On Interpretation of Graffiti Digits and Commands for eBooks: Neural Fuzzy Network and Genetic Algorithm Approach[1]

H.K. Lam, K.F. Leung, S.H. Ling, F.H.F. Leung 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 – This paper presents a proposed neural fuzzy network tuned by genetic algorithm (GA). By introducing a switch to each rule, the optimal number of rules can be learned. The membership functions of the neural fuzzy network are also tuned by GA. After training, the proposed neural fuzzy network is employed to interpret graffiti number inputs and commands for Electronic Books (eBooks).

I. INTRODUCTION

GA is a powerful random search technique to handle optimization problems [1-2, 5]. This is especially useful for complex optimization problems with a large number of parameters that make global analytical solutions difficult to obtain. It has been widely applied in different areas such as fuzzy control [7-9, 13], path planning [10], greenhouse climate control [11], modeling and classification [12, 18] etc.

Neural fuzzy networks (NFNs) have been proved to be a universal approximator [14], which can approximate nonlinear functions to an arbitrary accuracy. Expert knowledge and experience can be incorporated into an NFN, which were successfully applied in areas such as prediction [18], system modeling and control [14]. 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 a learning algorithm. 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 changing our life. One of the changes is on our reading habit. 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 and 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. In this paper, an NFN with rule switches is proposed to perform the interpretation of graffiti. GA with arithmetic crossover and non-uniform mutation [5] will be employed to train the proposed NFN. The result is a graffiti digit and command interpreter. It can interpret the digits 0 to 9 and three commands, namely "Input a Backspace", "Input a Carriage Return" and "Input a Space". Through this NFN, the optimal fuzzy rules and membership functions can be tuned. It is then applied to an eBook reader experimentally.

This paper is organized as follows. The proposed NFN with rule switches is presented in section II. A graffiti interpreter that 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 the interpretation of graffiti digits and commands for eBooks will be given in section IV. A conclusion will be drawn in Section V.

II. NEURAL FUZZY NETWORK WITH RULE SWITCHES

We use a fuzzy associative memory (FAM) [17] 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 is proposed which can have an optimal number of rules and membership functions. 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 functions is defined as,

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

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 $i = 1, 2, \ldots, n_{in}$; n_{in} is the number of input variables; $j = 1, 2, \ldots, n_{out}$; n_{out} is the number of output variables. The

[1] The work described in this paper was substantially supported by a Research Grant of the Centre for Multimedia Signal Processing, The Hong Kong Polytechnic University (project number A432).

behaviour of y_j of the NFN is governed by m_f fuzzy rules of the following format;

R_g: IF $x_1(t)$ is $A_{1g}(x_1(t))$ AND $x_2(t)$ is $A_{2g}(x_2(t))$ AND … AND $x_{n_{in}}(t)$ is $A_{n_{in}g}(x_{n_{in}}(t))$

THEN $y_j(t)$ is w_{jg}, $g = 1, 2, \ldots, m_f$; $t = 1, 2, \ldots, n_d$ (2)

where n_d denotes the number of input-output data pairs; w_{jg}, $j = 1, 2, \ldots, n_{out}$, is the output singleton of the rule g; $g \in [1, 2, \ldots, m_f]$. In this NFN, the membership function is a bell-shaped function given by,

$$A_{i_g}(x_i(t)) = e^{\frac{-(x_i(t)-\bar{x}_{i_g})^2}{2\sigma_{i_g}^2}} \tag{3}$$

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

$$\mu_g(t) = A_{1_g}(x_1(t)) \times A_{2_g}(x_2(t)) \times \ldots \times A_{n_{in_g}}(x_{n_{in}}(t)) \tag{4}$$

The j-th output of the NFN, $y_j(t)$, is defined as,

$$y_j(t) = \frac{\sum_{g=1}^{m_f} \mu_g(t) w_{j_g} \delta(\varsigma_{j_g})}{\sum_{g=1}^{m_f} \mu_g(t)} \tag{5}$$

where ς_{j_g} denotes the rule switch parameter of the g-th rule.

