#6524

APPLICATION OF NEURAL NETWORKING MODELS TO PREDICT ENERGY USE

M. Anstett

J.F. Kreider, Ph.D., P.E.
Member ASHRAE

ABSTRACT

This paper discusses the application of an artificial neural network model to predict energy use in a complex institutional building. A neural network has been applied to daily data collected manually by building personnel. A previously developed energy management system used linear regression and other statistical measures to develop formulas to predict the energy use for the building. The predictions and actual consumption are compared to determine whether any discrepancies exist. Discrepancies between the predicted and observed energy use are investigated to determine whether they are due to unusual weather conditions, fluctuations in building use, malfunctioning systems, or other causes.

The motivations for incorporating neural networks into this system are twofold. First is the desire to improve the predictive performance of the system. With more accurate predictions, problems with the building's energy system can be diagnosed and corrected in a timely fashion.

Neural networks can be developed that automatically update their learned knowledge over time. This provides adaptability to changes in the building's use and energy plant configuration. This automatic learning facility would reduce the amount of expert time required to analyze, build, and modify predictors of energy use.

This paper addresses the predictive performance issue. A comparison of the predictive ability of a neural network and a traditional statistical approach is presented. Neural network application issues are discussed along with results.

INTRODUCTION

A large building (UMC) at a Colorado university is used for many student activities. It has a cafeteria and bowling alley, houses many student organizations, and also provides facilities for entertainment and conferences. Operation of the building requires a substantial amount of energy resources. An energy monitoring and prediction system, called BEACON (Haberl et al. 1987; Haberl 1990) is used to aid the building operator.

The BEACON system, as applied in the UMC, is a spreadsheet program containing daily meter readings, environmental readings, occupancy measures, and formulas. The meters for electric, water, natural gas, and steam are read daily by UMC personnel. The worksheet calculates energy use predictions based on environmental and occupancy measures. These predicted energy use measures can be plotted, along with the observed uses and the differences between them. This provides the operator of the building with a meaningful representation of expected versus observed energy use. Variations are easily noted and investigated. Equipment malfunctions have been brought to the operator's attention due to this system.

The BEACON system also provides the building operator with daily energy costs. It is a useful tool in predicting what energy and economic resources will be required for maintaining the building. It aids in the process of estimating energy use costs for yearly budgets. This system, coupled with energy audits, has recommended energy conservation options that have been implemented and have reduced operational costs (Anderson et al. 1989). The system also provides a reminder to the building operator to periodically train staff to maintain an awareness of energy use. Two recent equipment problems have been pointed out by the BEACON system. Steam leaks have been brought to the building operator's attention due to discrepancies between measured and predicted values. A utility meter that had become faulty was also quickly noticed due to the BEACON system and daily readings.

In other buildings on campus, the BEACON system has been augmented with a rule-based expert system that can analyze discrepancies between the predicted and observed energy uses. This expert system provides reports with possible causes and suggested actions (Cooney et al. 1987; Haberl and Claridge 1987).

A key part of the BEACON system, and energy diagnostic systems in general, is the ability to accurately predict energy use. Formulas for these predictors are a result of many hours of expert analysis of the building data, coupled with close communication with the building's operator and staff. The result is a set of good, handcrafted predictors. For instance, the predictor for

Michael Anstett is a graduate research assistant and **Jan F. Kreider** is a professor and director of the Joint Center for Energy Management, Department of Civil, Environmental, and Architectural Engineering, University of Colorado, Boulder.

steam use is dependent mostly on whether the steam heating system is active. To create this predictor, the data were separated into two groups that have been regressed against varying degree-days to determine the best fit. The other predictors are similar in their construction. They also consider whether the college is in session and what day of the week it is.

These predictors are reasonably accurate. However, there are several factors that can cause the predictors to be off. A large driving force of energy use is related to building occupancy, which varies a great deal during the year. The measures for building occupancy are the indirect measures of sales in the cafeteria and scheduled operating hours. There is no information in the data base to take special events into consideration, which may significantly impact energy use.

Energy Monitoring, Prediction, and Diagnostic Approaches

Neural networks have several advantages over traditional approaches for predicting energy use. Relative to many standard energy prediction methodologies, the neural network model may be easier to apply. Considerable expertise is usually required to apply energy models such as thermal networks (Sonderegger 1977), Fourier analysis (Shurcliff 1984), ARMA (Subbaro 1985), computer simulation (DOE-2 or BLAST), or differential equations (Rabl 1988).

Expert systems have also been applied to monitoring and predicting energy use (Anderson et al. 1989; Haberl and Claridge 1987; Kreider et al. 1990a, 1990b). Hourly direct monitoring of sensors on energy system components and environmental conditions can be made (Kreider et al. 1990a, 1990b). The expert system applies rules to determine whether any mechanical problems have occurred and what the likely cause is. The initial investment for installing sensors, a computer data logger, and an analytical system can be recovered by savings from improved operation. Such a system is very sophisticated in its approach but requires an intensive effort to create the expert rule system. Although much of the expert system knowledge is adaptable to different building configurations, a substantial core of knowledge is building-dependent. This points out a problem with expert systems in general. They can be brittle and performance may not degrade gracefully when their knowledge base does not have the correct rule to apply.

Energy predictors are often based on statistical or numerical modeling of historical data. It is a time-consuming job to create the formulas to calculate these predictors. Some systems have semi-automated updating of these formulas (Anderson et al. 1991). This automation becomes more difficult to achieve if major changes occur in building use, or changes in the energy plant operation or configuration are made.

Artificial Neural Networks

Artificial neural networks (ANNs) are simplified mathematical models of biological neural systems. There are many types of ANNs that model different neural functions. As in biological systems, a great deal of parallelism exists. It is this parallelism that allows many simple actions to combine and produce a complex output. ANNs use simple processing units to combine data and store relationships between independent and dependent variables.

The artificial neural network requires parameters to be sampled, which reasonably model the building's energy use environment. A significant amount of time may be needed to train the network, but this can be done unsupervised by computer. Neural networks can also be set up to continue learning, once they are in use. The system can automatically adapt to new equipment or other changes in the building's energy system.

One facet of neural networks is that a statistical understanding of the relationships between the independent and the dependent variables is not needed. However, as with any modeling method, improved performance for a network can be expected when well-chosen independent variables are used. Contribution analysis of the independent variables can lead to well-chosen network inputs. An advantage of neural networks is their ability to degrade more gracefully at the edge of their knowledge base when compared to some rule-based solutions. They often give reasonable results when some data are missing or are in error.

