Short-Term Electrical Load Forecasting using Constructive Feed-Forward Neural Network

Kazi Rafiqul Islam, Md. Monirul Kabir, and Kazuyuki Murase

Abstract—This paper presents a new electrical load forecasting (ELF) model based on constructive approach using feed-forward neural network (FFNN). The vital aspect of this model is to determine the FFNN architecture automatically during training in order to forecast the electrical load. Thus, the strength of standard FFNN increases in forecasting the electrical load. Furthermore, the proposed model overcomes efficiently the existing shortcomings of FFNN to predict loads of holidays and fast load changes. We call this model as constructive approach for electrical load forecasting (CAELF) as per short term basis. In order to evaluate the performance of CAELF, the daily electrical load demand data of Spain has been used. Experimental result shows that CAELF has a significant capability to forecast the electrical load compared to the other standard FFNN models.

Keywords—Electrical load forecasting, Short-term, neural network, constructive technique, partial training, and CAELF.

I. INTRODUCTION

The economy of the operation and control of power systems is sensitive to system demand; large savings can be obtained by increasing the accuracy of demand forecast. The effect of a large forecast error is reflected in terms of over conservative or over risky operation. It implies that, over estimation leads to the startup of too many units or excessive energy purchase, thereby supplying an unnecessary level of reserve. On the other hand, under estimation persuades insufficient preparation of spinning reserve and causes the system to operate at a risk region to the disturbance. Thus, improvement in load forecasting accuracy leads to the cost savings and increases in the system security [1].

A number of approaches exist in the literature (e.g., [1]-[11]), where they try to solve the short term electrical load forecasting (STELF) problem using neural networks (NNs). It has been confirmed that, the usage of NN in STELF always outperforms any human-based computational analysis in terms of accuracy, easy maintenance for users. Because, NN has a good capability for mapping between input and output although load (i.e., output) is being increased day by day [12]. Feed forward NN (FFNN) has been used in [7]-[11] to solve the ELF problem for different regions with a reasonable computational cost. It is noted that, FFNNs are much suitable for mapping static relationships between inputs and outputs and ultimately providing good results in ELF. However, FFNNs need large historical data and have a limited capability to predict loads of holidays and fast load changes [13]. To overcome the shortcomings of FFNN, a number of efforts have been done in [2], [4]-[6] recently, among which echo state NN, radial basis function NN, recurrent NN, and nonlinear autoregressive NN are used, respectively. It is noted here that, the performances of aforementioned NN models are satisfactory in predicting the electrical load comparing to the FFNN, but computationally expensive. Thereby, huge requirements are necessary for the hardware setups as well as experts are needed for maintainances.

This paper describes a new single-stage online ELF approach using FFNN, called constructive ELF approach (CAELF). This approach differs from previous works in a way that, CAELF determines the appropriate NN architecture in advance before the ELF starts using constructive NN training. In contrast to the previous approaches (e.g., [7]-[11]), they generally use a fixed NN architecture with randomly selecting the hidden neuron in the hidden layer during training before the ELF starts. It well known that, the random selection of hidden neurons affects the generalization performance of NNs. The reason is that, the performance of any NN is greatly dependent on its architecture [14][15]. Thus determining hidden neurons’ number automatically provide a novel approach in building learning models using NNs for ELF.

The remainder of this paper is organized as follows. Section II describes about the feed-forward neural network, whereas, a detailed description about CAELF has been presented in Section III. Section IV discusses the results of our experimental study. Finally, Section V concludes the paper with a brief summary and a few remarks.

II. NEURAL NETWORK

Artificial Neural Network is massively parallel interconnected networks of simple adaptive elements and their hierarchical organizations [16]. These networks are intended to interact with the objects of the real world in the same way as biological nervous systems do. They constitute an alternative knowledge representation paradigm for artificial intelligence. Particularly, NNs are made up of processing units, called neurons. This neuron performs a weighted summation over the outputs of the neurons that are connected to its inputs. Then,
III. THE PROPOSED FORECASTING MODEL (CAELF)

CAELF uses a training approach in association with incremental training to find a minimum number of hidden neurons for NN models. Hidden neurons (HNs) are added simultaneously one by one in constructive fashion during the training process of a NN. If the addition of HN does not improve the NN’s accuracy, it is then removed.

The major steps of CAELF are summarized in Fig. 1, which are explained further as follows.

Step 1) At first, choose a feed-forward NN with minimal size. Precisely, size of the input layer and output layer are decided by the total number of input variables and the output load of the given ELF data samples, respectively, whereas, size of the hidden layer is initialized using one hidden neuron.

Step 2) Start the partial training of NN on the training data sample up to \( \tau \) epoch using the back-propagation (BP) algorithm [17]. The number of training epochs, \( \tau \), is specified by the user. Partial training, which was first used in conjunction with an evolutionary algorithm [18], means that the NN is trained for a fixed number of epochs regardless whether it has converged or not.