III. Tuning of Membership Functions and Rules of the Neural Fuzzy Network

In this section, the proposed NFN is employed to interpret graffiti digits and commands for eBooks. Fig. 3 shows the block diagram of the graffiti digit and command interpreter with m graffiti inputs. It consists of m NFNs and a graffiti determiner. The input-output relationships of the NFNs are trained using the GA with arithmetic crossover and non-uniform mutation [5]. The input-output relationship of one of the m neural networks in Fig. 3 is described by,

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

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

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

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

The objective is to maximize the fitness value of (7) by setting the chromosome to be $[\bar{x}_{i_g} \;\; \sigma_{i_g} \;\; \varsigma_g \;\; w_{j_g}]$ for all i, j and g. It can be seen from (6), (7) and (8) that a larger fitness value implies a smaller error value. From (7) and (8), the NFN is trained such that the outputs is 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 NFNs, which is defined as,

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

where,

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

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

$\bar{\mathbf{y}}_i$ 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 i-th 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 j-th graffiti, is the most likely input graffiti.

IV. Application Example and Results

The interpretation of graffiti digits and commands for eBooks by the proposed NFN will be presented in this section. A point on the eBook screen is characterized by a number. Ten sampled points of the graffiti will be taken as the inputs of the graffiti digit and command interpreter. The interpretation process is achieved by an NFN (with 10 inputs 10 outputs, 15 membership functions) with rule switches for each graffiti. The ten inputs nodes, z_i, $i = 1, 2, \ldots, 10$, are taken uniformly from the input graffiti. In our eBook application, digits 0 to 9 and three graffiti commands (inputs of backspace, carriage return and space) can be interpreted. These graffiti are shown in Fig. 4. The eBook graffiti interpreter thus has 13 NFNs. To train these NFNs, 100 sampled points for each set of graffiti are used.

In these NFNs, the number of membership functions is 15. Referring to (5), the proposed neural fuzzy network used for the interpretation process is governed by,

$$y_j(t) = \frac{\sum_{g=1}^{15} \mu_g(t) w_{j_g} \delta\left(\varsigma_{j_g}\right)}{\sum_{g=1}^{15} \mu_g(t)}, j = 1, 2, \ldots, 10 \quad (13)$$

The fitness function for each NFN is defined as follows,

$$fitness = \frac{1}{1 + err} \quad (14)$$

$$err = \sum_{k=1}^{10} \frac{\sum_{t=1}^{100} \left| \frac{y_k(t)}{\|\mathbf{y}(t)\|} - \frac{y_k^d(t)}{\|\mathbf{y}^d(t)\|} \right|}{100} \quad (15)$$

The GA with arithmetic crossover and non-uniform mutation [5] is employed to tune the membership functions and numbers of rules of the NFN of (13). The objective is to maximize the fitness function of (14). The best fitness value is 1 and the worst one is 0. The population size is 10. The lower and the 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 $\left[\bar{x}_{i_g} \quad \sigma_{i_g} \quad \varsigma_g \quad w_{j_g}\right]$ for all $i = 1, 2, \ldots, 10$; $j = 1, 2, \ldots, 10$; $g = 1, 2, \ldots, 15$. The initial values of the link weights are randomly generated. The number of iterations to train the neural networks is 2000. The training time for one NFN is around 4.94 hours when a 1.4GHz P4 system is used. 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 neural fuzzy network has 150 rules initially. Fig. 5 shows the similarity values of each NFN for the 390 (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" training graffiti. We have successfully implemented the graffiti digit and command interpreter in the eBook shown in Fig. 1. The image of inputting a digit "9" to the eBook reader using the proposed graffiti digit and command interpreter was captured and is shown in Fig. 6.

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 commands for eBooks.

References

[1] J.H. Holland, *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press, 1975.

[2] D.T. Pham and D. Karaboga, *Intelligent optimization techniques, genetic algorithms, tabu search, simulated annealing and neural networks.* Springer, 2000.

[3] Y. Hanaki, T. Hashiyama, and S. Okuma, "Accelerated evolutionary computation using fitness estimation," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, vol. 1, 1999, pp. 643-648.

[4] K.A. De Jong, *Ph.D. Thesis: An analysis of the behavior of a class of genetic adaptive systems.* Ann Arbor, MI: University of Michigan, 1975.

