PAPER

Grey Neural Network and Its Application to Short Term Load Forecasting Problem

Cheng-Hsiung HSIEH[†], *Regular Member*

SUMMARY In this paper, a novel type of neural networks called grey neural network (GNN) is proposed and applied to improve short term load forecasting (STLF) performance. This work is motivated by the following observations: First, the forecasting performance of neural network is affected by the randomness in STLF data. That is, poor performance results from large randomness and vice versa. Second, the grey first-order accumulated generating operation (1-AGO) is reported having randomness reduction property. By the observations, the GNN is proposed and expected to have better STLF performance. The GNN consists of grey 1-AGO, the piecewise linear neural network (PLNN), and grey first-order inverse accumulated generating operation (1-IAGO). Given a set of STLF data, the data is first converted by grey 1-AGO and then is put into the PLNN to perform forecasting. Finally, the predicted load of GNN is obtained through grey 1-IAGO. For comparison, the original STLF data is also put into the PLNN itself. With identical training conditions, the simulation results indicate that with various network structures the GNN, as expected, outperforms the PLNN itself in terms of mean squared error.

key words: *grey 1-AGO, piecewise linear neural network (PLNN), short term load forecasting (STLF), grey neural network*

1. Introduction

The short term load forecasting (STLF) problem has been widely studied in the field of electrical power and energy systems. The reason is that accurate forecasting helps in the real-time power generation, efficient energy management, and economic cost saving. Up to present, approaches proposed for STLF problem can be roughly divided into four types: time series approach, regression approach, expert-based approach, and neural network based approach. Since 1990, the neural network based approach is getting more and more attention for its promising results in STLF. In the beginning, researchers are trying to demonstrate the feasibility of applying neural networks to STLF problem in power engineering. Recently, efforts are put to improve the forecasting performance of neural networks. A wide variety of methods to improve STLF performance have been reported as in [1]–[19] which include network structure modification, learning algorithm innovation, hybrid system, and data preprocessing. Here we concentrate ourselves on the data preprocessing approaches. In [20], it is noticed that the randomness in the STLF training data affects the forecasting performance of neural networks, i.e., increasing the randomness degrades the performance and vice versa. Thus, it is expected that better forecasting performance can be achieved if the randomness in the training data can be reduced. An approach to reduce the randomness inherent in the training data is the preprocessing technique, clustering. Given training data set $\Gamma^{(0)}$, the objective of clustering is to collect homogeneous data together and to divide $\Gamma^{(0)}$ into subsets, based on a similarity measure. Then each data subset is used to train a corresponding neural network. Note that in general the randomness in each clustered subset of $\Gamma^{(0)}$ is less than that in $\Gamma^{(0)}$. Consequently, better forecasting performance of neural networks is expected. Up to present, several clustering approaches have been proposed. In [12], a heuristic clustering approach to partition the training data by the days of the week is proposed while the training data is hourly partitioned in [11]. In [10], a 'follow the leader' approach is used to cluster the training data. In [9], a fuzzy approach partitions the training data based on fuzzy curve. In [2], a neural approach based on modified Kohonen clustering skill is employed to partition load data while neural gas is used in [15].

This paper introduces the grey preprocessing approach, the first order accumulated generating operation (1-AGO) [21], to reduce the randomness in STLF data. Then a grey neural network (GNN) which is based on the piecewise linear neural network (PLNN) [22] is applied to perform load forecasting. There are two main reasons to explain why PLNN is applied in this paper. First, PLNN is a type of modular neural networks. Many researchers have indicated that modular neural networks have better load forecasting performance than global type neural networks such as multilayer perceptron (MLP) neural networks [23], since the training data set is partitioned. Moreover, it is demonstrated in [22] that PLNN usually performs as well as an MLP with equivalent required multiplies in many data sets. Second, as described later in Sect. 4 an on-line training is demanded in the given STLF example and PLNN meets the requirement because it is of fast convergence and training efficiency properties [22]. Since training an MLP is a time-consuming task, thus MLP is not appropriate because an on-line training is a key

Manuscript received May 18, 2001.

Manuscript revised September 21, 2001.

