Graffiti Commands Interpretation for eBooks Using a Self-Structured Neural Network and Genetic Algorithm[1]

K. F. Leung, H. K. Lam, 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 the interpretation of graffiti commands for electronic books (eBooks). A neural network will be employed to perform the graffiti interpretation. By introducing a switch to each link of the neural network, the structure of the neural network can be obtained and tuned automatically by the genetic algorithm (GA) with arithmetic crossover and non-uniform mutation. Simulation results on interpreting graffiti commands for eBooks using the proposed neural network will be shown.

I. INTRODUCTION

Notebook Computers and Personal Digital Assistants (PDAs) have been modifying our life, including 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 effects, 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. As many functions are implemented in a single eBook Reader, it is not convenient to access these functions through menus without hot keys. However, even when hot keys (as icons on the screen) are employed, time is needed to access them, especially when the number of hot keys is large. Thanks to the touch screen, a one-step commanding process using graffiti is proposed for eBooks. For instance, when a user draws an arbitrary straight line on the screen, the eBook Reader is able to respond to the command represented by the straight line. However, computers are only good at numerical, manipulation, while the interpretation of graffiti commands is a symbolic manipulation process. Thus, a way to let the eBook reader understand symbolic information should be found. In this paper, a neural network with link switches is proposed to perform the interpretation of graffiti commands. GA with arithmetic crossover and non-uniform mutation [5] will be used to train the proposed neural network.

GA is a powerful random search technique to handle optimization problems [1-6, 17]. 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 [9-11, 15], path planning [12], greenhouse climate control [13], modeling and classification [7-8, 14] etc..

Neural networks have been proved to be a universal approximator [16]. A 3-layer feed-forward neural network can approximate any nonlinear function to an arbitrary accuracy. Applications like prediction [7], system modeling and control [16] have been reported. In view of its specific structure, a neural network can be used to realize a learning process [2]. In general, learning is carried out in two steps: (1) a network structure is defined, and (2) an algorithm is derived for realizing the learning process. Usually, the structure of the neural network is fixed for a learning process. However, this fixed structure may not provide the best performance. If the neural network structure is too complicated, the implementation cost will be high.

In this paper, an eBook graffiti command interpreter is proposed. A three-layer neural network with link switches introduced in each link is proposed to facilitate the tuning of the optimal network structure. This neural network with link switches is the basic component of the proposed eBook graffiti command interpreter. GA employing arithmetic crossover and non-uniform mutation [5] is used to tune the structure and the parameters of the neural network. The proposed graffiti command interpreter is then applied to an eBook reader experimentally.

This paper is organized as follows. The proposed 3-layer neural network with link switches is presented in section II. The eBook graffiti command interpreter is presented in section III. Results on the interpretation of graffiti commands using the proposed interpreter for eBooks will be given in section IV. A conclusion will be drawn in Section V.

II. NEURAL NETWORK WITH LINK SWITCHES

In this section, a neural network with link switches [19-20] is presented. By introducing a switch to each link, not only the parameters but also the structure of the neural network can be tuned using GA [5].

A multiple-input-multiple-output three-layer neural network as shown in Fig. 2 is proposed. Specifically, a unit step function is introduced to each link to realize a link switch. Such a unit step function is defined as,

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

Referring to Fig. 2, the input-output relationship of the proposed multiple-input-multiple-output three-layer neural network is as follows,

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

[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).

$z_i(t)$, $i = 1, 2, \ldots, n_{in}$, are the inputs which are functions of a variable t; n_{in} denotes the number of inputs; v_{ij}, $i = 1, 2, \ldots, n_{in}$; $j = 1, 2, \ldots, n_h$, denotes the weight of the link between the i–th input and the j-th hidden node; n_h denotes the number of the hidden nodes; s_{ij}^1, denotes the parameter of the link switch from the i-th input to the j-th hidden node. s_{jk}^2, $j = 1, 2, \ldots, n_h$, $k = 1, 2, \ldots, n_{out}$, 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:

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

$y_k(t)$ is the k-th output of the proposed neural network. By introducing the switches, the weights v_{ij} and the switch states can be tuned. In 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. EBOOK GRAFFITI COMMAND INTERPRETER

