Gain Estimation for an AC Power Line Data Network Transmitter Using a Self-Structured Neural Network and Genetic Algorithm[1]

H.K. Lam[2], S.H. Ling, F.H.F. Leung, P.K.S. Tam and Y.S. Lee

Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

***Abstract* - This paper presents the estimation of the transmission gain for the AC power line data network in an intelligent home. The estimated gain ensures the transmission reliability and efficiency. A neural network with link switches is proposed to perform the estimation. Genetic algorithm with arithmetic crossover and non-uniform mutation is employed to tune the parameters and the structure of the proposed neural network. An application example will be given.**

I. INTRODUCTION

Nowadays, homes [1] should offer smart features to ensure a higher security, more entertainment and comfort to residents. A reliable communication channel among electrical appliances and users has to be present to support these features. Also, the energy can be used in a more efficient manner. At present, many researchers and companies are developing intelligent home systems. A phone-based remote controller facilitates home users to issue control commands to their home appliances through telephones [2]. A small two-arm mobile robot in a home can be controlled via an ISDN link [3]. In the U.S., X-10 systems are commonly used to support low-cost and slow-rate AC power line data networks.

Without relying on the manufacturers of electrical appliances and installing a LAN, one simple way to realize the communication channel [4-5] for home appliances and users is to make use of the AC power line. We have successfully implemented a power line data network [7] based on spread-spectrum technology [6], which facilitates communications at 10 Kbps in the noisy and signal-distorting environment of AC power lines. This network serves as a backbone for an intelligent home system through which electrical appliances can be controlled via line/mobile phones, PDAs, keypads or personal computers anytime and anywhere, inside or outside the home. One of the major issues of the power line data network is the reliability, which ensures the sent information to be correctly received. However, the electric power line at home has many appliances connected to it, and each appliance has different characteristics that affect the power line conditions. When using an AC power line as a networking medium [7], one has to deal with problems such as electromagnetic interference, varying impedance, narrow frequency impairments (due to noise), and signal attenuation. To increase the network reliability, a higher gain for transmitters should be used; however, the transmission rate has to be reduced and the power for data transfer will increase. Thus, the gain of the transmitters in a power line data network is an important factor to ensure the transmission reliability and efficiency. In this paper, a neural network with link switches [8-9] is proposed to estimate the transmitter gain. By introducing the link switches, an optimal network structure can be obtained after tuning to provide an optimal performance. Genetic algorithm (GA) with arithmetic crossover and non-uniform mutation [10] is employed for the tuning process.

This paper is organized as follows. The neural network with link switches is presented in Section II. Turing of the parameters and the structure of the proposed neural network for estimating the transmitter gain will be presented in Section III. Section IV will present an example. A conclusion will be drawn in Section V.

II. NEURAL NETWORK WITH LINK SWITCHES

Neural network was proved to be a universal approximator [14]. A 3-layer feed-forward neural network can approximate any nonlinear continuous function to an arbitrary accuracy. Neural networks are widely applied in areas such as prediction [12], system modeling and control [14]. Owing to its particular structure, a neural network is good at learning [11] using some algorithms such as GA [10] and back propagation [11]. In general, the learning steps of a neural network are as follows. First, a network structure is defined with fixed numbers of inputs, hidden nodes and outputs. Second, an algorithm is chosen to realize the learning process. However, a fixed structure may not provide the optimal performance within a given training period. A small network may not provide good performance owing to its limited information processing power. A large network, on the other hand, may have some of its connections redundant [15-16]. Moreover, the implementation cost for a large network is high. To obtain the network structure automatically, constructive and destructive algorithms can be used [15]. The constructive algorithm starts with a small network. Hidden layers, nodes and connections are added to expand the network dynamically [16-21]. The destructive algorithm starts with a large network. Hidden layers, nodes and connections are then deleted to contract the network dynamically [22-23]. The design of a network structure can be formulated into a search problem. Genetic algorithms [8, 24] were employed to obtain the solution. Pattern-classification approaches for designing the network structure[25] can also be found.

In this paper, a three-layer neural network with switches introduced in some links is proposed to facilitate the tuning of

[1] The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administration Region, China (Project No. PolyU 5098/01E).

