Dayalbagh Educational Institute

From the Selected Works of D. K. Chaturvedi Dr.

Summer April 12, 2010

Short-Term Load Forecasting Using Soft Computing Techniques

D. K. Chaturvedi, Dayalbagh Educational Institute

Available at: https://works.bepress.com/dk_chaturvedi/43/
Short-Term Load Forecasting Using Soft Computing Techniques

D. K. Chaturvedi¹, Sinha Anand Premdayal¹, Ashish Chandiok²
¹Department of Electrical Engineering, D. E. I., Deemed University, Agra, India
²Department of Electronics and Communication, B. M. A. S., Engineering College, Agra, India
Email: dkc_foe@rediffmail.com
Received November 10, 2009; revised December 18, 2009; accepted January 21, 2010

Abstract

Electric load forecasting is essential for developing a power supply strategy to improve the reliability of the ac power line data network and provide optimal load scheduling for developing countries where the demand is increased with high growth rate. In this paper, a short-term load forecasting realized by a generalized neuron–wavelet method is proposed. The proposed method consists of wavelet transform and soft computing technique. The wavelet transform splits up load time series into coarse and detail components to be the features for soft computing techniques using Generalized Neurons Network (GNN). The soft computing techniques forecast each component separately. The modified GNN performs better than the traditional GNN. At the end all forecasted components is summed up to produce final forecasting load.

Keywords: Wavelet Transform, Short Term Load Forecasting, Soft Computing Techniques

1. Introduction

Short-term load forecasting (STLF) is an essential technique in power system planning, operation and control, load management and unit commitment. Accurate load forecasting will lead to appropriate scheduling and planning with much lower costs on the operation of power systems [1–6]. Traditional load forecasting methods, such as regression model [7] gray forecasting model [8,9] and time series [10,11] do not consider the influence of all kind of random disturbances into account. At recent years artificial intelligence are introduced for load forecasting [12–17]. Various types of artificial neural network and fuzzy logic have been proposed for short term load forecasting. They enhanced the forecasting accuracy compared with the conventional time series method. The ANN has the ability of self learning and non-linear approximations, but it lacks the inference common in human beings and therefore requires massive amount of training data, which is an intensive time consuming process. On the other hand fuzzy logic can solve uncertainty, but traditional fuzzy system is largely dependent on the knowledge and experiences of experts and operators, and is difficult to obtain a satisfied forecasting result especially when the information is incomplete or insufficient.

This paper aims to find a solution to short term load forecasting using GNN with wavelet for accurate load forecasting results. This paper is organized as follows: Section 2 discusses various traditional and soft computing based short term load forecasting approaches. Concept of wavelet analysis required for prediction will be discussed in Section 3 while elements of generalized neural architecture needed will be described in Section 4. A prediction procedure using wavelets and soft computing techniques and its application to time series of hourly load forecasting consumption is discussed in Section 5. Section 6 includes discussion and concluding remarks.

2. Traditional and Soft Computing Techniques for Short Term Load Forecasting

2.1. Traditional Approaches

Time Series Methods

Traditional short term load forecasting relies on time series analysis technique. In time series approach the model is based on past load data, on the basis of this model the forecasting of future load is done. The techniques used for the analysis of linear time series load signal are:

1) Kalman Filters Method
The kalman filter is considered as the optimal solution to many data prediction and trend matching. The filter is constructed as a mean square minimization which requires the estimation of the covariance matrix. The role of the filter is to extract the features from the signal and ignore the rest part. As load data are highly non linear and non stationary, it is difficult to estimate the covariance matrix accurately [18].

2) Box Jenkins Method

This model is called as autoregressive integrated moving average model. The Box Jenkins model can be used to represent the process as stationary or non stationary. A stationary process is one whose statistical properties are same over time, which means that they fluctuate over fixed mean value. On other hand non stationary time series have changes in levels, trends or seasonal behavior. In Box Jenkins model the current observation is weighted average of the previous observation plus an error term. The portion of the model involving observation is known as autoregressive part of the model and error term is known as moving average term. A major obstacle here is its slow performance [19].

3) Regression Model

The regression method is widely used statistical technique for load forecasting. This model forms a relationship between load consumptions done in past hour as a linear combination to estimate the current load. A large data is required to obtain correct results, but it requires large computation time.

4) Spectral Expansion Technique

This method is based on Fourier series. The load data is considered as a periodic signal. Periodic signal can be represented as harmonic summation of sinusoids. In the same way electrical load signal is represented as summation of sinusoids with different frequency. The drawback of this method is that electrical load is not perfect periodic. It is a non stationary and non linear signal with abrupt variations caused due to weather changes. This phenomenon results in the variation of high frequency component which may not be represented as periodic spectrum. This method is not suitable and also requires complex equation and large computation time.

