Supervised Adaptive Resonance Theory Neural Network for Modelling Dynamic Systems

D. T. Pham and M. F. Sukkar
Intelligent Systems Research Laboratory
Systems Engineering Division
School of Engineering
University of Wales Cardiff
PO Box 917, Newport Road, Cardiff CF2 1XH, U.K.

ABSTRACT

A supervised neural network, SMART2, has been developed which can be used with the ART2 algorithm for modelling discrete dynamic systems. A new layer has been added as a higher transformation stage to provide an output mapping field. The connection between the new field and the category field has been made by Long Term Memory (LTM) adaptive filters. Top-down adaptive filters in the new field have been employed to code the output prototype. Error equations have been derived to trace errors in the model and train the new network. The proposed network has been shown in simulation to be able to represent arbitrary dynamic systems. Results presented in this paper demonstrate the effectiveness of the network.

Keywords: Adaptive Resonance Theory; Long Term Memory; Short Term Memory; ART2 network; SMART2 network.

1. INTRODUCTION

Adaptive Resonance Theory (ART) networks which were developed by Grossberg and Carpenter are self-organising neural networks, that is they make no use of the class information associated with a training pattern [1]. ART nets automatically detect clustering and form classes of the data structure [2]. The ART architecture is reasonably suited for pattern recognition tasks [3, 4]. An extension to incorporate supervised learning to enable the architecture to act as a mapping network has been developed as a modified version of ART [5, 6].

This paper presents a different type of ART2 network which is also a supervised network suitable for mapping applications, the Supervised Mapping ART2, or SMART2, network. The paper discusses the application of SMART2 to the modelling of dynamic systems. This is a task that has so far been implemented mainly using other kinds of networks such as the multilayer perceptron or the Elman and Jordan networks (see [7-10] for example).

The remainder of the paper is organised as follows. Section 2 describes SMART2, and how it has been designed for dynamic system modelling. Section 3 reports on the use of SMART2 to model different plants. The paper concludes with Section 4. The paper assumes the reader is reasonably familiar with the ART2 network, a detailed description of which can be found in [2].

2. SMART2 AND DYNAMIC SYSTEM MODELLING

Supervised Function

SMART2 is a supervised learning network because there is a definite input-output mapping that the network must learn and the network receives a required output from its environment. The architecture of SMART2 is shown in Figures 1 and 2.

