On Interpretation of Graffiti Digits and Characters for eBooks: Neural-Fuzzy Network and Genetic Algorithm Approach

K. F. Leung, Frank H. F. Leung, *Senior Member, IEEE*, H. K. Lam, *Member, IEEE*, and S. H. Ling, *Student Member, IEEE*

***Abstract*—This paper presents the rule optimization, tuning of the membership functions, and optimization of the number of fuzzy rules, of a neural-fuzzy network (NFN) using a genetic algorithm (GA). The objectives are achieved by training a proposed NFN with rule switches. The proposed NFN and GA are employed to interpret graffiti number inputs and commands for electronic books (eBooks).**

***Index Terms*—Electronic books (eBooks), genetic algorithm (GA), neural-fuzzy networks (NFNs).**

I. Introduction

THE genetic algorithm (GA) is a powerful random search technique to handle optimization problems [1]–[3]. This is especially useful for complex problems with a large number of parameters that make the global analytical solutions difficult to obtain. It has been widely applied in different areas such as fuzzy control [4]–[6], [10], path planning [7], greenhouse climate control [8], modeling and classification [9], [13] etc.

Neural-fuzzy networks (NFNs) have been proved to be a universal approximator [11], which can approximate nonlinear functions to an arbitrary accuracy. Expert knowledge and experience can be incorporated into an NFN [11], [13]. In view of its specific structure, an NFN can be used to realize a learning process [2]. In general, learning involves two aspects: 1) defining a network structure based on fuzzy rules and 2) choosing an algorithm to realize the learning process. Usually, the network structure is fixed for a learning process as the number of rules is fixed. However, this fixed structure may not provide the best performance within a given training period. If the structure of the NFN is too complicated, the training period will be long and the implementation cost will be high.

Notebook computers and personal digital assistants (PDAs) are widely accepted by the public. In particular, electronic books (eBooks) are winning their popularity as a kind of media that can offer rich contents and features such as multimedia presentations, instant dictionaries, bookmark functions, etc., within a small handheld device. As shown in Fig. 1, an eBook reader should have no keyboard or mouse. The main input device is a touch screen. Typing can be done by using an on-screen keyboard. Yet, this is not a convenient way of input. One natural way of inputting information to the eBook is to write directly on the touch screen. However, computers are only good at numerical manipulation, while the interpretation of graffiti is a symbolic manipulation process. Thus, a way to convert a symbolic manipulation process to a numerical manipulation process should be found. Different methodologies for recognizing handwritten characters can be found in the literature. In general, four different approaches [14] are used: template matching, statistical techniques, structural techniques, and neural/NFNs. The idea of the template matching approach is to determine the best match between the stored templates and the input. For statistical techniques, statistical decision theory is employed to determine the class to which the input belongs. Hidden Markov modeling [15] is one of the popular statistical techniques for handwritten character recognition. On applying structural techniques, some complex patterns are represented by some simpler patterns. Based on these simpler patterns, the input can be classified. Examples of structural techniques include grammatical [16] and graphical [17] methods. In general, the main idea of the neural/neural-fuzzy network approaches [18] is to learn

Fig. 1. eBook reader.

Manuscript received January 7, 2002; revised February 7, 2003. Abstract published on the Internet January 13, 2004. This work was supported by a Research Grant of the Centre for Multimedia Signal Processing, The Hong Kong Polytechnic University, under Project A432.

The authors are with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

Digital Object Identifier 10.1109/TIE.2004.825285

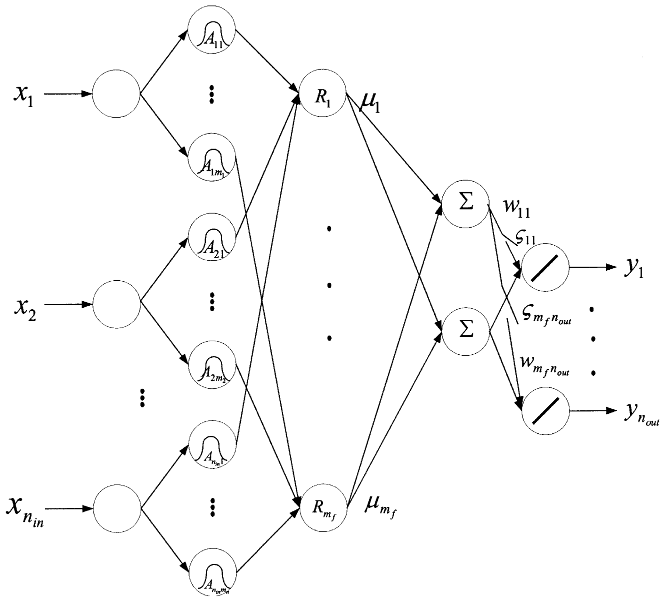


Fig. 2. Proposed NFN.