[†]The author is with the Department of Electronic Engineering of Chien Kuo Institute of Technology, Changhua, Taiwan, R.O.C.

issue in the given STLF problem.

This paper is organized as follows: A brief review of grey 1-AGO and first order inverse AGO (1-IAGO) [21] is given in Sect. 2. Next, the structure of GNN is introduced and its cost function is discussed as well. In Sect. 4, the proposed GNN is applied to an STLF example where the forecasting results of PLNN and GNN are compared. Finally, conclusions and further researches are described in Sect. 5.

2. Grey 1-AGO and 1-IAGO

Let $\Gamma^{(0)} = \{(\mathbf{x}_p^{(0)}, y_p^{(0)}),$ for $1 \le p \le N_p\}$ be a power load forecasting data set where M-dimensional $\mathbf{x}_p^{(0)}$ is the pth input vector of historical power load, $y_p^{(0)}$ is the pth desired output or k-step ahead power load to be predicted, and N_p is the total number of patterns. To effectively reduce the randomness in data, the grey 1-AGO needs satisfy the following two conditions: (i) data is of same sign, and (ii) the ratio between adjacent data in $\Gamma^{(0)}$ should be within 1 order in magnitude. Assume that Conditions (i) and (ii) are satisfied. From $\Gamma^{(0)}$, a new data set Γ is formed by grey 1-AGO as follows.

The grey 1-AGO converted input vector, $\mathbf{x}_p$, of $\mathbf{x}_p^{(0)}$ is given as

$$x_p(k) = \sum_{n=1}^{k} x_p^{(0)}(n) \tag{1}$$

for $1 \le k \le M$, where $x_p^{(0)}(k)$ and $x_p(k)$ are elements of $\mathbf{x}_p^{(0)}$ and $\mathbf{x}_p$, respectively. As for desired output $y_p^{(0)}$, by grey 1-AGO the new output y_p is found as

$$y_p = y_p^{(0)} + \sum_{n=1}^{M} x_p^{(0)}(n) \tag{2}$$

By (1) and (2), the new data set $\Gamma = \{(\mathbf{x}_p, y_p),$ for $1 \le p \le N_p\}$ is formed.

From (1), it is easy to recover $x_p^{(0)}(k)$ from $x_p(k)$ as

$$x_p^{(0)}(k) = x_p(k) - x_p(k-1) \tag{3}$$

Similarly, $y_p^{(0)}$ can be obtained from (2) as

$$y_p^{(0)} = y_p - \sum_{n=1}^{M} x_p^{(0)}(n) \tag{4}$$

The reverse operation of grey 1-AGO is called grey 1-IAGO.

It is reported that grey 1-AGO is able to reduce the randomness in data. To understand and to visualize the idea, the data $S^{(0)} = \{1, 4, 2, 5, 3\}$, which satisfies Conditions (i) and (ii) described previously, is given as an example. By (1), the grey 1-AGO converted data

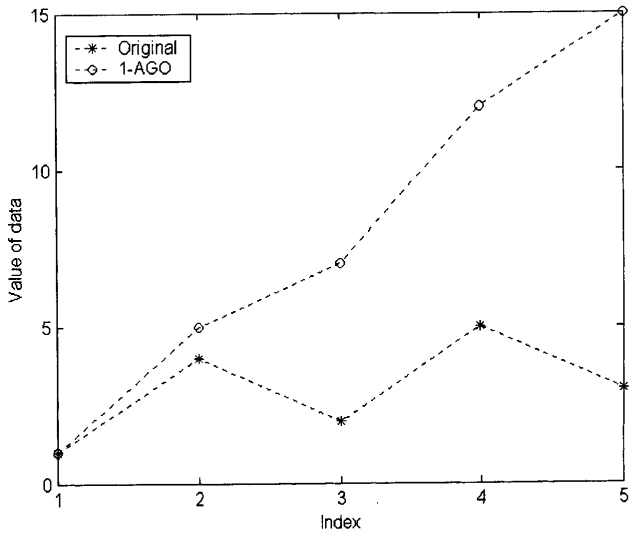


Fig. 1 The original data and grey 1-AGO converted data.

is found as $S = \{1, 5, 7, 12, 15\}$. Both $S^{(0)}$ and S are plotted in Fig. 1 which indicates that the grey 1-AGO converted data S is monotonic increasing and less random than the original data $S^{(0)}$. This example helps understanding that the grey 1-AGO is able to reduce the randomness in data.

