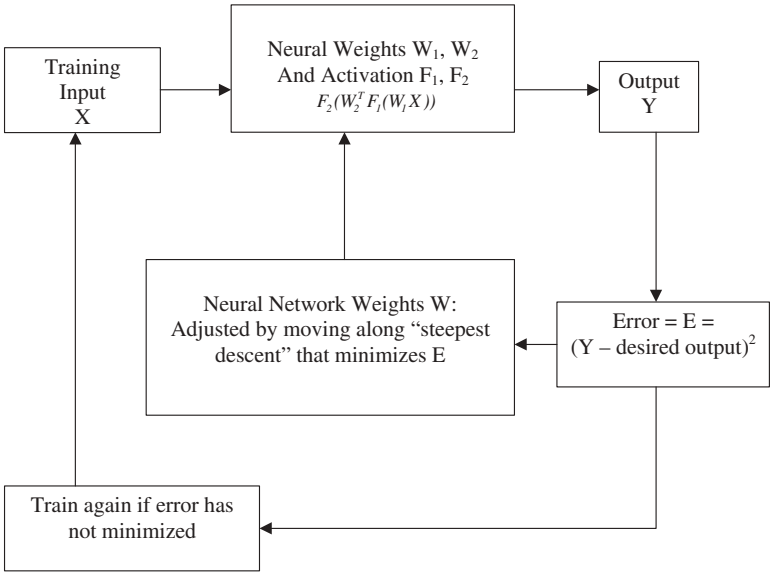

Figure 3: Neural Network Backpropagation Training

The problem is to determine which connections in the entire network were generating the error. This is a process of blame assignment. The basic idea is to specify an expression of the overall performance of the system and subsequently to find a way to optimize that performance (Rumelhart et al., 1994).

In the case of neural networks, the performance criterion is the minimization of error—to be more precise, minimization of squared error. Therefore, the expression for the total error of the system is as follows:

$$\text{Error} = \text{E} = \sum_{p,i} \left(t_{ip} - y_{ip} \right)^2,$$

where i indexes the output units, p indexes the input-output pairs to be learned, t_{ip} indicates the desired output, and y_{ip} is the actual output of the network. The object is to minimize this function. If the output functions are differentiable, blame assignment becomes simplified. A particular unit can be assigned blame in proportion to the degree to which change in that unit's activity leads to change in the error. That is, we change the weights of the system in proportion to the derivative of the error with respect to the weights (Rumelhart et al., 1994). Therefore, the change in W_{ij} (from p. 240) is proportional to $\frac{\partial E}{\partial y_i} \times \frac{\partial y_i}{\partial w_{ij}}$.

Backpropagation can be achieved by implementing the following equation. We start with a set of arbitrary weights $\mathbf{W}_0$ and update them by implementing the formula

$$h(\mathbf{X}, \mathbf{W}) = \text{output function of the network}$$

$$\text{W}_t = \text{W}_{t-1} + \eta \, \nabla \boldsymbol{h}(\mathbf{X}_t, \text{W}_{t-1}) \, (\mathbf{Y}_t - h(\mathbf{X}_t, \text{W}_{t-1})) \; t = 1, 2, 3, \ldots,$$

η = learning rate
$\nabla \boldsymbol{h}$ = vector containing partial derivatives of h with respect to $\mathbf{Y}_t$
$\mathbf{X}_t$ = vector of inputs
W = weights
t = time index (White, 1989)

As shown above, using a differentiable function as an output function considerably enhances the capabilities and the ease of design of backpropagation of a neural network. The sigmoid function illustrated earlier,

$$F(x) = \frac{1}{1 + e^{-x}},$$

has a first derivative of the form

$$\frac{e^{-x}}{\left(1 + e^{-x}\right)^2}.$$

The popularity of the sigmoid function can be attributed to its simplicity, differentiability, and nonlinearity.

Learning Parameters

In training neural networks, the following three training parameters are required: learning rate, momentum, and training tolerance.

Learning rate limits or multiplies the extent of weight changes in any given iteration. A high learning rate, one that reacts very quickly to input changes, tends to make the network unstable. The changes are too radical, and the network's ability to predict will suffer. However, if the learning rate is lower than optimum, the network will take a substantially longer time to learn (Garson, 1998; C. Kuo & Reitsch, 1995). A large learning rate is helpful to accelerate learning until the weight search starts to plateau. However, at the same time, the possibility that the weight search jumps over a minimum error condition and moves into undesirable regions is increased. When this happens, backpropagation learning may fail (Yu & Chen, 1997).

There is a consensus among researchers that adaptive learning rates can stabilize and accelerate convergence to a desired solution (Looney, 1996). Keeping the learning rate η constant is inefficient. A learning rate eventually declining to zero is minimally required for backpropagation to settle down. It has also been shown that the learning rate can be generalized from a scalar quantity to a matrix of a very specific form (White, 1989).

The momentum factor determines the proportion of the last weight change that is added to the new weight change.[6] Low momentum often causes oscillation of weights and renders the network unstable, and learning is never completed. High momentum corresponds to a lack of flexibility and adaptability on the part of the network. Changes are slow to happen in te face of new information (C. Kuo & Reitsch, 1995). In general, the momentum factor should be less than one (unity) to stabilize backpropagation. When error oscillations happen, a momentum factor close to unity is needed to smooth

the error oscillations (Yu & Chen, 1997). On steep slopes during the middle of training, it is desirable to have a small momentum factor. Toward the end of training, the momentum factor should be relatively large (Looney, 1996).

The training tolerance factor specifies the margin of error allowable when network output patterns are compared to training patterns. A training tolerance of zero indicates that network outputs must exactly match training patterns. A training tolerance close to zero will adversely affect the ability of the model to generalize, as a high degree of accuracy in the model is desired relative to training data. However, a high training tolerance factor is also not recommended. This will result in inaccurate results because the specified accuracy is low (C. Kuo & Reitsch, 1995).

Balance must be achieved in specifying the training parameters for a neural network. The training parameters are application specific and are usually determined by trial and error.

Overtraining

In general, artificial neural networks use more parameters than conventional statistical procedures and are thus more susceptible to overtraining. Figure 4 shows the training and overtraining of neural networks. The overtraining phenomenon is observed when the mean squared error of the network continues to increase while the network performance is still improving in learning training patterns. This is highly undesirable as it signifies that the network cannot recognize unknown patterns and its generalization ability is hampered (Tzafestas, Dalianis, & Anthopoulos, 1996).

There are two common methods of dealing with the overtraining phenomenon. The simplest method is to fit the model using only one part of the data and using the other part of the data to evaluate the model's performance.

