THE ENERGY JOURNAL

IAEE International Association *for* Energy Economics

WWW.IAEE.ORG

Papers

Short Term Energy Forecasting with Neural Networks

Stuart McMenamin and Frank Monforte

Volume 19, Number 4 **1998**

International Association for Energy Economics

Short Term Energy Forecasting with Neural Networks

J. Stuart McMenamin and Frank A. Monforte***

Artificial neural networks are beginning to be used by electric utilities to forecast hourly system loads on a day-ahead basis. This paper discusses the neural network specification in terms of conventional econometric language, providing parallel concepts for terms such as training, learning, and nodes in the hidden layer. It is shown that these models are flexible nonlinear equations that can be estimated using nonlinear least squares. It is argued that these models are especially well suited to hourly load forecasting, reflecting the presence of important nonlinearities and variable interactions. The paper proceeds to show how conventional statistics, such as the BIC and MAPE statistics can be used to select the number of nodes in the hidden layer. It is concluded that these models provide a powerful, robust and sensible approach to hourly load forecasting that will provide modest improvements in forecast accuracy relative to well-specified regression models.

1. INTRODUCTION

Electricity system operators require near-term forecasts of hourly system loads. These forecasts provide the basis for production and maintenance scheduling. In addition to the standard tools of regression and time-series analysis, approaches using artificial neural networks are being applied to these forecasting problems (see Khotanzad et al. 1993, Papalexopolous et al. 1994, and Ramanathan et al. 1997).

The Energy Journal, Vol. 19, No. 4. Copyright © 1998 by the IAEE. All rights reserved.

* Senior Vice President, Regional Economic Research Inc., 12520 High Bluff Drive, Suite 220, San Diego, CA 92031-2062. E-mail: stuart@rer.com

** Research Manager, Regional Economic Research Inc., 12520 High Bluff Drive, Suite 220, San Diego, CA 92031-2062.

Although neural network models are being used, many analysts in the industry treat them as black boxes, and users often do not understand what these models are or how they work. In part, this reflects the fact that the models are developed and presented using a language that is foreign to utility forecasters. The first purpose of this paper is to present and explain neural network models in terms of conventional econometric language. Specifically, building on the thinking of White (1989), we will show that neural network models can be thought of as flexible nonlinear models, and we will discuss estimation of these models using nonlinear least squares.

The second purpose is to discuss the nature of the hourly load forecasting problem. We will show that the problem requires a nonlinear specification with a wide range of variable interactions. We argue that this type of problem is well suited to the neural network approach, since estimation of network parameters will identify the most useful nonlinearities and interactions without specifying them explicitly in advance.

One issue in specifying neural network models is the degree of flexibility, which is controlled by the number of nodes in the hidden layer. The third purpose of this paper is to demonstrate how conventional sample statistics, like the Bayesian Information Criterion (BIC), can be related to out-of-sample forecasting performance, as measured by MAD and Mean Absolute Percent Error (MAPE) statistics. The results suggest that the BIC can be used to help determine the optimal number of nodes.

Finally, the estimation results of the neural network model need to be analyzed to develop an understanding of the model implications. Specifically, model slopes and elasticities can be computed for each observation, and examination of these results can provide insights about the time profile of model sensitivities as well as the key variable interactions. Using this type of information, neural network models can be used to help structure the nonlinearities and interactions to be included in regression models.

Our general conclusion is that neural networks provide a strong modeling capability for short-term hourly load forecasting. Since load responses to weather variables are nonlinear, and since there are important interactions between weather variables and calendar variables, the problem is well suited to the neural network approach. As long as neural network models are not overspecified, they can provide excellent explanatory power in sample and out of sample.

2. NEURAL NETWORK SPECIFICATION

Although neural networks can take many functional forms (see Kuan and White, 1994), the form that we will use for modeling hourly loads is as follows:

$$Y^t = B_0 + \sum_{n=1}^{N} \left(B_n \times \frac{1}{1 + e^{-(a_{n,0} + \sum_{k=1}^{K} a_{n,k} X_k^t)}} \right) + u^t \tag{1}$$

To understand this nonlinear function, consider the following simple example. If the number of explanatory variables (*K*) is three, and the number of nodes (*N*) is two, then the network function takes the following form:

$$\begin{aligned} Y^t = B_0 &+ B_1 \times \frac{1}{1 + e^{-(a_0 + a_1 X_1^t + a_2 X_2^t + a_3 X_3^t)}} \\ &+ B_2 \times \frac{1}{1 + e^{-(b_0 + b_1 X_1^t + b_2 X_2^t + b_3 X_3^t)}} + u^t \end{aligned} \tag{2}$$

Although this function is clearly nonlinear in the explanatory variables and most of the parameters, it has linear components. This is clear if we rewrite the network function as follows:

$$Y^t = B_0 + B_1 \times H_1^t + B_2 \times H_2^t + u^t \tag{3}$$

Expressed in the language of the neural network literature, each *H* in equation (3) represents a node in the hidden layer. And, for the particular specification we are using here, the output function, which is shown in equation (3), is linear in these values.

