10.3 Short Term Electric Load Forecasting Using Neural Networks

This paper was published in Proceedings of the Iowa State University Electric Power Research Center Power Affiliates Research Program, May 1993.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
SHORT-TERM ELECTRIC LOAD FORECASTING
USING NEURAL NETWORKS

Eric Daugherty
Eric Bartlett

ABSTRACT

Short-term electric load forecasting (STELF) plays an important role in electric utilities, and several techniques are used to perform these predictions and system modelings. Recently, artificial neural networks (ANN's) have been implemented for STELF with some success. This paper will examine improved STELF by optimization of ANN techniques. The strategy for the research involves careful selection of input variables and utilization of effective generalization. Some results have been obtained which show that, with the selection of another input variable, the ANN's use for STELF can be improved.

INTRODUCTION

According to Weigend, Huberman, and Rumelhart, predicting the future is the prime reason that the laws for certain phenomena are so diligently sought [1]. The prediction of certain phenomena depends on one of two types of knowledge. The first type is an understanding of the laws for the given phenomena. With the laws well understood, a mathematical model can be obtained and subsequently used to make predictions. The second type applies when the underlying laws are not well understood but empirical patterns are apparent. In this case, predictions can still be obtained.

For some cases, not only are the underlying laws of a system unknown, but there is also little or no apparent empirical relation. In addition, many cases are often represented with noise, which makes any relations more difficult to obtain. For the cases where there is little or no empirical relation and where noise is present, several methods are employed to obtain models. Some of these methods include autoregressive techniques, general exponential smoothing, and Kalman filtering.

Another method that has been applied to obtain models is the use of artificial neural networks (ANN's). Lapedes and Farber have demonstrated the feasibility of implementing ANN's for prediction and system modeling [2]. Bartlett has provided more examples of the implementation of ANN's for prediction and system modeling by utilizing them for nuclear power
plant diagnostics and for the modeling of chaotic animal populations \([3,4]\). ANN's have been implemented in a wide spectrum of prediction and system modeling cases and their use for short-term electric load forecasting is becoming more prevalent.

**SHORT-TERM ELECTRIC LOAD FORECASTING**

Short-term electric load forecasting (STELF) plays an important role in the operation of an electric utility. STELF ranges from predictions involving minutes to those involving a few weeks and is performed for several reasons such as day-to-day operations and scheduling, reserve margin allocation, and economic planning. Thus, understanding the variables involved and their relation with STELF is important for accurate and reliable predictions.

For this reason, many methods have been implemented. Moghram and Rahman \([5]\) have stated the five most common methods for STELF as: (1) multiple linear regression, (2) stochastic time series, (3) general exponential smoothing, (4) state space method, and (5) knowledge-based approach. But, as mentioned before, ANN's have been implemented for STELF and are becoming increasingly popular for this task. An ANN was implemented for STELF and obtained comparable results, within a few percentage better and worse, with standard methods \([6]\).

**ARTIFICIAL NEURAL NETWORKS**

ANN's have many applications, STELF being just one. ANN's have the ability to generalize, i.e., the ability to have knowledge of a thing never encountered before based on it's similarities with things already known. ANN's are also capable of complex function mapping and noise insensitivity. These qualities are the motivation for utilizing ANN's for STELF.

ANN's are a type of artificial intelligence that were inspired from the brain \([7]\). ANN's are computer algorithms which consist of highly interconnected processing elements called neurons that produce either weak, strong, or intermediate signals based on the weighted sum of the input signals they receive. These neuron output signals are either the inputs for other nodes or the outputs of the ANN. One way the ANN obtains the correct outputs is by learning from a set of examples. This type of ANN, which uses examples or training sets, is almost always associated with feed forward back propagation (FFBP) ANN's.

FFBP ANN's are widely used today and have been the most widely applied and investigated type of ANN. Figure 1 illustrates a schematic of an ANN. The circles represent the neurons and the lines connecting the circles represent the interconnections, also referred to as
Figure 1. A neural network schematic.
weights. During the learning process, the examples, one set at a time, are presented to the ANN via the input layer, which just outputs the value. These values are then "fed forward" via the connections (i.e., multiplied by the weights) so that the neurons in the hidden layer receive a weighted sum of the input values. Using a transfer function, with the value of the weighted sum input, the neurons in the hidden layer produce an output signal. The same process is then repeated for the top set of weights and the output layer. The values from the output layer represent the system output for the ANN. Thus, the examples are fed forward to examine the ANN output with the desired output from the example.

Since the neurons need to produce either a weak, strong, or intermediate value, a sigmoid transfer function is most commonly used. This function is chosen so that the output can be close to one, close to zero, or somewhere in between, depending on the value of the summed input. This allows the neuron to have varying levels of excitation for its output. The sigmoid function is

\[ f = \frac{1}{1 + \exp(-s \cdot t)} \]  

(1)

where \( t \) is some gain (usually 1) and \( s \) is the weighted sum of the inputs to that neuron.

