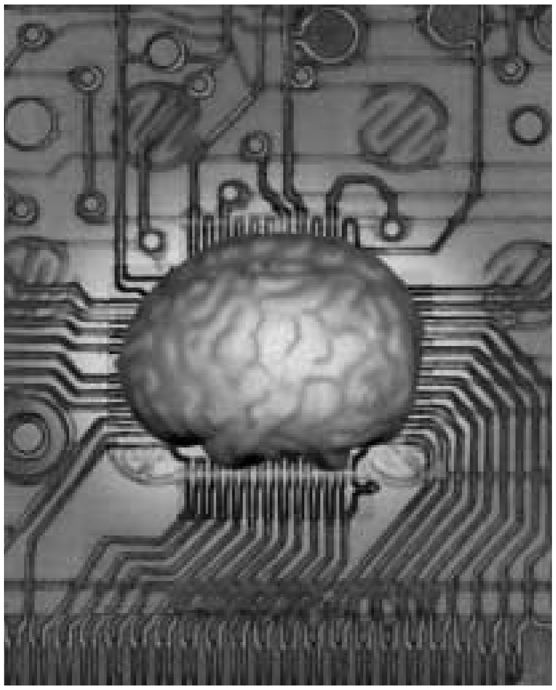
BY LOUISE FRANCIS

The Basics of Neural Networks Demystified

ARTIFICIAL NEURAL NETWORKS are the intriguing new high-tech tool for mining hidden gems in data. Data mining—which also includes techniques such as decision trees, genetic algorithms, regression splines, and clustering—is used to find patterns in data. Data mining techniques, including neural networks, have been applied to portfolio selection, credit scoring, fraud detection, and market research.


ARTVILLE/RUSSELL THURSTON

Neural networks are among the more glamorous of the data mining techniques. They originated in the artificial intelligence discipline where they're often portrayed as a brain in a computer. Neural networks are designed to incorporate key features of neurons in the brain and to process data in a manner analogous to the human brain. Much of the terminology used to describe and explain neural networks is borrowed from biology.

Data mining tools can be trained to identify complex relationships in data. Typically the data sets are large, with the number of records at least in the tens of thousands and the number of independent variables often in the hundreds. Their advantage over classical statistical models used to analyze data, such as regression and ANOVA, is that they can fit data where the relationship between independent and dependent variables is nonlinear and where the specific form of the nonlinear relationship is unknown.

Artificial neural networks (hereafter referred to as neural networks) share the same advantages as many other data mining tools, but also offer advantages of their own. For instance, decision tree, a method of splitting data into homogenous clusters with similar expected values for the dependent variable, are often less effective when the predictor variables are continuous than when they're categorical. Neural networks work well with both categorical and continuous variables.

Many other data mining techniques, such as regression splines, were developed by statisticians. They're computationally intensive generalizations of classical linear models. Classical linear models assume that the functional relationship between the independent variables and the dependent variable is linear. Classical modeling also allows linear relationships that result from a transformation of dependent or independent variables, so some nonlinear relationships can be approximated. Neural networks and other data mining techniques don't require that the relationships between predictor and dependent variables be linear (whether or not the variables are transformed).

The various data mining tools differ in their approaches to approximating nonlinear functions and complex data structures. Neural networks use a series of neurons in what is known as the hidden layer that apply nonlinear activation functions to approximate complex functions in the data.

Despite their advantages, many statisticians and actuaries are reluctant to embrace neural networks. One reason is that they're considered a "black box": Data goes in and a prediction comes out, but the na-

LOUISE FRANCIS IS A PRINCIPAL OF FRANCIS ANALYTICS AND ACTUARIAL DATA MINING, INC. IN PHILADELPHIA. HER ORIGINAL PAPER, "NEURAL NETWORKS DEMYSTIFIED," WAS AWARDED THE 2001 MANAGEMENT DATA AND INFORMATION PRIZE BY THE CASUALTY ACTUARIAL SOCIETY AND THE INSURANCE DATA MANAGEMENT ASSOCIATION. IT IS AVAILABLE AT THE CAS WEBSITE AT WWW.CASACT.ORG/ABOUTCAS/MDIPRIZE.HTM.

ture of the relationship between independent and dependent variables is usually not revealed.