Looking more closely at the first node (H_1), we see that the exponent in the denominator is a linear weighted sum of the explanatory variables and we will rewrite this linear weighted sum as follows:

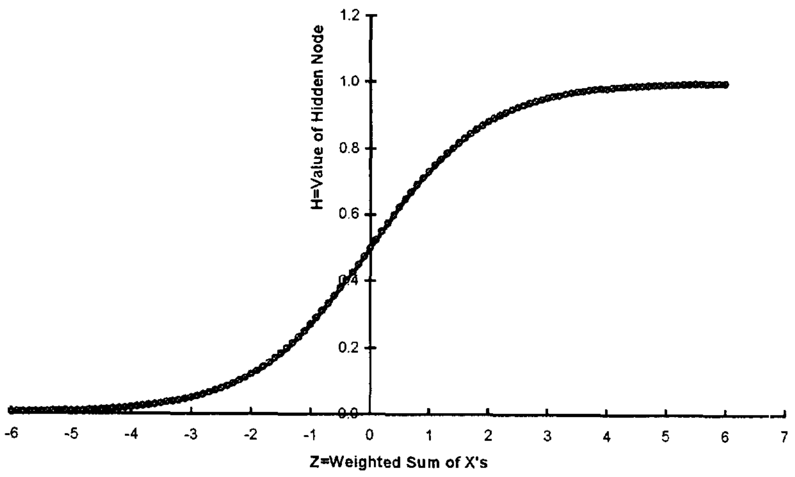
$$Z_1^t = a_0 + a_1 X_1^t + a_2 X_2^t + a_3 X_3^t \tag{4}$$

Then, with some rearrangement, we have the following:

$$H_1^t = \frac{1}{1 + e^{-Z_1^t}} = \frac{e^{Z_1^t}}{1 + e^{Z_1^t}} \qquad (5)$$

In this form, equation (5) is easily recognized as a binary logit, the workhorse of discrete choice models and market share modeling. If *Z* is a large negative number, *H* is close to 0. If *Z* is 0, *H* is 0.5. And if *Z* is a large positive number, *H* is close to 1.0. In between, it traces out an S-shaped function. Plotting *H* as a function of *Z* gives the result depicted in Figure 1.

Figure 1. Binary Logistic Function



This implies that there is an S-shaped relationship between each node value (*H*) and each explanatory variable (*X*) in that node. This response curve may be positively or negatively sloped, depending on the sign of the slope coefficient on the *X* variable.

In addition, the specification for each node is automatically interactive, since we can rewrite the exponential as follows:

$$e^{a_0 + a_1 X_1^t + a_2 X_2^t + a_3 X_3^t} = e^{a_0} e^{a_1 X_1^t} e^{a_2 X_2^t} e^{a_3 X_3^t} \qquad (6)$$

As a result, each *X* variable interacts with all other *X*'s that do not have zero slopes in the node. This is a strength of the specification if the underlying process has multiplicative interactions.

Implications of Parallelism

Returning to equations (1) and (2), it is clear that each explanatory variable appears several times on the right-hand side of the equation and in exactly the same algebraic form. This repetitive property is called parallelism, and it is one of the strengths of the approach.

This point is a major concern to forecasters schooled in econometrics. Repetition of explanatory functions on the right hand side of the equation looks like an acute form of multicollinearity. The flexibility of the neural network model, however, comes from the fact that different parameters will be found for each node, allowing each node to serve a specialized function. This requires an estimation approach that starts with a different set of initial parameter values for each node. From these initial starting points, the role of each node will evolve as estimation proceeds.

Since each node will process input data values differently, the specification allows for a variety of nonlinearities and for a range of variable interactions. For example, in the first node, variations in *X* could be causing movement in the linear part of the logistic equation (*Z* between -1 and +1), and in the second node, variations in *X* might be operating in the top part of the S-shape (*Z* between 1 and 4). This could produce results very similar to inclusion of both *X* and ln(*X*) as explanatory variables in a regression model.

As another example, if you combine two logistic curves, one positively sloped and one negatively sloped, the result could be a U-shaped response over the relevant range of an *X* variable, which is similar to inclusion of squared term for that *X* variable.

Neural Network Terminology

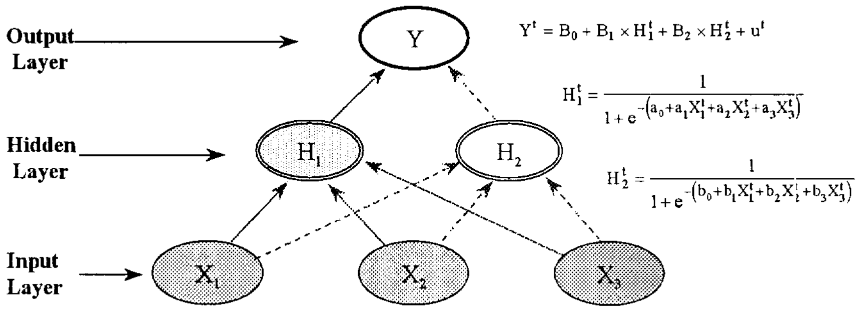
In neural network terms, equation (1) has the following properties:

- It is a feedforward artificial neural network with a single output.
- It has one hidden layer with multiple nodes.
- It uses logistic transfer functions in each node of the hidden layer.
- It uses a linear transfer function at the output layer.