Neural networks have certain disadvantages. Their knowledge is stored implicitly in the internal data structures of the net and is not readily available for inspection or interpretation. A certain amount of understanding of neural network models is required to properly apply them. It needs to be stated that neural networks are not entirely deterministic in nature and can exhibit random behavior. They do not always converge to an acceptable solution. Neural networks are computationally expensive to train.

A hybrid system containing both an expert system and a neural network component may be the best solution for overcoming the limitations of the individual approaches. We will be concentrating on the neural network component.

The ANN model that was used displays this parallel behavior and is called a pattern associator. It takes a pattern of data for input and learns an association between the input pattern and a desired output pattern. The learning is embodied in the internal structure of the network. The network is composed of three building blocks—units, weights, and biases (Figure 1). Weights are the interconnections (synapses) between units (neurons), and biases act as thresholding elements for units. Weights and biases are made to vary over time in reaction to supervised training, and to store the associations

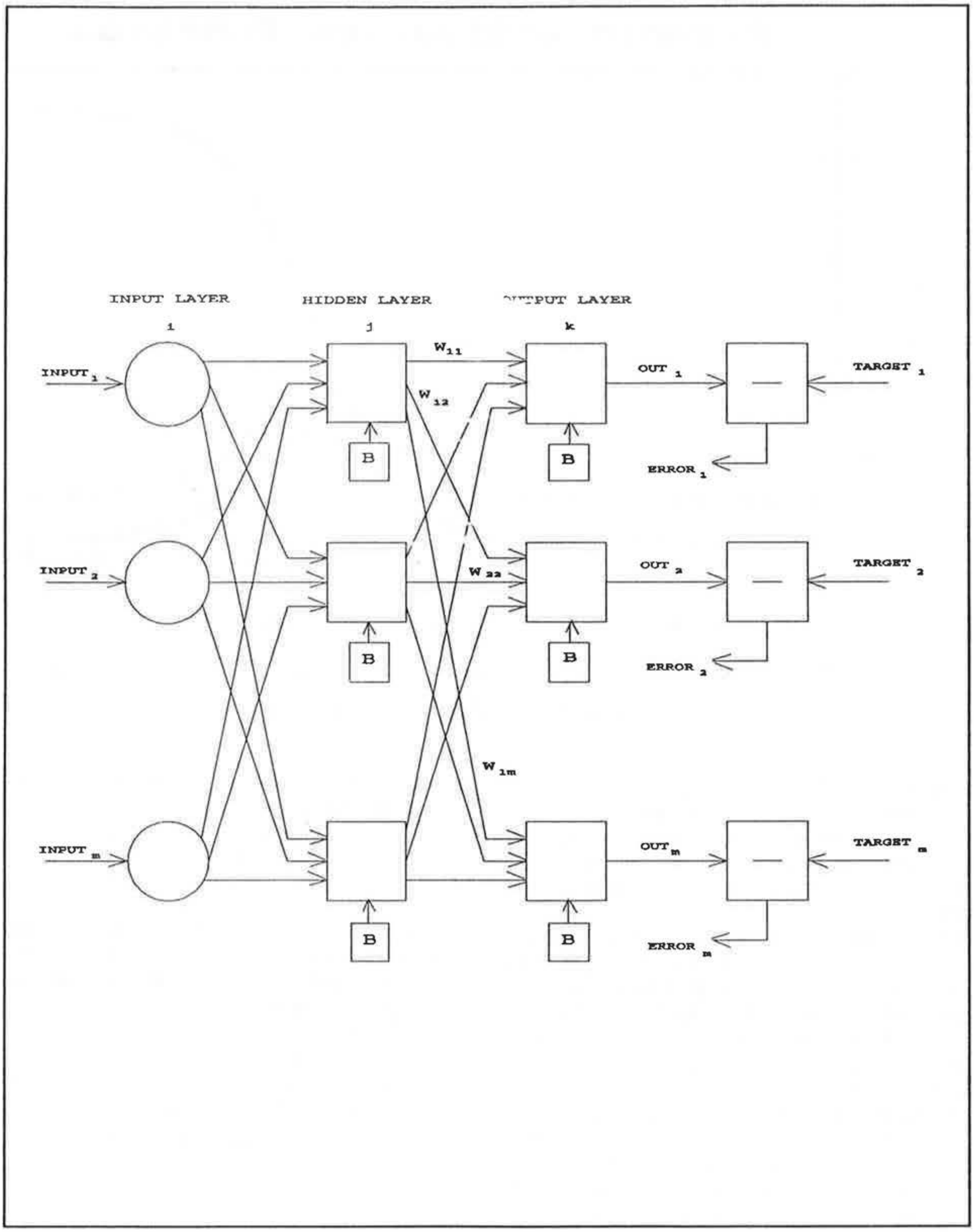

Figure 1 *Artificial neural network diagram (from Wang 1991).*

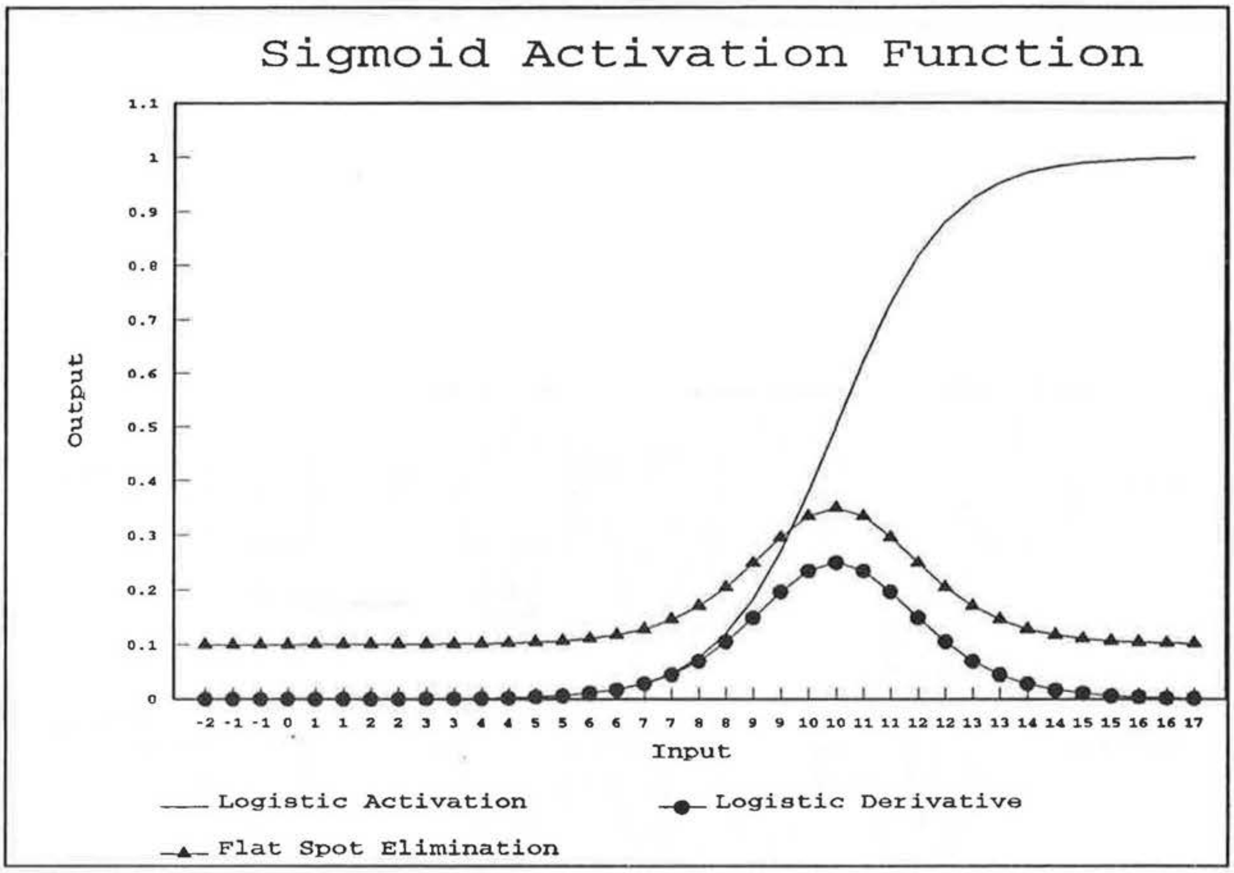

Figure 2 *Sigmoid activation function, its derivative, and flat spot elimination modification.*

learned by the network. Units act as processors, performing a combination function over their inputs. This combined input value, the unit's net input, is acted upon by an activation function to produce an output from the unit. The output of one unit into another is weighted by the weight between the units.

There are three types of units in our pattern associator network—input units, hidden units, and output units. They are arranged in layers from the input layer, to one or more hidden layers, to the output layer. Data are presented to the input units that propagate information to successive hidden layers until data exit the network through the output units.

Each input unit brings a single data element of a pattern into the network. Input units do not have bias inputs, combination functions, or activation functions. The activation level, or output of the unit, is identical to its input. The values brought into the net correspond to the independent variables of our energy prediction analysis.

Each hidden and output unit has a bias input. They also receive input from each of the nodes in the preceding layer. A combination function is used to determine the net input to each of the hidden and output units (Equation 1). The combination function for a unit takes the activation of each unit (a_j) that sends it input and weights it by the weight (w_{ij}) between that unit and itself. It sums these inputs along with the bias contribution.

Combination function:

$$\text{net}_i = \sum_{j=1}^{N} a_j\, w_{ij} + \text{bias}_i. \tag{1}$$