Because of the complexity of the functions used in the neural network approximations, neural network software typically does not supply the user with information about the nature of the relationship between predictor and target variables. The output of a neural network is a predicted value and some goodness-of-fit statistics. However, the functional form of the relationship between independent and dependent variables is not made explicit.

In addition, the strength of the relationship between dependent and independent variables, i.e., the importance of each variable, is also often not revealed. Classical models as well as other popular data mining techniques, such as decision trees, supply the user with a functional description or map of the relationships.

This article seeks to open that black box and show what's happening inside the neural networks. While I use some of the artificial intelligence terminology and description of neural networks, my approach is predominantly from the statistical perspective. The similarity between neural networks and regression will be shown. This article will compare and contrast how neural networks and classical modeling techniques deal with a specific modeling challenge, that of fitting a nonlinear function. Neural networks are also effective in dealing with two additional data challenges: 1) correlated data and 2) interactions. A discussion of those challenges is beyond the scope of this article. However, detailed examples comparing the treatment of correlated variables and interactions by neural networks and classical linear models are presented in Francis (Francis, 2001).

Feedforward Neural Networks

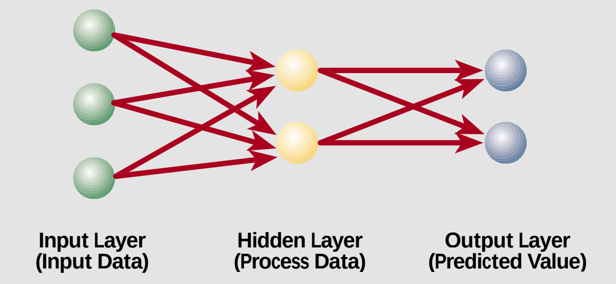
Though a number of different kinds of neural networks exist, I'll be focusing on feedforward neural networks with one hidden layer. A feedforward neural network is a network where the signal is passed from an input layer of neurons through a hidden layer to an output layer of neurons.

The function of the hidden layer is to process the information from the input layer. The hidden layer is denoted as hidden because it contains neither input nor output data and the output of the hidden layer generally remains unknown to the user.

The feedforward network with one hidden layer is one of the most popular kinds of neural networks. The one discussed in this article is known as a Multilayer Perceptron (MLP), which uses supervised learning. Some feedforward neural networks have more than one hidden layer, but such networks aren't common.

Neural networks incorporate either supervised or unsupervised learning into the training. A network that is trained using supervised learning is presented with a target variable and fits a function that can be used to predict the target variable. Alternatively, it may classify records into levels of the target variable when the target variable is categorical. This is analogous to the use of such statistical procedures as regression and logistic regression for prediction and classification.

FIGURE 1 Three-Layer Feedforward Neural Network



A network trained using unsupervised learning doesn't have a target variable. The network finds characteristics in the data that can be used to group similar records together. This is analogous to cluster analysis in classical statistics. This article focuses only on supervised learning feedforward MLP neural networks with one hidden layer.

Structure of a Feedforward Neural Network

Figure 1 displays the structure of a feedforward neural network with one hidden layer. The first layer contains the input nodes. Input nodes represent the actual data used to fit a model to the dependent variable, and each node is a separate independent