[5] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*, (2n ed.). Springer-Verlag, 1994.

[6] G.X. Yao and Y. Liu "Evolutionary Programming made Faster," *IEEE Trans. Evolutionary Computation*, vol. 3, no. 2, pp.82-102, July 1999.

[7] 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. Systems, Man and Cybernetics, Part B*, vol. 31 no. 1, pp. 32-53, Feb. 2001.

[8] Y.S Zhou and L.Y, Lai "Optimal design for fuzzy controllers by genetic algorithms," *IEEE Trans. Industry Applications*, vol. 36, no. 1, pp. 93-97, Jan.-Feb. 2000.

[9] C.F. Juang, J.Y. Lin and C.T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 30, no. 2, pp. 290–302, April 2000.

[10] H. Juidette and H. Youlal, "Fuzzy dynamic path planning using genetic algorithms," *Electronics Letters*, vol. 36, no. 4, pp. 374-376, Feb. 2000.

[11] R. Caponetto, L. Fortuna, G. Nunnari, L. Occhipinti, and M. G. Xibilia, "Soft computing for greenhouse climate control," *IEEE Trans. Fuzzy Systems*, vol. 8, no. 6, pp. 753-760, Dec. 2000.

[12] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: complexity and performance," *IEEE. Trans. Fuzzy Systems*, vol. 8, no. 5, pp. 509–522, Oct. 2000.

[13] K. Belarbi and F. Titel, "Genetic algorithm for the design of a class of fuzzy controllers: an alternative approach," *IEEE Trans. Fuzzy Systems*, vol. 8, no. 4, pp. 398-405, Aug. 2000.

[14] M. Brown and C. Harris, *Neuralfuzzy adaptive modeling and control.* Prentice Hall, 1994.

[15] S. Amin and J.L. Fernandez-Villacanas, "Dynamic Local Search," in *Proc. 2nd Int. Conf. Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp129-132, 1997.

[16] L.F.F. Wessels and E. Barnard, "Avoiding false local minima by proper initialization of connections," *IEEE Trans., Neural Network*, vol. 3, no.6, pp. 899-905, 1992.

[17] B. Kosko, *Neural Networks and Fuzzy System: A Dynamical Systems Approach to Machine Intelligence.* Prentice Hall, 1991.

[18] 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," *Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics Society, IECON'2001* (to be published).

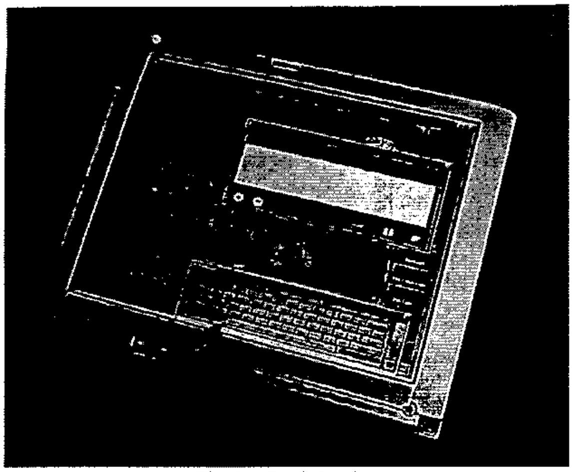

Fig. 1. eBook Reader.

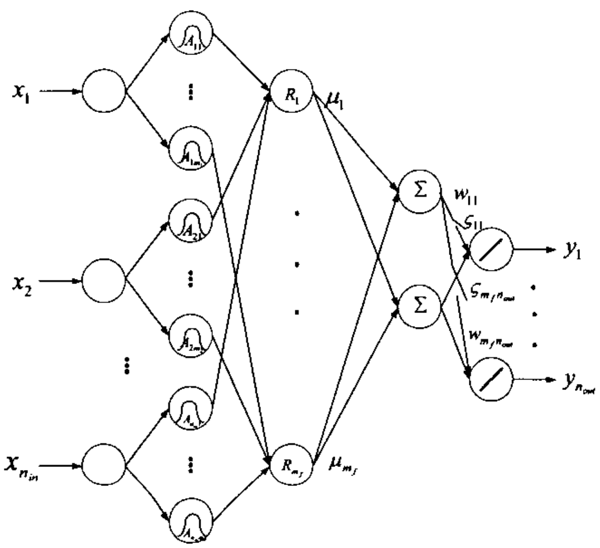

Fig. 2. Proposed neural fuzzy network.