All hidden and output units use the same activation function to determine their activation level or output. A sigmoidal-shaped function, the logistic function (Equation 2), is applied. This function is nonlinear and easily differentiable (Figure 2). The logistic function is used to constrain the output of the unit to a desired range. It also performs smoothing, and its derivative (Equation 3) is used in the learning mode, which will be discussed later.

Sigmoidal activation function:

$$a_{pi} = \frac{1}{1 + e^{-\text{net}_{pi}}}. \tag{2}$$

Derivative of activation function:

$$\frac{da_{pi}}{d\,\text{net}_{pi}} = a_{pi}(1 - a_{pi}). \tag{3}$$

The activations of the output units correspond to the dependent variable predictors. The network maps the independent variables, which are the inputs to our network, to these output predictors. In order for the weights and biases to be modified to perform this mapping correctly, the network needs to be trained.

Training applies a technique, called "back propagation," to systematically modify the network. Learning proceeds by applying the independent variable data, one pattern at a time, to the inputs of the network. The data are propagated through the network to the output units. The output activations (a_{pi}) are compared to target values of the dependent variables (t_{pi}) and an error term is calculated for each pattern (Equation 4).

Error function:

$$E_{pi} = t_{pi} - a_{pi}. \tag{4}$$

The aim of the training process is to reduce this error term, which is accomplished by back propagating the error through the net, beginning with the output units. The weights and biases in the system will be changed in proportion to the amount they contributed to the error, in order to minimize this error. The partial derivative of the error function, with respect to every weight in the network, will be computed (Equation 5). Each weight will be changed according to the generalized back propagation rule (Equation 6), where ε denotes a constant learning rate and a_{pj} is the activation of unit j in pattern p.

Weight error derivative:

$$\Delta w_{ij} = -K\frac{\partial E}{\partial w_{ij}}. \tag{5}$$

Back propagation rule:

$$\Delta w_{ij} = \varepsilon\ \delta_{pi}\ a_{pj}. \tag{6}$$

The change in weights refers to the weight from unit j to unit i. The term δ represents the effect of change in the net input to unit j on the activation of unit i in pattern p. The δ term is computed in two different ways, depending on whether the unit is an output unit (Equation 7) or a hidden unit (Equation 8). Output units use the difference between the target value and the output activation. This difference is multiplied by the derivative of the activation function with respect to a change in its input.

Delta error, output unit:

$$\delta_{pi} = a_{pi}(1 - a_{pi})(t_{pi} - a_{pi}). \tag{7}$$

Delta error, hidden unit:

$$\delta_{pi} = a_{pi}(1 - a_{pi})\sum_k \delta_{pk}\, w_{ki}. \tag{8}$$

A specific target value is not available for the hidden units. It is derived by summing the back-propagated deltas from all units directly connected to this hidden unit, times the respective weights between the hidden unit and the directly connected units.

The change to the weights can either be applied after each pattern or accumulated and applied after each epoch, which is the entire set of training patterns. This same back-propagation learning technique can also be applied to the biases in the network.