To understand why the neural network is called feedforward, it is useful to draw the classic neural net diagram. As shown in Figure 2, the explanatory variables (*X*) enter at the bottom in the input layer. The logistic transfer functions appear in the hidden layer. And the dependent variable (Y) appears in the output layer.

The idea is that the input variables feed into the nodes in the hidden layer, and there is no feedback. Further, the nodes do not feed sideways into each other. Instead, they feed onward to the output layer. There is no lagged feedback from the output layer to the hidden layer nodes. The absence of feedbacks or node-level interactions makes it a feedforward system. If there are lagged feedbacks from the output layer to the hidden layer, then it is called a recurrent system.

Figure 2. Network Diagram with Three Xs, and Two Nodes


Output Layer
Hidden Layer
Input Layer
Y
H_1
H_2
X_1
X_2
X_3
$Y^t = B_0 + B_1 \times H_1^t + B_2 \times H_2^t + u^t$
$H_1^t = \frac{1}{1+e^{-(a_0+a_1X_1^t+a_2X_2^t+a_3X_3^t)}}$
$H_2^t = \frac{1}{1+e^{-(b_0+b_1X_1^t+b_2X_2^t+b_3X_3^t)}}$

Although the term "hidden layer" is meaningful in neural science models, this terminology is of no particular value for time-series forecasting. The hidden layer is just part of the model specification. It is important to understand the functional specification that is used in this component since it is the source of all nonlinearities and interactions in the model.

Moving from the hidden layer to the output layer, equation (1) uses a linear transfer function. We could include some additional nonlinearity at this level. For example, for *Y* variables that have a discrete outcome, such as a binary (0,1) variable, a logistic activation function at the output layer would probably be desirable. But for forecasting problems with continuous output variables, it is not clear that there is any gain from further nonlinearity at this level.

3. ESTIMATION APPROACHES FOR NEURAL NETWORKS

In the neural network literature, the process of parameter estimation is called training. The goal of the training process is to find network parameters that make the model errors small. The estimation process is more complicated than for a regression model because the model is nonlinear and because the objective function is relatively complicated.

Many neural network programs use some variant of a method called backpropogation, following the lead of Rummelhart et al. (1986). For the type of forecasting problem addressed here, this approach is unnecessarily slow and cumbersome (see the discussion in Masters, 1995). In what follows, we use a conventional nonlinear least squares algorithm to find parameter solutions. Specifically, we use the Levenberg-Marquat (LM) algorithm in the IMSL library (IMSL, 1994).

The estimation algorithm works as follows. First, a set of random values are assigned to the model parameters. Given this set of parameter values, predicted values and residuals are computed for each observation, along with the derivatives of the residuals with respect to changes in each parameter. The LM algorithm uses this information to change the parameters to new values that will reduce the sum of squared errors. These revised values provide a new starting point, and the revision process is repeated. For the data examined here, the parameter values usually converge within reasonable tolerance within 100 iterations through the data.

Multiple Optima

Parameter estimation is a well behaved process for most common statistical problems, such as finding the solution to a nonlinear regression model or solving the parameters of an ARMA model. However, because of the parallelism property, which reflects the inclusion of multiple nodes in the hidden layer, it can be shown that the least squares objective function for a neural network is extremely complex with a huge number of local optima, as opposed to a single global optimum (Goffe, Ferrier, and Rogers, 1994).

Because of this complexity, it is necessary to explore a wide region of the parameter space to find a relatively good solution. Merely going downhill from a single random starting point to the nearest local optimum will not do the job. This is equally true for estimation approaches based on backpropogation as it is for approaches based on mathematical optimization. Using a multiple-seed approach, we have found that estimation from 20 alternative random starting points is fairly certain to produce several strong solutions for problems of the type studied here.

The rule that we use to select the final model parameters is based on an average of in-sample and out-of-sample error statistics. To develop these statistics, estimation from each random starting point is based on a subset of the sample data. Once a solution is found, these parameters are used to test the power of the estimated parameters, based on the observations that have been withheld from the estimation process. Usually, solutions that perform well in sample also perform well out of sample, but this is not always the case. Especially with a large number of nodes, some of the solutions will be more specialized to the specific cases in the sample, and some will be more stable and more useful out of sample.

The fact that there are many solutions that have comparable performance is neither surprising nor disturbing. This implies that there are a large number nonlinear specifications that provide similar performance. The same result holds for most regression models, in that minor variations in the specification usually do not alter the model results significantly.

Continued Learning

As new data become available, it is natural to re-estimate parameters with the extended sample period. In the neural network literature the update process to incorporate new data is called learning. This terminology stems from the backpropogation method, in which learning is treated as an extension of the process used in training. Although learning is not the focus of this paper, this is a point of confusion for many analysts who believe that learning is a unique property of neural networks.