3. Grey Neural Network

In this section, the piecewise linear neural network (PLNN) [22] is first reviewed from which the proposed grey neural network (GNN) is derived. Then the proposed GNN based on PLNN is given and the relationship of cost functions associated with GNN and PLNN is established.

3.1 Piecewise Linear Neural Network (PLNN)

For its simplicity and efficiency, the PLNN is employed in this paper. The structure of PLNN is depicted in Fig. 2. In the training stage of PLNN, it involves three steps: (i) initializing modules, (ii) expanding the number of modules, and (iii) eliminating less useful modules. Here, we emphasize on the signal flow in PLNN. For details of PLNN, one may consult [22]. The function of each block in Fig. 2 is described in the following. Given data set $\Gamma = \{(\mathbf{x}_p, y_p),$ for $1 \le p \le N_p\}$, the input vector $\mathbf{x}_p$ is augmented by constant 1. Next the global linearity is removed from y_p where $\mathbf{A}_g$ is the coefficient vector for the global linear mapping from $\mathbf{x}_p$ to y_p. Then the augmented input vector $\mathbf{x}_p$ is normalized and used to find clusters $\mathbf{c}_j$ for $1 \le j \le K$, where K is the prescribed maximum number of modules used. Once clusters $\mathbf{c}_j$ are obtained, weighted distance measures $d_j(\mathbf{x}_p, \mathbf{c}_j)$ for $1 \le j \le K$ are calculated. Then selection function $s(\cdot)$ directs $\mathbf{x}_p$ to the module whose d_j is minimum and disables other modules. Consequently, an estimate of y_p, $\hat{y}_p$, is found through the selected

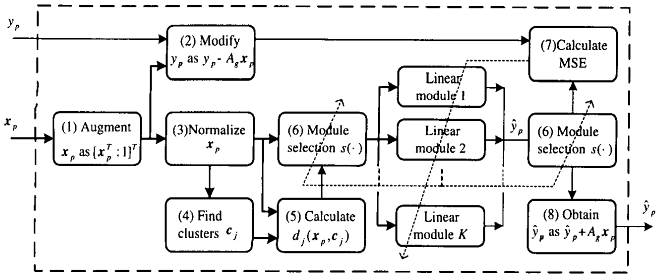


Fig. 2 The structure of piecewise linear neural network (PLNN).

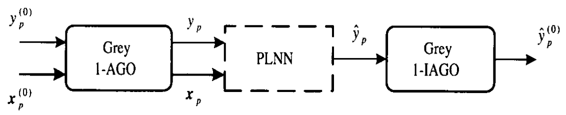


Fig. 3 The structure of grey neural network (GNN).

module. For each module, it performs linear mapping and therefore is called linear module. When $\hat{y}_p$ is obtained, the squared error of y_p and $\hat{y}_p$ is accumulated to find mean squared error (MSE) which is used to update weights in linear modules during training stage. The output of PLNN $\hat{y}_p$ is obtained by adding back the component removed in Block 2.

3.2 Grey Neural Network and Its Cost Function

The structure of grey neural network (GNN) based on PLNN is depicted in Fig. 3. With the PLNN given in Fig. 2, the proposed GNN is constructed by cascading grey 1-AGO in the input side and grey 1-IAGO in the output side of PLNN. In other words, the grey 1-AGO preprocesses input vector $\mathbf{x}_p^{(0)}$ as in (1) and desired output $y_p^{(0)}$ as in (2) while the grey 1-IAGO post-processes estimated output $\hat{y}_p$ as in (4). Note that the randomness in STLF data set affects the forecasting performance of neural networks. Increasing the randomness degrades the performance and vice versa. The purpose of grey 1-AGO is to reduce the randomness inherent in the STLF data set. Consequently, there is a hope that GNN has better forecasting performance than PLNN.