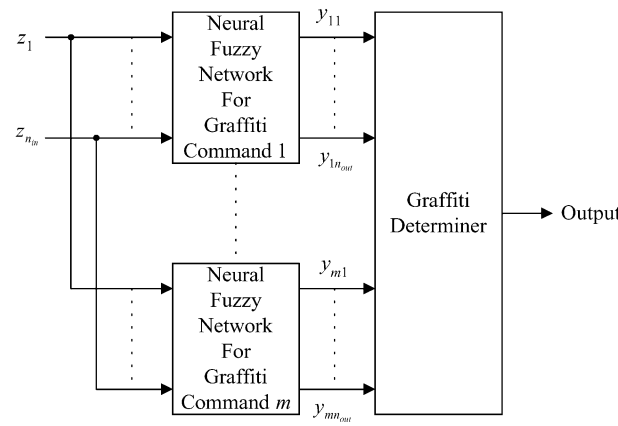


Fig. 3. Block diagram of the graffiti digit and character interpreter.

Digits or Characters	Strokes	Digits or Characters	Strokes
0(a)		6	
0(b)		7	
1		8(a)	
2		8(b)	
3		9	
4		Backspace	
5(a)		Carriage Return	
5(b)		Space	

Fig. 4. Graffiti digits and characters (with the dot indicating the starting point of the graffiti).

the features of the training patterns through some processes. The input's features can then be recognized using the trained neural/NFN. The objective of this paper is to develop an algorithm recognizing handwritten digits and characters. An NFN with rule switches is proposed to perform the interpretation of graffiti. A GA with arithmetic crossover and nonuniform mutation [3] will be employed to train the proposed NFN. The result is an interpreter that can interpret the digits 0–9 and three (control) characters, namely, *Backspace*, *Carriage Return*, and *Space*. Through this NFN, the optimal fuzzy rules and membership functions can be tuned. It is applied to an eBook reader experimentally.

This paper is organized as follows. The proposed NFN with rule switches will be presented in Section II. A graffiti interpreter, which is formed by the proposed NFN, is proposed in Section III. The tuning of the membership functions and rules of the NFN will be presented. Application results on interpreting graffiti digits and characters for eBooks will be given in Section IV. A conclusion will be drawn in Section V.

II. NFN With Rule Switches

We use a fuzzy associative memory (FAM) [12] type of rule base for the NFN. An FAM is formed by partitioning the universe of discourse of each fuzzy variable according to the level of fuzzy resolution chosen for the antecedents, thereby generating a grid of FAM elements. The entry at each grid element in the FAM corresponds to a fuzzy premise. An FAM is thus interpreted as a geometric or tabular representation of a fuzzy logic rule base. For an NFN, the number of possible rules may be too large. This makes the network complex while some rules may not be necessary. The implementation cost is also unnecessarily high. Thus, a multi-input multi-output NFN, which can have an optimal number of rules and membership functions, is proposed. The main difference between the proposed network and the traditional network is that a unit step function is introduced to each rule. The unit step function is defined as

$$\delta(\varsigma) = \begin{cases} 0, & \text{if } \varsigma \le 0 \\ 1, & \text{if } \varsigma > 0 \end{cases}, \qquad \varsigma \in \Re. \tag{1}$$

This is equivalent to adding a switch to each rule in the NFN. The rule is used if the corresponding rule switch is closed. Otherwise, the rule is not necessary. Referring to Fig. 2, we define the input and output variables as x_i and y_j respectively; where $i = 1, 2, \ldots, n_{\text{in}}$; n_{in} is the number of input variables; $j = 1, 2, \ldots, n_{\text{out}}$; and n_{out} is the number of output variables. The behavior of y_j of the NFN is governed by m_f fuzzy rules in the following format;

$$\begin{aligned} &R_g: \text{IF } x_1(t) \text{ is } A_{1_g}(x_1(t)) \text{ AND } x_2(t) \text{ is } A_{2_g}(x_2(t)) \\ &\text{AND } \ldots \text{ AND } x_{n_{\text{in}}}(t) \text{ is } A_{n_{\text{in}_g}}(x_{n_{\text{in}}}(t)) \\ &\text{THEN } y_j(t) \text{ is } w_{j_g}, \qquad g=1,2,\ldots,m_f; \quad t=1,2,\ldots,n_d \end{aligned} \tag{2}$$

where m_f also denotes the number of membership functions; n_d denotes the number of input-output data pairs; w_{j_g}, $j =$