[2] email: hkl@eie.polyu.edu.hk

0-7803-7474-6/02/$17.00 ©2002 IEEE

the network structure. As a result, a given fully connected feedforward neural network may become a partly connected network after learning. This implies that the cost of implementing the proposed neural network, in terms of hardware implementation and processing time, can be reduced. The network structure and parameters will be tuned simultaneously using the GA with arithmetic crossover and non-uniform mutation [13]. The proposed multiple-input-multiple-output three-layer neural network is shown in Fig. 1. The main different point is that a unit step function is introduced to each link. Such a unit step function is defined as,

$$\delta(\alpha) = \begin{cases} 0 & \text{if } \alpha < 0 \\ 1 & \text{if } \alpha \geq 0 \end{cases}, \ \alpha \in \Re \tag{1}$$

This 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 as follows,

$$y_k(t) = \sum_{j=1}^{n_h} \delta(s_{jk}^2) w_{jk} \, logsig\left[\sum_{i=1}^{n_{in}} \left(\delta(s_{ij}^1) v_{ij} z_i(t) - \delta(s_j^1) b_j^1\right)\right] - \delta(s_k^2) b_k^2$$
$$, k = 1, 2, \ldots, n_{out} \tag{2}$$

$z_i(t)$, $i = 1, 2, \ldots, n_{in}$, are the inputs which are functions of a variable t; n_{in} denotes the number of inputs; n_h denotes the number of the hidden nodes; w_{jk}, $j = 1, 2, \ldots, n_h$; $k = 1, 2, \ldots, n_{out}$, denotes the weight of the link between the j-th hidden node and the k-th output; v_{ij} denotes the weight of the link between the i-th input and the j-th hidden node; s_{ij}^1 denotes the parameter of the link switch from the i-th input to the j-th hidden node; s_{jk}^2 denotes the parameter of the link switch from the j-th hidden node to the k-th output; n_{out} denotes the number of outputs of the proposed neural network; b_j^1 and b_k^2 denote the biases for the hidden nodes and output nodes respectively; s_j^1 and s_k^2 denote the parameters of the link switches of the biases to the hidden and output layers respectively; $logsig(\cdot)$ denotes the logarithmic sigmoid function:

$$logsig(\alpha) = \frac{1}{1 + e^{-\alpha}}, \ \alpha \in \Re \tag{3}$$

$y_k(t)$, $k = 1, 2, \ldots, n_{out}$, is the k-th output of the proposed neural network. By introducing the switches, the weights w_{jk} and v_{ij}, and the switch states can be tuned. It can be seen that the weights of the links govern the input-output relationship of the neural network while the switches of the links govern the structure of the neural network.

III. Tuning Parameters and Structure

The proposed neural network can be employed to learn the input-output relationship of an application using the improved GA. The input-output relationship is described by,

$$\mathbf{y}^d(t) = \mathbf{g}\left(\mathbf{z}^d(t)\right), \ t = 1, 2, \ldots, n_d \tag{4}$$

where $\mathbf{z}^d(t) = \left[z_1^d(t) \quad z_2^d(t) \quad \cdots \quad z_{n_{in}}^d(t)\right]$ and $\mathbf{y}^d(t) = \left[y_1^d(t) \quad y_2^d(t) \quad \cdots \quad y_{n_{out}}^d(t)\right]$ are the given inputs and the desired outputs of an unknown nonlinear function $\mathbf{g}(\cdot)$ respectively; n_d denotes the number of input-output data pairs. The fitness function is defined as,

$$fitness = \frac{1}{1 + err} \tag{5}$$

$$err = \sum_{k=1}^{n_{out}} \frac{\sum_{t=1}^{n_d} \left| y_k^d(t) - y_k(t) \right|}{n_d} \tag{6}$$

The objective is to maximize the fitness value of (5) using the GA by setting the chromosome to be $\left[s_{jk}^2 \quad w_{jk} \quad s_{ij}^1 \quad v_{ij} \quad s_j^1 \quad b_j^1 \quad s_k^2 \quad b_k^2\right]$ for all i, j, k. It can be seen from (5) and (6) that a larger value of *fitness* implies a smaller value of *err*.

