Short-Term Daily Load Forecasting in an Intelligent Home with GA-Based Neural Network

S. H. Ling, F. H. F. Leung, H. K. Lam and P. K. S. Tam
Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract - Daily load forecasting is essential to improve the reliability of the AC power line data network and provide optimal load scheduling in an intelligent home system. In this paper, a short-term daily load forecasting realized by a GA-based neural network is proposed. A neural network with a switch introduced to each link is employed to minimize forecasting errors and forecast the daily load with respect to different day types and weather information. Genetic algorithm (GA) with arithmetic crossover and non-uniform mutation is used to learn the input-output relationships of an application and the optimal network structure. Simulation results on a short-term daily load forecasting in an intelligent home will be given.

I. INTRODUCTION

Nowadays, homes can have smart features to ensure a high degree of security and comfort. Reliable channels for the communications among electrical appliances and users can be realized. Moreover, with a home network, electrical appliances can be used in an efficient way such that the wastage of energy can be reduced. This paper is based on an intelligent home system [15]. In this system, the AC power line network is used not only for supplying electric power, but also serving as the data communication channel among electrical appliances. With this AC power line data network, a short-term load forecasting can be realized. An accurate load forecasting can bring the following benefits to the intelligent home.

1) Increasing the reliability of AC power line data network. One problem faced by the power line data network is the possible low impedance of the power line in the operating bandwidth [16-17] for data transmission. When the line impedance is too low, the maximum transmission rate will be affected, and the reliability and throughput of the AC power line data network will decrease. The attenuation of the data signal in an AC power line is proportional to the loads connected to it. The reliability of the power line data network can be enhanced if the load is kept at an optimal level through forecasting and balancing. We can also adaptively set a suitable data transmission rate based on the forecasted load condition in order to reduce the overhead owing to retransmissions of data.

2) Optimal load scheduling - At present, the peak demands of electricity are met by operating costly auxiliary generators, or by purchasing power from other utility companies. The cost for supplying peak power is therefore much higher than that for supplying the average power. A reduction in the peak value of electricity demand can be achieved if we can realize load forecasting, and schedule the demands on the utility company accordingly. This has to be supported by batteries installed in the intelligent home that are responsible for sharing the load demand.

Computational intelligence techniques have been applied in daily load forecasting. Artificial neural networks have been considered as very promising tools to short-term load forecasting [18-25]. However, the gradient-descent (GD) algorithm for parameter training of the neural networks suffers from the common problems of convergence to local minima and sensitivity to initial values of the parameters. Global search techniques such as Genetic Algorithm (GA) may solve these problems. Genetic algorithm (GA) is a powerful searching algorithm to handle optimization problems [1-3]. It is particularly useful for complex optimization problems with a large number of tuned parameters. Applications of GA can be found in fuzzy control [6-10], path planning [9], greenhouse climate control [10], modeling and classification [11], etc.

Neural network with fixed structure trained by GA for short-term load forecasting was reported in [26]. Other application areas of neural networks include prediction [4-5], system modelling and control [13]. Thanks to its particular structure, neural networks are very good in learning [2] using some learning algorithms such as GA [1] and back propagation [2]. Usually, the structure of a neural network is fixed for a learning process. However, a fixed structure may not provide the best performance within a given training period. If the neural network structure is too complicated, the training period will be long and the implementation cost will be high.

In this paper, a three-layer neural network with a switch introduced in each link is proposed to facilitate the tuning of the optimal network structure. GA with arithmetic crossover and non-uniform mutation [5] is used to help tuning the structure as well as the parameters of the proposed neural network. The proposed neural network is then used to forecast the daily load in an intelligent home. Simulation results will be given to illustrate the performance of the proposed neural network.

This paper is organized as follows. The GA-based neural network with link switches is introduced in section II. A short-term daily load forecasting by using the proposed GA-based neural network is presented in section III. A conclusion will be drawn in section IV.