Define the cost function of PLNN with data set $\Gamma^{(0)}$ as

$$MSE_{PLNN} = \sum_{k=1}^{K}\left[\frac{1}{N_k}\sum_{p=1}^{N_k}(\hat{y}_p^{(0)} - y_p^{(0)})^2\right] \tag{5}$$

where $\hat{y}_p^{(0)}$ is an estimate of $y_p^{(0)}$ and N_k is the number of patters assigned to module k for $1 \le k \le K$. The sum of N_k is equal to N_p. Similarly, the cost function of GNN related to data set Γ is given as

$$MSE_{GNN} = \sum_{k=1}^{K}\left[\frac{1}{N_k}\sum_{p=1}^{N_k}(\hat{y}_p - y_p)^2\right] \tag{6}$$

For GNN, note that training data set $\Gamma^{(0)}$ is converted by grey 1-AGO and the new training data set Γ is put into PLNN. Therefore, the relationship of cost functions or MSE for GNN and PLNN needs to be established. The relationship ensures that the comparison between GNN and PLNN is made under same ground. To relate MSE_{PLNN} and MSE_{GNN}, (6) is rewritten as

$$\begin{aligned} MSE_{GNN} =& \sum_{k=1}^{K}\left\{\frac{1}{N_k}\sum_{p=1}^{N_k}\left[\left(\hat{y}_p - \sum_{i=1}^{M}x_p^{(0)}(i)\right)\right.\right. \\ &\left.\left. - \left(y_p - \sum_{i=1}^{M}x_p^{(0)}(i)\right)\right]^2\right\} \\ =& \sum_{k=1}^{K}\left[\frac{1}{N_k}\sum_{p=1}^{N_k}(\hat{y}_p^{(0)} - y_p^{(0)})^2\right] \\ =& MSE_{PLNN} \end{aligned} \tag{7}$$

where grey 1-IAGO is applied in the second equality. From (7), it is obvious that the forecasting performances of GNN and PLNN are compared on the same basis since $MSE_{GNN} = MSE_{PLNN}$.

4. Application of GNN to STLF Problem

In this section, the proposed GNN is applied to an STLF example. First, the STLF data set used here is described. Next the reason to use PLNN in the simulation is stated and then the determination of the number of inputs and the number of modules is described. Finally, simulations are performed and the results of PLNN and GNN are compared.

4.1 Data Description

The STLF data set used in the simulation is called TU.10. This data set was obtained from TU Electric Company in Texas. The first ten inputs are last ten minutes power load in megawatts (MW) for entire TU electric utility, which cover a large part of north Texas. The output is power load fifteen minutes in the future from the current time. All powers were originally sampled every fraction of a second, and averaged over one minute to reduce noise. The total number of patterns is 1415. In the data set TU.10, $M = 10$ and $N_p = 1415$. Note that data set TU.10 is of positive sign which satisfies Condition (i) of grey 1-AGO described in Sect. 2. Also, Condition (ii) is met in TU.10 since the power load does not change abruptly in general.

The data set TU.10 is used to train neural networks such that the problem of load frequency control (LFC) in a multi-area interconnected power system [24] can be relieved. One major problem in LFC is that area load demand fluctuations change faster than the area generation response rate which results in load upsets. In [25], it indicates that an appropriate load forecasting of the next fifteen to thirty minutes gives better performance of LFC. With the forecasted load, generation control can be accomplished beforehand and therefore reduces operation costs of multi-area interconnected power system. This is why the power load fifteen minutes ahead from the current time is used as the desired output.

4.2 Determination of Neural Network and Its Structure

Since the predictive LFC forecasts power load fifteen minutes ahead, in general an on-line training is required such that a satisfactory forecasting performance can be achieved. As for on-line training, fast convergence and network structure are two key issues. As reported in [22], PLNN is of fast convergence and training efficiency properties. Consequently, it is employed in the simulation. Note that the complexity of network structure in PLNN is a function of the number of inputs N_{inp}, the number of modules K, and the number of outputs N_{out}. In this example, only N_{inp} and K should be determined since N_{out} is fixed at one. As for N_{inp}, a small number is not appropriate for it may contain insufficient information and result in poor forecasting performance. On the other hand, a large N_{inp} requires more training time. Also, it is noted that a further input away from predicted data has less effect on forecasting performance in the k-step ahead time series prediction in general. Consequently, $N_{inp} = 10$ is employed in the given data set.

