Journal of Electromagnetic Waves and Applications, Vol. 9, No. 1/2, 17–36, 1995

Design of gratings and frequency selective surfaces using Fuzzy ARTMAP neural networks

C. G. Christodoulou, J. Huang, M. Georgiopoulos, and J. J. Liou

Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816, USA

Abstract–This paper presents a study of the Fuzzy ARTMAP neural network in designing cascaded gratings and frequency selective surfaces (FSS) in general. Conventionally, trial and error procedures are used until an FSS matches the design criteria. One way of avoiding this laborious process is to use neural networks (NNs). A neural network can be trained to predict the dimensions of the elements comprising the FSS structure, their distance of separation, and their shape required to produce the desired frequency response. In the past, the multi-layer perception architecture trained with the back-prop learning algorithm (back-prop network) was used to solve this problem. Unfortunately, the back-prop network experiences, at times, convergence problems and these problems become amplified as the size of the training set increases. In this work, the Fuzzy ARTMAP neural network is used to address the FSS design problem. The Fuzzy ARTMAP neural network converges much faster than the back-prop network, and most importantly its convergence to a solution is guaranteed. Several results (frequency responses) from cascaded gratings corresponding to various angles of wave incidence, layer separation, width strips, and interstrip separation are presented and discussed.

1. INTRODUCTION

Frequency selective surfaces (FSS) have numerous applications as electromagnetic system devices, such as polarizers, filters, radomes, dichroic reflectors, infrared sensors and beam tuners for optical systems. Currently, there is no closed-form solution that can directly relate a desired frequency response to the corresponding FSS. Trial and error procedures are used until a frequency selective surface matches the design criteria. One way to avoid this process and obtain a synthesis procedure is to utilize the training capabilities of neural networks. A neural network can be trained to predict the dimensions of the metallic patches (or apertures), their distance of separation and their shape in order for an FSS to produce the desired frequency response.

Previous work is limited to designing an FSS using the back-prop neural network ([1,2]). The back-prop neural network is a multi-layer feed-forward neural network trained with the back-prop learning algorithm ([3]). During the training of this network the geometric information pertaining to an FSS (e.g., the dimensions, shapes, etc.) is fed as input to the neural network. The corresponding frequency response for each dimension and shape is also provided to the neural network as its desired output. Each pair (input, desired output) constitutes a

training pair and it is utilized to train the neural network architecture. Once the neural network is trained, a desired response is provided at the neural network output, and through an algorithmic process, called inversion ([4]), the network produces at its inputs the appropriate dimensions and element shapes of an FSS that can generate this desired response.

In this paper, we propose an alternative method to tackle the aforementioned FSS problem. This method relies on another neural network architecture, named Fuzzy ARTMAP [5]. This neural network possesses several advantages compared with the back-prop neural network. First, the Fuzzy ARTMAP neural network is faster to train than the back-prop neural network. Second, once Fuzzy ARTMAP is trained with a list of training data, it does not require extensive retraining with the old data when new training data are added to the training list. On the other hand, extensive retraining of the back-prop network with the old training data is often required when new training data are added to the training list. Finally, Fuzzy ARTMAP is guaranteed to converge to a solution for any collection of training data provided to the network, whereas the back-prop network has often problems to converge to the right solution.

2. THE FUZZY ARTMAP NEURAL NETWORK

2.1 Preliminaries

As most of the existing neural network models the Fuzzy ARTMAP neural network consists of a number of nodes (neurons) that are interconnected with each other via connections of varying strength, called *weights*. Fuzzy ARTMAP is capable of learning arbitrary mappings from an input space of arbitrary dimensionality to an output space of arbitrary dimensionality.

In particular, the Fuzzy ARTMAP neural network consists of two Fuzzy ART ([6]) modules designated as ART_a and ART_b, as well as an inter-ART module as shown in Fig. 1. Inputs from an input space are presented at the ART_a module, while outputs from an output space are presented at the ART_b module. The inter-ART module includes a MAP-layer F_{ab}, whose purpose is to determine whether the mapping between the inputs and the outputs is the desired one.