II. GA-BASED NEURAL NETWORK WITH LINK SWITCHES

In this section, a neural network with link switches is presented. The tuning of the network parameters and structure using GA will also be formulated in this section.
A. Neural Network with Link Switches

The proposed three-layer network is shown in Fig. 1. Specifically, a unit step function is introduced to each link. This unit step function is defined as,

\[
\delta(\alpha) = \begin{cases} 
0 & \text{if } \alpha < 0 \\
1 & \text{if } \alpha \geq 0 
\end{cases}, \quad \alpha \in \mathbb{R} \tag{1}
\]

The introduction of the step function is equivalent to adding a switch to each link of the neural network. Referring to Fig. 1, the input-output relationship of the proposed multiple-input-multiple-output three-layer neural network is given by,

\[
y_k(t) = \sum_{i=1}^{n_i} \delta(s^i_k)w_{ij} \logsig \left[ \sum_{j=1}^{n_i} \left( \delta(s^i_j) + \delta(s^i_j) - \delta(s^i_j) \right) \right] - \delta(s^i_k) b_k^i \\
\text{, } k = 1, 2, \ldots, n_{out} \tag{2}
\]

\[\text{where } y^d(t) = \begin{bmatrix} y^d_1(t) & y^d_2(t) & \cdots & y^d_{n_d}(t) \end{bmatrix}\text{ is the desired output corresponding to the input }
\begin{bmatrix} z^d_1(t) & z^d_2(t) & \cdots & z^d_{n_d}(t) \end{bmatrix}\text{ of an unknown nonlinear function } g( ) \text{ respectively. } n_d \text{ denotes the number of input-output data pairs. The fitness function is defined as,}
\]

\[
\text{fitness} = \frac{1}{1 + err} \tag{5}
\]

\[
\text{err} = \frac{1}{n_d} \sum_{i=1}^{n_d} \left| y^d_i(t) - y^a_i(t) \right| \tag{6}
\]

The objective is to minimize the mean absolute percentage error (MAPE) using GA by setting the chromosome to be \[\begin{bmatrix} \varphi & w_k & s^i_k & b_k^i \end{bmatrix}\] for all \(i, j, k\).

The range of (5) is from 0 to 1. A larger value of the fitness function indicates a smaller value of (6).

III. SHORT-TERM DAILY LOAD FORECASTING SYSTEM

The proposed short-term daily load forecasting system in an intelligent home will be discussed in this section. The idea of daily load forecasting is to construct seven multi-input multi-output neural networks, one for each day. Each neural network has 24 outputs representing the expected hourly load for a day. The most important work in the short-term daily load forecasting in an intelligent home is the selection of the input variables. In this forecasting system, there are three main kinds of input variables:

1. Historical load data: hourly loads were collected and used as historical load inputs. The historical load data reflect the habit of the family on power consumption.

2. Temperature inputs: the average temperature of the previous day and the present day are used as two inputs in this forecasting system. The value of the average temperature of the present day is not from the temperature forecasting of the weather observatory, assuming that their forecasting accuracy is good.

3. Rainfall index inputs: the average rainfall index of the previous day and the present day (again predicted by the weather observatory) are used as two inputs in this forecasting system. The range of the rainfall index is from 0 to 1. 0 represents no rain and 1 represents heavy rain.

A diagram of the daily load forecasting system is shown in Fig. 2. Each of the seven neural networks (for Monday to Sunday) has 28 inputs, 24 outputs with link switches. Among the 28 inputs nodes, the first 24 input nodes (\(z_1, \ldots, z_{24}\)) represent the previous 24 hourly loads [21] and is

\[0-7803-7278-6/02/$10.00 \odot 2002~IEEE\]