TABLE I
RESULTS OF THE PROPOSED NFNs FOR INTERPRETING GRAFFITI AFTER TRAINING FOR 30 TIMES

Neural fuzzy network	Best fitness value	Number of rules for the best network	Training error (MSE) for the best network	Number of recognition error of the best network for the training pattern	Recognition error rate (%) of the best network for the testing pattern	Testing error (MSE) for the best network	Average fitness value
0(a)	0.9986	80	0.0027	0	6.6667%	0.9972	0.9968
0(b)	0.9997	113	0.0003	0	0%	0.9995	0.9995
1	0.9978	89	0.0022	1	0%	0.9978	0.9963
2	0.9994	113	0.0006	0	0%	0.9990	0.9991
3	0.9995	109	0.0005	0	3.3333%	0.9969	0.9992
4	0.9988	92	0.0012	0	3.3333%	0.9956	0.9979
5(a)	0.9992	100	0.0008	0	3.3333%	0.9974	0.9986
5(b)	0.9985	103	0.0015	0	0%	0.9966	0.9975
6	0.9992	93	0.0008	0	6.6667%	0.9948	0.9979
7	0.9991	113	0.0009	1	3.3333%	0.9989	0.9988
8(a)	0.9995	108	0.0005	0	0%	0.9962	0.9992
8(b)	0.9997	98	0.0003	1	6.6667%	0.9834	0.9994
9	0.9990	107	0.0010	0	0%	0.9987	0.9984
Back Space	0.9996	97	0.0004	0	0%	0.9976	0.9991
Return	0.9996	113	0.0004	0	0%	0.9988	0.9994
Space	0.9995	91	0.0005	0	0%	0.9992	0.9990

$1, 2, \ldots, n_{\text{out}}$, is the output singleton of the rule g. In this NFN, the membership function is a bell-shaped function given by

$$A_{i_g}(x_i(t)) = e^{-(x_i(t)-\bar{x}_{i_g})^2/2\sigma_{i_g}^2}, \quad i = 1, 2, \ldots, n_{\text{in}}; \quad g = 1, 2, \ldots, m_f \tag{3}$$

where $\bar{x}_{i_g}$ and σ_{i_g} are the mean value and the standard deviation of the membership function, respectively. The grade of membership of each rule is defined as

$$\mu_g(t) = A_{1_g}(x_1(t)) \times A_{2_g}(x_2(t)) \times \cdots \times A_{n_{\text{in}_g}}(x_{n_{\text{in}}}(t)), \quad g = 1, 2, \ldots, m_f. \tag{4}$$

The jth output of the NFN, $y_j(t)$, is defined as

$$y_j(t) = \frac{\sum_{g=1}^{m_f} \mu_g(t) w_{j_g} \delta(\varsigma_{j_g})}{\sum_{g=1}^{m_f} \mu_g(t)}, \qquad j = 1, 2, \ldots, n_{\text{out}} \tag{5}$$

where ς_{j_g} denotes the rule switch parameter of the gth rule.

III. TUNING OF MEMBERSHIP FUNCTIONS AND RULES OF THE NFN

In this section, the proposed NFN is employed to interpret graffiti digits and characters for eBooks. Fig. 3 shows the block diagram of the interpreter with m graffiti inputs. It consists of m NFN's and a graffiti determiner. The input–output relationships of the NFNs are trained using the GA with arithmetic crossover and nonuniform mutation [3]. The input–output relationship of one of the m neural networks in Fig. 3 is described by

$$\mathbf{y}^d(t) = \mathbf{z}(t) \equiv \frac{\mathbf{x}(t)}{\|\mathbf{x}(t)\|}, \qquad t = 1, 2, \ldots, n_d \tag{6}$$

where $\mathbf{y}^d(t) = [\, y_1^d(t) \quad y_2^d(t) \quad \cdots \quad y_{n_{\text{in}}}^d(t)\,]$ and $\mathbf{z}(t) = [\, z_1(t) \quad z_2(t) \quad \cdots \quad z_{n_{\text{in}}}(t)\,]$ are the desired outputs and the inputs of the NFN, respectively. $\mathbf{x}(t) = [\, x_1(t) \quad x_2(t) \quad \cdots \quad x_{n_{\text{in}}}(t)\,]$ is the vector of uniformly sampled points of the graffiti. $\|\cdot\|$ denotes the l_2 norm. The fitness function is defined as

$$\text{fitness} = \frac{1}{1 + \text{err}} \tag{7}$$

and err which is the mean square error (MSE) is defined as

$$\text{err} = \sum_{k=1}^{n_{\text{in}}} \frac{\sum_{t=1}^{n_d} \left(\frac{y_k(t)}{\|\mathbf{y}(t)\|} - \frac{y_k^d(t)}{\|\mathbf{y}^d(t)\|} \right)^2}{n_{\text{in}} n_d}. \tag{8}$$

The objective is to maximize the fitness value of (7) by setting the chromosome to be $[\, \bar{x}_{i_g} \quad \sigma_{i_g} \quad \varsigma_g \quad w_{j_g}\,]$ for all i, j and g. It can be seen from (6)–(8) that a larger fitness value implies a smaller error value. From (7) and (8), the NFN is trained such that the outputs are similar to its inputs. As shown in Fig. 3, we have m sets of graffiti training samples for m NFNs correspondingly. Each set of graffiti training samples is used to train its corresponding NFN. During the operation, the sampled points of the input graffiti will be fed to all the m neural-fuzzy networks. The output of the m neural-fuzzy networks will be fed to the graffiti determiner to generate the final result that indicates the possible graffiti input. The graffiti determiner measures the similarity between the input graffiti and the outputs of the NFN's. The similarity of an input graffiti to the output of an NFN is defined as