The ART_a module consists of two layers of nodes designated by F_1^a and F_2^a. Weights, called *bottom-up weights*, connect the nodes of layer F_1^a and the nodes of layer F_2^a. Similarly, weights, called *top-down weights*, connect the nodes of layer F_2^a and the nodes of layer F_1^a. The ART_b module consists of two layers of nodes designated by F_1^b and F_2^b. Weights, called *bottom-up weights*, connect the nodes of layer F_1^b and the nodes of layer F_2^b. Similarly, weights, called *top-down weights*, connect the nodes of layer F_2^b and the nodes of layer F_1^b. The Inter-ART module consists of a layer of nodes, called the MAP-layer F_{ab}. There are connections emanating from the nodes of layer F_2^a and converging to the nodes of the MAP-layer F_{ab}, as well as weights between the nodes of the MAP-layer F_{ab} and the nodes of layer F_2^b.

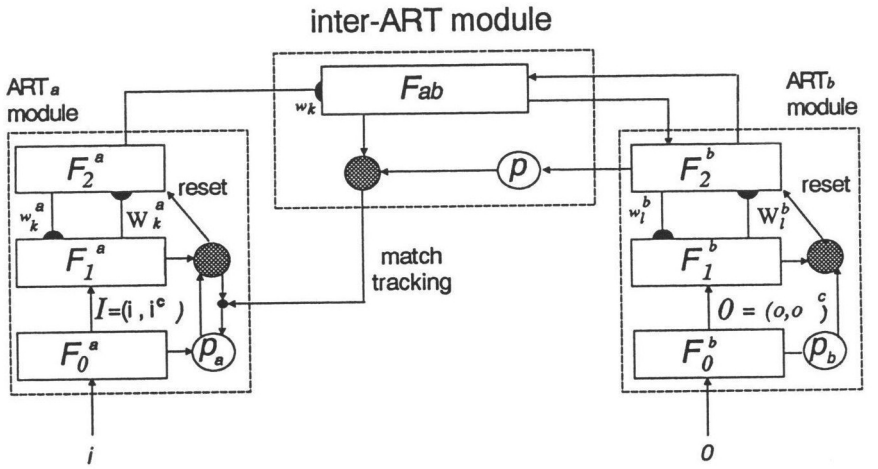


Figure 1. Fuzzy ARTMAP architecture.

Before an input is presented to the ART_a module it is normalized so that each one of its components has values lying in the interval $[0, 1]$. Furthermore, additional preprocessing of the normalized inputs, occurring at layer F_0^a of ART_a, is required; this type of preprocessing, called *complement coding*, takes a normalized input vector $\mathbf{i}$, and it produces a vector $\mathbf{I}$, such that

$$\mathbf{I} = (\mathbf{i}, \mathbf{i}^c) \tag{1}$$

where $\mathbf{i}^c = \mathbf{1} - \mathbf{i}$, and $\mathbf{1}$ stands for a vector with all of its components equal to 1. Similar type of normalization and preprocessing stages are required at the ART_b module. That is if $\mathbf{o}$ designates a normalized output, the complement coded output $\mathbf{O}$ presented at the F_1^b layer of ART_b is equal to

$$\mathbf{O} = (\mathbf{o}, \mathbf{o}^c) \tag{2}$$

where $\mathbf{o}^c = \mathbf{1} - \mathbf{o}$, and $\mathbf{1}$, as before, is a vector with all of its components equal to 1.

As it is the case with other neural network models we distinguish two phases with the Fuzzy ARTMAP neural network. The *training phase* and the *performance phase*. In the training phase we have a collection of input output pairs designated as $(\mathbf{I}^1, \mathbf{O}^1), (\mathbf{I}^2, \mathbf{O}^2), \ldots (\mathbf{I}^P, \mathbf{O}^P)$ that we refer to as the *training list*. We want to train the Fuzzy ARTMAP neural network to learn the following mapping: $\mathbf{I}^1$ to $\mathbf{O}^1$, $\mathbf{I}^2$ to $\mathbf{O}^2$, and eventually $\mathbf{I}^P$ to $\mathbf{O}^P$. To do so, we present the training list repeatedly to the Fuzzy ARTMAP architecture. That is we present $\mathbf{I}^1$ to ART_a and $\mathbf{O}^1$ to ART_b, then $\mathbf{I}^2$ to ART_a and $\mathbf{O}^2$ to ART_b, and eventually $\mathbf{I}^P$ to ART_a and $\mathbf{O}^P$ to ART_b; this corresponds to one list presentation. We