998

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on July 7, 2009 at 03:01 from IEEE Xplore. Restrictions apply.
denoted by $z_i = L_i^j (t-1)$, where $i = 1, 2, \ldots, 24$. Nodes 25
($z_{25}$) and nodes 26 ($z_{26}$) represent the average temperature of
the previous day and present day respectively. Nodes 27
($z_{27}$) and Nodes 28 ($z_{28}$) represent the average rainfall index
at previous day and present day respectively. The output
layer consists of 24 output nodes that represent the forecasted
24 hourly loads of a day and is denoted by $y_i(t) =
L_i(t)$, $j = 1, 2, \ldots, 24$. There are two methods of realizing
the daily load forecasting system. One is by training the
forecasting neural network off-line and the other is by
training on-line. The offline training is a time consuming
processing. However, once trained, the system can make the
forecast quickly (as a lower number of iterations is needed).
In this example, we use 12 sets historical data for off-line
training. Once trained off-line, the forecasting system
operates in an on-line mode, and the weights of neural
network will be updated day by day with 500 iterations.
From (2), the proposed neural network used for the daily
load forecasting is governed by,

$$y_i(t) = \sum_{j=1}^{24} \delta (s_{ij}) w_{ij} \log \left[ \sum_{k=1}^{24} \delta (s_{kj}) v_k z_k(t) - \delta (s_{ij}) b_i^j \right] - \delta (s_{ij}) b_i^j$$

where $k=1, 2, \ldots, 24$. (7)

The number of hidden nodes ($n_h$) is changed from 14 to 19
in order to compare the learning performance. GA is
employed to tune the parameters and structure of the neural
network of (7). The fitness function is defined as follows,

$$fitness = \frac{1}{1 + err}$$

(8)

$$err = \frac{1}{12} \sum_{t=1}^{24} \frac{|y_i^o(t) - y_i(t)|}{y_i^o(t)}$$

(9)

The objective is to maximize the fitness function of (8). A
larger value of the fitness function indicates a smaller value of
err of (9). The best fitness value is 1 and the worst one is
0. The population size used for the GA is 10. The lower and
the upper bounds of the link weights are defined as

$$-3 \leq w_{ij}, v_j, s_{ij}^j, b_i^j \leq 3$$

and

$$-1 \leq s_{ij}^j, s^j, s_{ij}^k, b_i^k, b_i^j, b_i^k \leq 1$$

where $i = 1, 2, \ldots, 3; j = 1, 2, \ldots, n_h, k = 1, 2, \ldots, 24$ [14].

The chromosomes used for the GA are $[s_{ij}^j, w_{ij}, v_j, s_{ij}^j, b_i^j, s_{ij}^k, b_i^k, b_i^j, b_i^k]$ for all $i$, $j$, $k$.

The initial values of the link weights are set at 0.03. The
probability of crossover and mutation are set at 0.8 and 0.03
respectively and the shape parameter of the non-uniform
mutation operation is set at 5. The number of the iterations
to train the neural network is 2000. The simulation results are
tabulated in Table I and Table II. Table I shows the
simulation results of daily load forecasting on Wednesday
and Table II shows the daily load forecasting on Sunday.
These tables show the fitness value, the average training
error, the average forecasting error (in term of MAPE) and
the percentage of reduction of the number of links of the
neural network. Referring to these two tables, the best result
is obtained when the number of hidden node is equal to 16.
From Table I, $n_h = 16$, the number of link reduced is 111
after learning (the number of links of a fully connected
network is 872). It is about 12.73% reduction of the links.
The average training error and forecasting error in term of
MAPE from Monday to Sunday are shown in Table III.
The average errors of training and forecasting are 2.2403 and
2.8677 respectively. Fig.3 and Table IV show the simulation
results of the daily load forecasting on Sunday using the
proposed network. In Fig.3, the solid line represents
the forecasted result and the dashed line is the actual load.
Referring to Table IV, we can observe that the percentage
error for each hour is within 0 to 3.8% (the average error is
2.3485%).