The second approach to the problem is to use a network pruning algorithm to reduce the network size and hence the number of parameters to be estimated (Hill, Marquez, O'Connor, & Remus, 1994). The benefits of pruning are

1. reduction in operational size,
2. reduction in introduced extraneous noise, and
3. improvement in the rate of successful recognition through generalization.

Pruning should be done at the end of training, and then further training should be undertaken (Looney, 1996; Seitsma & Dow, 1988).

Alternatives to Backpropagation

Several alternatives have also been proposed to backpropagation as a method of learning and correcting weights in an artificial neural network. A sampling of some of the alternative techniques includes the probabilistic neural network, which involves a one-pass learning procedure and can be directly implemented in the neural network architecture (Specht, 1991). Another alternative to backpropagation, projection pursuit learning, beats conventional backpropagation in learning accuracy, is more parsimonious (requires fewer neurons for achieving the same level of accuracy), and requires a shorter training period to achieve the same level of accuracy (Hwang et al.,

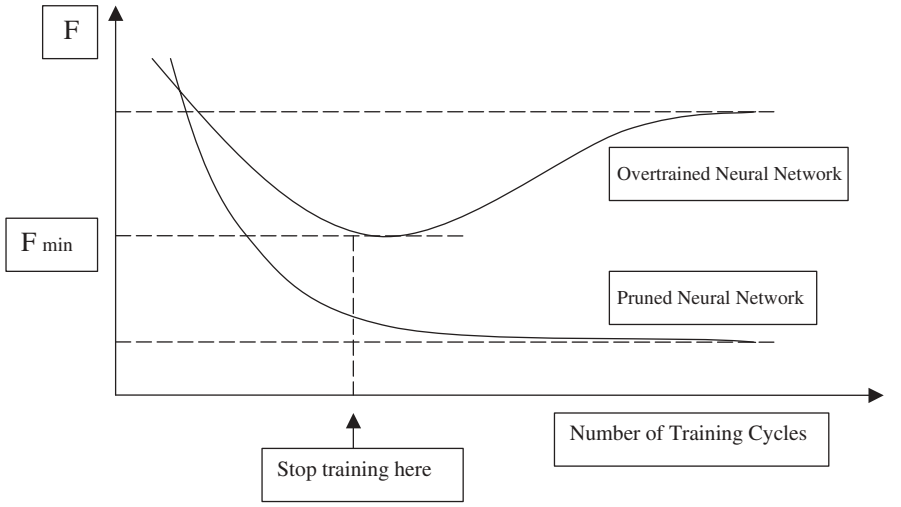

Figure 4: Training and Overtraining of Neural Networks

1994). In addition, a sequential learning procedure has been adopted that has the ability to train networks with mixed neuron types.

Networks with mixed neurons can give more accurate predictions and require a smaller number of neurons for comparable accuracy. Most important, the number of neurons required for a problem may be determined (J. Zhang & Morris, 1998). A nonlinear form of backpropagation has been developed in which the derivatives of the activation function need not be calculated. Furthermore, backpropagation of errors is through the same nonlinear network of forward propagation and not linearized as in standard backpropagation (Golomb & Hertz, 1997). In essence, several modifications and alternatives have been proposed to the standard method of backpropagation.

The genetic algorithm is another method of training neural networks. The genetic algorithm can not only serve as a global search algorithm, but also by correctly defining the objective function, it can achieve a parsimonious architecture (Sexton, Dorsey, & Johnson, 1998).

Applications of Neural Networks in Statistical Analysis

Many neural network models are similar to conventional statistical methods. Some of the conventional statistical techniques that can be emulated by neural networks are generalized linear models, polynomial regression, nonparametric regression and discriminant analysis, projection pursuit regression, principal components, and cluster analysis. Neural networks are especially good as statistical models when the emphasis is on prediction and/or classification of complicated phenomena rather than on explanation.

Even though a neural network can arbitrarily accurately produce the same results as conventional regression, a neural net is inherently a different mathematical approach. With this caveat in mind, some features of neural nets and regression will be

compared—noting that strict mathematical equality could generally occur only in the case of an extremely simple, degenerate neural net.

A neural network with linear activation functions can mimic a linear regression model. A perceptron with a logistic activation function is analogous to a logistic regression model. A perceptron with a threshold activation function can model a linear discriminant function.[7] With multiple outputs, the function becomes a multiple discriminant function. Multiple linear regression has a closed form solution for the coefficients whereas a neural network generally uses an iterative process (Cheng & Titterington, 1994; Sarle, 1994). A nonlinear additive model can also be implemented as a neural network.

With a small number of hidden neurons, a multilayer neural net is a parametric model and may be considered as an alternative to polynomial regression. With a moderate number of hidden neurons, a multilayer neural net can be thought of as a quasi-parametric model similar to projection pursuit regression. A multilayer neural net with one hidden layer is similar to a projection pursuit regression model except that a multilayer neural net uses a predetermined functional form for the activation function and projection pursuit regression uses a flexible nonlinear smoother. If the number of hidden neurons is increased with sample size, a multilayer neural net becomes a nonparametric sieve that provides a useful alternative to methods such as kernel regression (Sarle, 1994; White, 1992).

Advantages of Neural Networks

In the field of exploratory multivariate modeling, neural networks have several major advantages over conventional multiple linear regression. Artificial neural networks have the capabilities for learning to identify patterns between independent variables and dependent variables in a data set. In addition, they possess the ability to specify and estimate a specialized regression model or model adjustment for each pattern. Neural networks can also deal very effectively with nonlinear transformations and data discontinuities. Furthermore, neural networks are adaptive in the sense that they can trigger and choose models appropriately as patterns recur. Multimodel coordination may also be achieved through artificial neural networks (Gorr, 1994). Principal successes of neural networks lie in large-scale optimization and pattern recognition problems (Stern, 1996).

Several researchers have used neural networks in regression analysis (Marquez et al., 1991; Specht, 1991; Wu & Yen, 1992). Many researchers have also compared conventional multiple regression and neural-network-based regression (Bansal, Kauffman, & Weitz, 1993; De Veaux, 1995; Duliba, 1991). In a significant majority of cases, neural-network-based analysis outperformed conventional regression techniques. As a matter of fact, linear regression is a simple case of a single-layer neural network with linear activation functions. The most significant advantages of neural networks when compared to regression are illustrated below.

Nonlinear

Neural networks by their nature are nonparametric and nonlinear in nature. They can deal with data sets exhibiting significant conventionally uncharacterizable

nonlinearity. Neural networks can automatically transform and represent highly complex nonlinear relationships more effectively than regression (Gorr, Daniel, & Szczypula, 1994; C. Kuo & Reitsch, 1995; Marquez et al., 1991).