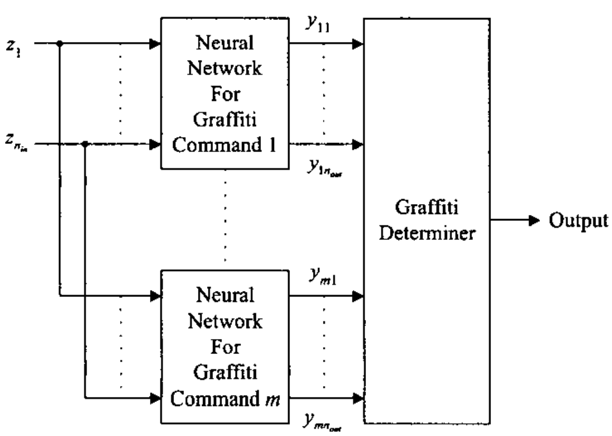

Fig. 3. Block diagram of the graffiti digit and command interpreter.

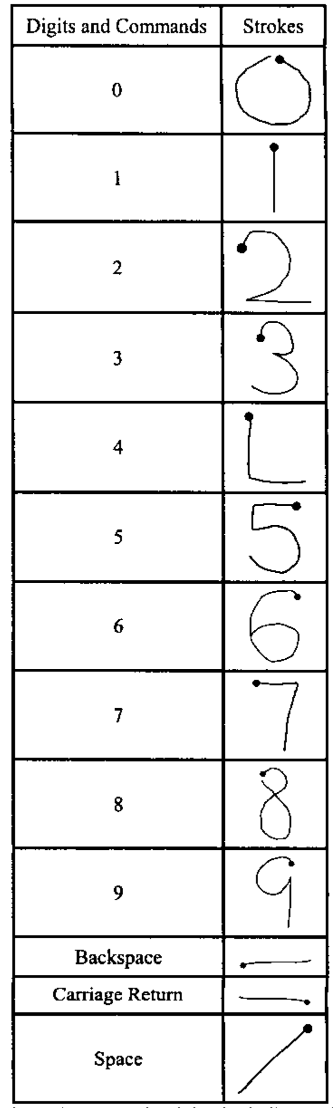

Fig. 4. Graffiti digits and commands. (The dot indicates the starting point of the graffiti).

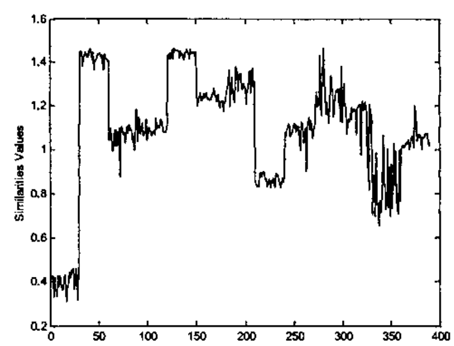

Fig. 5(a). Similarity values of the neural network for digit "0".

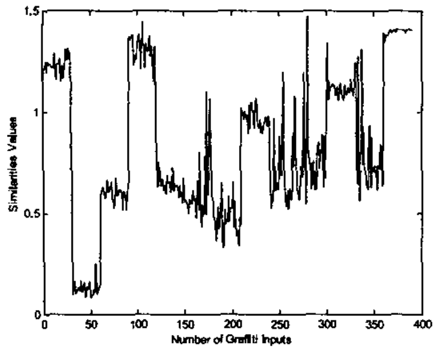

Fig. 5(b). Similarity values of the neural network for digit "1".

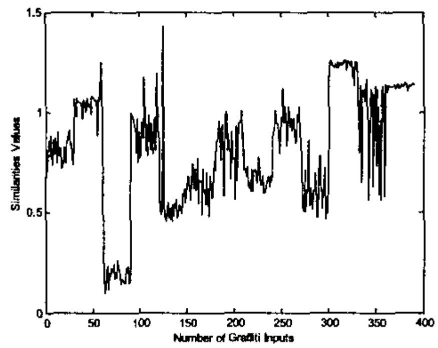

Fig. 5(c). Similarity values of the neural network for digit "2".