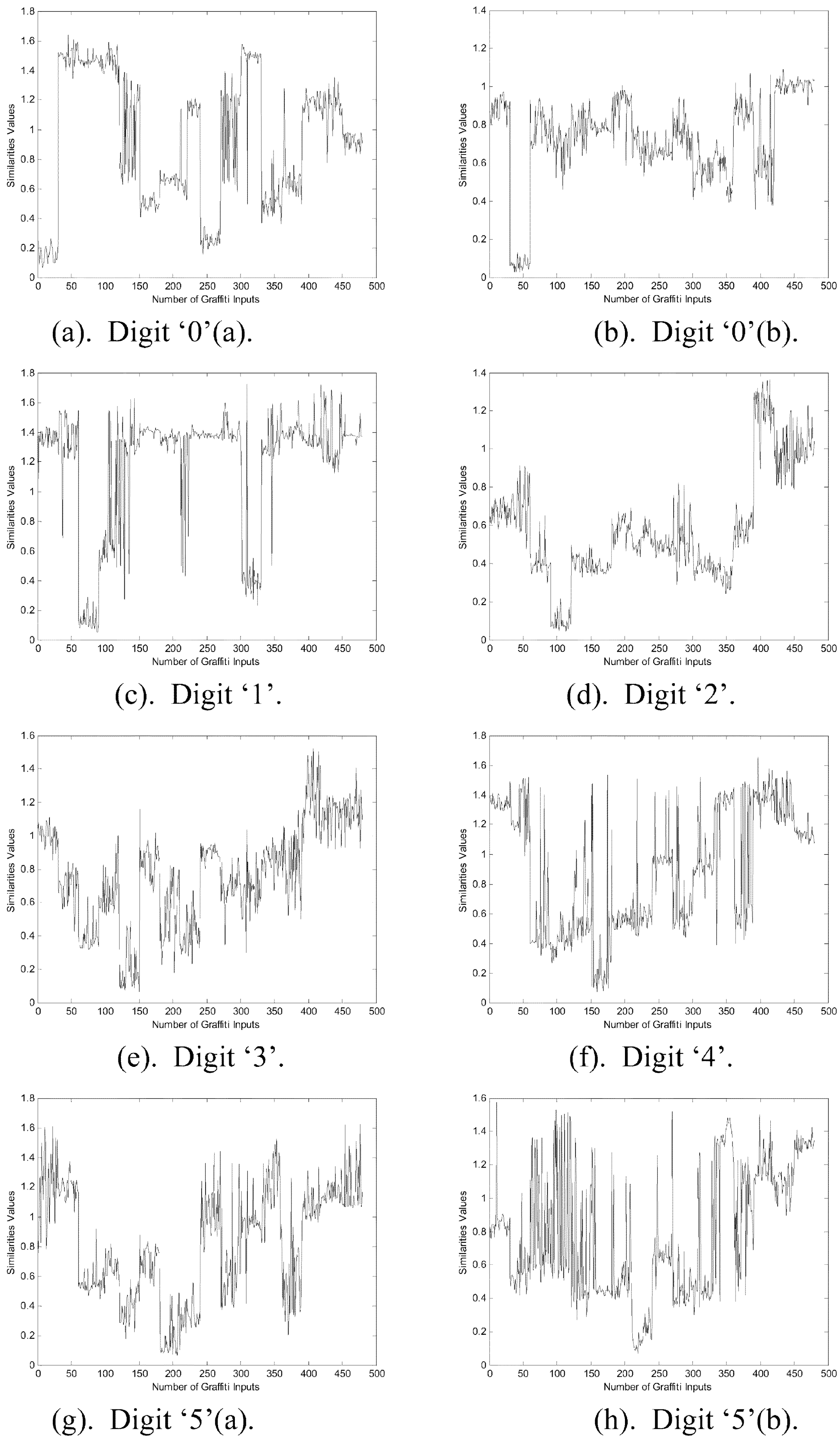
$$S_i = \|\bar{\mathbf{y}}_i - \bar{\mathbf{z}}\|, \qquad i = 1, 2, \ldots, m \tag{9}$$

where

$$\bar{\mathbf{y}}_i = \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|} = [\, \bar{y}_1(t) \quad \bar{y}_2(t) \quad \cdots \quad \bar{y}_{n_{\text{in}}}(t)\,], \quad i = 1, 2, \ldots, m \tag{10}$$

$$\bar{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|} = [\, \bar{z}_1(t) \quad \bar{z}_2(t) \quad \cdots \quad \bar{z}_{n_{\text{in}}}(t)\,] \tag{11}$$

$\bar{\mathbf{y}}_i$ and $\bar{\mathbf{z}}$ denote the normalized outputs and the normalized input of the NFNs, respectively. A smaller value of S_i implies a closer match of the input graffiti to the graffiti represented by the ith



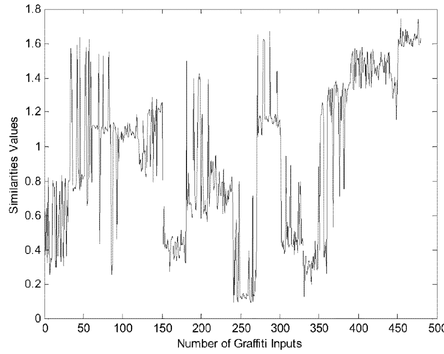
(a). Digit '0'(a). (b). Digit '0'(b).