The nonlinear nature of neural networks does not mean that they cannot abstract linear patterns or that they cannot emulate linear methods when the situation requires them to. Holcomb and Morari (1992) developed an architecture where linear regression has been beneficially merged with neural networks. The inclusion of one linear unit in the architecture has been shown to perform as well as linear methods. Networks lacking this feature will have a difficult time recovering linear performance in situations well treated by linear methods. Problems that are nearly linear or that are linear over a portion of the input space will benefit from the inclusion of the linear unit. Therefore, network architectures may be designed that give "linear" performance as a lower bound without compromising the capability of the network to reproduce nonlinear functions.

Thus, the prudent design of neural networks can ensure superior performance for data exhibiting all degrees of nonlinearity.

Prior Model Specification Is Not Required

While conducting artificial neural-network-based regression, the model to be optimized need not be specified in advance.[8] Thus, neural network–based regression does not require explicit a priori relationships between inputs and outputs (Mukesh, 1997). This feature eliminates the guesswork involved in model specification when using conventional regression techniques (Marquez et al., 1991). In the simulation conducted by Marquez et al. (1991), the neural network performed close to the true model when the relationship was not sufficiently defined. In fact, Denton (1995) found that neural networks performed better than regression when the model was misspecified.

Neural Networks Do Not Require the Assumptions of Regression

Neural network–based regression is not constrained by the assumptions required of conventional regression analysis. Neural network–based regression does not have to make assumptions about underlying population distributions (C. Kuo & Reitsch, 1995). Neural networks are an extremely flexible, almost nonparametric tool less prone to the curse of dimensionality (Intrator & Intrator, 1993). Neural network–based analysis methods need not assume that independent variables are not correlated with each other. This solves the problem of multicollinearity that arises in multiple linear regression. Neural networks do not assume that residuals of independent observations are independent of each other. In addition, they do not require the assumptions that residuals are normally distributed with zero mean and constant variance (Denton, 1995). In the simulations conducted by Wang (1998), only two properties were considered essential to check the curve-fitting results of an artificial neural network–based regression, that the probability distribution of the error term ε is normal and $E(\varepsilon) = 0$.

Adaptive

Neural networks have the capability to learn from experience, allow adaptive adjustment to the predictive model as new examples become available, and generalize the results (Tam & Kiang, 1992; Rumelhart et al., 1994). As new information comes in, past information is not ignored; instead, its importance will be reduced incrementally as new examples are fed into the network (Tam & Kiang, 1992). Furthermore, neural networks can automatically detect different states of phenomena through independently variable data patterns and switch on/off model components as appropriate (Gorr et al., 1994). Data patterns that repeat more often are reinforced, and those that do not are weakened.

See Through Noise and Irrelevant Data

In general, neural networks can see through noise and distortion. They have the ability to abstract essential characteristics in the presence of irrelevant data (Lippman, 1987; Marquez et al., 1991). Neural networks are useful for analyzing data contained in "fuzzy" data sets, data in which there are stable patterns that are subtle or deeply hidden (Mukesh, 1997). Simulations conducted by Bansal et al. (1993) show that the predictive capability of a neural network was unaffected by data quality.

Neural networks and neural network–based regression are particularly good at analyzing data sets in which outliers are present (Denton, 1995; Marquez et al., 1991). This is due to a network's ability to abstract only the essential characteristics of a data set. In analyses conducted by Oja and Wang (1996), neural fitting models clearly outperformed linear methods including least squares and total least squares in impulse noise and colored noise environments and in the presence of outliers. In simulations conducted by Marquez et al. (1991), at the highest levels of noise and smallest sample size, the neural network model outperformed even the noiseless model used to generate the underlying model.

As mentioned before, part of the reason a neural net is so good at rejecting outliers and anomalous data is because the activation function is typically chosen to limit each neuron's response.

High Degree of Robustness

Neural networks offer a high degree of robustness when compared to conventional regression. Even when model assumptions are violated, neural networks exhibit a high degree of robustness and fault tolerance because of primarily local connections (Lippman, 1987; Wu & Yen, 1992). Even if areas of the neural network model break down, the overall performance of the model remains largely unaffected. The observed fault tolerance is primarily due to the highly connectionist architecture and the distributed nature of the computations (Tam & Kiang, 1992).

Limited Data Availability

Neural networks outperform regression when data is limited (C. Kuo & Reitsch, 1995). Neural networks are superior to regression analysis for forecasting, particularly

in situations with limited data availability (Chiang, Urban, & Baldridge, 1996). Many managers like this feature of neural nets. Used as a black box, neural nets just produce "the answer," whereas regression tends to emphasize more the why of the problem (Hall & Krumm, 1999).

Highly Efficient in Dealing With Missing Data

The neural network approach to dealing with missing values in a data set and to reconstructing a data set is more efficient than conventional regression-based methods (C. Kuo & Reitsch, 1995). Gupta and Lam (1996) showed that a backpropagation neural network outperformed both the average method and the iterative regression analysis to compute missing values. There are several possible reasons for the superiority of the neural network approach. The average approach can skew the data distribution, and multiple regression cannot account for the nonlinear relationships in the data set. Furthermore, for both average and regression methods, the presence of outliers or influential observations in the data set may shift fitted values. In this respect, neural networks are better as they can account for nonlinear relationships and do not require any distribution assumptions concerning the underlying data set. Furthermore, neural nets have a big advantage in that they are facile at switching models as needed. In case some elements of an input data vector are missing, the neural net, through past experience, can switch models to one that best predicts or deals with the missing data.

Model Combinations

A single neural network has the capability of combining two or more different model specifications. For instance, unlike regression, neural networks can incorporate both a fixed effects and random effects model (Duliba, 1991). A properly designed neural network can choose a linear model, a binary yes/no output model, and a nonlinear model. It makes the choice by learning which type of input needs which type of model. In the case in which an input is some combination, the neural net will combine the models. As an example, a neural net can give an answer such as "the influence tactic was 80% job-focused and 20% supervisor-focused."

Capability of Real-Time Applications

Being adaptive, neural networks offer online processing capabilities. Although expert systems are satisfactory for offline processing, neural networks are superior for online applications (Tam & Kiang, 1992). In addition, an expert system needs good a priori knowledge of the problem to be programmed. The neural net can shadow an expert and learn during the project in real time.

Compared to regression, neural networks (a) deal with both linear and nonlinear data, (b) formulate the correct data model without a priori specification by the researcher, (c) require less stringent assumptions than regression, (d) learn from experience, (e) see through noise and irrelevant data, (f) offer a high degree of robustness, (g) perform well with limited data, (h) deal effectively with missing data, (i) combine two or more models at the same time, and (j) complete real-time applications.

Limitations of Neural Networks

Numerous advantages accrue to organizational scholars by using neural networks. Neural networks grant the capability to analyze complex problems for which conventional techniques may be inadequate. However, researchers should understand the several inherent limitations of neural networks before using them in analysis.