FIGURE 2 Scatterplot of X and Y

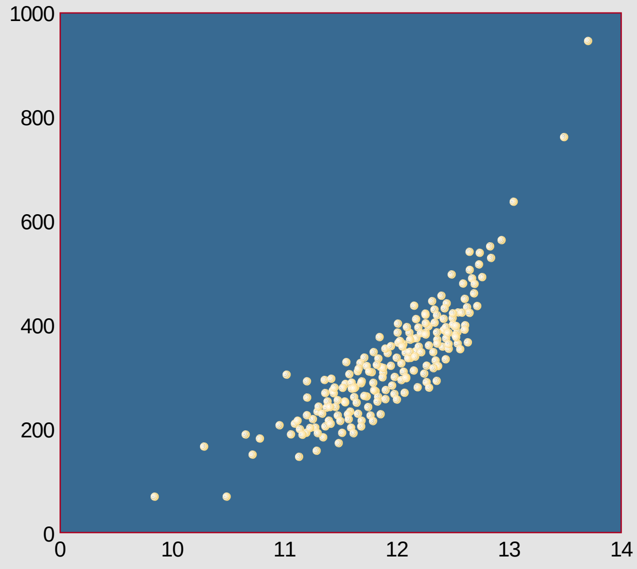


FIGURE 3 Scatterplot of X and Y with “True” Y

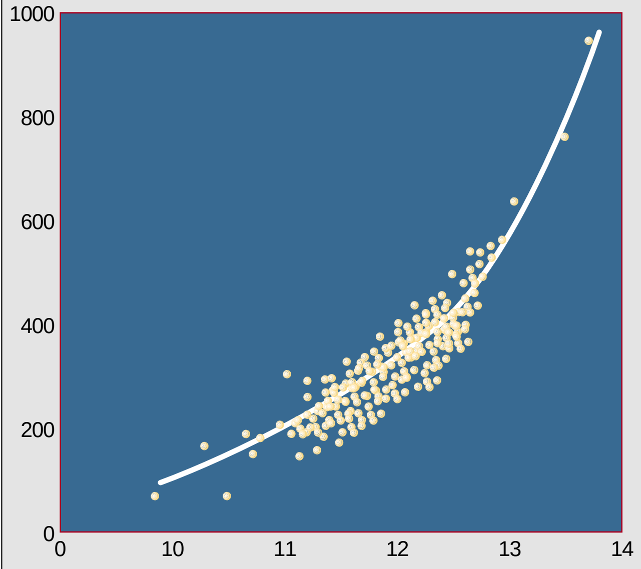
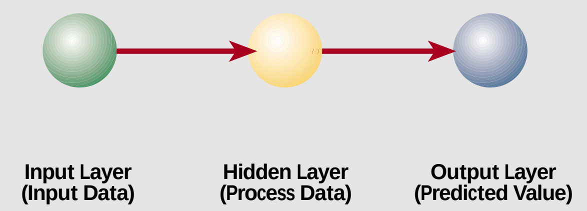


FIGURE 4 Simple Neural Network
One Hidden Node



variable. These are connected to another layer of neurons called the hidden layer or hidden nodes, which modifies the data.

The nodes in the hidden layer connect to the output layer. The output layer represents the target or dependent variable(s). It's common for networks to have only one target variable, or output node, but there can be more. An example would be a classification problem where the target variable can fall into one of a number of categories. Sometimes each of the categories is represented as a separate output node.

Generally, each node in the input layer connects to each node in the hidden layer and each node in the hidden layer connects to each node in the output layer.

The artificial intelligence literature views this structure as analogous to biological neurons. The arrows leading to a node are like the axons leading to a neuron. Like the axons, they carry a signal to the neuron or node. The arrows leading away from a node are like the dendrites of a neuron, and they carry a signal away from a neuron or node. The neurons of a brain have far more complex interactions than those displayed in the diagram, but the developers of neural networks view them as abstracting the most relevant features of neurons in the human brain.

Neural networks “learn” by adjusting the strength of the signal coming from nodes in the previous layer connecting to it. As the neural network better learns how to predict the target value from the input pattern, each of the connections between the input neurons and the hidden or intermediate neurons and between the intermediate neurons and the output neurons increases or decreases in strength.

A function called a threshold or activation function modifies the signal coming into the hidden layer nodes. In the early days of neural networks, this function produced a value of 1 or 0, depending on whether the signal from the prior layer exceeded a threshold value. Thus, the node or neuron would only fire if the signal exceeded the threshold, a process thought to be similar to that of a neuron.