In the practical application of PLNN, we use pattern capacity [26], which is the total number of training patterns memorized perfectly, to help in the determination of K. The pattern capacity of PLNN is given as

$$C_p = K(N_{inp} + 1) \tag{8}$$

As a rule of thumb, C_p is chosen as $N_p/10$ for better generalization of trained PLNN. In the case of $N_{inp} = 10$, K is obtained as

$$K = \frac{141.5}{11} = 12.86 \tag{9}$$

Thus $K = 13$ is set in the actual STLF application.

4.3 Simulations and Comparison

The simulation procedure is described in the following. To compare the performance of PLNN and GNN, the original data set TU.10 is first put into PLNN with structure 10-K-1, for $11 \leq K \leq 20$, where 10 is the number of inputs, K is the number of modules prescribed, and 1 is the number of output. Since the PLNN is a linear type of neural networks, it is of fast convergence property. Consequently, 15 training iterations are sufficient for all 10-K-1 structures of PLNN. The MSE_{PLNN} for each different structure is recorded.

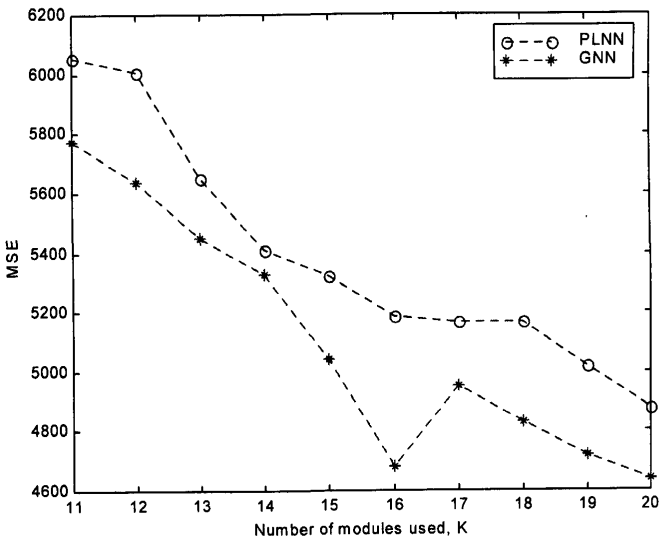


Fig. 4 Comparison of MSE_{PLNN} and MSE_{GNN}.

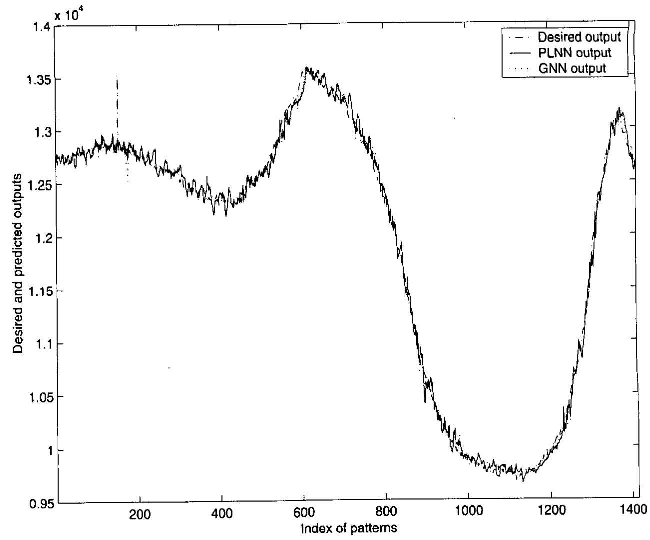


Fig. 5 The predicted outputs and desired output.