Prior to training the network, the weights and biases have been initialized to small random values. This random initial state affects the training time and the accuracy of the resultant network. It is necessary for the weights to be set to different initial values in order for the back propagation method to work.

The opposing goals of minimizing learning time and maximizing predictor accuracy need to be dealt with. Much effort is being applied to develop techniques to

TABLE 1
Energy Prediction Independent and Dependent Variables

Inputs	**node**	**Independent Variables**
	0	Month of year, (1-12).
	1	Day of month, (1-31).
	2	Day of week, (0-6).
	3	Scheduled number of operating hours.
	4	High Temperature Outside in degrees Fahrenheit.
	5	Low Temperature Outside in degrees Fahrenheit.
	6	Average Temperature Outside in degrees Fahrenheit.
	7	City Water Temperature in degrees Fahrenheit.
	8	Grill sales in dollars.
	9	Catering sales in dollars.
	10	Building Heat, 0 = off, 1 = on.
Outputs	**node**	**Dependent Variables**
	0	Steam in lbs.
	1	Electric in kwh.
	2	Natural Gas in CCF.
	3	Water in gallons.

optimize the back-propagation learning method (Falman 1988; McClelland and Rumelhart 1988; Wilson 1991). Three of these techniques are applied to minimize learning time. The first technique uses change of weight information, from the previous learning pass (Equation 9) (McClelland and Rumelhart 1988), to alter our basic back-propagation rule. A new parameter, momentum (α), has been added. This allows the learning rate to be increased in most cases. This equation is the default learning method.

Modified back-propagation rule:
$$\Delta w_{ij}(n+1) = \varepsilon\, \delta_{pi}\, a_{pi} + \alpha\, \Delta w_{ij}(n). \tag{9}$$

The second technique, called "flat spot elimination" (Falman 1988), alters the derivative of the sigmoidal activation function to minimize a problem called "sticking weights." The derivative of the logistic function goes to zero as the unit's output approaches 0.0 or 1.0. The formula is altered by adding a constant .1 to the derivative of the sigmoid function (Figure 2). This results in a useful error signal being back propagated, even when the derivative of the sigmoid function is approaching zero.

The third technique applied to decrease learning time is a heuristic called "descending epsilon" (Wilson 1991). The number of presentations required for the network to learn an input pattern varies from pattern to pattern. Some patterns are learned early in the training cycle. Descending epsilon determines when to stop training a pattern based on when the error signal for that pattern goes below a certain threshold. This technique has been

TABLE 2
Neural Network Parameters and Configurations

Network configurations.	Vary number of hidden layers. Vary the number of hidden units per layer.
Normalization Interval.	(0.0 - 1.0) versus (-.9 - .9)
Training pattern order.	Random versus Sequential.
Weight update schedule.	After each pattern, after each Epoch.
ε - learning rate.	Vary learning rate.
α - momentum.	Vary momentum.
Flat spot elimination.	Compare to default back propagation.
Descending Epsilon.	Compare to default back propagation.

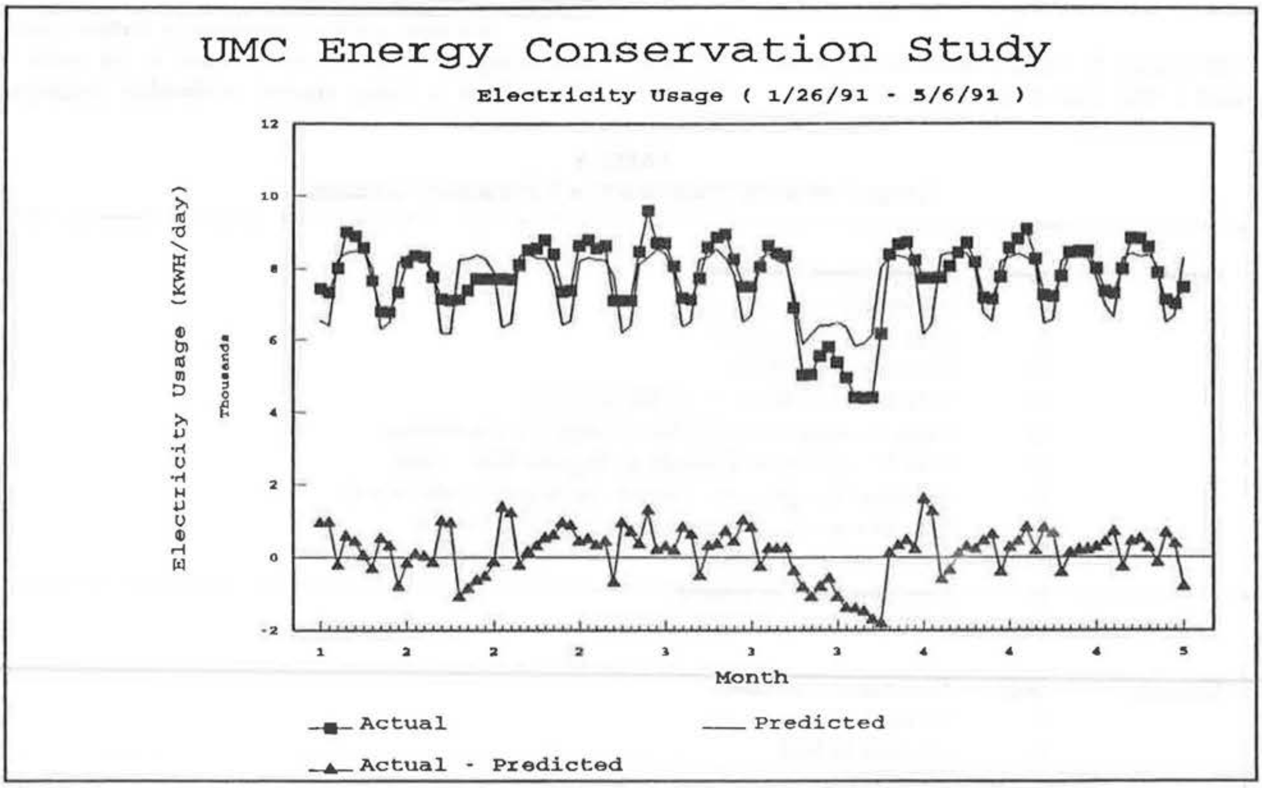

Figure 3 Neural network prediction of electrical consumption.

modified here to periodically check the patterns to make sure the error is still below the threshold. If the error goes above the threshold, then training resumes on that pattern.