IV. Application Example

The proposed neural network will be employed to estimate the gain of the transmitter in a power line data network. A network with 48 inputs and 48 outputs is employed to perform the estimation. The inputs of the proposed neural network are the gains in every half-hour of the previous day, while the outputs are the estimated gains in every half-hour of the present day. Seven neural networks will be used for estimating the transmission gain. These neural networks are named Sunday-Monday, Monday-Tuesday, Tuesday-Wednesday, Wednesday-Thursday, Thursday-Friday, Friday-Saturday neural networks. For instance, the Sunday-Monday neural network makes use of the transmission gains of Sunday to estimate the transmission gains of Monday. To perform the training, we have to collect some testing patterns. 48 transmission gains for every half-hour at each day (24 hours) will be measured. To measure the optimal gain, a data packet is continuously sent from the transmitter to the receiver while the transmission gain value is increased gradually. The increment in the gain value will stop at the point when the data packet can be correctly received. At that moment, the value of the transmission gain is the smallest possible one that is not susceptible to interference, and the optimal transmission gain would be set at a value a bit higher than this smallest one.

In this application example, transmission gains for 7 weeks are collected as the testing patterns and employed to train the proposed neural network using the GA with arithmetic crossover and non-uniform mutation [10]. Take the Sunday-Monday neural network for instance, the measured transmission gains for Sundays of the previous 7 weeks will serve as the inputs and the desired outputs are the measured transmission gains of Mondays in the previous 7 weeks respectively. The rationale for this arrangement of training is based on the assumption that relation between the gain pattern of Sunday and that of Monday is approximately constant in the most recent seven weeks. Seven neural networks will be used to derive 7 different relations for the days in a week. The number of hidden nodes is chosen to be 5. The probabilities of crossover and mutation are selected to be 0.9 and 0.005 respectively. The training will last for 2000 iterations. The upper and lower bounds of each parameter are 1 and −1 respectively. The initial values of all the parameters are chosen to be 0.2. The shaping parameter of the GA for non-uniform mutation [10] is selected to be 2. All the 7 neural networks are trained in the same way. The input and output patterns are normalized such that the elements of the input vector and the desired output vector are between 0 and 1. The fitness function for training is defined in (5) and (6). By minimizing the errors between the desired outputs and the proposed neural network's outputs, the characteristics of the transmission gain pattern are learnt. After training, the proposed neural network will be employed to estimate the transmission gain. The 48 transmission gains of the previous day will be fed to the trained network. The 48 neural network outputs will be the estimated transmission gains for every half-hour of the present day. The transmission gain employed by the transmitter in every minute is obtained by a linear interpolation equation:

$$\tilde{g}(\xi+30\tau)=\frac{\hat{g}(\tau+1)-g(\tau)}{30}\xi+g(\tau)\,,\ \xi=1,2,\ldots,30,\ \tau=0,1,\ldots,47 \tag{7}$$

where $\tilde{g}(\xi+30\tau)$ denotes the transmission gain employed by the transmitter at $\xi+30\tau$ minutes counted from 00:00 am; $\hat{g}(\tau+1)$ denotes the τ+1-th output of the corresponding neural network (i.e. the estimated transmission gain at $30(\tau+1)$ minutes counted from 00:00 am); $g(\tau)$ denotes the measured transmission gain at 30τ minutes counted from 00:00 am. Note that we measure the optional gain every 30 minutes.

For comparison purpose, a traditional 3-layer feedforward neural network [29] will be trained by back-propagation with momentum and adaptive learning rate [29] under the same condition. The learning parameters are as follows: the learning rate is 0.2, the ratio to increase the learning rate is 1.05, the ratio to decrease the learning rate is 0.7, the maximum validation failures is 5, the maximum performance increase is 1.04, and the momentum constant is 0.9. Table I shows the simulation results of proposed and traditional approaches for the Sunday-Monday neural network among 10 trails. To obtain the fitness value for the testing patterns, the measured transmission gains for Sunday of week 8 are fed to the trained neural network to obtain the estimated transmission gains for Monday of week 8. The fitness value is then obtained by (5) and (6) based on the estimated and measured transmission gains. The number of links for the proposed network is 486 while that of the traditional network is 533. Fig. 2 shows the actual (solid line) and predicted (dotted line) normalized gains for Monday of week 8 using the proposed approach. Fig. 3 shows the actual (solid line) and predicted (dotted line) normalized gains for Monday of week 8 using the traditional approach. It can be seen that the performance of the proposed approach is better than that of the traditional approach in terms of the fitness values and number of links.