In this section, the proposed neural network is employed to interpret graffiti commands for eBooks. Fig. 3 shows the block diagram of the graffiti command interpreter with m graffiti commands. It consists of m neural networks and a graffiti command determiner. The input-output relationships of the proposed neural networks are trained using 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,

$$\mathbf{y}^d(t) = \mathbf{z}(t) \equiv \frac{\mathbf{x}(t)}{\|\mathbf{x}(t)\|},\ t = 1, 2, \ldots, n_d \qquad (4)$$

where $\mathbf{y}^d(t) = \begin{bmatrix} y_1^d(t) & y_2^d(t) & \cdots & y_{n_{in}}^d(t) \end{bmatrix}$ and $\mathbf{z}(t) = \begin{bmatrix} z_1(t) & z_2(t) & \cdots & z_{n_{in}}(t) \end{bmatrix}$ are the desired outputs and the inputs of the neural network respectively. $\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_{n_{in}}(t) \end{bmatrix}$ is the sampled points of the graffiti commands. $\|\cdot\|$ denotes the l_2 norm. n_d denotes the number of input-output data pairs. The fitness function is defined as,

$$fitness = \frac{1}{1+err} \qquad (5)$$

$$err = \sum_{k=1}^{n_{in}} \frac{\sum_{t=1}^{n_d} \left| \frac{y_k(t)}{\|\mathbf{y}(t)\|} - \frac{y_k^d(t)}{\|\mathbf{y}^d(t)\|} \right|}{n_d} \qquad (6)$$

The objective is to maximize the fitness value of (5) using GA with arithmetic crossover and non-uniform mutation [5] by setting the chromosome to be $\begin{bmatrix} s_{jk}^2 & w_{jk} & s_{ij}^1 & v_{ij} & s_j^1 & b_j^1 & s_k^2 & b_k^2 \end{bmatrix}$ for all i, j, k. It can be seen from (5) and (6) that a larger fitness value implies a smaller error value. From (5) and (6), the neural network 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 neural networks correspondingly. Each set of graffiti training samples is used to train a neural network. During the operation, the sampled points of the input graffiti command will be fed to all the m neural networks. The output of the m neural networks will be fed to the graffiti command determiner to generate the final result which indicates the possible input graffiti command. The graffiti command determiner will measure the similarity between the input graffiti command and the outputs of the neural networks. The similarity of an input graffiti command to an output of a neural network is defined as,

$$S_i = \|\bar{\mathbf{y}}_i - \bar{\mathbf{z}}\|,\ i = 1, 2, \ldots, m \qquad (7)$$

where,

$$\bar{\mathbf{y}}_i = \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|} = \begin{bmatrix} \bar{y}_1(t) & \bar{y}_2(t) & \cdots & \bar{y}_{n_{in}}(t) \end{bmatrix},\ i = 1, 2, \ldots, m \qquad (8)$$

$$\bar{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|} = \begin{bmatrix} \bar{z}_1(t) & \bar{z}_2(t) & \cdots & \bar{z}_{n_{in}}(t) \end{bmatrix} \qquad (9)$$

$\bar{\mathbf{y}}_i$ and $\bar{\mathbf{z}}$ denote the normalized outputs and the normalized input of the neural networks respectively. A smaller value of S_i implies that the input graffiti matches more closely to the graffiti represented by the i-th neural network. The smallest similarity values among the m neural networks is defined as,

$$S_j = \min_i S_i \qquad (10)$$

The index j of (10) is the output of the graffiti command determiner which indicates the j-th graffiti is the most likely input graffiti command.

IV. APPLICATION EXAMPLE AND RESULTS

The interpretation of graffiti commands for eBooks by the proposed neural network 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 command interpreter. The interpretation process is achieved by a three-layer neural network (10-input-10-output, 11 hidden nodes) with link switches for each graffiti. The ten inputs nodes, z_i, $i = 1, 2, \ldots, 10$, represent ten sampled points of the input graffiti command. These ten sampled points are taken uniformly from the input graffiti commands. In our eBook application, three graffiti commands are defined: rectangle, triangle and straight line which are shown in Fig. 4. The eBook graffiti command interpreter thus has three neural networks for interpreting three graffiti commands. To train the neural networks, 500 sampled points for each set of the graffiti command are used.