present the training list to Fuzzy ARTMAP as many times as it is necessary for the architecture to learn the desired map. The order of presentation of the input pairs within the list is not essential, as long as all the pairs in the list are presented within every list presentation. It is important, for the aforementioned training scenario, to determine when the learning process is completed. We say that the learning process is complete if the adaptable weights in the Fuzzy ARTMAP neural network do no not change in one training list presentation. This is an indication that all the desired mappings between inputs and corresponding outputs from the training list have been established by the neural network architecture. The adaptable weights in the Fuzzy ARTMAP neural are the bottom-up and top-down weights in ART_a and ART_b, and the connections from the F_2^a layer to the MAP-layer F_{ab}.

During the training phase of the Fuzzy ARTMAP neural network an input vector (e.g., input $\mathbf{I}^p$) from the training list is presented at the ART_a module, while the corresponding output vector (e.g., $\mathbf{O}^p$) from the training list is presented at the ART_b module. A compressed representation of the input vector $\mathbf{I}^p$ is established at the F_2^a layer of the ART_a module, and a compressed representation of the output vector $\mathbf{O}^p$ is established at the F_2^b layer of the ART_b module. The purpose of the inter-ART module is to determine whether the compressed representation of the input $\mathbf{I}^p$ at F_2^a is mapped to the compressed representation of the output $\mathbf{O}^p$ at the F_2^b. If the correct mapping is established the adaptable network weights are modified to incorporate this mapping as part of their knowledge. If the correct mapping is not established then the network looks for another compressed representation of the input $\mathbf{I}^p$ at F_2^a which establishes the correct mapping with the compressed representation of output $\mathbf{O}^p$ at F_2^b; then modification of the adaptable network weights ensues.

The performance of the network is investigated once the training phase is completed. During the performance phase of the Fuzzy ARTMAP neural network a test input is provided at the ART_a module and the corresponding output that this test input is mapped to, at the ART_b module, is observed. By presenting to the network architecture a collection of test inputs with known desired outputs we can evaluate the performance of the network on network inputs that it has never seen before.

2.2 FSS Design Using the Fuzzy ARTMAP Neural Network

The Fuzzy ARTMAP neural network is applied to the specific problem of cascaded gratings, with different strip widths, interstrip distance of separation, layer separation, dielectric slab thickness, and angles of wave incidence.

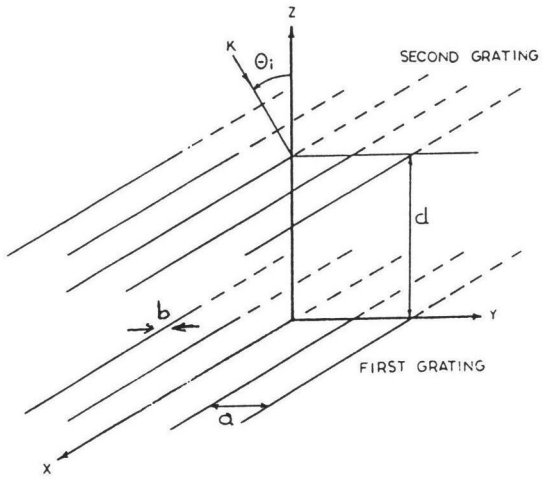
The training inputs for the network correspond to the geometrical information of the cascaded gratings shown in Fig. 2. In particular, 66 sets corresponding to cascaded gratings and 48 sets corresponding to cascaded gratings with dielectric slabs were used as our training inputs. Furthermore, the corresponding frequency responses were used as the training outputs. As a training output to the neural network the magnitude of the transmission coefficient, as a function of cell distance

a, is used. This transmission coefficient curve is referred to in this paper as the *frequency response.* The data for the transmission coefficient as a function of the dielectric slab thickness, number of cascaded layers, wave incidence, and other geometrical parameters were produced by using the Spectral Domain FFT method ([7]) and the cascaded network theory ([8]). Actually, data for the transmission coefficient can be calculated using any other available method.