With linear least squares models, updating the parameters with additional data is a full re-estimation of the model, and the new data typically are given the same weighting as the earlier data. For nonlinear least squares, the re-estimation process can begin with the same set of initial guesses that were used to start the original estimation process or it can begin with the solution from that estimation. Either way, for problems with a well-behaved objective function, the final solution will be the single set of parameters that correspond to the global optimum based on the expanded data set.

For neural networks, which are known to have a large number of local optima, the situation is a bit different. In this case, it seems natural to start with the parameters from the training process and re-optimize with the new data included. However, starting at the training solution implies starting with the specific set of nonlinearities and variable interactions represented by that solution. From this starting point, re-estimation with the expanded data typically leads to minor changes in the estimated parameters. This implies that you are staying at the same local optimum at which you started, and that the location of this solution does not move much because of the new data. In this sense, the new data play less of a role than the earlier data. In essence, the functional form was determined in the training process, which looked at many solutions and selected one, and the new data are used only to refine parameters given that functional form. To give the new data equal play with the earlier data, it would be necessary to repeat the entire training process with the expanded data set.

4. SUMMARY STATISTICS AND TEST STATISTICS FOR NEURAL NETWORKS

There are a wide variety of standard statistics that can be used to evaluate the performance of a neural network model. Once parameters are estimated, a full set of residuals can be computed for in-sample and out-of-sample observations. Then traditional measures of fit, such as R-square statistics, the model standard error, MAD, MAPE, and BIC can be computed.

Properties of the error term have not received much attention in the context of neural networks. In part, this reflects the type of problems to which they have most often been applied, many of which fall into the area of pattern recognition. In a time-series context, however, the error term for a neural network model faces the same potential set of issues as in a linear model. There may be excluded variables, errors in variables, or a systematic error process, implying that there is still information remaining in the estimated residuals.

Regarding error processes, the residuals of the neural network model can be used to compute all of the standard statistics, including the Durbin-Watson statistic, the Ljung-Box statistic, and the ACF and PACF patterns. If these statistics indicate that the residuals follow a time-series process, such as an AR process or a short MA process, then the nonlinear estimation algorithm can be extended to include estimation of the parameters of the ARMA process.

Model Sensitivities

In a linear model, each explanatory variable has a slope parameter, and this slope equals the derivative of the predicted value with respect to that variable. The slopes can be converted to estimated elasticities in each period by multiplying the slope by the ratio of the explanatory variable to the dependent variable (X/Y) for that observation.

For neural networks, the result is similar, but significantly more complicated. Because of the nonlinearities and interactions inherent in the specification, the derivative of the predicted value with respect to an explanatory variable will usually have a different value for each observation. The elasticity value in each period can then be computed using the derivative value in that period.

In fact, one of the potential uses of neural networks is to do exploratory analysis. For example, you can place your data in a network with a small number of nodes, do some training, and look at the pattern of the slopes and elasticities over time. Analysis of these results can lead to a better understanding of the factors that determine how model sensitivities vary over time and which variable interactions are important.

Number of Nodes

Although there is no hard and fast rule, for most time-series problems that we have looked at, the optimal number of nodes appears to be between two and five. Of course, as you add nodes, the in-sample fit always improves. That is, the sum of squared errors will always decline if you add more parameters. However, beyond a point, the coefficients have the freedom to specialize in order to explain specific events in the sample period, and these specialized results do not necessarily generalize to out-of-sample conditions.

There are no established statistics or procedures for deciding on the number of nodes. From the econometric side, the following types of statistics are relevant to this issue:

- Adjusted R Square,
- Akaike Information Criterion (AIC), and
- Bayesian Information Criterion (BIC).

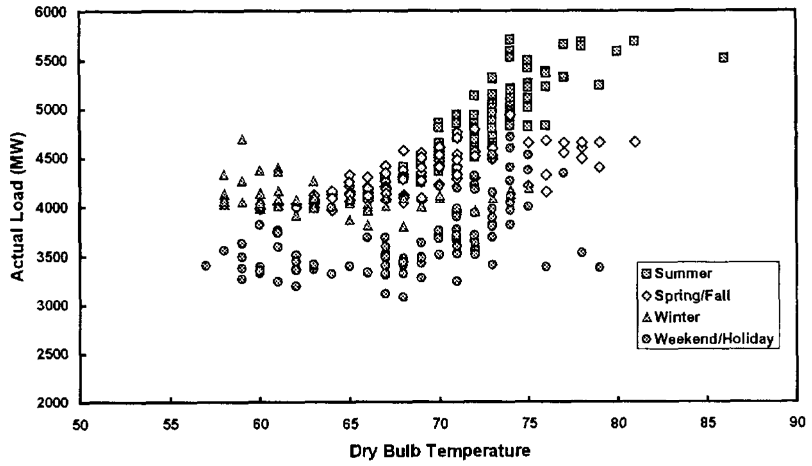
All of these statistics improve when the sum of squared errors is reduced, but impose a penalty for the increased number of coefficients. As will be seen below, we find that of these statistics the BIC provides a useful guide for model specification purposes.