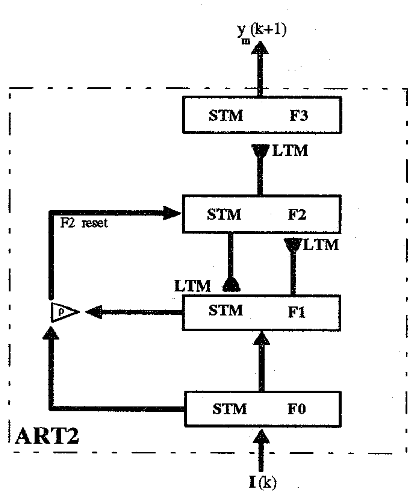

Figure 1. Overall structure of SMART2 network

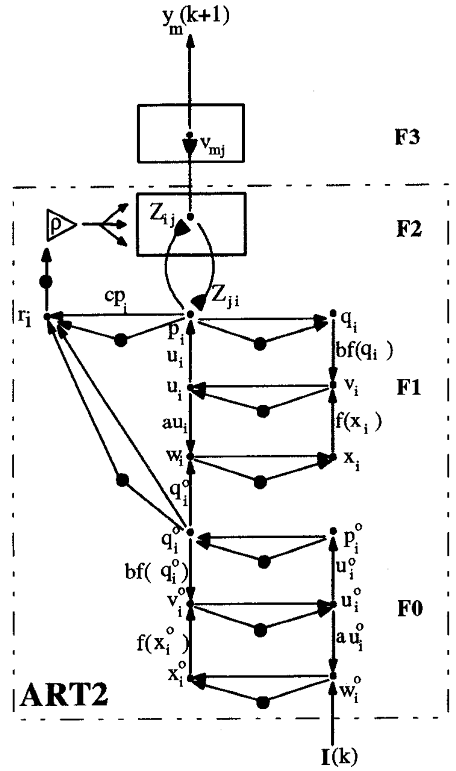

Figure 2. Detailed architecture of SMART2 network

In Figure 2, p_i^0, u_i^0, w_i^0, q_i^0, v_i^0, x_i^0, p_i, u_i, w_i, q_i, v_i and x_i are the STM for node i in the preprocessors of the F0 and F1 layers. a and b are internal feedback parameters for the F0 and F1 layers. Feedback is employed to perform contrast enhancement and noise suppression. c is a parameter controlling the sensitivity of the network to the reset operation. d is a top-down feedback parameter determining the amount of weight change during learning. Z_{ij} and Z_{ji} are the LTM for the bottom-up and top-down adaptive filters between F1 and F2. r_i is the i-th component of the reset vector **r** and ρ is the vigilance parameter. f is a non-linear or piecewise linear feedback activation function used in the F0 and F1 preprocessors. θ is the threshold in f.

To allow supervised learning of dynamic patterns, a new layer (F3) with adaptive filter connections has been added to the original ART2 network. The number of nodes, m, in F3 is equal to the dimension of the system output. The adaptive filters introduced by F3 are contained in pathways leading from the nodes in F3 to the category representation field F2. During training, the ART2 module receives a series of input patterns **I**(k) and the top-down adaptive filter F3 is supplied with a series of corresponding output patterns **y**(k+1). These outputs are stored only in the new Long Term Memory (LTM) traces used in the top-down adaptive filter leading from the output field F3 to the category field F2.

F2 selects the node to be fired as the winning node receiving the largest total input and quenching activity in all other nodes. During learning, after the winning node j in F2 has been chosen, the top-down LTM traces equations of F3 are:

$$v_{mj} = y_m(k+1) \tag{1}$$

if the j-th F2 node has not been reset on the current trial, or

$$v_{mj} = 0 \tag{2}$$

otherwise.

In the above equations, v_{mj} is the LTM trace of node m in F3 due to node j in F2 and $y_m(k+1)$ is the m-th output of the system at time k.

If during recall, at time (k+1), the j-th node in F2 has been selected for firing by the winner-take-all rule, the Short Term Memory (STM) $y_m(k+1)$ of the m-th node in F3 obeys the following equation:

$$y_m(k+1) = v_{mj} \tag{3}$$

Learning Scheme and Error Equations

The scheme for training SMART2 to model a dynamic plant is illustrated in Figure 3. Given a time series input u(k) (k= 0, 1, 2, ...), SMART2 is taught to generate the corresponding time series output y(k+1) (k= 0, 1, 2, ...) that the plant will produce if supplied with u(k).

As the plant is dynamic, at a particular time (k+1), y(k+1) depends not only on u(k) but on previous output values and possibly previous input values also. To reflect this, in SMART2, the input vector **I**(k) is made up of the current input u(k) and previous input u(k-1) plus the current output y(k) and previous output y(k-1), namely:

$$\mathbf{I}(k) = [u(k) \quad u(k-1) \quad y(k) \quad y(k-1)] \tag{4}$$

This is illustrated in Figures 3 and 4. The latter Figure shows SMART2 acting as an independent model after training.

The recognition category for **I**(k) is represented at node j in field F2 by the weight vector $\mathbf{Z}_j(k)$, where:

$$\mathbf{Z}_j(k) = [Z_{j0}(k) \quad Z_{j1}(k) \quad Z_{j2}(k) \quad Z_{j3}(k)] \tag{5}$$