Next, TU.10 is preprocessed by grey 1-AGO as given in (1) and (2). The grey 1-AGO converted data set is denoted as TU.10a. Then the GNN is trained with identical training conditions as that for PLNN. For each different structure the MSE_{GNN} is recorded. Finally, the MSE_{PLNN} and MSE_{GNN} are compared.

By the simulation procedure described in the previous paragraph, the MSE_{PLNN} and MSE_{GNN} for different K are plotted in Fig. 4. As expected, it indicates the GNN outperforms the PLNN for all cases. With the desired outputs, the predicted outputs of PLNN and GNN in the case of $K = 13$ are depicted in Fig. 5. The scatter plots associated with PLNN and GNN for the case of $K = 13$ are, respectively, given in Figs. 6 and 7 where the diagonal line is an auxiliary line for better observation.

One interesting result in Fig. 4 is that the MSE_{GNN} for $K = 16$ is less than that for $17 \leq K \leq 19$. One possible reason is that the case of $K = 16$ has

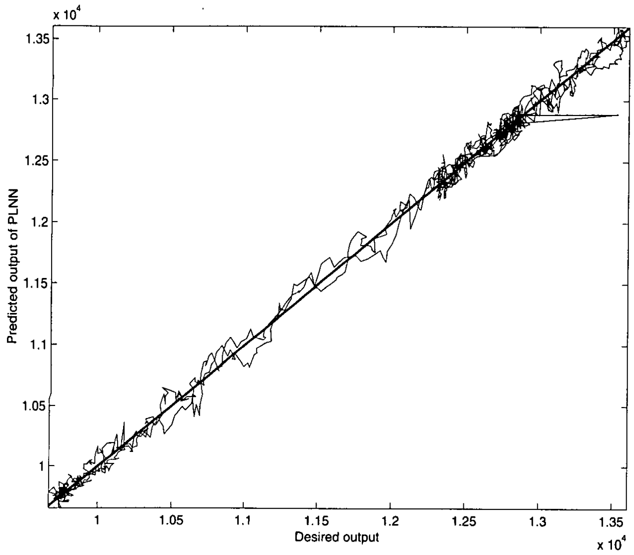


Fig. 6 The scatter plot for PLNN.

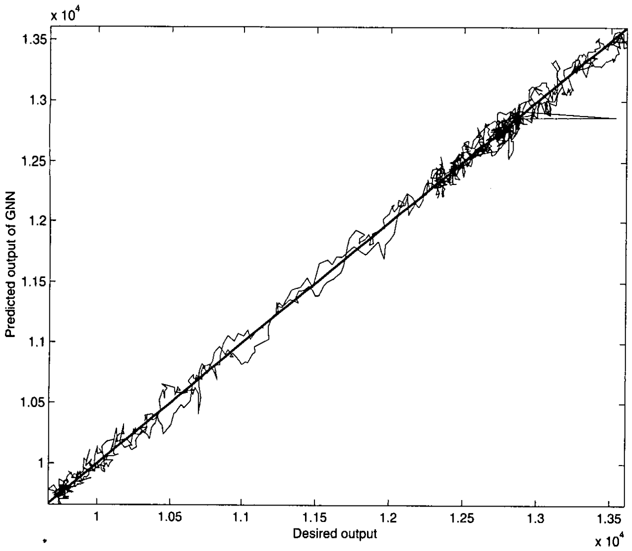


Fig. 7 The scatter plot for GNN.

better clusters and therefore has better performance. The simulation results of PLNN and GNN are summarized and compared in Table 1 where $E\%$ denotes the improvement on MSE in percentage and is defined as

$$E\% = \frac{MSE_{PLNN} - MSE_{GNN}}{MSE_{PLNN}} \times 100\% \qquad (10)$$

5. Conclusions and Further Researches

This work is motivated by two observations in the following. First, the forecasting performance of neural network is affected by the randomness in STLF data. Second, the grey 1-AGO is of randomness reduction property. By the observations, in this paper we first proposed the GNN whose structure consists of the grey 1-AGO, PLNN, and the grey 1-IAGO. Next, the GNN is applied to a STLF problem. As expected, the GNN

Table 1 Simulation results of PLNN and GNN.