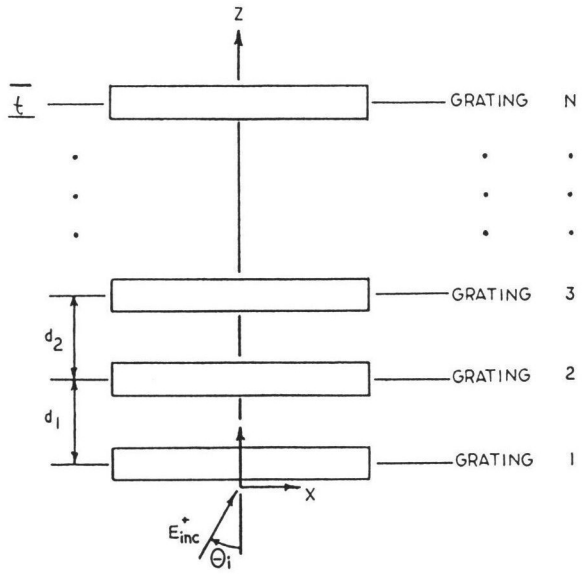
As we emphasized in the previous subsection the inputs and outputs of the training list are normalized prior to their presentation at the ART_a and ART_b modules, respectively. Also, complement coding of the inputs and the outputs is implemented at the preprocessing layers F_0^a and F_0^b of the Fuzzy ARTMAP neural network, respectively.

In our application the F_0^a layer of Fuzzy ARTMAP has 3 nodes, corresponding to the normalized input components θ, b and d; θ designates the angle of incidence, b denotes the strip width, and d corresponds to the distance of separation of cascaded gratings in the FSS. The F_1^a layer has six nodes to accommodate the normalized input vector (θ, b, d) and its complement vector (θ^c, b^c, d^c). The F_0^b layer of Fuzzy ARTMAP has 17 inputs corresponding to 17 distinct values of the desired frequency response, obtained by sampling the desired frequency response at 17 distinct equally distributed points over its domain of definition. Consequently, the F_1^b layer has 34 nodes to accommodate the aforementioned 17-D vector and its complement. The number of nodes in F_2^a and F_2^b are chosen to be large enough to produce the compressed representations of the inputs and outputs presented at layers F_1^a and F_1^b, respectively.

The *performance phase* of the Fuzzy ARTMAP in the FSS problem differs from the standard performance phase of this network, described in the previous subsection. This is due to the fact that in the performance phase we are provided with test outputs (desired frequency responses) instead of test inputs (FSS parameter values). During this phase we apply the test output at the ART_b module of Fuzzy ARTMAP. A compressed representation of this output will be chosen in the F_2^b layer of ART_b, which in turn, through the inter-ART weight connections, will point us to a node in the F_2^a layer of ART_a. This node in F_2^a represents a compressed representation of FSS parameter values, and the top-down weights emanating from this node will provide for us the FSS parameter values that can hopefully generate the frequency response applied at the ART_b module. The test outputs used in the performance phase of the Fuzzy ARTMAP were 6 desired frequency responses pertaining to FSS cascaded gratings parameter values. The performance of the Fuzzy ARTMAP neural network is evaluated by (i) utilizing the FSS parameter values that the network provides to us, (ii) producing the frequency response (using the equations in [7] and [8]) that these parameters generate, and (iii) comparing the frequency response in step (ii) with the desired frequency response.



(a)



(b)

Figure 2. a) Geometry of two cascading gratings b) Cascading of N gratings or dielectric slabs.

Since with the Fuzzy ARTMAP neural network applied to the FSS problem a many-to-one mapping is implemented, the situation where many collections of FSS parameter values are mapped to identical or almost identical frequency responses is possible. Hence, in the performance phase of Fuzzy ARTMAP, as we present a desired frequency response at the ART_b module it is possible that the Fuzzy ARTMAP architecture, through the $F_2^a \rightarrow F_{ab}$ and $F_{ab} \leftrightarrow F_2^b$ weights, points us to more than one collection of FSS parameter values that are mapped to this desired frequency response.

A more detailed description of the training and performance phase of Fuzzy ARTMAP neural network, applied to the FSS problem, is included in Appendix A.

3. SIMULATION RESULTS

In this section, we present simulation results that demonstrate the performance of the Fuzzy ARTMAP neural network applied to the particular FSS design problem mentioned earlier.