In this neural network, the number of hidden nodes is 11. Referring to (2), the proposed neural network used for the interpretation process is governed by,

$$y_k(t) = \sum_{j=1}^{11} \delta(s_{jk}^2) w_{jk} \, logsig\left[\sum_{i=1}^{10}\left(\delta(s_{ij}^1) v_{ij} z_i(t) - \delta(s_j^1) b_j^1\right)\right] - \delta(s_k^2) b_k^2, \quad k = 1, 2, \ldots, 10 \tag{11}$$

The fitness function for each neural network is defined as follows,

$$fitness = \frac{1}{1+err} \tag{12}$$

$$err = \sum_{k=1}^{10} \frac{\sum_{t=1}^{500} \left| \frac{y_k(t)}{\|\mathbf{y}(t)\|} - \frac{y_k^d(t)}{\|\mathbf{y}^d(t)\|} \right|}{500} \tag{13}$$

The GA with arithmetic crossover and non-uniform mutation [5] is employed to tune the parameters and structure of the neural network of (11). The objective is to maximize the fitness function of (12). 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 $\frac{-3}{\sqrt{n_h}} \geq v_{ij}, w_{jk}, b_j^1, b_1^2 \geq \frac{3}{\sqrt{n_h}}$, $n_h = 11$, $-1 \geq s_{j1}^2, s_{ij}^1, s_j^1, s_1^2 \geq 1$, $i = 1, 2, \ldots, 10$; $j = 1, 2, \ldots, 11$, $k = 1, 2, \ldots, 10$ [18]. The chromosomes used are $\begin{bmatrix} s_{j1}^2 & w_{jk} & s_{ij}^1 & v_{ij} & s_j^1 & b_j^1 & s_1^2 & b_1^2 \end{bmatrix}$ for all *i*, *j* and *k*. The initial values of the link weights are randomly generated. The learning processes are carried out by a personal computer with P4 1.4GHz CPU and 256M RAM. The number of iterations to train the neural networks is 2000. Training for one neural network takes around 10.11 hours. After training, 50 graffiti samples of each kind of graffiti (50 × 3 = 150 testing graffiti commands) are used to test the performance of the trained neural networks. The simulation results are tabulated in Table I. From this Table, it can be observed that the numbers of connected links in the three neural networks are reduced after learning. A fully connected neural networks has 241 links, including the bias links. Fig. 5 shows the similarity values for each neural network of the 150 testing graffiti commands. It can be seen that the neural network trained by a particular graffiti command will provide a smaller similarity values for that kind of graffiti command. For example, in Fig. 5(a), the similarity values of the first 50 testing graffiti (rectangles) are smaller, as that neural network is trained by 500 rectangular training graffiti. We have successfully implemented the graffiti command interpreter in the eBook shown in Fig. 1. The rectangle is for using the pen application (Fig. 5), triangle for using the bookmark application (Fig. 6) and straight line for using the magnification application (Fig. 7).

V. Conclusion

By introducing a switch to each link, a neural network that facilitates the tuning of its structure has been proposed. Using GA with arithmetic crossover and non-uniform mutation, the proposed neural network can learn the input-output relationship of an application and the optimal network structure. A fully connected neural network will become a partly connected neural network after learning. This implies a lower cost of implementation. The proposed neural network with link switches trained by GA with arithmetic crossover and non-uniform mutation has been applied to interpret graffiti commands for eBooks. It has been shown that our proposed network can meet the stated objectives.

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] 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.

[8] T.J. Cholewo, J.M. Zurada, "Sequential network construction for time series prediction," in *Proc. Int. Conf. Neural Networks*, vol.4, 1997, pp.2034-2038.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] H. K. Lam, S. H. Ling, K. F. Leung, and F. H. F. Leung, "On Interpretation of Graffiti Commands for eBooks using a Neural Network and an Improved Genetic Algorithm," *Proceedings of the tenth International Conference on Fuzzy Systems, IEEEFUZZ'2001* (to be published).

[20] 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).

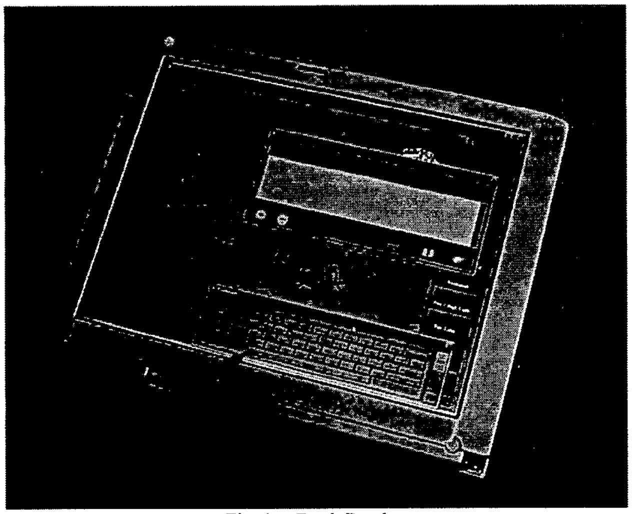


Fig. 1. eBook Reader.