	$MSE_{PLNN}(\mathrm{MW})^2$	$MSE_{GNN}(\mathrm{MW})^2$	$E\%$
$K = 11$	6055.22	6008.73	4.6
$K = 12$	6007.92	5636.42	6.1
$K = 13$	5651.48	5448.76	3.5
$K = 14$	5412.43	5329.88	1.5
$K = 15$	5321.55	5041.29	5.2
$K = 16$	5185.90	4682.50	9.7
$K = 17$	5169.96	4952.16	4.2
$K = 18$	5169.57	4831.55	6.5
$K = 19$	5016.19	4717.62	5.9
$K = 20$	4879.28	4638.49	4.9

outperforms the PLNN. The simulation results indicate that the GNN with different network structures has better forecasting performance, whose range from 1.5% up to 9.7%, than the corresponding PLNN. With simple preprocessing grey 1-AGO and post-processing grey 1-IAGO, the forecasting performance of PLNN has been improved.

In the given STLF example, data set TU.10 consists of power load feature only. However, some useful features, such as weather conditions, day information, and so forth, may help in more accurate forecasting. With these extra features, the conditions required in grey 1-AGO could fail to be met and thus the GNN may not improve the forecasting performance. Consequently, our further researches on the proposed GNN will concentrate on the release of Conditions (i) and (ii) required in grey 1-AGO such that other useful features can be included in STLF data set and the GNN can be extended to mapping problems other than STLF problem.

Acknowledgement

This work was supported by the National Science Council of the Republic of China under grants NSC 89-2218-E-270-001 and NSC 90-2213-E-270-006.

References

[1] M.H. Choueiki, C.A. Mount-Campbell, and S.C. Ahalt, "Implementing a weighted least squares procedure in training a neural network to solve the short-term load forecasting problem," IEEE Trans. Power Systems, vol.12, no.4, pp.1689–1694, 1997.

[2] I. Erkmen and A. Ozdogan, "Short term load forecasting using genetically optimized neural network cascaded with a modified Kohonen clustering process," The 12th IEEE International Symposium on Intelligent Control, pp.107–112, Istanbul, Turkey, 1997.

[3] K.-L. Ho, Y.-Y. Hsu, and H.-H. Yang, "Short term load forecasting using a multilayer neural network with an adaptive learning algorithm," IEEE Trans. Power Systems, vol.7, no.1, pp.141–149, 1992.

[4] M.H. Choueiki, C.A. Mount-Campbell, and S.C. Ahalt, "Building a 'Quasi optimal' neural network to solve the short-term load forecasting problem," IEEE Trans. Power Systems, vol.12, no.4, pp.1432–1439, 1997.

[5] D.K. Ranaweera, N.F. Hubele, and A.D. Papalexopoulos,

"Application of radial basis function neural network model for short-term load forecasting," IEE Proc. Generation, Transmission, and Distribution, vol.142, no.1, pp.45–50, 1995.

[6] A.G. Bakirtzis, J.B. Theocharis, S.J. Kiartzis, and K.J. Satsios, "Short term load forecasting using fuzzy neural networks," IEEE Trans. Power Systems, vol.10, no.3, pp.1518–1524, 1995.

[7] K.-H. Kim, J.-K. Park, K.-J. Hwang, and S.-H. Kim, "Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy experts systems," IEEE Trans. Power Systems, vol.10, no.3, pp.1534–1539, 1995.

[8] T.W.S. Chow and C.-T. Leung, "Nonlinear autoregressive integrated neural network model for short-term load forecasting," IEE Proc. Generation, Transmission, and Distribution, vol.143, no.5, pp.500–506, 1996.

[9] S.F. Papadakis, J.B. Theocharis, S.J. Kiartzis, and A.G. Bahirtzis, "A novel approach to short-term load forecasting using fuzzy neural networks," IEEE Trans. Power Systems, vol.13, no.2, pp.480–492, 1998.

[10] M. Djukanovic, B. Babic, D.J. Sobajic, and Y.-H. Pao, "Unsupervised/supervised learning concept for 24-hour load forecasting," IEE Proc. Generation, Transmission, and Distribution, vol.140, no.4, pp.311–318, 1993.