(c). Digit '1'. (d). Digit '2'.

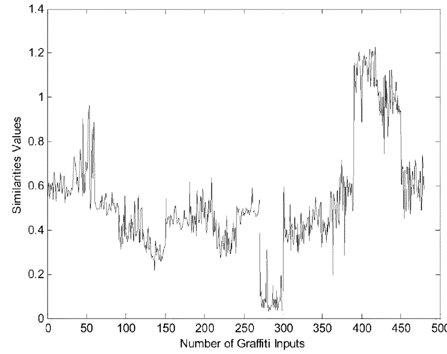
(e). Digit '3'. (f). Digit '4'.

(g). Digit '5'(a). (h). Digit '5'(b).

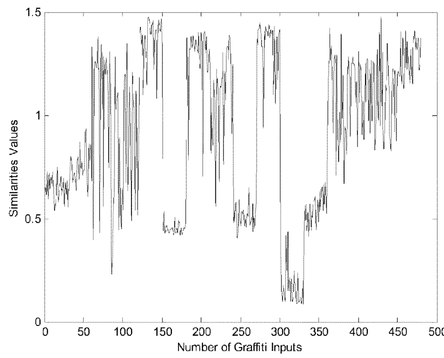
Fig. 5. Similarity values of the 16 proposed neural-fuzzy networks for the 480 (30 for each type) testing graffiti.



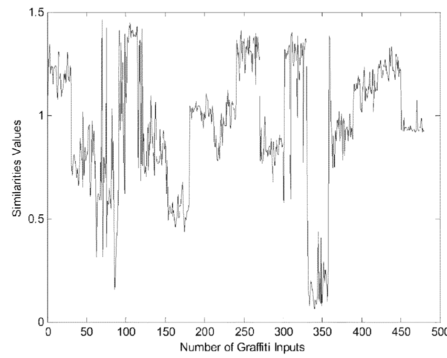
(i). Digit ‘6’.



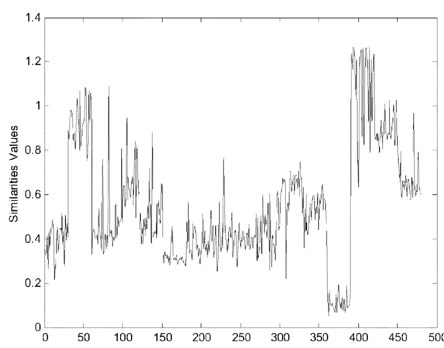
(j). Digit ‘7’.



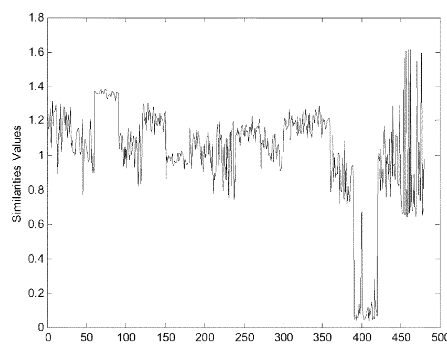
(k). Digit ‘8’(a).



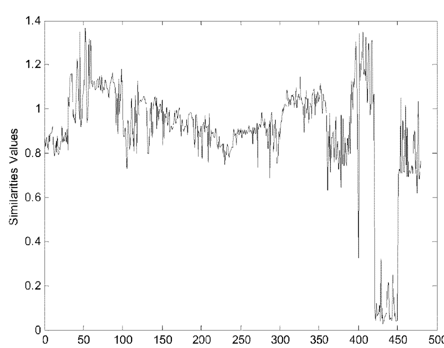
(l). Digit ‘8’(b).



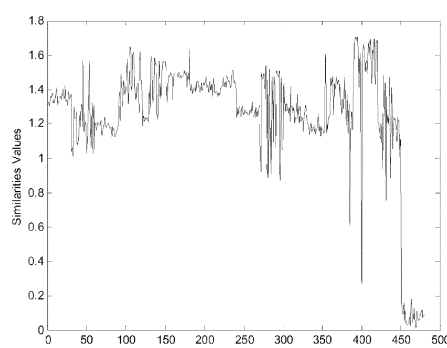
(m). Digit ‘9’.



(n). Character ‘backspace’.



(o). Character ‘carriage return’.



(p). Character ‘space’.

Fig. 5. (*Continued*.) Similarity values of the 16 proposed neural-fuzzy networks for the 480 (30 for each type) testing graffiti.

NFN. The smallest similarity value among the m NFNs is defined as

$$S_j = \min_i S_i. \tag{12}$$

The index j of (12) is the output of the graffiti determiner, which indicates the jth graffiti is the most likely input graffiti.