5. DATA

The dependent variable data is the hourly system load for an electric utility in the Southwest U.S. for 1993, giving a total of 8,760 observations. Corresponding data for hourly temperatures, humidity levels, cloud cover, and wind speed were obtained from the National Oceanic and Atmospheric Administration. In addition to these data, the calendar for that year gives a variety of day-type variables, such as day of the week and the timing of holidays. Finally, data on the time of sunrise and sunset provides other important seasonal information.

The model is applied separately to data for each hour of the day. For purposes of presentation, we focus on the data for 3 p.m. Figure 3 shows a scatter plot of the load each day at 3 p.m. against the corresponding dry-bulb temperature at that hour. The points are coded with symbols that separate weekends and holidays from the weekdays in each season. As is clear from this plot, there is significant variation in the system load at this hour for a given temperature, leaving much to be explained by other conditions and calendar variables.

Figure 3. Loads Versus Temperature at 3 p.m.



Nonlinearities and Interactions

As is evident from Figure 3, the relationship between the hourly loads and temperature appears to be nonlinear. Specifically, in winter months when it is cold, increases in the temperature appear to reduce the load. This probably reflects reductions in electric heating loads. In contrast, in summer, increased temperature values appear to be strongly correlated with increased loads. Although this slope appears to be positive in spring and fall months, the weather response slope does not appear to be as large as in the summer months.

As indicated in Table 1, there are a significant number of potential explanatory variables. The exact set that are available for modeling depends on the timing of the forecast. For example, for a day-ahead forecast that must be developed by 4 p.m. each day, the most recent value that can be included as a lagged load is the system load as of 3 p.m. Of course, earlier values, such as the load in the morning, can also be included since these values will be known at the time of the forecast.

The difficult part of model specification for an econometric approach to this problem concerns the variable interactions. The most important examples are the interactions between weather variables and calendar variables, the interactions between lagged loads and calendar variables, and the interactions among weather variables. Brief examples of these interactions are discussed below.

Table 1. Summary of Explanatory Variables

Weather	Calendar	Lagged Loads
Coincident Temperature	Day of the Week	Morning (day-1)
Daily High	Month/Season	Afternoon (day-1)
Daily Low	Holidays	Same hour (day-1)
Cumulative Temperature	Days near holidays	Same hour (day-2)
Temperature Gradient	Sunrise & Sunset	
Humidity		
Wind Speed		
Cloud Cover		
<———	Interactions	———>

- ***Weather and Calendar Variables.*** It is clear from inspection of the data that an extra degree of temperature has a different impact on a weekday than on a weekend day or holiday. For that matter, the Saturday slope may be different from the Sunday slope, and, as discussed above, the weekday slopes in winter for a given temperature are different than the weekday slopes in summer. These facts suggest that temperature data must be interacted with day-of-the-week variables and seasons, at a minimum.

- ***Lagged Loads and Calendar Variables.*** Lagged loads are powerful explanatory variables in next-day forecasting exercises. And it is fair to include these variables in a model, since the data values are known at the time of the forecast. However, the relationship between yesterday's load and today's load differs significantly across days. On a Monday, the lagged load is for a Sunday, and therefore the slope is different than on a Tuesday, when the lagged load is for a Monday. For some Tuesdays, however, the lagged load is for a Monday holiday. These and other interactions must be allowed in the model so that the differential influence of lagged loads can be estimated across different day types.

- ***Weather Variable Interactions.*** Laws of thermodynamics and the presence of heating and cooling equipment suggest some important interactions. For example, the influence of increased wind speed in the summer should be negative, due to lowered cooling loads. However in the winter, increased wind implies wind chill which raises heating loads. Similarly, cloud cover in the summer lowers cooling loads, and cloud cover in the winter can increase heating loads. Further, the role of humidity may be different in the winter than in the summer. These interactions imply that the slopes on these types of weather variables can switch signs depending on other conditions.

6. MODEL ESTIMATES

As a point of reference, a linear regression model was estimated with various combinations of the variables and appropriate interactions. The regression model includes all of the factors listed above as explanatory variables. Specifics are as follows:

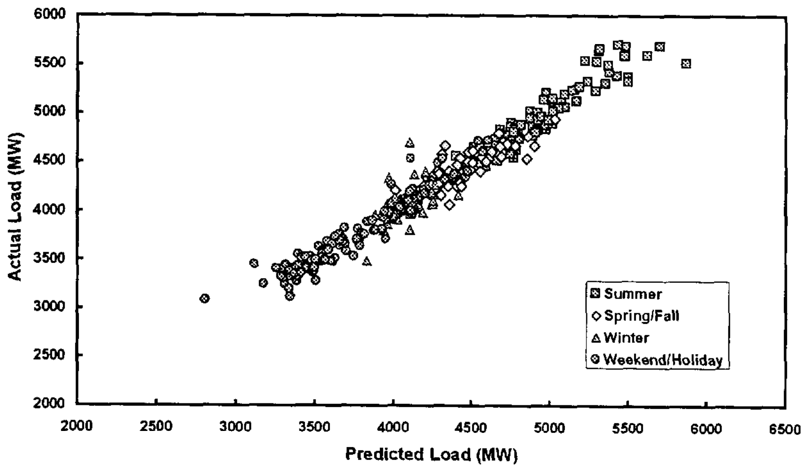
- Daily high and low temperature variables are included in the model, since this resulted in a better fit than use of the coincident temperature alone.