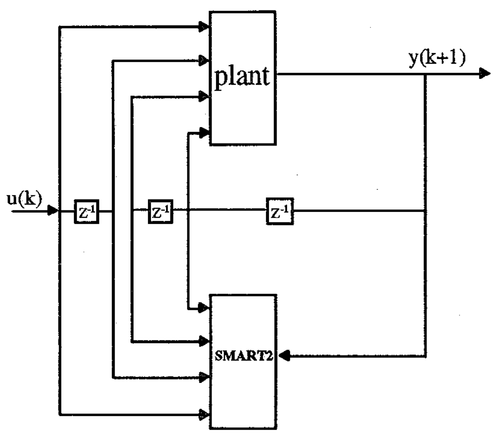

Figure 3. Identification structure

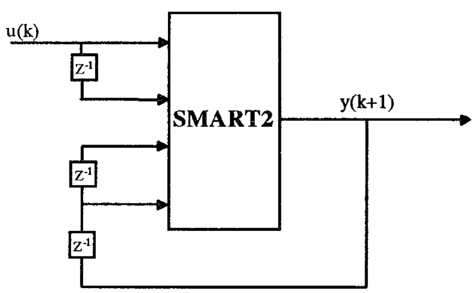

Figure 4. Recall structure

In equation (5), $Z_{ji}(k)$ $(0 \le i \le 3)$ is the weight (LTM trace) in the top-down adaptive filter from the j-th node of the F2 layer to the i-th node of the F1 layer.

At time (k+1), the input and weight vectors are:

$$\mathbf{I}(k+1)= [u(k+1) \quad u(k) \quad y(k+1) \quad y(k)] \tag{6}$$

$$\mathbf{Z}_{ji}(k+1)= [Z_{j0}(k+1) \quad Z_{j1}(k+1) \quad Z_{j2}(k+1) \quad Z_{j3}(k+1)] \tag{7}$$

Consequently, at time (k+2), the input and weight vectors are:

$$\mathbf{I}(k+2)= [u(k+2) \quad u(k+1) \quad y(k+2) \quad y(k+1)] \tag{8}$$

$$\mathbf{Z}_{j}(k+2)= [Z_{j0}(k+2) \quad Z_{j1}(k+2) \quad Z_{j2}(k+2) \quad Z_{j3}(k+2)] \tag{9}$$

Ideally, after the weights have settled down following the presentation of the input vector,

$$Z_{j0}(k+1)= Z_{j1}(k+2)= u(k+1) \tag{10}$$

$$Z_{j1}(k+1)= Z_{j0}(k)= u(k) \tag{11}$$

$$Z_{j2}(k+1)= Z_{j3}(k+2)= y(k+1) \tag{12}$$

$$Z_{j3}(k+1)= Z_{j2}(k)= y(k) \tag{13}$$

If SMART2 produces an incorrect output when a certain input is applied, the top-down weights $\mathbf{Z}_j$ can be adjusted to improve the output by using the difference $e_i(k+1)$ (i= 0 to 3) between the actual and ideal weights. That is:

$$\overline{Z}_{ji}(k+1)= Z_{ji}(k+1) + \alpha\, e_i(k+1) \tag{14}$$

where $\overline{Z}_{ji}(k+1)$ is the corrected value of $Z_{ji}(k+1)$, α is a learning rate $(0 < \alpha < 1)$ and

$$e_0= Z_{j1}(k+2) - Z_{j0}(k+1) \tag{15}$$

$$e_1= Z_{j0}(k) - Z_{j1}(k+1) \tag{16}$$

$$e_2= Z_{j3}(k+2) - Z_{j2}(k+1) \tag{17}$$

$$e_3= Z_{j2}(k) - Z_{j3}(k+1) \tag{18}$$

Note that adjustments to the bottom-up LTM adaptive filters from F1 to F2 are made in the same manner.

SMART2 Algorithm

The main steps of the algorithm are as follows:

1. Initialise all weights in the ART2 module according to the ART2 procedure [2] and set those in the new F3 layer to zero.

2. Present an input-output pair.

3. Determine the F2 node to be fired.

4. Start learning and update LTM traces between F1 and F2 using the ART2 procedure and those between F2 and F3 using Equations (1) and (2).

5. Adjust the weights between F1 and F2 according to Equation (14).

6. Go to step 2.

3. SIMULATIONS

SMART2 has been trained in simulation to model different types of plants. This section presents the results for a second-order plant, a third-order plant and a non-linear plant. A series of uniform random inputs $u \in [0, 1]$ was presented to the plants to produce a training data set of 80 points. Complement coding was used for negative values.