Difficulty in Choosing the Number of Units

Choosing the number of hidden units to include in a network is at best a trial and error exercise (Warner & Misra, 1996). Although some researchers have attempted to find a mathematical formula for the number of hidden units, there is no single accepted rule. At best, the complexity of the problem will determine the number of required hidden units. In cases in which one layer seems adequate, the number of nodes required may be prohibitive (Warner & Misra, 1996). This is a problem shared by conventional regression, too. Choosing the correct number of variables or degree of the fitting polynomial can be more art than science.

Overfitting may occur if the number of nodes is large relative to the training samples (Tam & Kiang, 1992). A larger than optimum structure may not improve results. In simulations conducted by Gorr et al. (1994), additions to the structure of the neural network did not improve predictive accuracy. Too small a network, on the other hand, can have longer training times, and failures to learn to discriminate are more frequent. If the architecture of the network is larger than required, there are obvious difficulties in selecting which neurons or connections to remove. The removal of inactive connections is unlikely to improve performance, and the removal of active connections may necessitate considerable retraining (C. W. Lee, 1997).

Model Parameters Are Unidentifiable

Unlike conventional multiple linear regression, regression-type model coefficients cannot be ascertained directly from neural network–based regression. Although it could be argued that neural network weights are the parameters for a neural network model, they cannot be interpreted in the same manner as regression weights are interpreted. For finite samples, neural network models are unidentifiable (Intrator & Intrator, 1993). In addition, the significance of individual inputs cannot be analyzed (Tam & Kiang, 1992). The difficulty in interpreting neural network weights arises due to the absence of marginal evaluators (Gorr et al., 1994). A neural network does not directly reveal the functional relationship among variables; they are buried in the summing of the sigmoidal functions (Warner & Misra, 1996). Especially in predictive situations, neural networks provide superior predictions but are difficult to interpret. The disadvantages are most evident in explanation research or when an underlying relationship among variables is to be found (Cheng & Titterington, 1994; Stern, 1996).

There are several solutions for this situation. One is to teach the network to calculate the regression coefficients, teaching it via conventional regression results. The other method is to estimate sensitivities by varying one input and watching how the output varies. A danger of this method is that the neural net can be quite nonlinear,

and this derivative-type information may be valid only locally. A slightly different input vector could potentially exhibit very different sensitivities. However, a recent study by Lu, AbouRizk, and Hermann (2001) showed that by analyzing the input sensitivity of the backpropagation neural network, the researchers were able to solve this problem and sort out the relationship between the output variable and the input parameters.

Superiority of Regression Under Ideal Conditions

Multiple linear regression is superior to neural network analysis in cases in which all the assumptions are met and the model is specified correctly. Regression performs better when the functional relationship is known (Warner & Misra, 1996). The best regression model performs slightly better than neural networks. Similarly, a simple statistical criterion for prediction intervals may be specified for regression, whereas tedious sensitivity analyses need to be conducted when using neural networks to specify prediction intervals. Model fitting through regression is much less computationally intense than neural network model fitting. Regression can accommodate a limited data set for model fitting through larger prediction, whereas neural networks require a large number of input/output samples for training (Chang & Su, 1995). When the model is correctly predicted in advance, regression requires fewer data points to converge or fit the model. Furthermore, regression is better at decomposition (Gorr, 1994).

Although neural network models can incorporate more than one model, correctly specified individual regression models seem to work better. In simulations conducted by Duliba (1991), a fixed effects regression model outperformed a neural network model incorporating both a fixed effects and random effects model. In addition, neural networks are very sensitive to R^2. In simulations conducted by Bansal et al. (1993), multiple linear regression performed better than a neural network when R^2 was the performance criterion.

Because neural networks are nonlinear in coefficients, the normal probability model is not applicable. Finally, regression methodology is more established; hence, managerial practitioners will be able understand and interpret the results of regression more easily than the outputs of a neural network.

Problems With Backpropagation and Sample Size

Backpropagation is the most widely used method of training neural networks. Classical backpropagation has several inherent limitations. Generally, a moderate to large sample size and number of iterations are required for the network to converge to a desired solution (Al-Deek, 2001; Nguyen & Cripps, 2001). The large number of iterations result in a training time that is unusually long (Specht, 1991; Tam & Kiang, 1992). Another major limitation of classical backpropagation is the tendency to converge to local rather than global minima (Stager & Agarwal, 1997). Because the mean square error function can be multimodal, it can be necessary to use nonlinear optimization algorithms. Furthermore, backpropagation fails to make use of all relevant statistical information (White, 1989). In addition, deciding when to stop training is another uncertain parameter (Warner & Misra, 1996).

Difficulty in Choosing Initial Parameters

Neural networks are highly sensitive to initial conditions. McCormack and Doherty (1993) noted that the initial architecture has a strong influence on the learning ability of a neural network and that the architecture depends on the data to be learned. Small changes in learning rates, network design, and initial conditions may produce large changes in network behavior (Refenes, Zapranis, & Francis, 1994).

There is a consensus among researchers that adaptive learning rates can stabilize and accelerate convergence and that a good starting weight set improves both the training speed and learning quality. However, choosing the learning rate, the momentum factor, the cost functions, and other network design elements is difficult because of a lack of formal rules (Looney, 1996; Warner & Misra, 1996).

Neural networks can simulate both linear and nonlinear regression and can overcome most of the drawbacks of regression. However, as the previous paragraphs have shown, neural networks also have drawbacks. One option is for researchers to use neural networks and regression in combination to take advantage of the strengths of both techniques. For example, a researcher can use the neural network to predict and regression to show the relative strength of the various inputs that predict the outcome variables.

Specific Applications of Neural Networks

This section reviews numerous data analyses that have been done in business research using artificial neural networks. Then, the article discusses several areas of management research that do not currently use neural networks but could benefit from using this technique.

Studies That Compare Neural Networks and Regression

Garson (1995) analyzed studies that compare neural networks to traditional statistical techniques. He described 8 studies that found that neural networks are not superior to other analysis techniques. He reviewed 35 studies that found neural networks are superior to traditional statistical techniques. Garson also lists more than 60 studies that use neural networks in the areas of business and economics. His conclusion is that neural networks present the researcher with a tool that is powerful across a wide range of applications.

Forecasting is one specific area of business research in which neural networks have been examined. For example, R. J. Kuo (2001) analyzed convenience store sales forecasting, comparing a fuzzy neural network with the conventional statistical method. Using time series data, he found that the genetic algorithm initiated fuzzy neural network performed more accurately than the conventional statistical method (the autoregressive and moving average technique). T. H. Lee and Jung (2000) found more mixed results. In their study of forecasting credit risk, they found that regression outperformed neural networks for urban accounts but that neural networks outperformed regression for rural accounts (T. H. Lee & Jung, 2000).