The ANN then uses the difference between the system output and desired output and employs the delta rule to adjust the weights [8]. The weights are adjusted so the ANN "learns" the examples. Thus, the errors are propagated back through the ANN, by using the delta rule. This process of feed forward and back propagation is repeated until the ANN output is within some desired error criteria with the examples. An RMS is usually defined for error examination. The delta rule is given by

\[ w(\text{new}) = w(\text{old}) + B \cdot d \cdot x/lxl^2 \]  

(2)

where \( d \) is the difference between the neuron output and the desired output, \( x \) is the neuron input, and \( B \) is defined as the learning constant. But since the desired outputs of the neurons in the hidden layer are not known, the difference cannot be computed. For this reason, the delta rule cannot be used to adjust the weights between the hidden layer and the input layer, and for this case the generalized delta rule is utilized [9].

For the hidden layer, the generalized delta rule gives a value of the difference, \( d \), for use in Equation (2) and is given by

\[ d = f' \cdot ES \]  

(3)
where ES is the sum of the output neuron differences multiplied by the weights connecting a hidden node to the output neurons. In Figure 1, for hidden neuron (2,1), the sum would include the two weights that emanate from it to the two output neurons. The derivative \( f' \) is given by

\[
f' = f \cdot (1 - f)
\]  

(4)

where \( f \) is the sigmoid transfer function.

It should also be noted that ANN’s can have any number of input, hidden, or output neurons. ANN’s can also have more than one hidden layer. Some other features include use of different learning parameters for different layers, use of gains, and even the use of different transfer functions. These examples are just a few variations that have made FFBP ANN’s more useful in some cases and more complex in others.

**DESCRIPTION OF EXPERIMENT**

Although ANN’s have had success for prediction and system modeling, there are some problems associated with them. First of all, back propagation (BP) tends to converge slowly. Secondly, since BP is a gradient-descent method, the problem of reaching a local minimum instead of a global minimum is a possibility [9]. Some methods for dealing with this include starting from a new position (i.e., training with a new set of weights) and/or using a momentum term with the delta rule. Also, depending on the situation, ANN’s might require a large number of examples for convergence. Even with these drawbacks, FFBP ANN’s are relatively simple to implement and have been successful in a wide variety of applications.

For STELF, there are special aspects to consider. First of all, the generalization performed by ANN’s is very important so that the STELF is accurate and reliable. If ANN’s become large with too many hidden nodes, they will tend to memorize the examples instead of learning them. This could cause a problem when the ANN is utilized for problems other than the learned examples. The ideal of generalization points out the need for input selection because the best generalization will give the best mapping. Inclusion of the most important inputs will also lead to a more accurate generalization which, in turn, will give the ANN more reliable results.

The experiments performed for this research will deal with optimum ANN’s for STELF by examination of generalization techniques and consideration and selection of input variables. Also, ANN’s that deal with inherent aspects of STELF, like chaotic and multivariate time series, will be examined and implemented to determine if they provide better results when applied to STELF.
RESULTS

An ANN was trained using a basic time series for one-hour ahead predictions. The input consisted of 12 sequential hourly loads and the output was the next hour or the 13th hour. The ANN examples were the load curves for the month of November 1991. Thus, for 12 inputs with 1 output there were 708 examples, since November has 30 days, each with 24 hours. Figure 2 illustrates the results of the ANN, and the RMS was 0.04539. Next, the ANN was tested on a different month in a different year to examine the generalization characteristic. Figure 3 illustrates the results of this trial, and the RMS was 0.1087. Since the RMS was higher, the generalization characteristic tended towards memorization.

Another ANN was trained on the same set of examples in which the first network trained. This ANN was also a time series with 12 inputs, but another input was included to indicate the day of the predicted load. This ANN trained to an RMS of 0.05213 and is illustrated in Figure 4. Even with a higher RMS during training, the test of this ANN on the "test" month gave an RMS of 0.09246, which was lower than the ANN with the lower training RMS. The testing of the second ANN is shown in Figure 5. Thus, the inclusion of the extra variable improved the generalization characteristic.

CONCLUSIONS AND FUTURE WORK

ANN's applied to STELF have provided good results and could possibly prove to be a comparable method with Moghram and Rahman [5]. Early results indicate that optimum ANN's can increase the effectiveness for STELF. Future work will include the use of larger training sets, the further investigation of input variables, and the examination of generalization techniques that should provide even better results for STELF.

BIBLIOGRAPHY


Nov. 2, 1991 to Nov. 10, 1991
Electric loads

Figure 2. ANN time series prediction for training set.

June 18, 1989 to June 24, 1989
Electric loads

Figure 3. ANN time series prediction for test set.
Nov. 2, 1991 to Nov. 10, 1991
Electric loads

![Graph showing electric loads from Nov. 2, 1991 to Nov. 10, 1991 with labeled axes and lines for Actual, ANN output, and Abs. error.]

Figure 4. ANN modified time series prediction for training set.

June 18, 1989 to June 24, 1989
Electric loads

![Graph showing electric loads from June 18, 1989 to June 24, 1989 with labeled axes and lines for Actual, ANN output, and Abs. error.]

Figure 5. ANN modified time series prediction for test set.