METHODS

An artificial neural network (ANN) simulator was developed in C++ on a 80×86 computer system to allow different types of neural networks to be trained and evaluated. The simulator employs the standard back-propagation learning method (McClelland and Rumelhart 1988). In addition, it supports flat spot elimination (Falman 1988) and descending epsilon (Wilson 1991). Multiple networks were trained and evaluated using different network configurations and parameter settings. The input and target data need to be interval scaled to a range usable by the network (0.0 to 1.0). The predictors created by the network undergo a reversal of this interval scaling process.

Training of a network proceeds until certain criteria are met. Three criteria were used, and training continued until one of the criteria was met. Two of the criteria use the measured root mean square error (RMS) (Equation 10). RMS performs a summation of the error over the entire N patterns. The first criterion trains the net until RMS goes below a threshold level of .05. The second criterion trains the net until RMS remains constant within .0001 for 10 consecutive epochs and a minimum number of epochs have been trained. The third and default criteria are to end training when a maximum number of epochs have been trained.

$$\text{RMS error:} \quad \text{RMS} = \sqrt{\frac{\sum_{i=1}^{N} (t_i - a_i)^2}{N}} . \tag{10}$$

The following measures were used to rate the different networks: RMS error, maximum error, fraction of variance (r^2) (Equation 11), elapsed training time, and the number of epochs trained. Maximum error is simply the largest error noted while testing a set of patterns. An r^2 of 1.0 denotes perfect prediction.

$$\text{Fraction of variance explained:} \quad r^2 = 1 - \frac{\sum_{i=1}^{N} (t_i - a_i)^2}{\sum_{i=1}^{N} (t_i)^2} . \tag{11}$$

Description of Training Set Data

The data used for training came from the BEACON data set covering the time period of 7/1/89 to 11/15/90 for a total of 503 data patterns. Table 1 shows the independent variables and the energy use predictors that were generated. The data were viewed graphically to find and correct data entry errors. A number of parameters and configurations of the network were evaluated (Table 2).

The first network trained contained no hidden units. The configuration was (11:4), which denotes 11 input units and 4 output units. This was done to compare the relative contributions that each independent variable had on the energy predictors. This is only an approximation because a neural network with no hidden layers is not able to map many relationships between variables. The weights generated by this network correspond to the expectations of which variables contributed the most for each predictor (Table 3). The absolute value of the weight denotes the relative contribution to the result. The top 3 contributing independent variables for each dependent variable, are denoted in parentheses. The building uses steam for heating. Therefore, for steam use, the

TABLE 3
Weighted Contributions of Independent Variables Toward Energy Predictors
(No Hidden Layers)

	Predictors, weight and (ranking).			
Independent Variables.	**Electric**	**Water**	**Steam**	**Natural Gas**
Month of Year.	-.05	-.3	.1	.14
Day of Month.	-.2	-.3	.0	-.12
Day of week.	.18	.14	.02	-.02
Scheduled hours.	-.35	-.58	.0	-.46
High outdoor temperature.	-.21	.1	-.41	-.11
Low outdoor temperature.	-.72 (3)	-.86 (3)	-1.65 (1)	-.16
Average outdoor temperature.	-.24	-.17	-1.09 (3)	.04
Water Temperature.	.35	.77	-.92	-.68 (3)
Cafeteria Sales.	1.25 (1)	2.07 (1)	.48	1.96 (1)
Catering Sales.	.83 (2)	1.20 (2)	.12	1.23 (2)
Building Heat on.	-.26	-.34	1.6 (2)	-.14

dominant variables were the outdoor temperature and whether the building heat was on or off. For natural gas, electricity, and water use, the prime contributors were the indirect building use variables, the cafeteria, and catering sales.

Network Configurations

As the number of hidden layers and hidden units increases, the training time increases and the predictive ability of the network improves. We trained different configurations, which are summarized in Table 4. There is a trade-off between improved accuracy and increased training time. We found that the configuration (5:5:1) for electricity use was a reasonable compromise. Too few hidden units leads to overgeneralization and poor performance. Too many hidden units leads to a waste of computational effort and the potential to overfit the data.

The next configurations tested compared the effects of having a single network predict all of the energy use variables versus having four networks each predict one of the four variables (Table 5). The (11:11:11:4) network takes the longest to train of the five and provides four times as much information. The predictors are not as accurate as the single-variable networks. The discrepancy in accuracy is not large but is significant. Therefore, for our application, the single-network approach produces the better result.

Normalization Interval

Two different ranges were used to interval scale the data for training the network. The results coincide with measures made by Wang (1991). The measures were made using the standard back-propagation algorithm with the network configured (11:11:1) for electricity use (Table 6). For our data set, we found the range (.1 to .9) provided the better results. Training time was shorter for the same level of accuracy. Had a different activation function been used, such as a hyperbolic arctangent, the effective range that could be used for normalization would be increased. This would allow the training time to be decreased further (Falman 1988).

Training Pattern Order, Weight Update Schedule

Two methods were used for applying the patterns for training, sequential order, and random order. Random ordering changes the order of [illegible] after each epoch. The weights and biases in the network can be updated after each training pattern or the change can be accumulated and applied after each epoch. Measures were taken using the standard back-propagation algorithm and the network (11:11:11:4) (Table 7). The training time was significantly shorter when updating the weights after each epoch. We noted a larger variance in the final RMS for networks trained with this method than with weight

TABLE 4
Effect of Different Number of Hidden Layers and Hidden Units

Configuration	Training Time(min.)	Max. Error	Epochs	RMS	r^2
(5:1)	5	.31	500	.0912	.942
(5:5:1)	19	.29	500	.0902	.962
(5:5:5:1)	32	.3	500	.0915	.959
(5:3:3:1)	19	.3	500	.0932	.957
(5:5:5:5:1)	49	.35	522	.0949	.951

TABLE 5
Comparison of Single versus Multiple Output Networks

Configuration		Training Time(min.)	Max. Error	Epochs	RMS	r^2
(11:11:11:4)	water	167	.61	1000	.17	.98
	steam					.96
	electric					.97
	natural gas					.98
(11:11:11:1)	water	70	.447	500	.092	.96
(11:11:11:1)	steam	70	.46	500	.10	.95
(11:11:11:1)	electric	70	.46	500	.89	.96
(11:11:11:1)	natural gas	71	.47	500	.10	.96