It's now known that biological neurons are more complicated than previously believed. A simple all-or-none rule doesn't describe the behavior of biological neurons. Currently, activation functions are typically sigmoid in shape and can take on any value between 0 and 1 or between –1 and 1, depending on the particular function chosen. The modified signal is then output to the output layer nodes, which also apply activation functions. Thus, the information about the pattern being learned is encoded in the signals carried to and from the nodes. These signals map a relationship between the input nodes (the data) and the output nodes (the dependent variable(s)).

Fitting a Nonlinear Function

A simple example illustrates how neural networks perform nonlinear function approximations. This example will provide detail about the activation functions in the hidden and output layers to facilitate an understanding of how neural networks work.

In this example, the true relationship between an input variable X and an output variable Y is exponential and is of the following form:

$Y = e^{x/2+\mu}$ or

$$Y = \frac{x}{e^2} + \varepsilon$$

where $\varepsilon \sim N(0,75)$, $X \sim N(12,.5)$, and N (μ, σ) is understood to denote the normal probability distribution with parameters μ, the mean of the distribution and σ, the standard deviation of the distribution.

A sample of 500 observations of X and Y was simulated. A scatterplot of the X and Y observations is shown in Figure 2. It's not clear from the scatterplot that the relationship between X and Y is nonlinear. The scatterplot in Figure 3 displays the "true" curve for Y as well as the random X and Y values.

A simple neural network with one hidden layer was fit to the simulated data. In order to compare neural networks to classical models, a regression curve was also fit. The result of that fit will be discussed after the discussion of the neural network model. The structure of this neural network is shown in Figure 4.

As neural networks go, this is a relatively simple network with one input node. In biological neurons, electrochemical signals pass between neurons. In neural network analysis, the signal between neurons is simulated by software, which applies weights to the input nodes (data) and then applies an activation function to the weights.

The weights are used to compute a linear sum of the independent variables. Let Y denote the weighted sum:

$$Y = w_0 + w_1 * X_1 + w_2X_2 \ldots + w_nX_n$$

The activation function is applied to the weighted sum and is typically a sigmoid function. The most common of the sigmoid functions is the logistic function:

$$f(Y) = \frac{1}{1+e^{-Y}}$$

The logistic function takes on values in the range 0 to 1. Figures 5 displays a typical logistic curve. This curve is centered at an X value of 0, (i.e., the constant w_0 is 0). Note that this function has an inflection point at an X value of 0 and f (x) value of .5, where it shifts from a convex to a concave curve.

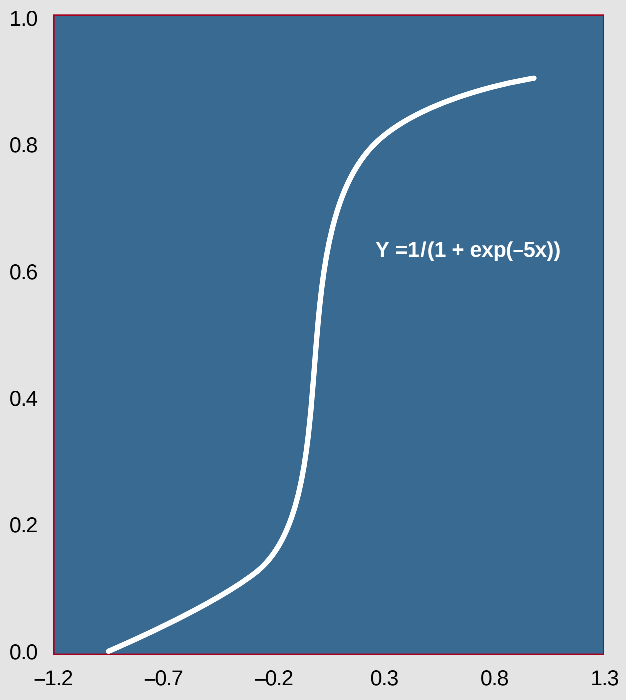
Also note that the slope is steepest at the inflection point where small changes in the value of X can produce large changes in the value of the function. The curve becomes relatively flat as X approaches both 1 and –1.