It should be noted that the proposed networks will be trained every week using the measured data of most recent 7 weeks. Consequently, some gradual environment changes such as the seasonal effects can be considered.

V. CONCLUSION

A neural network with link switch has been proposed. The link switches facilities the tuning of the structure of a neural network. GA with arithmetic crossover and non-uniform mutation has been employed to tune the parameters and the structure of the proposed neural network, which is used to estimate the transmission gain of the transmitter in the AC power line data network of an intelligent home.

REFERENCES

[1] C. Douligenis, "Intelligent home systems," *IEEE Communications Magazine*, vol. 31, Oct. 1993, pp. 52-61.
[2] E.M.C. Wong, "A phone-based remote controller for home and office automation," *IEEE Transactions on Consumer Electronics*, vol. 40, no. 1, Feb. 1994, pp. 148-152.
[3] J. Gray, D.G. Caldwell, and N. Linge, "A new perspective on the intelligent home," *Impact of Multimedia Services on the Home Environment, IEE Colloquium*, 1996, pp. 7/1-7/4.
[4] H.C. Ferrerira, H.M. Grove, O Hooijen and AJ HanVinc, "Power line communication: overview," in *Proc. IEEE AFRICON 4th*, vol. 2, 1996, pp. 558-563.
[5] D.R. Amitava, "Networks for homes," *IEEE Spectrum*, vol. 36, Dec. 1999, pp.26-33.
[6] D. Radford, "Spread-spectrum data leap through ac power line wiring," *IEEE Spectrum*, Nov. 1996, pp. 48-53.
[7] L.K. Wong, S.H. Ling, F.H.F. Leung, Y.S. Lee, S.H. Wong, T.H. Lee, H.K. Lam, Kai-Hong Ho, D.P.K. Lun, T.C. Hsung, "An intelligent home," in *Proc. Workshop on Service Automation and Robotics, Hong Kong*, June 2000, pp. 111-119.
[8] H.K. Lam, S.H. Ling, F.H.F. Leung and P.K.S. Tam, "Tuning of the Structure and Parameters of Neural Network using an Improved Genetic Algorithm," in *Proc. 27th Annual Conference of the IEEE Industrial Electronics Society, IECON'2001*, Denver, Nov, 2001, pp. 25-30.
[9] S.H. Ling, F.H.F. Leung, H.K. Lam, and P.K.S. Tam, "Short-term daily load forecasting in an intelligent home with GA-based neural network," in *Proce. IJCNN 2002, WCCI*, Honolulu, Hawaii, 12-17, May, 2002, pp.997-1002.
[10] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*, (2n ed.). Springer-Verlag, 1994.
[11] D.T. Pham and D. Karaboga, *Intelligent Optimization Techniques, Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Springer, 2000.
[12] M. Li, K. Mechrotra, C. Mohan, and S. Ranka, "Sunspot numbers forecasting using neural network," in *Proc. 5th IEEE Int. Symp. Intelligent Control*, 1990, pp.524-528.
[13] H.K. Lam, S.H. Ling, F.H.F. Leung and P.K.S. Tam, "Optimal and stable fuzzy controllers for nonlinear systems subject to parameter uncertainties using genetic algorithm," in *Proc. 10th Int. Conf. Fuzzy Systems, IEEEFUZZ'2001*, Melbourne, Australia, 2-5 Dec, 2001, pp. 908-911.
[14] M. Brown and C. Harris, *Neuralfuzzy Adaptive Modeling and Control*. Prentice Hall, 1994.
[15] X. Yao, "Evolving artificial networks," *Proceedings of the IEEE*, vol.

87, no. 7, pp. 1423-1447, 1999.

[16] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Tran. Neural Networks*, vol. 8, no. 3, pp. 694-713, 1997.

[17] F.J. Lin, C.H. Lin and P.H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *IEEE Trans. Fuzzy Systems*, vol. 9, no. 5, pp. 751-759, Oct. 2001.