IV. Application Example and Results

The interpretation of graffiti digits and characters for eBooks by the proposed NFN will be presented in this section. A point on the eBook screen is characterized by a number based on the x-y coordinates on a writing area. The size of the writing area is $x_{\max}$ by $y_{\max}$. The bottom left corner is set as (0, 0). Ten uniformly sampled points of the graffiti will be taken as the inputs of the interpreter. The points are taken in the following way. First, the input graffiti is divided into nine uniformly distanced segments characterized by ten points, including the start and the end points. Each point is labeled as (x_i, y_i), $i = 1, 2, \ldots, 10$. The first five points, (x_i, y_i), $i =$ 1, 3, 5, 7 and 9, taken alternatively are converted to five numbers ρ_i, respectively, by using the formula $\rho_i = x_i x_{\max} + y_i$. The other five points, (x_i, y_i), $i = 2, 4, 6, 8$ and 10, are converted to five numbers, respectively, by using the formula $\rho_i = y_i y_{\max} + x_i$. These ten numbers, z_i, $i = 1, 2, \ldots, 10$, will be used as the inputs of the proposed NFN (with ten inputs, ten outputs, and 15 membership functions) with rule switches for each graffiti pattern. In our eBook application, the digits 0–9 and three (control) characters (*Backspace*, *Carriage Return*, and *Space*) are interpreted. These graffiti are shown in Fig. 4 and the eBook graffiti interpreter thus has 16 NFNs. To train these NFNs, 100 sets of sampled points for each graffiti pattern are used. The input–output relationship of the NFNs is governed by (5). The fitness function of each NFN is given by (7), with

$$\text{err} = \sum_{k=1}^{10} \frac{\sum\limits_{t=1}^{100} \left(\frac{y_k(t)}{\|\mathbf{y}(t)\|} - \frac{y_k^d(t)}{\|\mathbf{y}^d(t)\|} \right)^2}{1000}. \tag{13}$$

The GA with arithmetic crossover and nonuniform mutation [3] is employed to tune the membership functions and numbers of rules of the NFN of (5). The objective is to maximize the fitness function of (7). The best fitness value is 1 and the worst one is 0. The population size is 10. The lower and upper bounds of the link weights are defined as $0 \geq \bar{x}_{i_g}, \sigma_{i_g}, \varsigma_g, w_{j_g} \geq 1$, $i = 1, 2, \ldots, 10$; $j = 1, 2, \ldots, 10$; $g = 1, 2, \ldots, 15$. The chromosomes of the GA process used are $[\,\bar{x}_{i_g} \quad \sigma_{i_g} \quad \varsigma_g \quad w_{j_g}\,]$ $i = 1, 2, \ldots, 10$; $j = 1, 2, \ldots, 10$; $g = 1, 2, \ldots, 15$. The initial values of the link weights are randomly generated. The number of iterations to train the neural networks is 2000, and the training is done 30 times. After training, 30 graffiti samples of each kind of graffiti are used to test the performance of the trained NFNs. The results are tabulated in Table I. From this Table, it can be observed that the numbers of connected links in the NFNs are reduced after learning (each NFN has 150 rules initially). Fig. 5

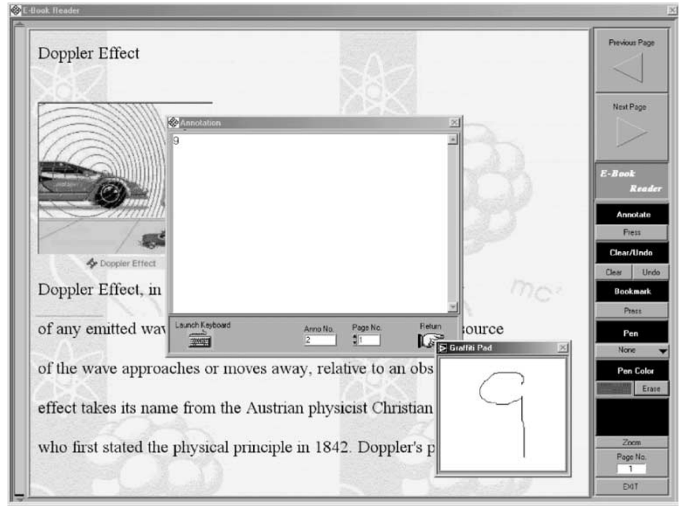


Fig. 6. Input of digit "9" to the annotation window using the graffiti pad of the eBook.

shows the similarity values of each NFN for the 480 (30 for each type of graffiti) testing graffiti. It can be seen that the NFN trained by a particular graffiti will provide a smaller similarity values for that kind of graffiti. For example, in Fig. 5(a), the similarity values of the first 30 testing graffiti (digit "0") are smaller, as that NFN is trained by 100 digit "0" graffiti pattern. We have successfully implemented the graffiti digit and character interpreter in the eBook shown in Fig. 1. The image of inputting a digit "9" to the eBook reader using the proposed graffiti interpreter was captured and is shown in Fig. 6. Referring to this figure, the graffiti interpreter of the eBook reader recognizes that the input digit is "9" and takes the corresponding action of the *annotation* feature. For comparison, a traditional neural-fuzzy network and two three-layer feedforward neural networks [11] are also employed as the networks in the graffiti interpreter of Fig. 3. The number of membership functions used for the traditional neural-fuzzy network is 15. The numbers of hidden nodes for the two neural networks are 15 and 21, respectively. The GA with arithmetic crossover and nonuniform mutation [3] using the same control parameters of the proposed approach is employed for training. The best results over 30 simulations of training are tabulated in Tables II–IV. For the neural network with hidden nodes of 21, the number of parameters to be tuned is nearly the same as that of the proposed NFN. It can be seen from Tables I–III that the proposed approach provides a better performance than that of the traditional approaches (using a traditional neural-fuzzy network with 15 membership functions and a traditional neural network with 15 hidden nodes) in terms of fitness value and/or number of rules. It can also be seen from Tables I and IV that the proposed approach provides a similar performance to that of the traditional neural network with 21 hidden nodes in terms of fitness value. Still, the proposed NFN, after training, will have fewer network parameters than the traditional neural network with 21 hidden nodes, resulting in a lower cost of implementation.