Another sigmoid function often used in neural networks is the hyperbolic tangent function that takes on values between –1 and 1:

$$f(Y) = \frac{e^Y - e^{-Y}}{e^Y + e^{-Y}}$$

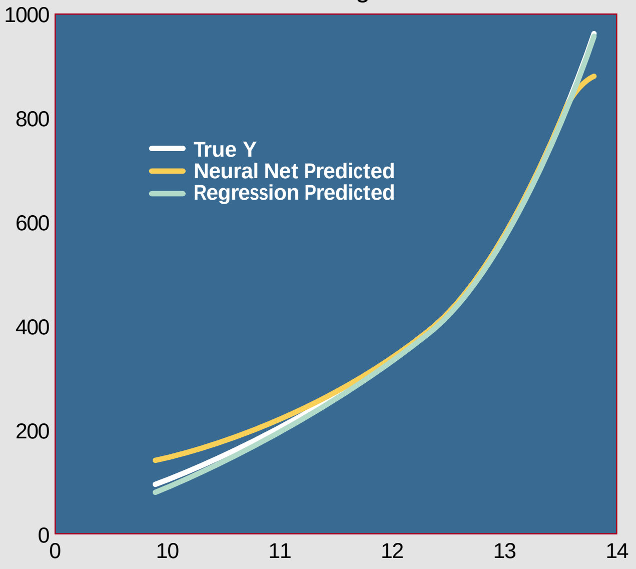
In this article, the logistic function will be used as the activation function. The Multilayer Perceptron is a multilayer feed-

FIGURE 5 Logistic Function



Neural networks are among the more

FIGURE 6 Fitted vs. "True" Y
Neural Network and Regression



forward neural network with a sigmoid activation function.

The logistic function is applied to the weighted input. In this example, there's only one input, so the activation function is:

$$h = f(X;w_0,w_1) = f(w_0 + w_1X) = \frac{1}{1+e^{-(w_0+w_1X)}}$$

This gives the value or activation level of the node in the hidden layer. Weights are then applied to the hidden node:

$$w_2 + w_3h$$

The weights w_0 and w_2 are like the constants in a regression and the weights w_1 and w_3 are like the coefficients in a regression. An activation function is then applied to this "signal" coming from the hidden layer:

$$o = f(h;w_2,w_3) = \frac{1}{1+e^{-(w_2+w_3h)}}$$

The output function, o, for this particular neural network with one input node and one hidden node can be represented as a double application of the logistic function:

$$f(f(X;w_0,w_1); w_2,w_3 = \frac{1}{1+e^{-(w_2+w_3\frac{1}{1+e^{-w_0+w_1X}})}}$$

From the formula above, it can be seen that fitting a neural network is much like fitting a nonlinear regression function. As with regression, the fitting procedure minimizes the squared deviation between actual and fitted values. Because a closed form solution does not exist, numerical techniques must be used to fit the function.

The use of sigmoid activation functions on the weighted input variables, along with the second application of a sigmoid function by the output node, is what gives the MLP the ability to approximate nonlinear functions.

One other operation is applied to the data when fitting the curve: normalization. The dependent variable X is normalized. Normalization is used in statistics to minimize the impact of the scale of the independent variables on the fitted model. Thus, a variable with values ranging from 0 to 500,000 does not prevail over variables with values ranging from 0 to 10, merely because the former variable has a much larger scale.

Various software products will perform different normalization procedures. The software used to fit the networks in this article normalizes the data to have values in the range 0 to 1. This is accomplished by subtracting a constant from each observation and dividing by a scale factor. It's common for the constant to equal the minimum observed value for X in the data and for the scale factor to equal the range of the observed values (the maximum minus the minimum).

Note also that the output function takes on values between 0 and 1 while Y takes on values between -∞ and +∞ (although for all practical purposes, the probability of negative values for the data in this particular example is nil). In order to produce predicted values, the output, o, must be renormalized by multiplying by a scale factor (the range of Y in our example) and adding a constant (the minimum observed Y in this example).

For comparison with a conventional curve fitting procedure, a regression was fit to the data. Due to the nonlinear nature of the relationship, a transformation was applied to Y. Since Y is an exponential function of X, the log transformation is a natural transformation for Y. However, because the error term in this relationship is additive, not multiplicative, applying the log transformation to Y produces a regression equation that is not strictly linear in both X and the error term.