In financial forecasting, Chiang et al. (1996) compared neural nets with linear and nonlinear regression analysis. Results of their study showed that neural nets significantly outperformed regression analysis in situations with limited data availability.

Similarly, Denton (1995) compared neural nets with multiple regression in causal forecasting. Results of his research showed that under ideal conditions, there was little difference in predictability between regression and neural nets. However, results indicated that in less than ideal conditions, neural networks do a better job (Denton, 1995). Also, C. Kuo and Reitsch (1995) performed a study that compared neural nets with several conventional forecasting models including regression analysis. The researchers found that "neural networks provide a superior method of forecasting in almost all cases" (p. 17). On the other hand, G. Zhang, Patuwo, and Hu (1998) analyzed 24 studies that compare neural networks to other forecasting methods and reported that the findings are not conclusive as to whether neural networks are superior to other methods.

Several studies have examined neural networks in managerial and MIS applications. Tam and Kiang (1992) analyzed managerial applications of neural networks. They used bank default data to compare neural networks with multivariate discriminant analysis (DA), logistic regression (logit), k nearest neighbor (kNN), and decision tree (ID3). Results of the study showed "that neural nets offer better predictive accuracy than DA, logit, kNN and ID3" (p. 942). Jain and Nag (1997) compared neural networks to logit models. They found that if using a realistic holdout sample, neural networks are superior to similar logit models.

Hardgrave, Wilson, and Walstrom (1994) compared neural nets to other techniques in predicting graduate student success. They evaluated five different models: least squares regression, stepwise regression, discriminant analysis, logistic regression, and neural nets. Results of their study showed that neural nets "perform at least as well as traditional methods and are worthy of further investigation" (p. 249). Similarly, Gorr et al. (1994) used neural networks to model the decision-making process of college admissions. Neural networks were compared with linear regression, stepwise polynomial regression, and an index used by the graduate admissions committee. These researchers found that "while the neural network identifies additional model structure over the regression models, none of the empirical methods was statistically significantly better than the practitioners' index" (p. 17).

Neural networks have also been compared to regression in operations research. Al-Deek (2001) compared a backpropagation neural network (BPN) with regression analysis for modeling freight movement at seaports. He found that the "BPN model is more accurate than the regression model" (p. 284) and can handle highly nonlinear problems. Thus, some studies have found neural networks to be superior, whereas other studies have found that other techniques work better. A critical factor is the type of problem to be solved. The next section discusses questions that could be answered using neural networks.

Areas of Organizational Research in Which Neural Networks Could Be Used

Many areas of business research will benefit from use of neural networks. As examples, the areas of managerial judgment, marketing management, communication patterns, group research, workplace composition, management compensation, and diffusion will be discussed.

Modeling of managerial judgment is a good area for the application of neural networks. The linear decision rule has been popular and has been successful in many

experimental and practical conditions (e.g., Deckro & Herbert, 1984; Goldberg, 1970; Mantel & Galman, 1982; Moskowitz & Miller, 1975; Remus, 1984). However, the linear decision rule is limited by several factors: Its complexity is limited to the linear combination of variables; without expertise it is possible to misspecify the model and fail to make the necessary data transformations. According to research by Remus and Hill (1990), neural networks can automatically handle nonlinearity and should perform better.

Neural networks could also be used to study marketing management and customer service. For instance, Hall and Krumm (1999) used neural networks to develop product return and sales forecasts. In their simulations, neural network performance was significantly better than regression when model R^2 and forecast R^2 were compared. Neural networks have also been used in data-mining applications to ascertain levels of customer satisfaction.

Neural networks can be especially useful for analyzing cyclical phenomena that occur often in business data. Neural networks may be used to analyze topics ranging from business cycles to communication patterns to negotiation patterns within organizations, such as many of the questions posed by Wade-Benzoni et al. (2002).

Neural networks can be used to predict managers' reactions to procedural fairness and justice. A paper by Taylor, Masterson, Renard, and Tracy (1998) used regression to analyze managers' satisfaction with more procedurally just systems. The complex nature of this type of problem lends itself to neural network analysis. Much of the future work that Brockman (2002) suggested is well suited to neural networks. Many procedural fairness variables have nonlinear properties, which make this area a prime candidate for neural networks.

Several areas in group and team research are particularly suitable to neural network analysis. A recent article by Gibson (1999) evaluated group efficacy and group effectiveness across tasks and cultures. With certain groups gaining additional importance in the organizational structure, neural network analysis may be more effective at predicting group behavior. Pelled, Eisenhardt, and Xin (1999) conducted a study analyzing workgroup diversity, conflict, and performance. Their article used regression analysis for predictions about diversity. A neural network may be beneficial in prediction if the data has nonlinear patterns. Furthermore, neural networks may be superior to linear regression if the object is to predict a threshold value such as the optimal amount of performance monitoring for the group or the optimum level of team diversity. Also, neural nets may be used to pick the optimum group of employees for a team.

An area of management research receiving increasing attention is the influence of workplace composition and workplace demographics on performance. A study conducted by Chatman, Polzer, Barsade, and Neale (1998) on the influence of demographic composition and organizational culture on work processes and outcomes could be replicated using neural network–based regression.

Another area of current interest is top management compensation. Several recently published academic papers discuss areas that may benefit from neural network–based regression analysis. For instance, Porac, Wade, and Pollack (1999) looked at industry categories and comparable firms in setting CEO compensation. Problems such as this one, in which categorical data are used and prediction is the objective, could be analyzed effectively with neural networks.

Neural networks are highly effective at analyzing patterns in data. Thus, research examining patterns in diffusion (O'Neill, Pouder, & Buchholtz, 1998), social net-

works (Brass, Butterfield, & Skaggs, 1998; Sparrowe, Liden, Wayne, & Kraimer, 2001), and communicative interactions (Larkey, 1996) could profit from the use of neural networks.

Neural networks are especially beneficial in problems in which the phenomenon exhibits nonlinear patterns. For instance, a recent publication by Barkema and Vermeulen (1998) showed that multinational diversity leads to foreign start-ups rather than acquisitions. Product diversity has a curvilinear effect on the tendency to use start-ups. Added insights into this type of nonlinear relationship may be possible with the use of neural network–based analyses.

Neural Network Software

Three commonly used neural network software packages are the Data Mining Solution by SAS, Neural Connection 2.0 by SPSS, and NeuroShell 2 by the Ward Systems Group. Many researchers will want to use the package that accompanies the statistical analysis software they normally use, such as SAS or SPSS. NeuroShell is a relatively inexpensive and easy-to-use package for those without strong preferences, who want to experiment with neural networks without making a high investment.