IV CONCLUSION

By introducing a switch to each link, a neural
network that facilitates the tuning of its structure has been
proposed. By employing GA with arithmetic crossover and
non-uniform mutation, a proposed GA-based neural network
is able to learn both the input-output relationship of an
application and the optimal network structure. As a result,
the fully connected neural networks will become a partly
connected neural network after learning. This implies a
lower cost of implementation. A short-term daily load
forecasting in an intelligent home using the proposed GA-
Based neural network has been presented. The performance
of the proposed network is satisfactory.

ACKNOWLEDGEMENT

The work described in this paper was substantially supported
by a Research Grant of The Hong Kong Polytechnic
University (project numbers G-V954 and A420).

REFERENCES


genetic algorithms, tabu search, simulated annealing and neural


Forecasting Using Neural Network,* 5th IEEE International Symposium

time series prediction,* International Conference on Neural

logic controller based on linguistic-hedge concepts and genetic

genetic algorithms,* IEEE Trans., Industry Applications, vol. 36,

learning through symbiotic evolution for fuzzy controller design,*
April, 2000, pp. 290-302.

genetic algorithms,* Electronics Letters, vol. 36, no. 4, Feb. 2000,
pp. 374-376
Fig. 1. The proposed 3-layer neural network.

Fig. 2. Neural network for daily load forecasting.

Fig. 3. Load forecasting results for Sunday (Week 13) given by the proposed network (solid line) and the actual load (dashed line).
<table>
<thead>
<tr>
<th>( n_d )</th>
<th>Fitness Values</th>
<th>Ave. training error MAPE(%)</th>
<th>Ave. forecasting error MAPE(%)</th>
<th>Ratio of the no. of links reduced to the total number of links (Percentage of reduction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0.977000</td>
<td>2.3541</td>
<td>2.8217</td>
<td>91/766 (11.88%)</td>
</tr>
<tr>
<td>15</td>
<td>0.976553</td>
<td>2.4072</td>
<td>3.1310</td>
<td>121/819 (14.77%)</td>
</tr>
<tr>
<td>16</td>
<td>0.981352</td>
<td>1.9062</td>
<td>2.5977</td>
<td>111/872 (12.73%)</td>
</tr>
<tr>
<td>17</td>
<td>0.977927</td>
<td>2.3493</td>
<td>2.9259</td>
<td>142/925 (15.35%)</td>
</tr>
<tr>
<td>18</td>
<td>0.978904</td>
<td>2.1551</td>
<td>2.8034</td>
<td>139/978 (14.21%)</td>
</tr>
<tr>
<td>19</td>
<td>0.969785</td>
<td>3.1156</td>
<td>3.7413</td>
<td>163/1031 (15.81%)</td>
</tr>
</tbody>
</table>

Table I. Learning results of the daily load forecaster for Wednesday.

<table>
<thead>
<tr>
<th>( n_d )</th>
<th>Fitness Values</th>
<th>Ave. training error MAPE(%)</th>
<th>Ave. forecasting error MAPE(%)</th>
<th>Ratio of the no. of links reduced to the total number of links (Percentage of reduction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0.989889</td>
<td>3.1035</td>
<td>3.0288</td>
<td>116/766 (15.14%)</td>
</tr>
<tr>
<td>15</td>
<td>0.973920</td>
<td>2.6778</td>
<td>2.6630</td>
<td>127/819 (15.51%)</td>
</tr>
<tr>
<td>16</td>
<td>0.977285</td>
<td>2.3243</td>
<td>2.3485</td>
<td>165/852 (18.92%)</td>
</tr>
<tr>
<td>17</td>
<td>0.977035</td>
<td>2.3405</td>
<td>2.3621</td>
<td>138/925 (14.92%)</td>
</tr>
<tr>
<td>18</td>
<td>0.972765</td>
<td>2.7997</td>
<td>2.8900</td>
<td>186/978 (19.02%)</td>
</tr>
<tr>
<td>19</td>
<td>0.974062</td>
<td>2.6629</td>
<td>2.4359</td>
<td>152/1031 (14.74%)</td>
</tr>
</tbody>
</table>

Table II. Learning results of the daily load forecaster for Sunday.