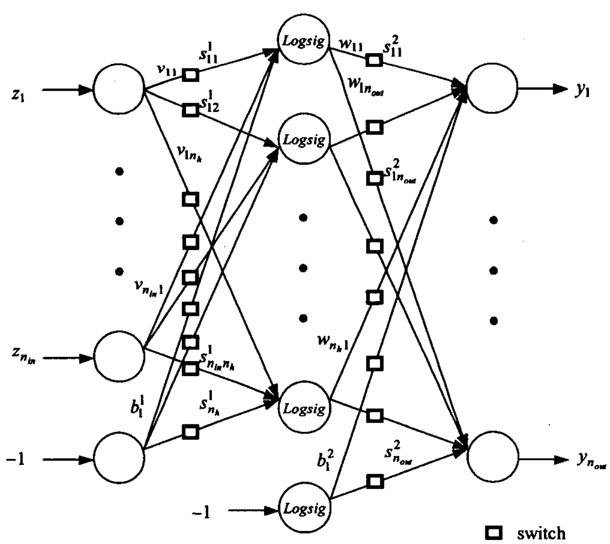


Fig. 2. 3-layer neural network with switches.

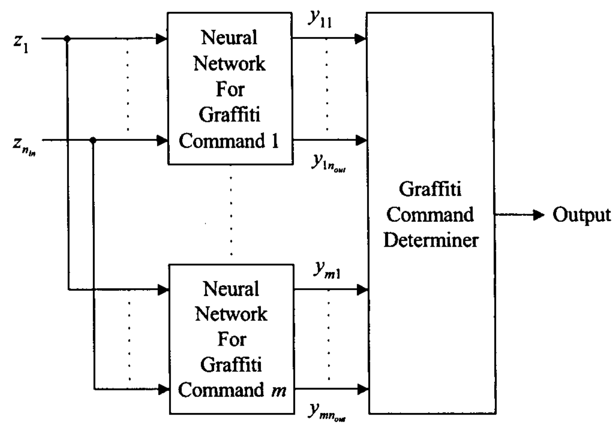
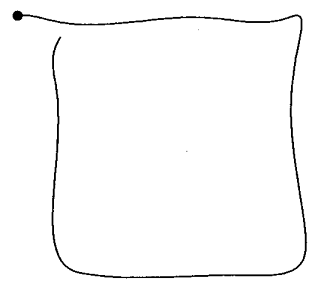
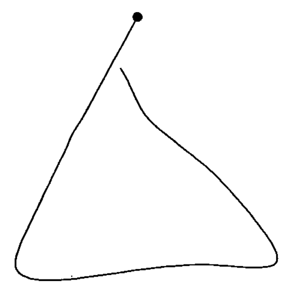


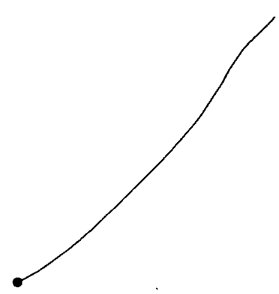
Fig. 3. Block diagram of the Graffiti Command Interpreter.



(a). Rectangle.

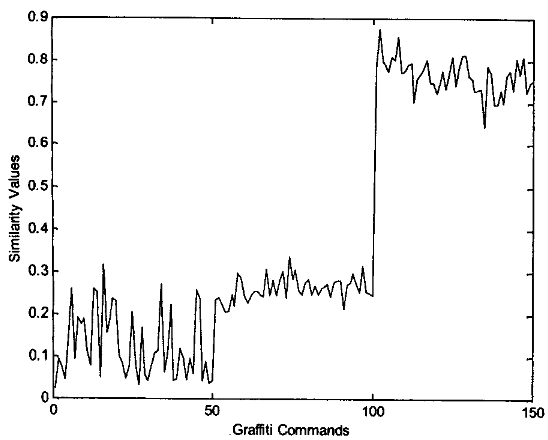


(b). Triangle.



(c). Straight line.

Fig. 4. Graffiti commands used in the eBook (the dot indicates the starting point).