Many other software options are available. Two additional software options, Neural Network Toolbox 3.0 (for MATLAB) and Practical Neural Network Recipes in C++ by Masters, are good for researchers with a computer science or programming background. Gencay and Selcuk (2001) provided a critical review of Neural Network Toolbox 3.0. In addition, there are dozens of free neural network software packages. The SAS Web site describes 45 free packages and 41 commercial packages at ftp://ftp.sas.com/pub/neural/FAQ.html. Another site that reviews 60 freeware and shareware packages and 34 commercial packages is www.it.uom.gr/pdp/DigitalLib/Neural/Neu_soft.htm.

Conclusion

This article has analyzed neural networks as statistical tools, predictive tools, and as alternatives to conventional statistical analysis. Research has shown that neural networks can overcome many of the shortcomings of conventional statistical techniques, especially regression. Furthermore, researchers can use both neural networks and regression in combination to take advantage of the strengths of both techniques. This article has examined the basic characteristics of and has discussed the business applications of neural networks.

Neural networks are efficient at analyzing problems that are solved by generating predictions and classifications of complicated phenomena rather than by generating explanations (Laguna & Marti, 2002). In terms of computational complexity, neural networks can handle problems that require the iterative use of data to detect patterns, where lots of example data are present, and it is difficult to specify a parametric model for the data (SAS Web site).

It should be noted that the body of neural network research has grown tremendously in the past 15 years. Although most of the early work was in the mathematical and computer sciences, recent research includes applications to the social sciences and management research. Research in organizational studies will also benefit through an understanding of and prudent application of the technique.

Notes

1. Activation functions may be considered analogous to the inverse of the link function in a generalized linear model. Activation functions are generally bounded but link functions often are not (McCullagh & Nelder, 1989; Sarle, 1994).

2. This relates to the degrees of freedom for the problem at hand. A larger neural network with more nodes has more degrees of freedom to model a more complicated underlying process. However, too many degrees of freedom in the neural network may cause problems of overfitting. Recent research shows that excess degrees of freedom cause little harm and can aid in convergence for some data sets (cf. http://www.neci.nec.com/~lawrence/papers/overfitting-aaai97/).

3. These choices are subject to the caveats of any social science investigation: Are the correct inputs and outputs choices being made?

4. Application of backpropagation was proposed as early as 1951 by Herbert Robbins and Sutton Munro, who called the procedure stochastic approximation (Robbins & Munro, 1951; White, 1989).

5. There are two common types of backpropagation learning: batch learning and sequential learning. The batch learning method updates weights after the presentation of the entire set of training data. Therefore, a training iteration incorporates one sweep through all the training patterns. Sequential learning, on the other hand, adjusts the network parameters as training patterns are presented, rather than after a complete pass through the training set. The sequential approach is a form of stochastic approximation put forward by Robbins and Munro (1951). The sequential method is generally preferred, as the batch method tends to monopolize computer resources. Also, the sequential, iterative method tends to be more stable due to smaller matrices. Finally, for virtually all cases of interest, no generality is sacrificed.

6. Although the momentum factor is an important learning parameter, sometimes little, if any, improvement may be observed due to the inclusion of momentum terms (Fahlman, 1988; Looney 1996). Riedmiller and Braun (1993) found that momentum sometimes hinders convergence. On the other hand, a good selection of parameters can significantly speed up convergence to the desired solution.

7. As stated earlier, the activation function in a perceptron is analogous to the inverse of the link function in a generalized linear model.

8. All continuous functions on compact subsets of R^P can be approximated by a two-layer network with sufficient neurons and sigmoid activation functions (Hornik, Stinchcombe, & White, 1989). This feature allows the network to fit any continuous, compact function.

References

Al-Deek, H. M. (2001). Comparison of two approaches for modeling freight movement at seaports. *Journal of Computing in Civil Engineering*, *15*, 284-291.

Bansal, A., Kauffman, R., & Weitz, R. (1993). Comparing the modeling performance of regression and neural networks as data quality varies: A business value approach. *Journal of Management Information Systems*, *10*, 11-32.

Barkema, H., & Vermeulen, F. (1998). International expansion through start-up or acquisition: A learning perspective. *Academy of Management Journal*, *41*, 7-26.

Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, *44*, 525-536.

Brass, D. J., Butterfield, K. D., & Skaggs, B. C. (1998). Relationships and unethical behavior: A social network perspective. *Academy of Management Review*, *23*, 14-31.

Brockman, J. (2002). Making sense of procedural fairness: How high procedural fairness can reduce or heighten the influence of outcome favorability. *Academy of Management Review*, *27*, 58-76.

Chang, C. A., & Su, C.-T. (1995). A comparison of statistical regression and neural network methods in modeling measurement errors for computer vision inspection systems. *Computers and Industrial Engineering*, *28*, 593-605.

Chatman, J. A., Polzer, J. T., Barsade, S. G., & Neale, M. A. (1998). Being different yet feeling similar: The influence of demographic composition and organizational culture on work processes and outcomes. *Administrative Science Quarterly*, *43*, 749-780.

Chen, L., & Sakaguchi, T. (2000). Data mining methods, applications, and tools. *Information Systems Management*, *17*, 65-70.

Cheng, B., & Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. *Statistical Science*, *9*, 2-54.

Chiang, W. C., Urban, T. L., & Baldridge, G.W. (1996). A neural network approach to mutual fund net asset value forecasting. *Omega: International Journal of Management Science*, *24*, 205-215.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematical Control Signal Systems*, *2*, 303-314.

Deckro, R. F., & Herbert, J. E. (1984). Goal programming approaches to solving linear decision rule based aggregate production planning models. *IEEE Transactions*, *16*, 308-316.

Denton, J. W. (1995). How good are neural networks for causal forecasting? *Journal of Business Forecasting Methods and Systems*, *14*(2), 17-21.

De Veaux, R. D. (1995). A guided tour of modern regression methods. *American Statistical Association, Proceedings of the Section on Physical and Engineering Sciences*, pp. 286-295.

Duliba, K. A. (1991). Contrasting neural nets with regression in predicting performance in the transportation industry. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences*, *4*, 163-170.

Fahlman, S. E. (1988). *An empirical study of learning speed in backpropagation* (Tech. Rep. No. CMU-CS-88-162). Pittsburgh, PA: Carnegie Mellon University.

Garson, G. D. (1995). A comparison of neural network and expert systems algorithms with common multivariate procedures from analysis of social science data. *Social Science Computer Review*, *9*, 399-435.

Garson, G. D. (1998). *Neural networks: An introductory guide for social scientists*. London: Sage.

Gencay, R., & Selcuk, F. (2001). Neural Network Toolbox 3.0 for MATLAB. *International Journal of Forecasting*, *17*, 305-317.

Gibson, C. B. (1999). Do they do what they believe they can? Group efficacy and effectiveness across tasks and cultures. *Academy of Management Journal*, *42*, 138-152.