TABLE 6
Comparison of Normalization Ranges

Range	Training Time(min.)	Max. Error	Epochs	RMS	r^2
.1 to .9	60	.35	500	.069	.973
0.0 to 1.0	61	.44	500	.087	.970

TABLE 7
Pattern Presentation Order, Weight Update Schedule

Presentation Order	Weight Update	Training Time(min.)	Max. Error	Epochs	RMS	r^2
Sequential	Epoch	167	.6	1000	.169	.97
Random	Epoch	167	.62	1000	.172	.97
Sequential	Pattern	263	.58	1000	.169	.97
Random	Pattern	263	.53	1000	.167	.97

TABLE 8
Effects of Learning Rate and Momentum Natural Gas Predictor (4:4:1),
Standard Back Propagation

Learning Rate	Momentum	Training Time(min.)	Max. Error	Epochs	RMS	r^2
.001	.9	14	.44	500	.137	.92
.005	.9	14	.42	500	.087	.968
.01	.9	14	.45	500	.087	.967
.05	.9	14	.4	500	.086	.95
.001	.7	14	.45	500	.143	.901
.005	.7	14	.42	500	.093	.96
.01	.7	14	.47	500	.087	.967
.05	.7	14	.46	500	.086	.963
.001	.5	14	.46	500	.146	.9
.005	.5	14	.42	500	.136	.92
.01	.5	14	.42	500	.089	.96
.05	.5	14	.44	500	.086	.96

TABLE 9
Flat Spot Elimination, Descending Epsilon
Standard Back Propagation

Method	Descending Epsilon	Training Time	Maximum Error	RMS	r^2
Back Propagation	ON	12	.46	.087	.969
Back Propagation	OFF	14	.46	.087	.969
Flat Spot Elimination	ON	12	.46	.088	.969
Flat Spot Elimination	OFF	14	.46	.086	.969

updates after each pattern. For our data set, random ordering of the application of patterns appeared almost equal to sequential ordering.

Learning Rate—ε, Momentum—α

The learning rate parameter, ε, can significantly impact how long a network takes for training and whether it will converge. The higher the value of ε, the more likely the network is to oscillate. If ε is too small, learning accuracy can suffer. The learning rate and momentum are closely related. To find optimal settings for these two parameters, a series of networks (4:4:1) was trained using standard back propagation (Table 8). As the momentum increases, the learning rate can also be increased. The main effect of increasing the learning rate is to lower the training time. Another effect is the loss of precision as ε approaches an oscillatory state. We found that a good compromise learning rate of .05 accompanied by a momentum of 0.9 worked well.

Flat Spot Elimination, Descending Epsilon

Flat spot elimination increases the ability of the network to back-propagate a useful error signal. We found that it compares well with standard back propagation. In general, we found a small decrease in training time with no loss of accuracy. Descending epsilon was a difficult procedure to tune. The results using this procedure show the training time is decreased but a certain amount of accuracy has been sacrificed. Table 9 summarizes our results when using a network (4:4:1) for natural gas. The learning rate is .01 and momentum is 0.9.

RESULTS

The data used for testing came from the BEACON data set and represented the period of 11/16/90 to 1/16/92. This set of data underwent the same interval scaling as our training data. The results we are presenting are from four networks trained with the flat spot elimination method. The result of a trained network is the weights and biases. To load a network, one merely loads a weight table that embodies the results of the training phase. These networks were trained with a learning rate of .02 and a momentum of .9 for 500 epochs with a total training time of 20 minutes. The results are graphically presented in Figures 3 through 6. As can be seen, the predictive quality of these nets is good.

Electricity use was modeled well with a training RMS of .07 and r^2 of .969 (Figure 3). For the test data, r^2 was .98. There was one period, shown in the graph, where the prediction is high for about a week. This corresponds to the spring break when classes are not in session. The neural network was unable to model it. The variables used for this (5:5:1) network were scheduled

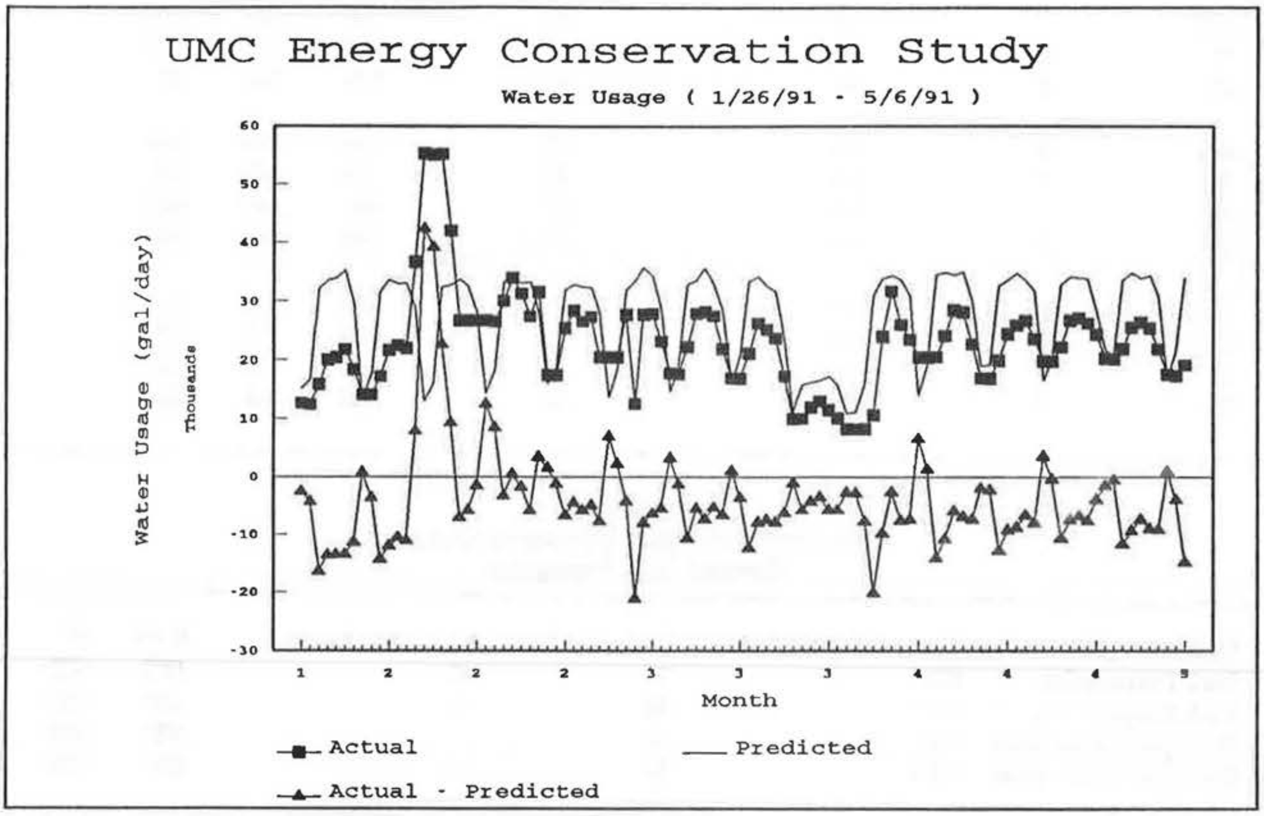

Figure 4 Neural network predicted water consumption.