(a). Rectangle.

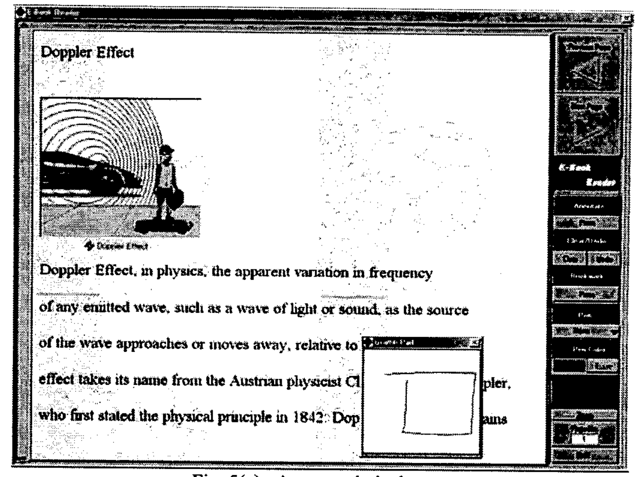
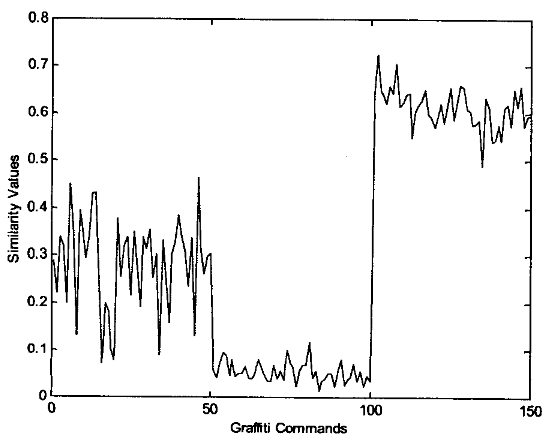


Fig. 5(a). A rectangle is drawn.



(b). Triangle.

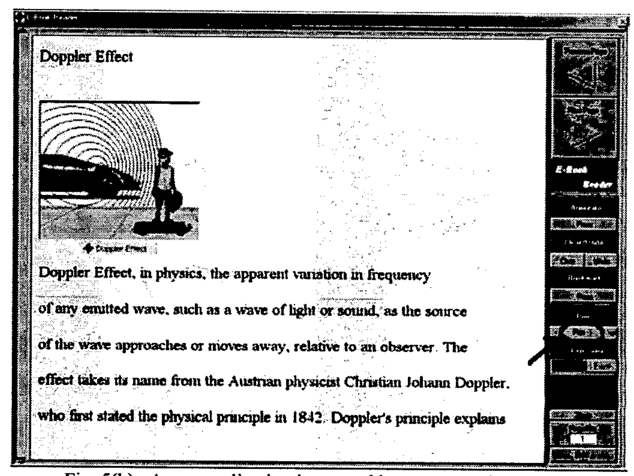
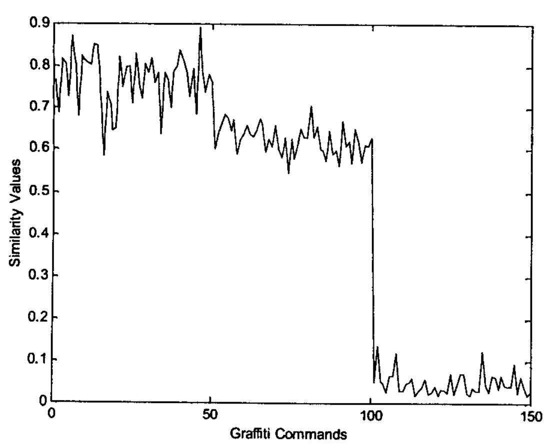


Fig. 5(b). A pen application is opened by a rectangular graffiti.



(c). Straight line.

Fig. 5. Similarity values of the three neural networks for the 150 testing graffiti. The first 50 testing graffiti commands are rectangles, the second 50 testing graffiti commands are triangles and the last 50 testing graffiti commands are straight lines.