For comparison purposes we attempted to solve the FSS design problem using the back-prop network with inversion, since this was a technique applied before in the literature. The back-prop network that we utilized consisted of one hidden layer of nodes. Despite our efforts we were unable to make this approach work satisfactorily. The back-prop network failed because in the training phase it did not satisfy the strict convergence criteria. When we loosened the convergence criteria in the training phase it converged to a solution, but during the performance phase (where the inversion algorithm was utilized) it did not perform satisfactorily (i.e., it did not converge to solutions for the input FSS parameters that produced as outputs the desired frequency responses). However, in order to obtain some back-prop results, pertaining to the FSS problem, we decided to simplify the problem and to train the back-prop on the inverse FSS problem. In particular, we trained the back-prop network with a smaller training list (i.e., the one consisting of the 66 FSS parameters corresponding to cascaded gratings). Furthermore, in the training phase the training inputs were the desired frequency responses and the training outputs were the FSS parameter values (i.e., θ, b and d) that generated these desired frequency responses. To be consistent with the training of Fuzzy ARTMAP the desired frequency responses were represented by 17 of their values obtained by sampling the desired frequency responses at 17 distinct, equally distributed over the domain of definition of a desired frequency response, points.

In the performance phase of the back-prop network, we presented as an input to the architecture one of the 6 test outputs used in the performance phase of Fuzzy ARTMAP. To evaluate the performance of the back-prop network: (i) we utilized the FSS parameter values that a desired frequency response produced at the network output, (ii) we used the FSS parameter values of step (i) to produce via the equations of references [7] and [8] the corresponding frequency response, and (iii) we compared the frequency response derived in part (ii) with the

desired frequency response. As we have emphasized above the back-prop results reported in this paper correspond to a back-prop network trained on the inverse FSS problem.

The back-prop network required 2-3 hours of training (40,000 - 60,000 training list presentations), while the Fuzzy ARTMAP neural network required only 10 seconds of training (2 - 3 training list presentations). Both approaches were implemented on a Sun-4 machine. Figures 3-8 illustrate the performance of the back-prop network and the performance of the Fuzzy ARTMAP network when they are presented with a desired frequency response from the test list. The numbers in the parentheses of these figures represent the geometric parameters of the gratings, (i.e., θ, b, d, where θ is in degrees and b and d are in terms of the interstrip distance a). A general observation that can be drawn from these figures is that both the back-prop and the Fuzzy ARTMAP networks yield satisfactory initial choices of the FSS parameter values that produce the desired frequency response; note that although the initial choices of the FSS parameter values obtained by these approaches are not close to the FSS parameter values that produced the desired frequency response these initial parameter choices generate a frequency response that is reasonably close to the desired frequency response.

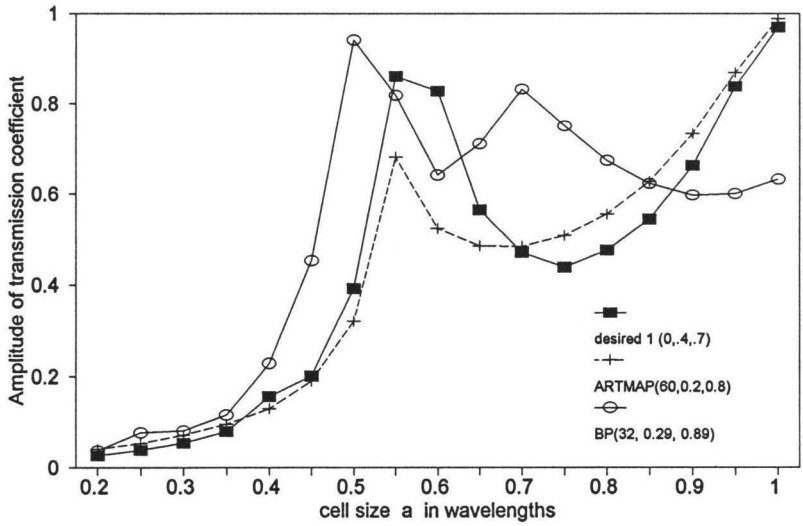


Figure 3. Transmission coefficient for two cascaded gratings. The desired response is compared with the obtained responses from the back-prop and ARTMAP neural nets for various geometric parameters (θ, b, and d).