2.2. Soft Computing Approach

Soft computing is based on approximate models working on approximate reasoning and functional approximation. The basic objective of this method is to exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness, low solution cost and best results for real time problems.

1) Artificial Neural Networks (ANN)

An artificial neural network is an efficient information processing system to perform non-linear modeling and adaptation. It is based on training the system with past and current load data as input and output respectively. The ANN learns from experience and generalizes from previous examples to new ones. It is able to forecast more efficiently the load as the load pattern are non linear and ANN is capable to catch trends more accurately than conventional methods.

2) Rule Based Expert Systems

An expert system is a logical program implemented on computer, to act as a knowledge expert. This means that program has an ability to reason, explain and have its knowledge base improved as more information becomes available to it. The load-forecast model can be built using the knowledge about the load forecast domain from an expert in the field. The knowledge engineer extracts this knowledge from the load domain. This knowledge is represented as facts and rules using the first predicate logic to represent the facts and IF-THEN production rules. Some of the rules do not change over time, some changes very slowly; while others change continuously and hence are to be updated from time to time [20].

3) Fuzzy Systems

Fuzzy sets are good in specialization, fuzzy sets are able to represent and manipulate electrical load pattern which possesses non-statistical uncertainty. Fuzzy sets are a generalization of conventional set theory that was introduced as a new way to represent vagueness in the data with the help of linguistic variable. It introduces vagueness (with the aim of reducing complexity) by eliminating the sharp boundary between the members of the class from nonmembers [21,22].

These approaches are based on specific problems and may represent randomness in convergence or even can diverge. The above mentioned approaches use either regression, frequency component or mean component or the peak component to predict the load. The prediction of the load depends upon both time and frequency component which varies dynamically. In this paper, an attempt is made to predict electrical load that combines the above mentioned features using generalized neurons and wavelet.

3. Elements of Wavelet Analysis

Wavelet analysis is a refinement of Fourier analysis [9–15,23–29] which has been used for prediction of time series of oil, meteorological pollution, wind speed, rainfall etc. [28,29]. In this section some important vaults relevant to our work have been described. The underlying mathematical structure for wavelet bases of a function space is a multi-scale decomposition of a signal, known as multi-resolution or multi-scale analysis. It is called the heart of wavelet analysis. Let L_2 (R) be the space of all signals with finite energy. A family \{V_j\} of subspaces of L_2(R) is called a multi resolution analysis of this space if
1) intersection of all $V_j, j = 1, 2, 3, \ldots$ be non-empty, that is $\bigcap_j V_j \neq \emptyset$

2) This family is dense in $L_2(R)$, that is, $= L_2(R)$

3) $f(x) \in V_j$ if and only if $f(2x) \in V_{j+1}$

4) $V_1 \subseteq V_2 \subseteq \ldots \subseteq V_j \subseteq V_{j+1}$

There is a function preferably with compact support of such that translates $\phi(x-k) k \in Z$. $\phi$ is called a scaling function or mother wavelet. The mother wavelet $\phi$ is obtained by building linear combinations of $\phi$. Further more $\phi$ and $\psi$ should be orthogonal, that is,

$$<\phi(-k), \psi(-l)> = 0, l, k \in Z \quad (2)$$

These two conditions given by (1) and (2) leads to conditions on coefficients $b_k$ which characterize a mother wavelet as a linear combination of the scaled and dilated father wavelets $\phi$:

$$\psi(x) = \sum_{k \in Z} b_k \phi(2x-k) \quad (3)$$

Haar, Daubechies and Coifmann are some well known wavelets. Haar wavelet (Haar mother wavelet) denoted by $\psi$ is given by

$$\psi(x) = \begin{cases} 1, & 0 \leq x \leq 1/2 \\ 1, & 1/2 < x \leq 1 \\ 0, & x < 0, x > 1 \end{cases} \quad (4)$$

Can be obtained from the father wavelet

$$\phi(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & x < 0, x > 1 \end{cases} \quad (5)$$

In this case coefficients $a_k$ in (1) are $a_0 = a_1 = 1$ and $a_k = 0$ for $k \neq 0, 1$. The Haar wavelets is defined as a linear combination of scaled father wavelets $\psi(x) = \phi(2x) - \phi(2x-1)$ which means that coefficients $b_k$ in (3) are $b_0 = 1$, $b_1 = -1$ and $b_k = 0$ otherwise, Haar wavelets can be interpreted as Daubechies wavelet of order 1 with two coefficients. In general Daubechies’ wavelets of order $N$ are not given analytically but described by $2N$ coefficients. The higher $N$, the smoother the corresponding Daubechies’ wavelets are (the smoothness is around $0.2^N$ for greater $N$). Daubechies’ wavelets are constructed in a way such that they give rise to orthogonal wavelet bases. It may be verified that orthogonality of translates of $\phi$ and $\psi$, requires that $\sum_k a_k = 2$ and $\sum_k b_k = 2$.