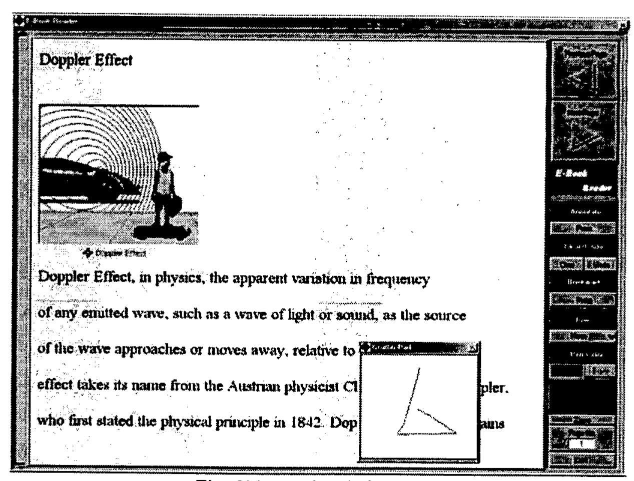


Fig. 6(a). A triangle is drawn.

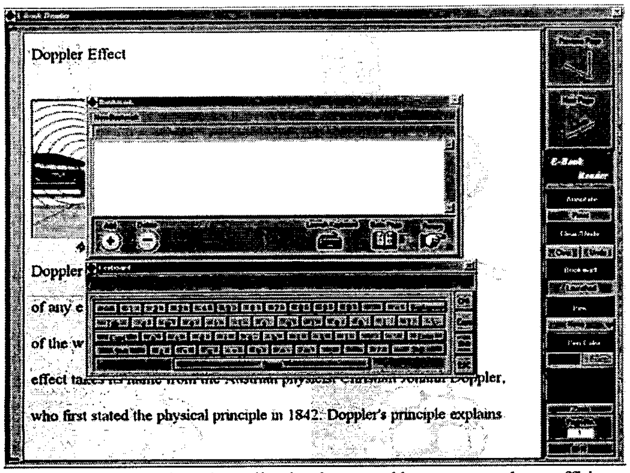


Fig. 6(b). A bookmark application is opened by a rectangular graffiti.

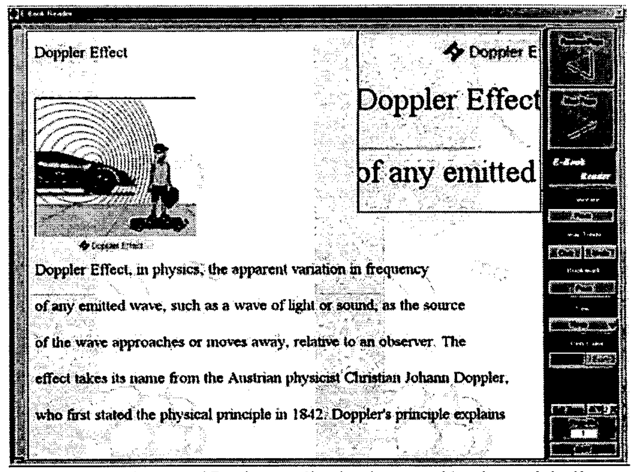


Fig. 7(b). The magnification application is opened by the straight-line graffiti.

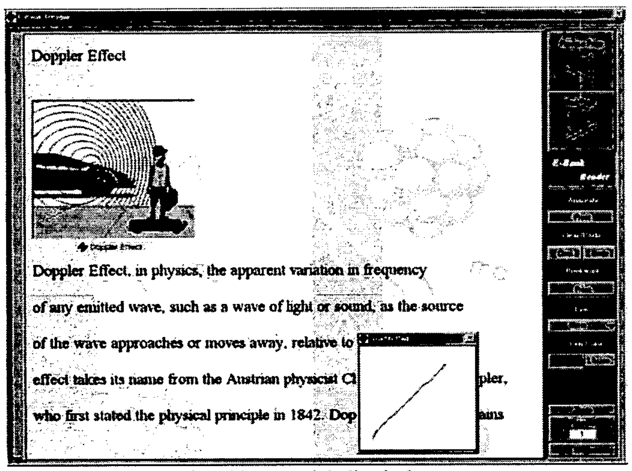


Fig. 7(a). A straight line is drawn.

Neural Network	Fitness Value	Number of Links	Training Error	Testing Error
Rectangle	0.9637	187	0.0363	0.0309
Triangle	0.9877	178	0.0123	0.0136
Straight Line	0.9879	188	0.0121	0.0119

Table I. Simulation results of the proposed neural networks for interpreting graffiti commands after training for 2000 iterations.

0-7803-7278-6/02/$10.00 ©2002 IEEE