[11] M.A. El-Sharkawi, R.J. Marks II, S. Oh, and C.M. Brace, "Data partitioning for training a layered perceptron to forecast electric load," ANNPS'93, pp.66–68, 1993.

[12] B.S. Kermanshahi, C.H. Poskar, G. Swift, P. Mclaren, W. Pedrycz, W. Buhr, and A. Silk, "Artificial neural network for forecasting daily loads of a Canadian electric utility," ANNPS'93, pp.302–307, 1993.

[13] J.R. McDonald and W. Rattray, "Experience with artificial neural network models for short-term load forecasting in electrical power systems: A proposed application of expert networks," The 3rd International Conference on Artificial Neural Networks, pp.123–127, 1993.

[14] S. Heine and I. Neumann, "Data analysis by means of Kohonen feature maps for load forecast in power systems," IEE Colloquium on Advances in Neural Networks for Control and Systems, pp.6/1–6/4, 1994.

[15] A. Piras, B. Buchenel, and Y. Jaccard, "Heterogeneous artificial neural network for short term electrical load forecasting," IEEE Trans. Power Systems, vol.11, no.1, pp.397–402, 1996.

[16] D.K. Ranaweera, G.G. Karady, and R.G. Farmer, "Effect of probabilistic inputs on neural network-based electric load forecasting," IEEE Trans. Neural Networks, vol.7, no.6, pp.1528–1532, 1996.

[17] A.S. Alfuhaid, M.A. El-Sayed, and M.S. Mahmoud, "Cascaded artificial neural networks for short-term load forecasting," IEEE Trans. Power Systems, vol.12, no.4, pp.1524–1529, 1997.

[18] S.J. Kiartzis, C.E. Zoumas, A.G. Bakirtzis, and V. Petridis, "Data pre-processing for short-term load forecasting in an autonomous power system using artificial neural networks," ICECS'96, pp.1021–1024, 1996.

[19] A. Khotanzad, R.-C. Hwang, A. Abaye, and D. Maratukulam, "An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities," IEEE Trans. Power Systems, vol.10, no.3, pp.1716–1722, 1995.

[20] H. Ma, A.A. El-Keib, and X. Ma, "Training data sensitivity of artificial neural network-based power load forecasting," Proc. 26th Southern Symposium on System Theory, pp.650–652, 1994.

[21] J.L. Deng, "Introduction to grey system theory," Journal of Grey Systems, vol.1, no.1, pp.1–24, 1989.

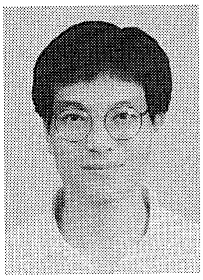
[22] H. Chanddrasekaran and M.T. Manry, "Convergent design of a piecewise linear neural network," IJCNN'99, vol.2, pp.1339–1344, 1999.

[23] S. Haykin, Neural Networks — A Comprehensive Foundation, Macmillan, 1994.

[24] N. Jaleeli, L.S. VanSlyck, D.N. Ewart, H.F. Lester, and A.G. Hoffmann, "Understanding automatic generation control," IEEE Trans. Power Systems, vol.7, no.3, pp.1106–1122, 1992.

[25] A.S. Debs, Modern Power Systems Control and Operation, Kluwer, Academic Publishers, 1988.

[26] K.K. Kim, "A local approach for sizing the multilayer perceptron (nearest neighbor mapper)," Ph.D. Dissertation, The University of Texas at Arlington, 1996.



Cheng-Hsiung Hsieh was born in Nantou, Taiwan, in 1963. He received the B.S. degree in Electronic Engineering from National Taiwan Institute of Technology, Taiwan, in 1989. In 1995, he earned the M.S. degree from the Department of Electrical Engineering of Tennessee Technological University, USA. He obtained his Ph.D. degree in Electrical Engineering from the University of Texas at Arlington, USA, in 1997. Currently, his research interests include grey systems, artificial neural networks, digital image processing, statistical signal processing, and digital communications.