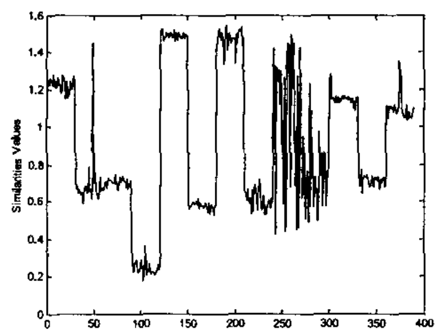

Fig. 5(d). Similarity values of the neural network for digit "3".

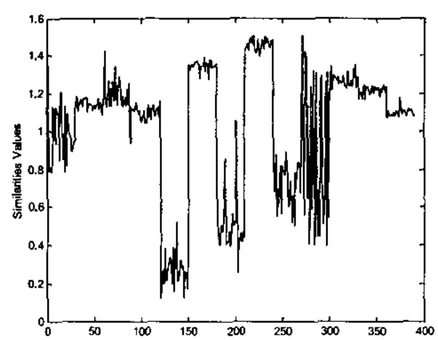

Fig. 5(e). Similarity values of the neural network for digit "4".

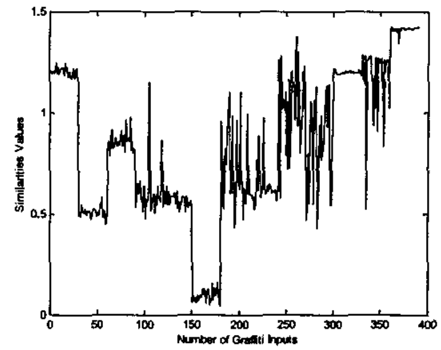

Fig. 5(f). Similarity values of the neural network for digit "5".

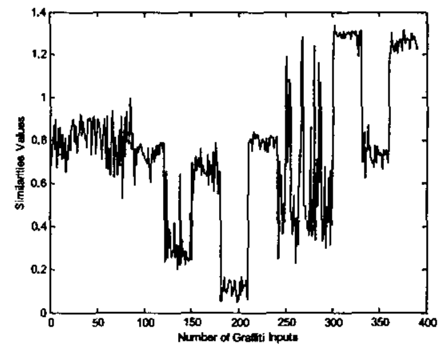

Fig. 5(g). Similarity values of the neural network for digit "6".

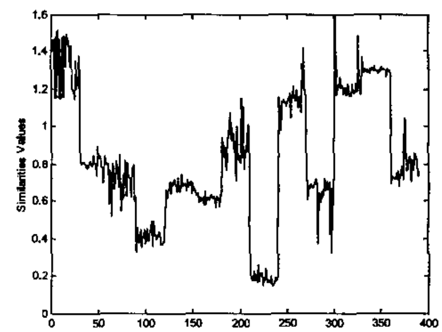

Fig. 5(h). Similarity values of the neural network for digit "7".

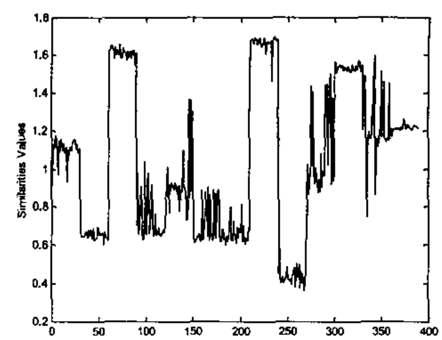

Fig. 5(i). Similarity values of the neural network for digit "8".

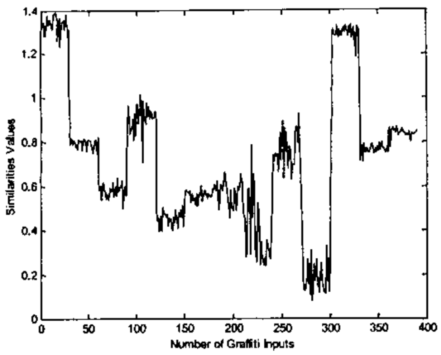

Fig. 5(j). Similarity values of the neural network for digit "9".