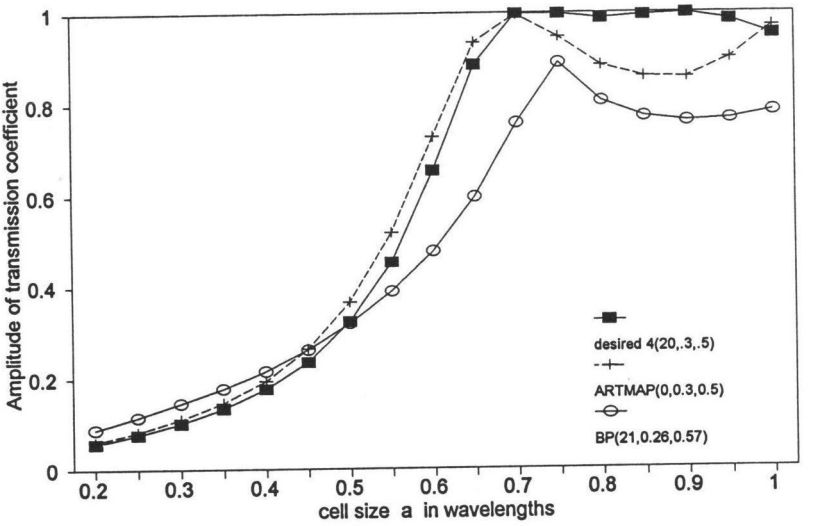


Figure 4. Transmission coefficient for two cascaded gratings. The desired response is compared with the obtained responses from the back-prop and ARTMAP neural nets for various geometric parameters (θ , b, and d).

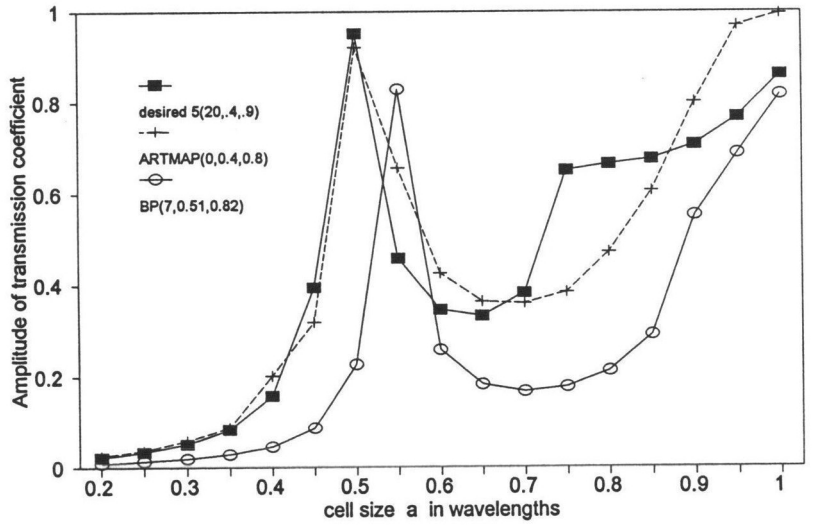


Figure 5. Transmission coefficient for two cascaded gratings. The desired response is compared with the obtained responses from the back-prop and ARTMAP neural nets for various geometric parameters (θ , b, and d).

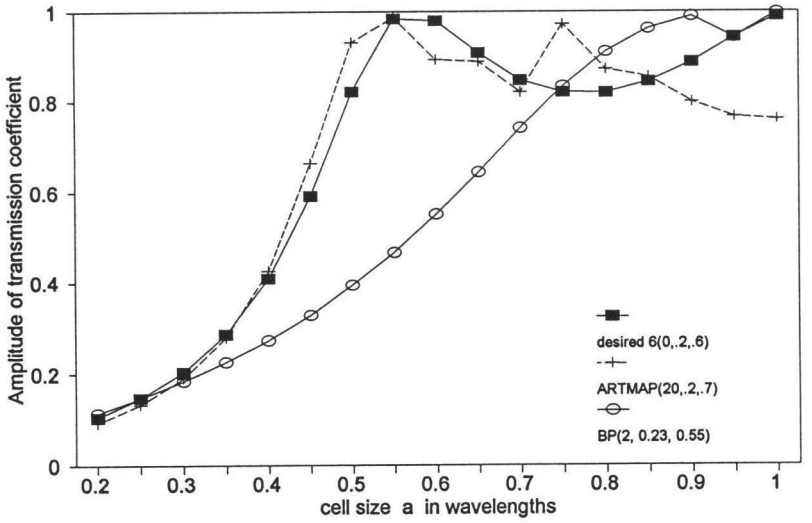


Figure 6. Transmission coefficient for two cascaded gratings. The desired response is compared with the obtained responses from the backprop and ARTMAP neural nets for various geometric parameters (θ , b, and d).