[18] Y. Hirose, K. Yamashita and S. Hijiya, "Back-propagation algorithm which varies the number of hidden units," *Neural Networks*, vol. 4, no. 1, pp. 61-66, 1991.

[19] A. Roy, L.S. Kim, and S. Mukhopadhyay, "A polynomial time algorithm for the construction and training of a class of multiplayer perceptions," *Neural Networks*, vol. 6, no. 4, pp. 535-545, 1993.

[20] N.K. Treadold and T.D. Gedeon, "Exploring constructive cascade networks," *IEEE Trans. Neural Networks*, vol. 10, no. 6, pp. 1335-1350, Nov, 1999.

[21] C.C. Teng and B.W. Wah, "Automated learning for reducing the configuration of a feedforward neural network," *IEEE Trans. Neural Networks*, vol. 7, no. 5, pp. 1072-1085, Sep, 1996.

[22] Y.Q. Chen, D.W. Thomás and M.S. Nixon, "Generating-shrinking algorithm for learning arbitrary classification," *Neural Networks*, vol. 7 no. 9, pp. 1477-1489, 1994.

[23] M.C. Moze and P. Smolensky, "Using relevance to reduce network size automatically," *Connect. Sci.*, vol. 1, no. 1, pp. 3-16, 1989.

[24] G.P. Miller, P.M. Todd and S.U. Hegde, "Designing neural networks using genetic algorithms," in *Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications*, 1989, pp. 379-384.

[25] N. Weymaere and J. Martens, "On the initialization and optimization of multiplayer perceptrons," *IEEE Trans. Neural Networks*, vol. 5, pp. 738-751, Sept. 1994.

[26] S.S. Haykin, *Neural Networks: A Comprehensive Foundation*, (2nd Ed.). Upper Saddle River, N. J.: Prentice Hall, 1999.

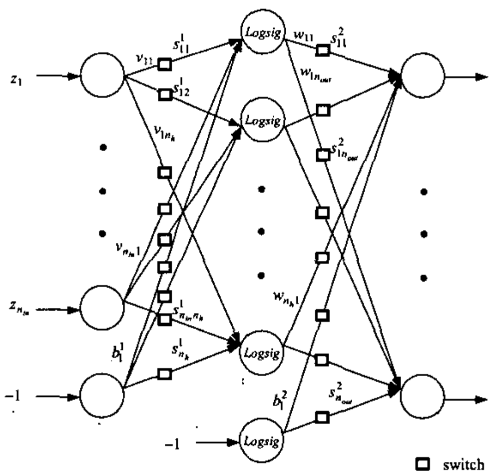

Fig. 1. The proposed 3-layer neural network.

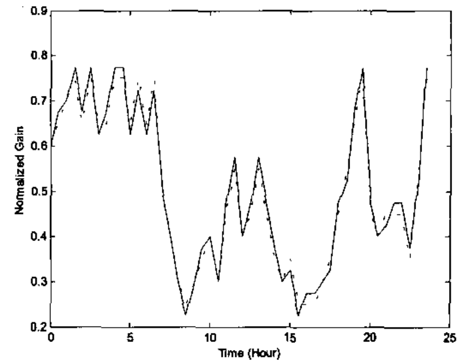

Fig. 2. The actual (solid line) and predicted (dotted line) normalized gains for week 8 using the proposed approach.

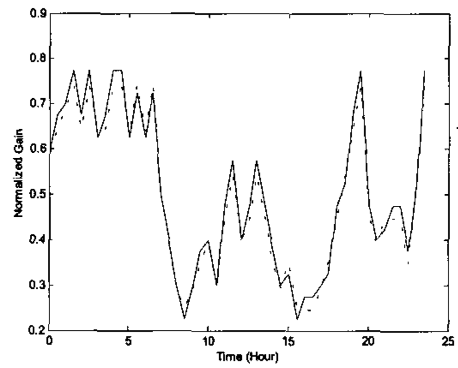

Fig. 3. The actual (solid line) and predicted (dotted line) normalized gains for week 8 using the traditional approach.

	Proposed Approach	Traditional Approach
Fitness Value (Training)	0.9838	0.9818
Fitness Value (Testing)	0.9816	0.9789
Number of Links	486	533

Table I. Simulation results based on the proposed and traditional approaches.