- For the high and low temperature variables, squared terms are included to allow for nonlinear responses, and different temperature slopes are estimated for weekdays versus weekend days and for each season.

- Lagged load values are included for 8 a.m. and 2 p.m. the preceding day. Both of these variables are interacted with day type and season variables.

The results of the regression are depicted in Figure 4. The overall mean absolute percent error (MAPE) was 1.90%, which is a good result for this type of data.

Figure 4. Actual Vs. Predicted Loads at 3 p.m. - Regression Model



Neural Network Models

Estimation was repeated using the neural network model described above. Variables included in the model were daily high and low temperatures, humidity, cloud cover, wind speed, lagged loads from the preceding day, and calendar variables including day of the week and season. All variables were included in all nodes and all nodes used an S-shaped activation function based on the logistic curve, as shown in equations (1) and (2).

To examine alternative levels of flexibility, the number of hidden layer nodes was varied from 1 to 5. For each level, 20 random starting points were utilized. The results from each of the 20 trials were compared, and the final specification was chosen to be the trial that produced the lowest average for the in-sample and out-of-sample MAPE statistics.

The results of these training exercises are summarized in Table 2.

Table 2. The Contribution of Additional Nodes

# of Nodes	Adjusted R Square	AIC	BIC	Sample MAPE (%)	Test MAPE (%)
1	0.939	9.95	10.18	2.35	2.24
2	0.959	9.61	10.07	1.87	1.81
3	0.961	9.60	10.28	1.72	1.71
4	0.966	9.51	10.41	1.56	1.77
5	0.968	9.49	10.61	1.45	2.16

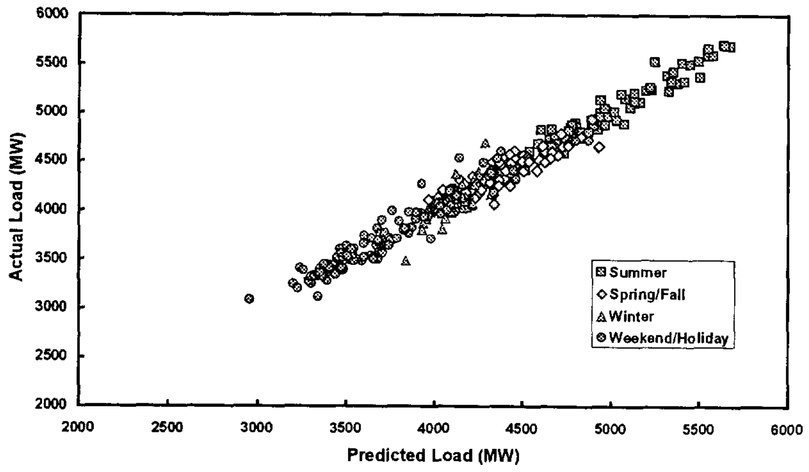
As shown in the table, as additional nodes are added, the in-sample MAPE declines steadily. Similarly, the adjusted R square increases steadily, and the AIC statistic decreases steadily. In both cases, this indicates that the improvement in model fit outweighs the penalty caused by an increased number of coefficients. In contrast, the BIC statistic is lowest for the specification with two nodes, suggesting a possible loss of predictive power as the number of nodes is increased beyond this range.

The out-of-sample MAPE value is at a minimum for the three-node specification, with a value of 1.71%. Although the lowest out-of-sample MAPE does not correspond exactly with the minimum BIC value, this result suggests that the number of nodes should not be extended far beyond the point where the BIC turns upward.

Both the in-sample and out-of-sample fit for the three-node specification are slightly better than we could obtain with a linear regression model (MAPE = 1.90%), despite efforts to include in the regression model what we expected to be the most important nonlinearities and interaction terms.

The plot of actual and predicted values for the 3-node specification is presented in Figure 5. As seen here the pattern of predicted values is very similar to the regression result presented in Figure 4. Examination of the residuals verified that the observations with the largest errors were the same for both modeling approaches.