TABLE II
RESULTS OF THE TRADITIONAL NFNs FOR INTERPRETING GRAFFITI AFTER TRAINING FOR 30 TIMES

Neural fuzzy network	Best fitness value	Number of rules	Training error (MSE) for the best network	Number of recognition error of the best network for the training pattern	Recognition error rate (%) of the best network for the testing pattern	Testing error (MSE) for the best network	Average fitness value
0(a)	0.9995	150	0.0005	0	10%	0.9974	0.9993
0(b)	0.9990	150	0.0010	0	0%	0.9976	0.9983
1	0.9994	150	0.0006	1	0%	0.9990	0.9990
2	0.9994	150	0.0006	0	0%	0.9987	0.9990
3	0.9995	150	0.0005	0	26.6667%	0.9897	0.9992
4	0.9997	150	0.0003	0	3.3333%	0.9977	0.9996
5(a)	0.9986	150	0.0014	1	6.6667	0.9962	0.9977
5(b)	0.9981	150	0.0019	0	33.3333	0.9740	0.9966
6	0.9993	150	0.0007	0	10%	0.9957	0.9988
7	0.9989	150	0.0011	1	6.6667%	0.9890	0.9960
8(a)	0.9996	150	0.0004	0	3.3333%	0.9968	0.9993
8(b)	0.9995	150	0.0005	0	13.3333%	0.9928	0.9993
9	0.9991	150	0.0009	1	6.6667%	0.9987	0.9987
Back Space	0.9996	150	0.0004	0	3.3333%	0.9945	0.9994
Return	0.9989	150	0.0011	0	3.3333%	0.9945	0.9981
Space	0.9996	150	0.0004	0	3.3333%	0.9958	0.9987

TABLE III
RESULTS OF THE TRADITIONAL NEURAL NETWORKS (15 HIDDEN NODES) FOR INTERPRETING GRAFFITI AFTER TRAINING FOR 30 TIMES

Neural fuzzy network	Best fitness value	Training error (MSE) for the best network	Number of recognition error of the best network for the training pattern	Recognition error rate (%) of the best network for the testing pattern	Testing error (MSE) for the best network	Average fitness value
0(a)	0.9993	0.0007	1	10%	0.9981	0.9991
0(b)	0.9993	0.0007	0	0%	0.9992	0.9991
1	0.9989	0.0011	2	0%	0.9989	0.9983
2	0.9976	0.0024	0	3.3333%	0.9962	0.9976
3	0.9981	0.0019	0	0%	0.9962	0.9977
4	0.9989	0.0011	2	0%	0.9972	0.9988
5(a)	0.9986	0.0014	0	3.3333%	0.9970	0.9984
5(b)	0.9986	0.0014	0	0%	0.9973	0.9984
6	0.9988	0.0012	4	23.3333%	0.9948	0.9987
7	0.9974	0.0026	4	3.3333%	0.9976	0.9967
8(a)	0.9988	0.0012	1	0%	0.9965	0.9986
8(b)	0.9989	0.0011	0	0%	0.9963	0.9988
9	0.9965	0.0035	13	3.3333%	0.9966	0.9957
Back Space	0.9986	0.0014	0	3.3333%	0.9965	0.9974
Return	0.9980	0.0020	0	0%	0.9970	0.9971
Space	0.9986	0.0015	0	0%	0.9985	0.9978

TABLE IV
RESULTS OF THE TRADITIONAL NEURAL NETWORKS (21 HIDDEN NODES) FOR INTERPRETING GRAFFITI AFTER TRAINING FOR 30 TIMES

Neural fuzzy network	Best fitness value	Training error (MSE) for the best network	Number of recognition error of the best network for the training pattern	Recognition error rate (%) of the best network for the testing pattern	Testing error (MSE) for the best network	Average fitness value
0(a)	0.9993	0.0007	1	6.6667%	0.9983	0.9992
0(b)	0.9993	0.0007	0	3.3333%	0.9981	0.9992
1	0.9989	0.0011	1	0%	0.9989	0.9986
2	0.9986	0.0014	0	0%	0.9983	0.9981
3	0.9989	0.0011	0	3.3333%	0.9971	0.9986
4	0.9992	0.0008	2	6.6667%	0.9963	0.9990
5(a)	0.9991	0.0009	0	0%	0.9979	0.9989
5(b)	0.9992	0.0008	0	0%	0.9980	0.9990
6	0.9989	0.0011	2	20%	0.9953	0.9987
7	0.9969	0.0031	5	3.3333%	0.9972	0.9966
8(a)	0.9987	0.0013	1	0%	0.9963	0.9986
8(b)	0.9989	0.0011	0	0%	0.9964	0.9988
9	0.9968	0.0032	13	6.6667%	0.9968	0.9957
Back Space	0.9988	0.0012	0	3.3333%	0.9965	0.9977
Return	0.9984	0.0016	0	0%	0.9974	0.9976
Space	0.9986	0.0014	0	0%	0.9986	0.9978