It is quite clear that in the higher case the scaled, translated and normalized versions of $\psi$ are denoted by

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j x - k) \quad (6)$$

With orthogonal wavelet $\psi$ the set $\{\psi_{j,k} | j, k \in Z\}$ is an orthogonal wavelet basis. A function $f$ can be represented as

$$f = \sum_{j, k \in Z} c_{j,k} \psi_{j,k} \quad (\psi_{j,k} \rightarrow <f>) \quad (7)$$

The Discrete Wavelet Transform (DWT) corresponds to the mapping $f \rightarrow c_{j,k}$. DWT provides a mechanism to represent a data or time series $f$ in terms of coefficients that are associated with particular scales [24,26,27] and therefore is regarded as a family of effective instrument for signal analysis. The decomposition of a given signal $f$ into different scales of resolution is obtained by the application of the DWT to $f$. In real application, we only use a small number of levels $j$ in our decomposition (for instance $j = 4$ corresponds to a fairly good level wavelet decomposition of $f$).

The first step of DWT corresponds to the mapping $f$ to its wavelet coefficients and from these coefficients two components are received namely a smooth version, named approximation and a second component that corresponds to the deviations or the so-called details of the signal. A decomposition of $f$ into a low frequency part $a_1$, and a high frequency part $d_1$ is represented by $f = a_1 + d_1$. The same procedure is performed on $a_1$ in order to obtain decomposition in finer scales: $a_1 = a_2 + d_2$. A recursive decomposition for the low frequency parts follows the directions that are illustrated in Figure 1.

The resulting low frequency parts $a_1, a_2, \ldots, a_n$ are approximations of $f$, and the high frequency parts $d_1, d_2, \ldots, d_n$ contain the details of $f$. This diagram illustrates a wavelet decomposition into $N$ levels and corresponds to

$$f = d_1 + d_2 + d_3 + \cdots + d_{N-1} + d_N + a_N \quad (8)$$

In practical applications, such decomposition is obtained by using a specific wavelet. Several families of wavelets have proven to be especially useful in various applications. They differ with respect to orthogonality, smoothness and other related properties such as vanishing moments or size of the support.
4. Neuro Theory of Generalized Neuron Model

The following steps are involved in the training of a summation type generalized neuron as shown in Figure 2.

4.1. Forward Calculations

Step 1: The output of the $\sum_1$ part of the summation type generalized neuron is

$$O_\Sigma = \frac{1}{1 + e^{-\lambda s_{net}}}$$  \hspace{1cm} (9)

where $s_{net} = \sum W_{ij}X_i + X_{o2}$

Step 2: The output of the $\pi$ part of the summation type generalized neuron is

$$O_{\Pi} = e^{-\lambda p_{i} p_{i_{net}}}$$  \hspace{1cm} (10)

where $p_{i_{net}} = \prod W_{ii}X_i * X_{str}$

Step 3: The output of the summation type generalized neuron can be written as

$$O_{pi} = O_{\Pi} * (1-W) + O_\Sigma * W$$  \hspace{1cm} (11)

4.2. Reverse Calculation

Step 4: After calculating the output of the summation type generalized neuron in the forward pass, as in the feed-forward neural network, it is compared with the desired output to find the error. Using back-propagation algorithm the summation type GN is trained to minimize the error. In this step, the output of the single flexible summation type generalized neuron is compared with the desired output to get error for the $i$th set of inputs:

$$\text{Error } E_i = (Y_i - O_i)$$  \hspace{1cm} (12)

Then, the sum-squared error for convergence of all the patterns is

$$E_p = 0.5 \sum E_i^2$$  \hspace{1cm} (13)

A multiplication factor of 0.5 has been taken to simplify the calculations.

Step 5: Reverse pass for modifying the connection strength.