Goldberg, L. R. (1970). Man versus model of man: A rationale plus some evidence for a method of improving clinical inferences. *Psychological Bulletin*, *73*, 422-432.

Golomb, D., & Hertz, J. (1997). How well can we estimate the information carried in neuronal responses from limited samples? *Neural Computation*, *9*, 649-665.

Gorr, W. L. (1994). Research prospective on neural network forecasting. *International Journal of Forecasting*, *10*, 1-4.

Gorr, W. L., Daniel, N., & Szczypula, J. (1994). Comparative study of artificial neural network and statistical models for predicting student grade point averages. *International Journal of Forecasting*, *10*, 17-34.

Gupta, A., & Lam, M. S. (1996). Estimating missing values using neural networks. *Journal of the Operational Research Society*, *47*, 229-238.

Hagiwara, M. (1993). Removal of hidden units and weights for backpropagation networks. *Proceedings of the International Joint Conference on Neural Networks*, *1*, 351-354.

Hall, Q. P., & Krumm, K. (1999). Developing product return and sales forecasts using neural nets. *Proceedings of the 28th Annual Meeting of the Western Decision Sciences Institute*, pp. 609-613.

Hardgrave, B. C., Wilson, R. L., & Walstrom, K. A. (1994). Predicting graduate student success: A comparison of neural networks and traditional techniques. *Computers and Operations Research*, *21*, 249-263.

Harrington, P. (1993). Sigmoid transfer functions in backpropagation neural networks. *Analytical Chemistry*, *65*, 2167-2168.

Hernandez, E., & Arkun, R. (1992). Study of the control relevant properties of backpropagation neural network models of nonlinear dynamical systems. *Computers and Chemical Engineering*, *16*, 227-240.

Hill, T., Marquez, L., O'Connor, M., & Remus, W. (1994). Artificial neural network models for forecasting and decision making. *International Journal of Forecasting*, *10*, 5-15.

Holcomb, T. R., & Morari, M. (1992). PLS/neural networks. *Computers and Chemical Engineering*, *16*, 393-411.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, *81*, 2554-2558.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359-366.

Hwang, J., Lay, S., Maechler, M., Martin, R. D., & Schimert, J. (1994). Regression modeling in backpropagation and projection pursuit learning. *IEEE Transactions on Neural Networks*, *5*, 342-353.

Intrator, O., & Intrator, N. (1993). Using neural networks for interpretation of nonlinear models. *American Statistical Association, Proceedings of the Statistical Computing Section*, pp. 244-249.

Jain, B. A., & Nag, B. N. (1997). Performance evaluation of neural network decision models. *Journal of Management Information Systems*, *14*, 201-217.

Johnson, D. E. (1989, November). Harnessing the power of multiple regression. *Chemical Engineering*, pp. 176-188.

Joseph, B., Wang, F. H., & Shieh, S. S. (1992). Exploratory data analysis: A comparison of statistical methods with artificial neural networks. *Computers and Chemical Engineering*, *16*, 413-423.

Kruschke, J. K. (1989). Improving generalization in backpropagation networks with distributed bottlenecks. *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*, *1*, 443-447.

Kung, S. Y., & Hwang, J. N. (1988). An algebraic projection analysis for optimal hidden units size and learning rate in backpropagation learning. *Proceedings of the IEEE/INNS International Conference on Neural Networks*, *1*, 363-370.

Kuo, C., & Reitsch, A., (1995). Neural networks vs. conventional methods of forecasting. *Journal of Business Forecasting Methods and Systems*, *14*(4), 17-22.

Kuo, R. J. (2001). A sales forecasting system based on fuzzy neural network with initial weights generated by genetic algorithm. *European Journal of Operational Research*, *129*, 496-517.

Laguna, M., & Marti, M. (2002). Neural network prediction in a system for optimizing simulations. *IIE Transactions*, *34*, 273-282.

Landis, R. S., & Dunlap, W. P. (2000). Moderated multiple regression tests are criterion specific. *Organizational Research Methods*, *3*, 254-266.

Larkey, L. K. (1996). Toward a theory of communicative interactions in culturally diverse workgroups. *Academy of Management Review*, *21*, 463-491.

Lee, C. W. (1997). Training feedforward neural networks: An algorithm giving improved generalization. *Neural Networks*, *10*, 61-68.

Lee, T. H., & Jung, S.-C. (2000). Forecasting creditworthiness: Logistic versus artificial neural net. *Journal of Business Forecasting Methods and Systems*, *18*(4), 28-30.

Lippman, R. P. (1987, April). An introduction to computing with neural nets. *IEEE ASSP Magazine*, pp. 4-22.

Littell, B. (2000). How will intelligent systems and expert systems impact the estate planning field? *Journal of Financial Service Professionals*, *54*, 26-27.

Looney, C. G. (1996). Advances in feedforward neural networks: Demystifying knowledge acquiring black boxes. *IEEE Transactions on Knowledge and Data Engineering*, *8*, 211-226.

Lu, M., AbouRizk, S. M., & Hermann, U. H. (2001). Sensitivity analysis of neural networks in spool fabrication productivity studies. *Journal of Computing in Engineering*, *15*, 299-308.

Mantel, R. J., & Galman, G. J. (1982). Optimal linear decision rules for a work force smoothing model with seasonal demand. *International Journal of Production Economics*, *6*, 167-174.

Marquez, L., Hill, T., Worthley, R., & Remus, W. (1991). Neural network models as an alternative to regression. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences*, *4*, 129-135.

Masahiko, A. (1989). Mapping abilities of three layer neural networks. *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*, *1*, 419-423.

Mavrovouniotis, M. L., & Chang, S. (1992). Hierarchical neural networks. *Computers and Chemical Engineering*, *16*, 347-369.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115-133.

McCormack, C., & Doherty, J. (1993). Neural network super architectures. *Proceedings of the International Joint Conference on Networks*, *1*, 301-304.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). London: Chapman & Hall.

Minsky, M., & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Mitchell, R. (2001). The ghosts in the machine: Can technology find terrorists? *American Spectator*, *34*(8), 52-59.

Moskowitz, H., & Miller, J. G. (1975). Information systems and decision systems for production planning. *Management Science*, *22*, 359-371.

Mukesh, D. (1997, March). Hate statistics? Try neural networks. *Chemical Engineering*, pp. 96-104.

Neal, W. D., & Wurst, J. (2001). Advances in market segmentation. *Marketing Research*, *13*, 14-18.

Nguyen, N., & Cripps, A. (2001). Predicting housing value: A comparison of multiple regression analysis and artificial neural networks. *Journal of Real Estate Research*, *22*, 313-336.

Oja, E., & Wang, L. (1996). Robust fitting by nonlinear neural units. *Neural Networks*, *9*, 435-444.