hours, low outdoor temperature, water temperature, catering sales, and cafeteria sales.

The predictions for water use (Figure 4) are not very accurate. The network (5:5:1) trained to an RMS of .093 and an r^2 of .956. However, the r^2 for the testing period dropped off dramatically to .84. The predicted values are consistently high. This predictor may be improved by increasing the number of units in the network. The variables used for this network were the same as those used in the electricity use network.

The network for steam use trained to an RMS of .057 and an r^2 of .972 (Figure 5). The r^2 for the testing phase was .967. The predictions are fairly accurate. There is a time lag between the outside temperature and the demand for steam. This predictor could probably be improved by using a measure of the previous days' temperatures in training the network. The (5:5:1) network used for steam had the temperature variables of building heat on/off and the day of week for inputs.

The natural gas network was trained to an RMS of .087 and an r^2 of .964. The variables used for the (4:4:1) network were cafeteria and catering sales, the water temperature, and the scheduled hours. Some periods were modeled well for the time frame shown in Figure 6. The natural gas predictor might also be improved by a different network topology and increased training time.

A comparison was made between the existing statistical predictor for electrical consumption and a neural network predictor. The statistical predictor was created by binning historical electrical use data into seven bins corresponding to the days of the week. The mean for each bin was computed to use as the electrical consumption predictor.

The neural network was assembled using two variables—sales in the cafeteria and catering sales. The network was trained for 1,000 epochs using a learning rate of .15, a momentum of .9, and a configuration of 2:2:1. The methods were compared using February 1990 for the training set and the following month for the test set. The r^2 for the neural network on the test set was .7 compared to an r^2 of .3 for the bin-mean method. The RMS error and maximum error were also significantly less for the neural network predictor.

CONCLUSIONS

Flat spot elimination was a useful modification and required little work to implement. Descending epsilon increased the instability of the training and was not tunable for our application. We found that a single network was the best for accuracy versus the four-network approach, where each predictor was created by a separate network. As the number of variables used to train the network increases, the generality of the network increases. Therefore, it is best to determine the pertinent independent variables and use them for training the network.

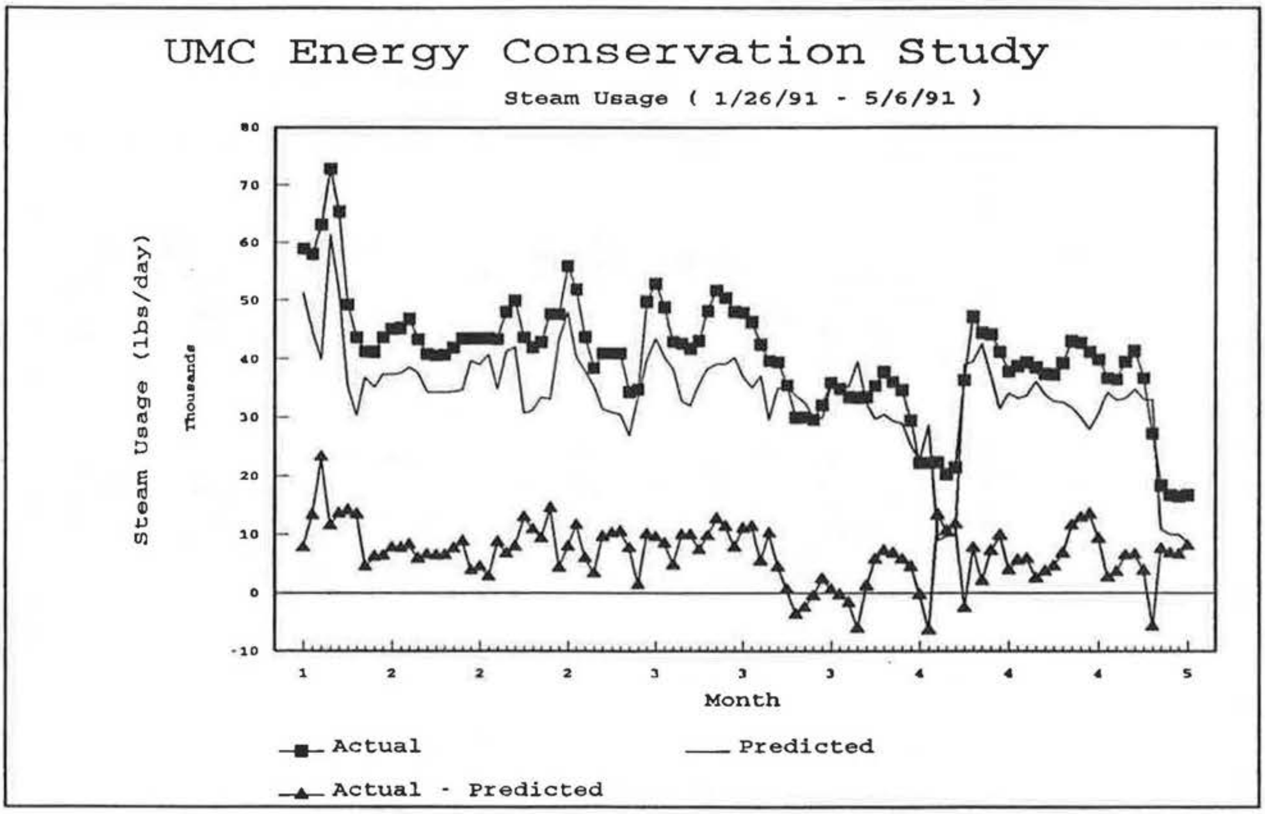

Figure 5 Neural network predicted steam consumption.