Figure 6 displays the regression fitted value and the neural network fitted value. It can be seen that both the neural network and the regression provide a reasonable approximation to the curve. The neural network and regression have approximately the same R^2 (.678). The regression predicted value for Y has a higher correlation with the "true" Y (.9999 versus .9955). This example demonstrates that linear models will perform well compared with neural networks when a transformation can be applied to the dependent or independent variable that makes the relationship approximately linear.

When an additional hidden node is added to the neural network, the correlation of its predicted value with the "true" Y becomes equal to that of the regression. This result illustrates an important feature of neural networks: the MLP neural network glamorous of the data mining techniques

with one hidden layer is a universal function approximator. Theoretically, with a sufficient number of nodes in the hidden layer, any nonlinear function can be approximated. In an actual application on data containing random noise as well as a pattern, it can sometimes be difficult to accurately approximate a curve no matter how many hidden nodes there are. This is a limitation that neural networks share with classical statistical procedures.

Summary

The article has attempted to remove some of the mystery from the neural network "black box." The author has described neural networks as a statistical tool that minimizes the squared deviation between target and fitted values, much like more traditional statistical procedures do. An example was provided that showed how neural networks are universal function approximators. Classical techniques can be expected to outperform neural network models when data is well behaved and the relationships are linear or data can be transformed into variables with linear relationships. However, neural networks seem to have an advantage over linear models when they are applied to complex nonlinear data. This is an advantage neural networks share with other data mining tools not discussed in detail in this article.

Note that the article does not advocate abandoning classical statistical tools, but rather adding a new tool to the actuarial tool kit. Classical regression performed well in the example in this article.

References

Berry, Michael J. A., and Linoff, Gordon, *Data Mining Techniques*, John Wiley and Sons, 1997.

Brocket, Patrick, Xia, Xiaohua and Derrig, Richard, "Using Kohonen's Self Organization Feature Maps to Uncover Automobile Bodily Injury Claims Fraud," *Journal of Risk and Insurance*, June, 1998, pp. 245 – 274.

Dhar, Vasant and Stein, Roger, *Seven Methods for Transforming Corporate Data into Business Intelligence*, Princeton Hall, 1997.

Derrig, Richard, "Patterns, Fighting Fraud With Data," *Contingencies*, Sept./Oct., 1999, pp. 40–49.

Francis, Louise, "Neural Networks Demystified," *Casualty Actuarial Society Forum*, Winter 2001, pp. 253-320.

Freedman, Roy S., Klein, Robert A. and Lederman, Jess, *Artificial Intelligence in the Capital Markets*, Probus Publishers 1995.

Lawrence, Jeannette, *Introduction to Neural Networks: Design, Theory and Applications*, California Scientific Software, 1994.

Potts, William J.E., Neural Network Modeling: Course Notes, SAS Institute, 2000.

Smith, Murry, *Neural Networks for Statistical Modeling*, International Thompson Computer Press, 1996.

Speights, David B, Brodsky, Joel B., Chudova, Durya I.,, "Using Neural Networks to Predict Claim Duration in the Presence of Right Censoring and Covariates," *Casualty Actuarial Society Forum*, Winter 1999, pp. 255-278.

Venebles, W.N. and Ripley, B.D., *Modern Applied Statistics with S-PLUS*, third edition, Springer, 1999.

Warner, Brad and Misra, Manavendra, "Understanding Neural Networks as Statistical Tools," *American Statistician*, November 1996, pp. 284 – 293.

ACKNOWLEDGMENTS: THE AUTHOR WISHES TO ACKNOWLEDGE THE FOLLOWING PEOPLE WHO REVIEWED THE ORIGINAL PAPER THIS ARTICLE IS TAKEN FROM AND PROVIDED MANY CONSTRUCTIVE SUGGESTIONS: PATRICIA FRANCIS-LYON, VIRGINIA LAMBERT, FRANCIS MURPHY, JANE TAYLOR, AND CHRISTOPHER YAURE.