Step 3) Check the termination criterion of NN training. If it is satisfied, the current NN architecture is the outcome of CAELF for a given data samples. Otherwise, follow the next step. In this work, calculate average training error [19], \( E_a \), on the validation samples. In other word, the average training error is considered here as mean squared error (MSE). Thus, the error, \( E_a \), is calculated as,

\[
E_a = \frac{1}{2p\tau} \sum_{p=1}^{P} \sum_{C=1}^{C} (t_c(p) - o_c(p))^2
\]  

(1)

Where, \( t_c(p) \) and \( o_c(p) \), respectively, are the actual and predicted responses of the \( c \)-th output neuron for the validation pattern \( P \). The symbols \( P \) and \( C \) represent the total number of validation patterns and of output neurons, respectively.

Step 4) Check the performance criterion of the network training. If the criterion is satisfied then the network is assigned to be trained further and go to Step 2. Otherwise, follow the next step.

Step 5) Add a hidden neuron to the network and go to Step 2 for following the partial training again.

Step 6) NN is then tested with the unseen testing pattern. Finally, get the electrical load forecasting from the current NN.

CAELF uses only one cost function that is the training error on validation data samples. CAELF finally tries to design a better load forecaster using NN. Details about some basic steps of CAELF are further given in the following sections.

A. Performance criterion of NN training:

If the average training error on validation samples reduces by a predefined amount \( \varepsilon \), after the training epoch \( \tau \), it is assumed that the training process is progressing well, thus further training is necessary and go to the Step 2. The reduction of training error can be described as,

\[
E_a(t-\tau) - E_a(t) > \varepsilon, \quad \tau = \tau, 2\tau, 3\tau, \ldots
\]

(2)

Where, \( \tau \) and \( t \) are positive integer number specified by the user.

B. Termination criterion of NN training:

Since CAELF adds hidden neurons one by one during the training process of a NN, the training error would reduce as the training process progresses. However, the objective of CAELF is to improve generalization ability of the NN. This means the training error may not be a right choice to be used for terminating the training process of the NN. Generally, a separate data samples, called the validation samples, is widely used for termination. It is assumed that the validation error gives an unbiased estimate because the validation data are not used for modifying the weights of the NN.

In order to achieve good generalization ability, CAELF uses average training error on validation samples in its termination criterion. It measures validation error after every \( \tau \) epochs of training, called strips. It terminates training when the average training error increases by a predefined amount \( \lambda \) for \( T \) successive times, which are measured at the end of each of \( T \)
successive strips [20]. Since the average training error on validation samples increases not only once but $T$ successive times, it can be assumed that such increases indicate the beginning of the final over fitting not just the intermittent. The termination criterion can be expressed as,

$$E_a(t + i) - E_a(t) > \lambda, \quad i = 1, 2, 3, \ldots, T$$

(3)

Where $\lambda$ and $T$ are the positive integer number specified by the user. Our model, CAELF tests the termination criterion after every $\tau$ epochs of training and stops the training when the condition described by the Eq. (3) is satisfied. In this work, the value of $T$ is chosen as 3.

C. Hidden neuron addition:

CAELF adds a hidden neuron to the existing network architecture according to the equation (4). The reason is that, the existing network architecture is not capable to acquire the all information of the data samples; thereby increasing the size of the network is necessary. Then, train the modified architecture for a certain number of $\tau$ epochs.

$$E_a(t + \tau) - E_a(t) \leq \varepsilon, \quad t = \tau, 2\tau, 3\tau, \ldots$$

(4)

Where $\varepsilon$ is the predefined amount specified by user.

IV. EXPERIMENTAL STUDIES

In this section, the performance of CAELF for predicting the electrical load at near future was presented using the daily load data sample. The data used in this study is the daily electrical demand in megawatts/hour in Spain [6], [21]. The CAELF’s performance was evaluated in terms of predicted error. Precisely, predicted error refers to the error of existing NN on testing data samples. For more clarification about the performance evaluation of CAELF, this section is organized by the following subsections.

A. Description of Data

The data used in the experimental analysis of this paper is the daily electrical demand in megawatts/hour in Spain from January 1, 1993 to June 30, 1998 for a total of 2007 days. To clarify about the data, Table I shows a partial sample of the data sheet. Using this data, the architecture of a feed-forward NN is also shown in Fig. 2.

<table>
<thead>
<tr>
<th>OBS</th>
<th>Log(Load)</th>
<th>HDD</th>
<th>CDD</th>
<th>Tue</th>
<th>Wed</th>
<th>--</th>
<th>Sun</th>
<th>Holiday</th>
<th>Easter</th>
<th>Feb</th>
<th>Mar</th>
<th>--</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/93</td>
<td>12.64</td>
<td>11.9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>31/12/97</td>
<td>12.93</td>
<td>7.47</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>1</td>
</tr>
</tbody>
</table>

B. Experimental Setup

The data used for training of the NN model in CAELF was from January 1, 1993 to December 31, 1997 for a total 1826 days, whereas the NN was validated during training using the data from July 1, 1998 to December 31, 1998 in total 184 samples. Precisely, these samples were called “in-sample” data as they used in NN model for training. On the other hand, the data sample period of 120 days from January 1, 1999 to April 30, 1999, that were used to test the forecasting performance by comparing model output (i.e., predicted load) with the actual load. These data samples are called as “out-of-sample” as they were not used during the training of NN.