1) Weight associated with the $\sum_1$ and $\sum_2$ part of the summation type Generalized Neuron is:

$$W(k) = W(k-1) + \Delta W$$  \hspace{1cm} (14)

Where

$$\Delta W = \eta \delta_i (O_z - O_{\Pi}) X_i + \alpha W(k-1)$$

and

$$\delta_i = \sum (Y_i - O_i)$$

2) Weights associated with the inputs of the $\sum_1$ part of the summation type Generalized Neuron are:

$$W_{ij}(k) = W_{ij}(k-1) + \Delta W_{ij}$$  \hspace{1cm} (15)

where

$$\Delta W_{ij} = \eta \delta_{ij} X_i + \alpha W_{ij}(k-1)$$

and

$$\delta_{ij} = \sum \delta_i W(1-O_z) * O_{zi}$$

3) Weights associated with the input of the $\pi$ part of the summation type generalized Neuron are:

$$W_{ii}(k) = W_{ii}(k-1) + \Delta W_{ii}$$  \hspace{1cm} (16)

where

$$\Delta W_{ii} = \eta \delta_{ii} X_i + \alpha W_{ii}(k-1)$$

and

$$\delta_{ii} = \sum \delta_i (1-W) * (2*p_{i_{net}}) * O_{\Pi}$$

$\alpha$ – Momentum factor for better convergence.

$\eta$ – Learning rate.

Range of these factors is from 0 to 1 and is determined by experience.

5. Generalized Neuron–Wavelet Approach

The Generalized Neuron–Wavelet approach has been used to predict the electrical load. In this approach, Daubechies wavelets Db8 have been applied in the decomposition for the give data pattern. There are four wavelet coefficients are used. All these wavelet coefficients are time dependent (the first three wavelet coefficients from
We observe the substantial difference of variability of the signals at different levels. The higher is the wavelet level, the lower variation of the coefficients and easier prediction of them. Our main idea is to substrate the prediction task of the original time series of high variability by the prediction of its wavelet coefficients on different levels of lower variability’s, and then using Equation (4) for final prediction of the power load at any time instant \( n \). Since most of the wavelet coefficients are of lower variability we expect the increase of the total prediction accuracy.

The wavelet tool available in Matlab is used for the process of wavelet decomposition of the time series representing average of the power load data for 120 hours. This step involves several different families of wavelets and a detailed comparison of their performance. In our case, The Daubechies wavelets of order 8 are performed.

Three level wavelet decomposition of the given time series \( X_N = f \) is performed

\[
 f = a_3 + d_1 + d_2 + d_3
\]

The smooth part of \( f \) is stored in \( a_3 \), and details on different levels are captured by \( d_1, d_2, d_3 \). Consequently a decomposition of the time series in three different scales is obtained. Figure 4 illustrates the decomposition of the original signals. The forecasting procedure methodology explained in Section 4 is used to predict the next value. The basic idea is to use the wavelet transforms and predict the data by soft computing techniques for individual coefficients of wavelet transform represented by \( a_3, d_1, d_2, d_3 \). The input to the architecture to predict the wavelet coefficients is explained in Figure 5.

The total predicted power load at an instant \( (i) \) is given by

\[
 F(i) = f_i(i) + f_2(i) + f_3(i) + f_4(i)
\]

6. Results and Discussions

The electric load data have been collected for 120 hours from Gujarat system and normalized them in the range 0–1. The Daubechies wavelet Db8 is used for decomposition and the wavelet coefficients \( d_1-d_3 \) and \( a_3 \) have been calculated. The trend of coefficients has been used for GN training and predicting the wavelet coefficients for future loads. So wavelet is used to extract the feature coefficients from data and then GN is implemented to predict the trend of the wavelet coefficient. The results of GN and actual load have been compared and shown in Figure 5.
Figures 5 and 6. The root means square error for training and testing results are .1127 and .1152 mega watts (MW) as in Table 1. When using generalized neuron and wavelet conjunction model, training each coefficient and combining to get the predicted output, a very high improvement is obtained in both training and testing results as shown in Figures 7 and 8. The root means square error for training and testing data are .0554 and .0763 respectively as in Table 1. The improvement in the results shows that accuracy of forecasting increases in the combined model and can give correct output for short term load forecasting.

Figure 6. Actual and predicted testing output using generalized neuron model (GNN).

Table 1. Comparison between GNN and GNN Wavelet technique.

<table>
<thead>
<tr>
<th>Type</th>
<th>Min. Error (Mw)</th>
<th>Max. Error (Mw)</th>
<th>RMSE (Mw)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNN (training)</td>
<td>0.0009521</td>
<td>0.3663</td>
<td>0.1127</td>
</tr>
<tr>
<td>GNNW (training)</td>
<td>0.002184</td>
<td>0.1706</td>
<td>0.0554</td>
</tr>
<tr>
<td>GNN (testing)</td>
<td>0.1306</td>
<td>0.2094</td>
<td>0.1152</td>
</tr>
<tr>
<td>GNNW (testing)</td>
<td>0.00462</td>
<td>0.1913</td>
<td>0.0673</td>
</tr>
</tbody>
</table>

Figure 7. Actual and predicted training output using generalized neuron wavelet model (GNNW).

Figure 8. Actual and predicted testing output using generalized neuron wavelet model (GNNW).

7. References