After being trained on that data, SMART2 was employed to generate outputs corresponding to a series of 300 inputs previously not encountered during training.

In all simulations, the F0 and F1 layers each had 4 nodes (the input vector dimension), the F2 layer had 80 nodes (the number of training data) and the F3 layer had 1 node (the output dimension).

Second-Order Plant

The second-order linear plant used in the simulations could be represented by the following difference equation:

$$y(k+1)= 1.72357y(k) - 0.74082y(k-1) + 0.009048u(k) - 0.008214u(k-1) \quad (19)$$

The plant input and output are denoted by u and y respectively. A piecewise linear activation function with θ= 0.018 was used. Internal feedback parameters a and b were both set to 9.4 to give stable STM preprocessor activities. The reset sensitivity parameter c was 0.1. The value of the feedback learning parameter d was 0.9. The vigilance parameter ρ was 0.96. The learning parameter α was 0.62. The values of these parameters were obtained by experimentation.

It was found that gain factors a and b only had a minor influence, the values used being necessary to ensure learning stability and allow the network to settle down. A piecewise linear activation function was used in preprocessors F0 and F1. The degree of contrast enhancement and noise suppression is determined by the threshold θ in the feedback activation function. Learning rules in F2 follow the membrane equation [2]. Slow learning was achieved using the Runge-Kutta method for solving that equation. The step size (h) was 0.1 and the maximum number of time steps (nsteps) was 100. Fast learning was implemented by applying Equations (1) and (2) in the LTM of the new F3 layer. The training time was approximately 10 seconds on a PC/AT 486 33 MHz microcomputer.

Figures 5 and 6 show the step and sinusoidal responses of SMART2 following training. They have been superimposed on those of the plant. It can be seen that there is little difference between them and thus the trained network is an accurate model of the plant.

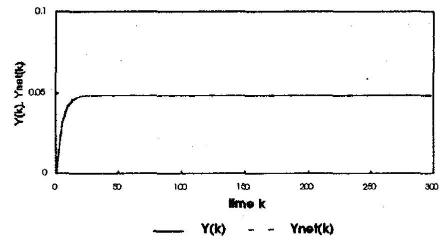

Figure 5. Step response of the second-order plant

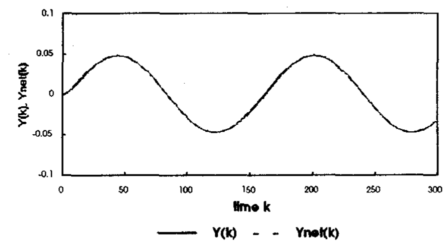

Figure 6. Sinusoidal response of the second-order plant

Third-Order Plant

The 3rd-order linear plant used could be represented by the following discrete input-output equation:

$$y(k+1)= 2.038y(k) - 1.366y(k-1) + 0.301y(k-2) + 0.0059u(k) + 0.018u(k-1) + 0.0033u(k-2) \quad (20)$$

The parameters employed in the SMART2 network were as follows: θ= 0.002, a= b= 8.2, c= 0.1, d= 0.9, ρ= 0.96, α= 0.51, h= 0.1 and nsteps= 100.

Figures 7 and 8 show the step and sinusoidal responses of SMART2 superimposed on those of the plant. Again, it can be noted that there are insignificant differences between the responses of SMART2 and those of the plant.

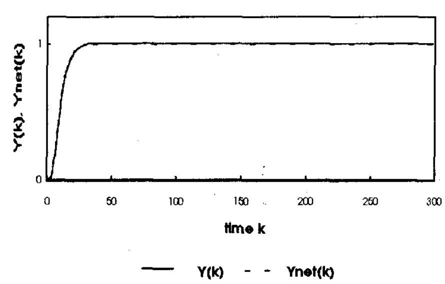

Figure 7. Step response of the third-order plant

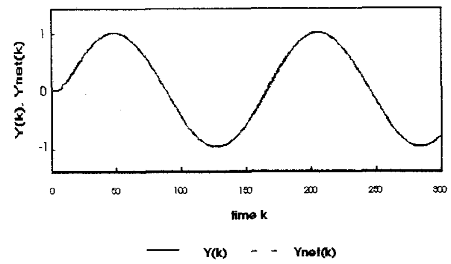

Figure 8. Sinusoidal response of the third-order plant

Non-linear Plant

The non-linear plant model adopted could be described by the following discrete input-output equation:

$$y(k+1) = \frac{y(k)}{1.5 + y^2(k)} - 0.3y(k-1) + 0.5u(k) \quad (21)$$

The parameter settings were as follows: $\theta= 0.2$, $a= b= 7.2$, $c= 0.1$, $d= 0.9$, $\rho= 0.98$, $\alpha= 0.39$, $h= 0.1$ and nsteps= 100.

Figures 9 and 10 give the step and sinusoidal responses of SMART2 which are virtually indistinguishable from those of the plant.

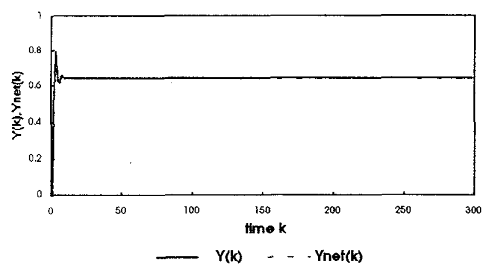

Figure 9. Step response of the non-linear plant

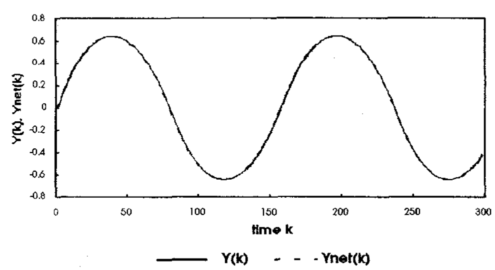

Figure 10. Sinusoidal response of the non-linear plant

4. CONCLUSION

SMART2, a supervised version of the ART2 neural network designed for modelling discrete dynamic systems, has been demonstrated in simulation on different types of plants. The advantages of SMART2 are its ability to model both linear and non-linear plants accurately, its requirement for a low number of inputs (only four input nodes are employed in all cases), its need for little training data (only 80 data points are used in all cases) and its fast training speed.

5. REFERENCES

[1] G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine", *Computer Vision, Graphics, and Image Processing*, Vol. 37, 1987, pp. 54-115.

[2] G. A. Carpenter and S. Grossberg, "ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns", *Applied Optics*, Vol. 26, No. 23, 1987, pp. 4919-4930.

[3] G. A. Carpenter and S. Grossberg, "The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network", *Computer*, Vol. 21, No. 3, 1988, pp. 77-88.

[4] G. A. Carpenter and S. Grossberg, "Search Mechanisms for Adaptive Resonance Theory (ART) Architectures", *IJCNN, Washington D.C.*, Vol. 1, 1989, pp. 201-205.

[5] G. A. Carpenter, S. Grossberg and J. H. Reynolds, "ARTMAP: A Self-Organizing Neural Network Architecture for Fast Supervised Learning and Pattern Recognition", *IJCNN, Seattle, WA*, Vol. 1, 1991, pp. 863-868.

[6] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds and D. B. Rosen, "FUZZY ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps", *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, 1992, pp. 698-713.

[7] D. T. Pham and X. Liu, "State-Space Identification of Dynamic Systems Using Neural Networks", *Eng. Appli. of AI*, Vol. 3, 1990, pp. 198-203.

[8] D. T. Pham and X. Liu, "Neural Networks for Discrete Dynamic System Identification", *J. Syst. Eng.*, Vol. 1, 1991, pp. 51-60.

[9] D. T. Pham and X. Liu, "Dynamic System Modelling Using Partially Recurrent Neural Networks", *J. Syst. Eng.*, Vol. 2, 1992, pp. 90-97.

[10] D. T. Pham and S. J. Oh, "A Recurrent Backpropagation Neural Network for Dynamic System Identification", *J. Syst. Eng.*, Vol. 2, 1992, pp. 213-223.

6. ACKNOWLEDGEMENT

The authors would like to thank the University of Aleppo (Syria) for supporting M. F. Sukkar.