V. CONCLUSION

By introducing a switch to each rule, an NFN can be tuned to obtain the optimal number of rules, and learn the input–output relationship of an application using GA. This implies a lower cost of implementation. The proposed NFN with rule switches trained by GA has been applied to interpret graffiti digits and characters for eBooks.

REFERENCES

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.
[2] D. T. Pham and D. Karaboga, *Intelligent Optimization Techniques, Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Berlin, Germany: Springer-Verlag, 2000.
[3] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*, 2nd ed. Berlin, Germany: Springer-Verlag, 1994.
[4] B. D. Liu, C. Y. Chen, and J. Y. Tsao, "Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 32–53, Feb. 2001.
[5] Y. S. Zhou and L. Y. Lai, "Optimal design for fuzzy controllers by genetic algorithms," *IEEE Trans. Ind. Applicat.*, vol. 36, pp. 93–97, Jan./Feb. 2000.
[6] C. F. Juang, J. Y. Lin, and C. T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, pp. 290–302, Apr. 2000.
[7] H. Juidette and H. Youlal, "Fuzzy dynamic path planning using genetic algorithms," *Electron. Lett.*, vol. 36, no. 4, pp. 374–376, Feb. 2000.
[8] R. Caponetto, L. Fortuna, G. Nunnari, L. Occhipinti, and M. G. Xibilia, "Soft computing for greenhouse climate control," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 753–760, Dec. 2000.
[9] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: Complexity and performance," *IEEE. Trans. Fuzzy Syst.*, vol. 8, pp. 509–522, Oct. 2000.
[10] K. Belarbi and F. Titel, "Genetic algorithm for the design of a class of fuzzy controllers: An alternative approach," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 398–405, Aug. 2000.
[11] M. Brown and C. Harris, *Neuralfuzzy Adaptive Modeling and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[12] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[13] F. H. F. Leung, H. K. Lam, S. H. Ling, and P. K. S. Tam, "Tuning of the structure and parameters of neural network using an improved genetic algorithm," *IEEE Trans. Neural Networks*, vol. 14, pp. 79–88, Jan. 2003.
[14] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *IEEE Trans. Syst.s, Man, Cybern. C*, vol. 31, pp. 216–233, May 2001.
[15] M. A. Mohamed and P. Gader, "Generalized hidden Markov models – Part II: Application to handwritten word recognition," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 82–95, Feb. 2000.
[16] M. Shridhar and F. Kimura, "High accuracy syntactic recognition algorithm for handwritten numerals," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 152–158, Jan. 1985.
[17] H. Y. Kim and J. H. Kim, "Handwritten Korean character recognition based on hierarchical random graph modeling," in *Proc. Int. Workshop Frontiers in Handwriting Recognition*, 1998, pp. 577–586.
[18] R. Buse, Z. Q. Liu, and J. Bezdek, "Word recognition using fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 65–76, Feb. 2002.

K. F. Leung received the B.Eng. (Hons.) degree in electrical engineering from The Hong Kong Polytechnic University, Hong Kong, in 1999.

He is currently a Research Student in the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His research interests include neural network, fuzzy logic, speech recognition, and electronic books.

Frank H. F. Leung (M'92–SM'03) was born in Hong Kong in 1964. He received the B.Eng. degree and the Ph.D. degree in electronic engineering from The Hong Kong Polytechnic University, Hong Kong, in 1988 and 1992, respectively.

In 1992, he joined The Hong Kong Polytechnic University, where he is currently an Associate Professor in the Department of Electronic and Information Engineering. He has published over 100 research papers on computational intelligence, control, and power electronics. At present, he is actively involved in research on the Intelligent Multimedia Home and eBook.

Dr. Leung is a reviewer for many international journals and has helped organize many international conferences. He is a Chartered Engineer and a Corporate Member of the Institution of Electrical Engineers, U.K.

H. K. Lam (M'95) received the B.Eng. (Hons.) and Ph.D. degrees from the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, in 1995 and 2000, respectively.

He is currently a Postdoctoral Fellow in the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His current research interests include intelligent control and systems, computational intelligence, and robust control.

S. H. Ling (S'03) received the B.Eng. (Hons.) and M.Phil. degrees in 1999 and 2002, respectively, from the Department of Electrical Engineering and the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, where he is currently working toward the Ph.D. degree in the Department of Electronic and Information Engineering.

He is currently a Research Student in the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His current research interests include evolutionary computation, fuzzy logic, neural networks, and intelligent homes.