In all experiments, one bias unit with a fixed input +1 was connected to the hidden layer and output layer. The learning rate and momentum term for training of NN were chosen as 0.05–0.1 and 0.4–0.7, respectively. The initial connection weights for an NN were randomly chosen in the range between -1.0 and 1.0. A sigmoid function was used as an activation function.

Fig. 2 Model of FFNN for forecasting the electrical load. Here, HDD and CDD refer to the exogenous variables of degree days that are calculated as heating degree days and cooling degree days, in that order. On the other hand, $W_d$ and $M_t$ are the dummy variables that represent all the weeks and monthly seasonalities, respectively. For more information about these input variables can be found in [21].
C. Experimental Results

The performance of CAELF in terms of forecasting the out-of-samples was measured by making a comparison between actual values and model outputs during the same period. Furthermore, we measured the mean absolute percentage error (MAPE) for the best relative accuracy measure among the various forecasting accuracy criteria [22]. However, MAPE was calculated here as,

\[
MAPE = \frac{1}{N} \sum_{n=1}^{N} \left| \frac{P_n - \bar{P}_n}{P_n} \right| \times 100
\]

Where, the \( P_n \) and \( \bar{P}_n \) represent the actual and predicted electrical load, respectively and \( N \) is the total number of samples available.

In this context, Fig. 3, Fig.5 and Fig. 6 represent the forecasting analysis for the period of 120 days (including errors in percentage), 30 days, and 7 days, respectively. Particularly, a comparison between actual load and predicted load (i.e., forecasting load) was made in Fig. 3 (a) using CAELF. We calculated MAPE between these two loads and found that it was 0.2132. In addition, more analytical forecasting results of CAELF can be found in Figs. 5 and 6, where the forecasting of electrical load was done in between 30 days and 7 days, respectively. In a closed observation among these figures, it has been found that, forecasting of electrical load using CAELF is satisfactory as the predicted load curve is very much closed to overlap the actual load curve.

D. Comparison with other works

The obtained forecasting result of CAELF on Spanish daily electrical load data has been compared with the results of three electrical load forecasting models, such as, (i) standard electrical load forecasting (SELF), (ii) NNELF-1[6], and (iii) NARx-2[6]. The first two models used standard feed-forward NN for electrical load forecasting, where a fixed number of hidden neuron in the hidden layer of NN and a fixed number of iteration for the NN training have been considered. The third one is the nonlinear autoregressive model that composed by two parts: first, the true available output is fed as an input to train the NN; second, the resulting network has a purely feed forward architecture and BP algorithm is used for training. We used one parameter for comparisons here, that is to say, MAPE.

In SELF, the whole setup of CAELF was used except the constructive approach and partial training. In this case, 5 hidden neurons were considered in the hidden layer of NN and 200 iterations for the NN training. For comparisons, we run SELF from 10 times and averaged the forecasting results. On the other hand, the model NNELF-1[6] and NARx-2[6] used 10 hidden neurons in the hidden layer and the forecasting results were averaged by 20 individual runs. Table II shows the comparison result among these four models including CAELF. It has been found that, the value of MAPE is the most reduced one for CAELF among the other models. On the other hand, the minimum value of SD signifies the robustness of CAELF.

![Fig.3](image3.png)  (a) Comparison between the actual load and the predicted load obtained from CAELF used by constructive feed forward neural network and (b) corresponding their errors in percentage.

![Fig.4](image4.png)  (a) Comparison between the actual load and the predicted load for 120 days obtained from standard model (SELF) used by feed forward neural network and (b) corresponding their errors in percentage.

![Fig.5](image5.png)  Comparison between the actual load and the predicted load for 30 days obtained from CAELF used by constructive feed forward neural network.

![Fig.6](image6.png)  Comparison between the actual load and the predicted load for 7 days obtained from CAELF used by constructive feed forward neural network.
V. CONCLUSION

In this paper, a new short-term electrical load forecasting model has been proposed, called as CAELF that utilizes the constructive approach in feed-forward NN training. Thus, the FFNN automatically determines the size of the hidden layer during training on basis of the ELF problem domain.

Experimental results presented in Fig. 3, Fig.5 and Fig. 6 show that, CAELF performs well in ELF problem in respect to the short-term basis (e.g., 120 days, 30 days, and 7 days). The reason is that, the forecasting curve between predicted load and actual load (Fig.3 (a)) is closed to similar. Furthermore, most of the points of errors in this error curve showed in Fig. 4 (b) are closed to zero.

In comparison to the other models, Fig. 3 and Fig. 4 show that, the result of our model is much better than that of SELF. In addition to the concept of MAPE, Table II confirms that, our model’s performance is better than other models, such as, SELF, NNELF-1 [6], and NARx-2 [6].

Furthermore, in CAELF, the forecasting results are not so satisfactory in the last portion of out-of-sample showed in Fig. 3 (a). The possible reason might be the nonlinearity of the load variations in the data samples. In order to reduce such limitations, incorporating more heuristic techniques in CAELF is left for further works.

REFERENCES