The normalization interval (.1 to .9) worked best for our application. Updating weights after each epoch showed an improvement by reducing training time. The learning rate and momentum need to be set in order to achieve a balance between minimum training time and predictive accuracy.

Neural networks are useful for predicting energy consumption in buildings. The UMC BEACON system has been augmented with neural network predictors for electricity, steam, water, and natural gas use. They can be viewed graphically with plots similar to those in Figures 3 through 6. A neural network program can be incorporated into a spreadsheet. The weights and biases that are the result of training can be loaded as a table and updated periodically if building parameters change. Neural network predictors can be as accurate as handcrafted predictors.

Future pursuits may include application and comparison of more efficient and accurate learning algorithms (Falman 1988) or adaptive learning algorithms. In addition, algorithms related to automated learning in neural networks will be reviewed. Use of contribution analysis to aid selection of well-chosen network inputs will be assessed.

REFERENCES

Anderson, D. 1991. Energy consumption predictors based on single value decomposition. Master's thesis, University of Colorado.

Anderson, D., L. Graves, W. Reinhart, J. Kreider, J. Dow, and H. Wubbena. 1989. A quasi-real-time expert system for commercial building HVAC diagnostics. *ASHRAE Transactions* 96(2): 954-960.

Cooney, K., F. Stern, and J. Haberl. 1987. An action-oriented team approach to building energy conservation using regression techniques and expert systems. *Proceedings of Annual Winter Meeting of ASME*, Boston, December 13-18.

Falman, S. 1988. Faster learning variations on back-propagation: An empirical study. *Proceedings of the 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, T. Sejnowski, eds., pp. 38-51. San Mateo, CA: Morgan Kaufman Publishers.

Haberl, J. 1990. Memorandum on update of UMC BEACON predictors.

Haberl, J., and D. Claridge. 1987. An expert system for building energy consumption analysis: Prototype results. *ASHRAE Transactions* 93(2).

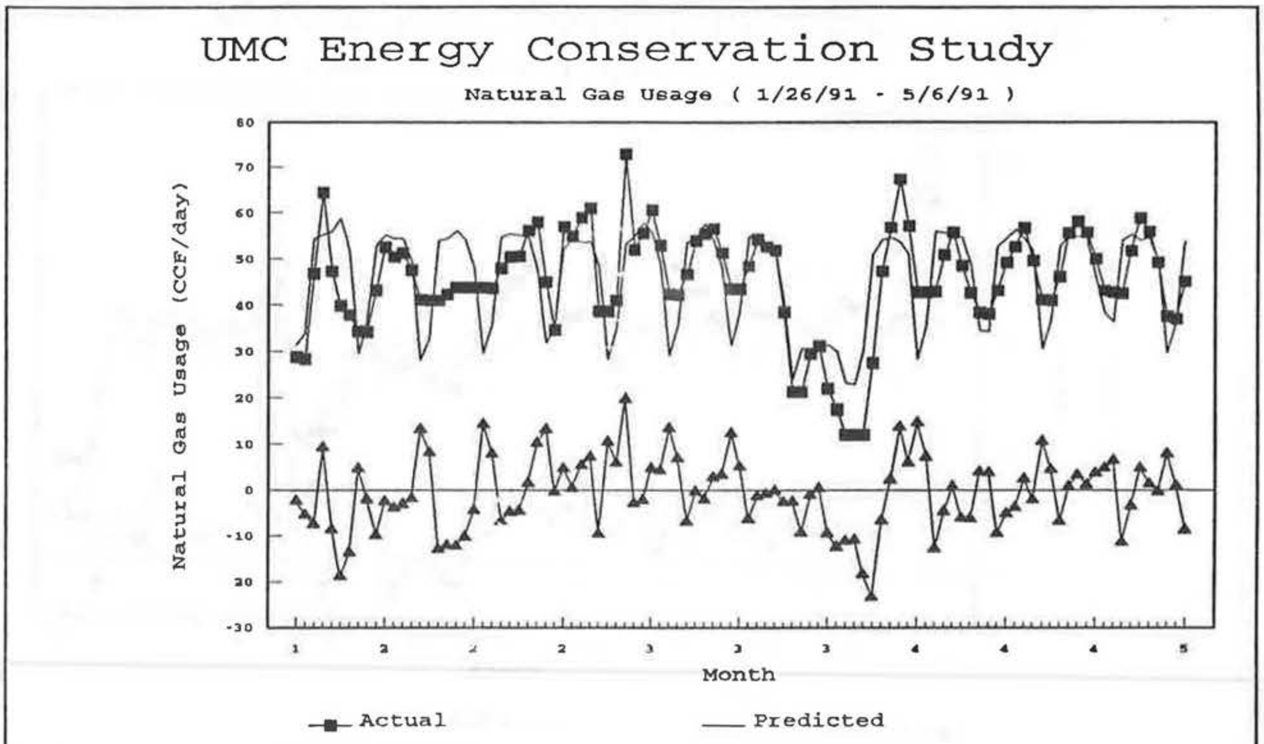

Figure 6 *Neural network predicted natural gas consumption.*

Haberl, J., L. Smith, K. Cooney, and F. Stern. 1988. An expert system for building energy consumption analysis: Applications at a university campus. *ASHRAE Transactions* 94(1).

Kreider, J., D. Anderson, and J. Dow. 1990a. Expert systems for commercial building HVAC diagnostics. DSC—Vol. 23, *Intelligent Control Systems*.

Kreider, J., K. Cooney, L. Graves, K. Meadows, F. Stern, and L. Weilert. 1990b. An expert system for commercial building HVAC and energy audits—A progress report. *ASHRAE Transactions* 96(1).

Kreider, J., and X. Wang. 1991. Artificial neural networks demonstration for automated generation of energy use predictors for commercial buildings. *ASHRAE Transactions* 97(1).

McClelland, J., and D. Rumelhart. 1988. *Explorations in parallel distributed processing*, QA 76.5.M295.

Rabl, A. 1988. Parameter estimation in buildings: Methods for dynamic analysis of measured energy use. *Transactions of the ASME*, Vol. 110, February.

Wilson, J. III. 1991. Back-propagation neural networks: A comparison of selected algorithms and methods of improving performance. *2d Workshop of Neural Networks*, pp. 39-46 QA 76.87 W7.