O'Neill, H. M., Pouder, R. W., & Buchholtz, A. K. (1998). Patterns in the diffusion of strategies across organizations: Insights from the innovation diffusion literature. *Academy of Management Review*, *23*, 98-114.

Parker, D. B. (1985). *Learning-logic* (Tech. Rep. No. 47). Cambridge, MA: MIT Center for Computational Research in Economics and Management Science.

Pelled, L. H., Eisenhardt, K. M., & Xin, K. R. (1999). Exploring the black box: An analysis of work group diversity, conflict, and performance. *Administrative Science Quarterly*, *44*, 1-28.

Pethokoukis, J. M. (2002). Robotrading 101: Sophisticated computer programs take the human element out of picking winners on Wall Street. *U.S. News and World Report*, *132*(3), 23-24.

Phua, P. K., Ming, D., & Lin, W. (2001). Neural network with genetically evolved algorithms for stocks prediction. *Asia-Pacific Journal of Operational Research*, *18*, 103-108.

Porac, J. F., Wade, J. B., & Pollock, T. G. (1999). Industry categories and the politics of the comparable firm in CEO compensation. *Administrative Science Quarterly*, *44*, 112-144.

Racine, J. (2001). On the nonlinear predictability of stock returns using financial and economic variables. *Journal of Business and Economic Statistics*, *19*, 380-382.

Refenes, A. N., Zapranis, A., & Francis, G. (1994). Stock performance modeling using neural networks: A comparative study with regression models. *Neural Networks*, *7*, 375-388.

Remus, W. E. (1984). An empirical investigation of the impact of graphical and tabular data presentations on decision making. *Management Science*, *30*, 533-542.

Remus, W., & Hill, T. (1990). Neural network models of managerial judgment. *Proceedings of the 23rd Annual Hawaii International Conference on System Sciences*, pp. 340-344.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for backpropagation learning: The RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks*, *1*, 586-591.

Robbins, H., & Munro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, *22*, 400-407.

Rosenblatt, F. (1962). *Principles of neurodynamics*. New York: Spartan.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*. (pp. 318-362). Cambridge, MA: MIT Press.

Rumelhart, D. E., Widrow, B., & Lehr, M. A. (1994). The basic ideas in neural networks. *Communications of the ACM*, *37*(3), 87-92.

Sarle, W. S. (1994). Neural networks and statistical models. *Proceedings of the SAS Users Group International Conference*, pp. 1528-1550.

Sarle, W. S. (2002). Ai-faq/neural-nets. Retrieved from ftp://ftp.sas.com/pub/neural/FAQ.html

Seitsma, J., & Dow, R. J. F. (1988). Neural net pruning—Why and how? *Proceedings of the IEEE International Conference on Neural Networks*, *1*, 325-333.

Sexton, R. S., Dorsey, R. E, & Johnson, J. D. (1998). Toward global optimization of neural networks. *Decision Support Systems*, *22*, 171-185.

Somers, M. J. (2001). Thinking differently: Assessing nonlinearities in the relationship between work attitudes and job performance using a Bayesian neural network. *Journal of Occupational and Organizational Psychology*, *74*, 47-61.

Sparrowe, R. T., Liden, R. C., Wayne, S. J., & Kraimer, M. L. (2001). Social networks and the performance of individuals and groups. *Academy of Management Journal*, *44*, 316-325.

Specht, D. F. (1991). A general regression neural network. *IEEE Transactions on Neural Networks*, *2*, 568-576.

Spining, M. T., Darsey, J. A., Sumpter, B. G., & Noid, D. W. (1994). Opening up the black box of artificial neural networks. *Journal of Chemical Education*, *71*, 406-411.

Stager, F., & Agarwal, M. (1997). Three methods to speed up the training of feedforward and feedback perceptrons. *Neural Networks*, *10*, 1435-1443.

Stern, H. S. (1996). Neural networks in applied statistics. *Technometrics*, *38*, 205-214.

Tam, K. Y., & Kiang, M. Y. (1992). Managerial applications of neural networks: The case of bank failure predictions. *Management Science*, *38*, 926-947.

Taylor, S. M., Masterson, S. S., Renard, M. K., & Tracy, K. B. (1998). Manager's reactions to procedurally just performance management systems. *Academy of Management Journal*, *41*, 568-579.

Tzafestas, S. G., Dalianis, P. J., & Anthopoulos, G. (1996). On the overtraining phenomenon of backpropagation neural networks. *Mathematics and Computers in Simulation*, *40*, 507-521.

Wade-Benzoni, K. A., Hoffman, A., Thompson, L. L., Moore, D. A., Gillespie, J. J., & Bazerman, M. H. (2002). Barriers to resolution in ideologically based negotiations: The role of values and institutions. *Academy of Management Review*, *27*, 41-57.

Walter, J. T., & Levy, E. (1979). Limitations in the logic of regression forecasting. *American Statistical Association, Proceedings of the Business and Economic Statistics Section*, pp. 541-545.

Wang, S. (1998). An insight into the standard backpropagation model for regression analysis. *Omega: International Journal of Management Science*, *26*, 133-140.

Warner, B., & Misra, M. (1996). Understanding neural networks as statistical tools. *American Statistician*, *50*, 284-293.

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, *1*, 339-356.

Werbos, P. J. (1991). Links between artificial neural networds (ANN) and statistical pattern recognition. In I. K. Sethi & A. K. Jain, *Artificial neural networks and statistical pattern recognition: Old and new connections* (pp. 11-31). New York: Elsevier Science.

White, H. (1989, December). Neural network learning and statistics. *AI Expert*, pp. 48-52.

White, H. (1992). *Artificial neural networks: Approximation and learning theory*. Oxford, UK: Blackwell.

Wu, F. Y., & Yen, K. K. (1992). Application of neural network in regression analysis. *Computers and Industrial Engineering*, *23*, 93-99.

Yu, X. H., & Chen, G. A. (1997). Efficient backpropagation learning using optimal learning rate and momentum. *Neural Networks*, *10*, 517-527.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, *14*, 35-62.

Zhang, J., & Morris, A. J. (1998). A sequential learning approach for single hidden-layer neural networks. *Neural Networks*, *11*, 65-80.

Kristen Bell DeTienne, Ph.D., is an associate professor in the Department of Organizational Leadership and Strategy at Brigham Young University. Her research examines the impact of technology on communication in organizations.

David H. DeTienne, Ph.D., is a senior principle engineer at ORINCON-Sygenex. His research examines simulation of communications and processing systems, and intelligent mobile software agents.

Shirish A. Joshi is a financial analyst in the Mergers and Acquisitions Group of Morgan Stanley. He specializes in analyzing and executing strategic transactions for companies in the general industrial, energy, and natural resources sectors.