Figure 5. Actual Vs. Predicted Loads at 3 p.m. - Neural Network Model

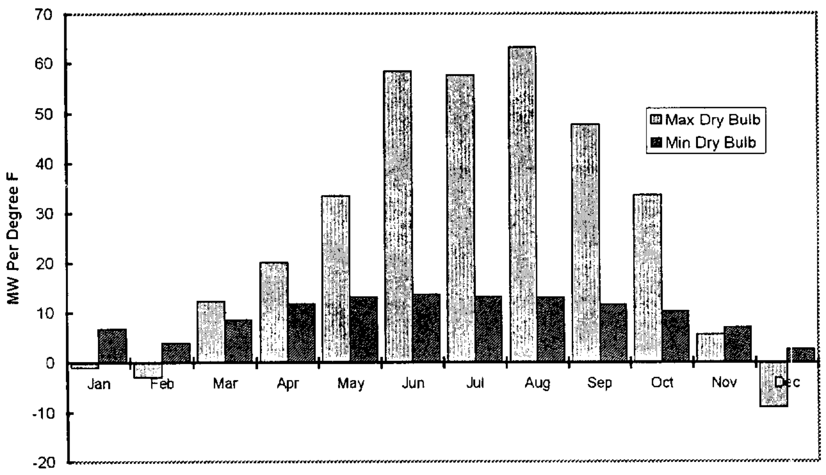


In addition to the summary statistics presented above, the model residuals were examined for presence of autocorrelation. The Durbin-Watson statistic for the 3-node model was 2.04, indicating absence of first order autocorrelation. Similarly, the Ljung-Box statistic was 23, and the probability of a greater value was 67%, indicating lack of evidence of higher-order autocorrelation problems.

Model Derivatives

As discussed above, the neural network model is both nonlinear and interactive by construction. The results from the network with three nodes were examined in detail, including computation and analysis of the period-by-period slopes and elasticities. The following focuses on the derivatives of load with respect to daily high and low temperatures. The monthly average values for these derivatives are plotted in Figure 6.

Figure 6. Plot of Model Derivatives of Load With Respect to Temperature



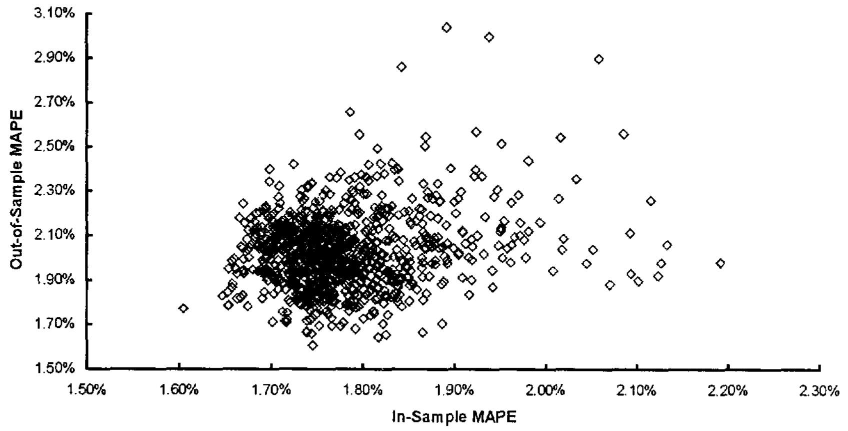
As seen in the figure, the slope with respect to the daily high temperature varies significantly through the year. In the winter months, this slope is a small negative value. In the summer months, the slope increases to a relatively stable value of about 50 MW per degree F. There is a steady transition upward in the spring months of March through May, and a steady downward transition from September through November.

The slope with respect to the daily low is a small positive value in the winter months. It remains relatively stable from April through October, between 10 and 15 MW per degree.

Model Training

To evaluate the relationship between in-sample and out-of-sample statistics, the 3-node model was estimated using 1,000 random starting points. The results are summarized in Figure 7. As shown, the in-sample summary statistics range from a low of 1.6% to a high value of 2.2%. Similarly, the out-of-sample fit statistics range from a low of 1.6% to a high of over 3.0%. The model solutions toward the lower left-hand corner of the chart are the solutions that fit well for both the in-sample and out-of-sample periods.

Figure 7. In-Sample Versus Out-of-Sample MAPE



Other Hours

Applying the same method to other hours of the day provides a set of models that can be used to forecast next-day loads. Across hours, the in-sample and out-of-sample MAPE values range from a low of 1.3% to a high of 1.8%. Generally speaking, the later hours of the day have larger errors, reflecting the fact that they are "longer" forecasts than the early morning hours. That is, the most recent load information included in the model is the load at 2 p.m. As a result, the forecast for 1 a.m. the following day is an 11-hour ahead forecast. In comparison, the forecast for hour 24 the following day is a 34-hour ahead forecast.

To visualize the predictive power of this set of hourly models, actual and predicted values for the summer and winter peak weeks are shown in Figure 8 and Figure 9.