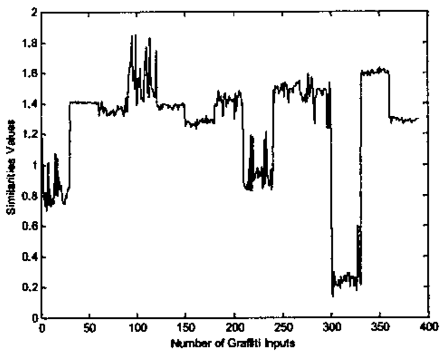

Fig. 5(k). Similarity values of the neural network for command "input a backspace".

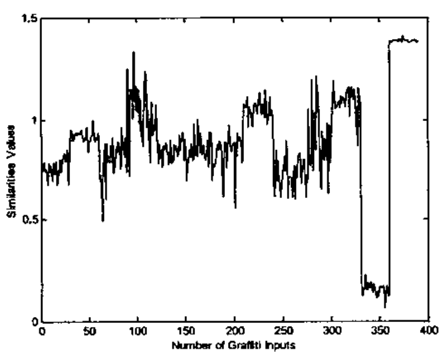

Fig. 5(l). Similarity values of the neural network for command "input a carriage return".

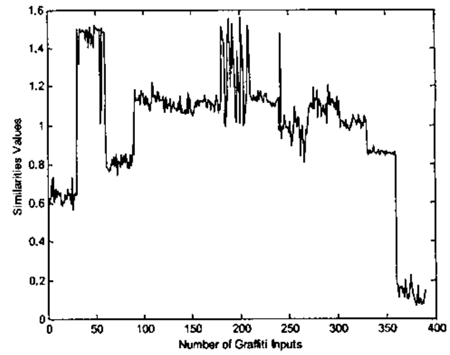

Fig. 5(m). Similarity values of the neural network for command "input a space".

Fig. 5. Similarity values of the 13 neural networks for the 390 (30 for each type) testing graffiti.

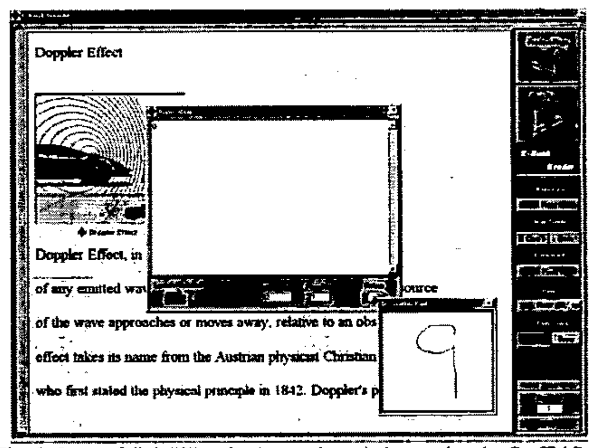

Fig. 6. Input of digit "9" to the Annotation windows using the Graffiti Pad of the eBook.

Fuzzy Network	Fitness Value	Number of Rules	Training Error	Testing Error
Left 0	0.9650	90	0.0350	0.0471
Right 0	0.9649	96	0.0351	0.0863
1	0.9738	72	0.0262	0.0251
2	0.9555	87	0.0445	0.0359
3	0.9451	73	0.0549	0.0534
4	0.9415	76	0.0585	0.0623
5a	0.9737	96	0.0263	0.0223
5b	0.9199	66	0.0801	0.0685
6	0.9755	114	0.0245	0.0457
7	0.9617	78	0.0383	0.0483
8a	0.9377	71	0.0623	0.0802
8b	0.9196	68	0.0804	0.0794
9	0.9572	103	0.0428	0.0503
Back Space	0.9735	69	0.0265	0.0477
Return	0.9846	102	0.0154	0.0374
Space	0.9766	67	0.0234	0.0243

Table I. Results of the proposed neural networks for interpreting graffiti digits and commands after training for 2000 iterations.