<table>
<thead>
<tr>
<th>Day type</th>
<th>Ave. training error MAPE(%)</th>
<th>Ave. forecasting error MAPE(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>2.1769</td>
<td>2.6963</td>
</tr>
<tr>
<td>Tuesday</td>
<td>2.3396</td>
<td>2.9289</td>
</tr>
<tr>
<td>Wednesday</td>
<td>1.9002</td>
<td>2.5977</td>
</tr>
<tr>
<td>Thursday</td>
<td>2.2937</td>
<td>3.5016</td>
</tr>
<tr>
<td>Friday</td>
<td>2.4001</td>
<td>3.2759</td>
</tr>
<tr>
<td>Saturday</td>
<td>2.2472</td>
<td>2.7349</td>
</tr>
<tr>
<td>Sunday</td>
<td>2.3243</td>
<td>2.3485</td>
</tr>
<tr>
<td>Average error:</td>
<td>2.2403</td>
<td>2.8677</td>
</tr>
</tbody>
</table>

Table III. Training error and forecasting error represented in MAPE at different day type (Mon-Sun) with \( n_d \) = 16.

<table>
<thead>
<tr>
<th>Sunday Hours</th>
<th>Normalized Loads</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of the day</td>
<td>Predicted</td>
<td>Desired</td>
</tr>
<tr>
<td>01:00</td>
<td>0.4509</td>
<td>0.4483</td>
</tr>
<tr>
<td>02:00</td>
<td>0.4513</td>
<td>0.4526</td>
</tr>
<tr>
<td>03:00</td>
<td>0.4595</td>
<td>0.4451</td>
</tr>
<tr>
<td>04:00</td>
<td>0.4582</td>
<td>0.4436</td>
</tr>
<tr>
<td>05:00</td>
<td>0.4513</td>
<td>0.4467</td>
</tr>
<tr>
<td>06:00</td>
<td>0.4517</td>
<td>0.4488</td>
</tr>
<tr>
<td>07:00</td>
<td>0.4504</td>
<td>0.4414</td>
</tr>
<tr>
<td>08:00</td>
<td>0.4595</td>
<td>0.4490</td>
</tr>
<tr>
<td>09:00</td>
<td>0.4771</td>
<td>0.4766</td>
</tr>
<tr>
<td>10:00</td>
<td>0.4595</td>
<td>0.4595</td>
</tr>
<tr>
<td>11:00</td>
<td>0.4595</td>
<td>0.4731</td>
</tr>
<tr>
<td>12:00</td>
<td>0.4769</td>
<td>0.4753</td>
</tr>
<tr>
<td>13:00</td>
<td>0.4822</td>
<td>0.4646</td>
</tr>
<tr>
<td>14:00</td>
<td>0.4783</td>
<td>0.4726</td>
</tr>
<tr>
<td>15:00</td>
<td>0.4773</td>
<td>0.4836</td>
</tr>
<tr>
<td>16:00</td>
<td>0.4772</td>
<td>0.4691</td>
</tr>
<tr>
<td>17:00</td>
<td>0.5637</td>
<td>0.5517</td>
</tr>
<tr>
<td>18:00</td>
<td>0.5714</td>
<td>0.5861</td>
</tr>
<tr>
<td>19:00</td>
<td>0.6067</td>
<td>0.6055</td>
</tr>
<tr>
<td>20:00</td>
<td>0.7218</td>
<td>0.7218</td>
</tr>
<tr>
<td>21:00</td>
<td>0.7536</td>
<td>0.7451</td>
</tr>
<tr>
<td>22:00</td>
<td>0.7762</td>
<td>0.7043</td>
</tr>
<tr>
<td>23:00</td>
<td>0.7143</td>
<td>0.7395</td>
</tr>
<tr>
<td>24:00</td>
<td>0.4595</td>
<td>0.4463</td>
</tr>
</tbody>
</table>

Table IV. Sunday hourly loads: predicted and actual figures.