Figure 8. 1993 Summer Peak Week

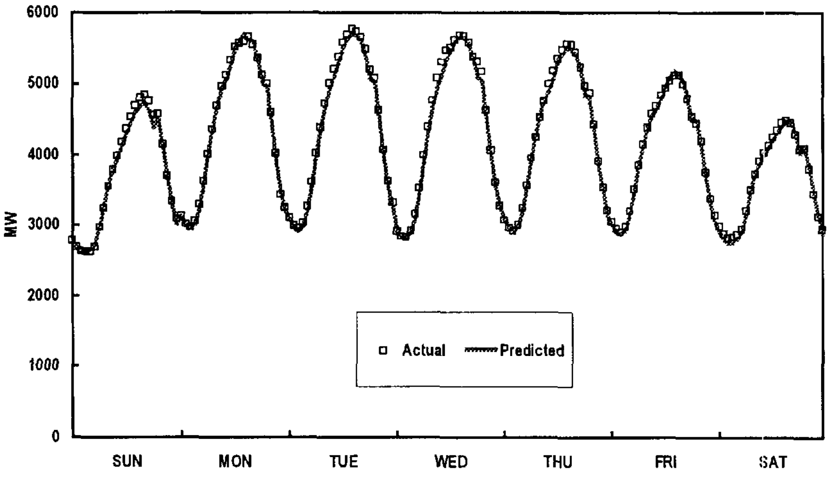
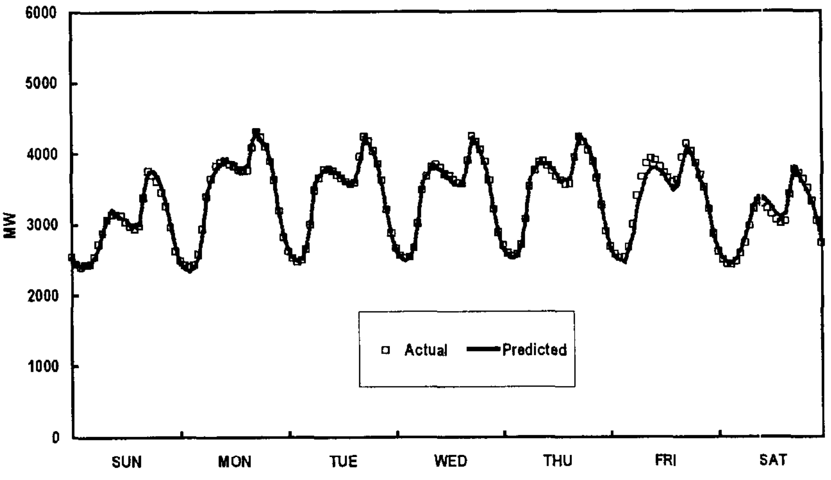


Figure 9. 1993 Winter Peak Week



7. CONCLUSION

Artificial neural networks provide a flexible nonlinear modeling framework. These models are well suited to short-term forecasting tasks, such as hourly system load forecasting for electric utilities, where there are important nonlinearities and variable interactions.

There is a danger, however, that the approach will be treated as a black box, and something that is difficult to understand. As we have shown here, these models have much in common with standard econometric approaches, and can be discussed in terms that are familiar to individuals who are comfortable with structural econometric forecasting. Also, the models can be evaluated and compared based on a variety of standard model statistics, such as R squares, MAD and MAPE values, and Durbin-Watson statistics.

The key difference is in the flexibility of these models. Estimation of parameters is a bigger task because the training exercise amounts to a search for a useful functional form. That is, for a given number of nodes, training involves the search for a set of nonlinearities and interactions that provide the best model fit to the historical data. The objective function for error minimization is exceedingly complex, because there are a large number of solutions that are locally optimal and that fit the data well. As a result, it is necessary to search the parameter space to find a good solution—one that works well both in sample and in reserved test periods.

As long as the number of nodes is kept to a reasonable level, the result is a forecasting model that is powerful, robust, and sensible. This method is useful for forecasting and, at least in our experience, it produces models that are slightly more accurate than a wide variety of regression model specifications. The neural network approach can also be used for exploratory analysis related to functional form. Specifically, examination of the pattern of estimated slopes and elasticities can be used to explore for important nonlinearities and variable interactions, and these insights can be used to strengthen regression model specifications.

REFERENCES

Goffe, William L., Gary D. Ferrier and John Rogers (1994). "Global Optimization of Statistical Functions with Simulated Annealing." *Journal of Econometrics* 60(1,2): 65-99. Jan/Feb.

Kuan, Chung-Ming and Halbert White (1992). "Artificial Neural Networks: An Econometric Perspective." University of California, San Diego, Department of Economics, Discussion Paper 92-11 (February).

Masters, Timothy (1995), *Advanced Algorithms for Neural Networks*, New York: John Wiley & Sons.

IMSL Stat/Library, Fortran Subroutines for Statistical Applications (1994). Boulder Colorado: Visual Numerics, Inc.

Khotanzad, Alireza, Rev-Chue Hwang and Dominic Maratukulam (1993). "Hourly Load Forecasting by Neural Networks." *Proceedings of the IEEE PES Winter Meeting*, February. Ohio.

Papalexopoulos, Alex D., Shangyou Hao and Tie-Mao Peng (1994). "An Implementation of Neural Network Based Load Forecasting Models for the EMS." *IEEE Transactions on Power Systems,* 9(4), November.

Rummelhart, D.E., B.E. Hinton, and R.J. Williams (1986). "Learning Internal Representations by Error Propagation." In D.E. Rumelhart and J.L. McClelland (eds.) *Parallel Distributed Processing: Explorations in the Microstrucutres of Cognition*. Cambridge, Mass: MIT Press.

White, Halbert (1989). "Neural-Network." *AI Expert* (December).