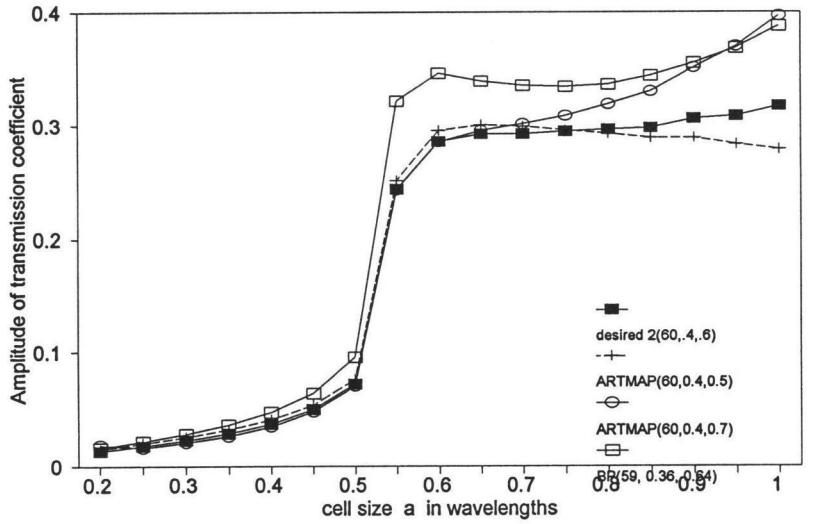


Figure 7. Transmission coefficient for two cascaded gratings. The desired response is compared with the obtained responses from the backprop and ARTMAP neural nets for various geometric parameters (θ , b, and d).

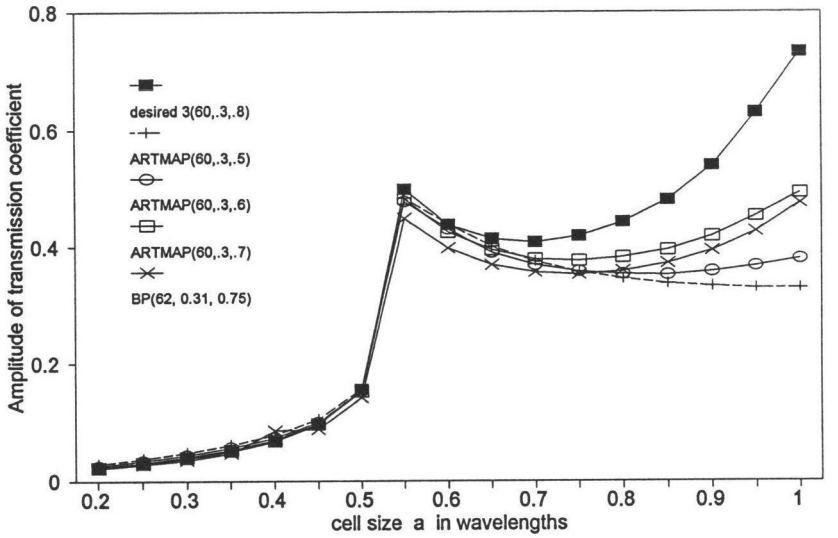


Figure 8. Transmission coefficient for two cascaded gratings. The desired response is compared with the obtained responses from the back-prop and ARTMAP neural nets for various geometric parameters (θ, b, and d).

It should also be noted that whereas the back-prop network yields only one choice of FSS parameter values, the Fuzzy ARTMAP network yields, at times, more than one set of FSS parameter values. In particular, in Figures 3-6, only one set of FSS parameter values is obtained by the back-prop and Fuzzy ARTMAP networks and the corresponding desired responses produced are compared against the desired frequency response. On the other hand, in Figures 7 and 8 Fuzzy ARTMAP produces 2 and three different sets of FSS parameter values, respectively. It is advantageous to obtain more than one prediction of the FSS parameter values that generate a desired frequency response, because then you can identify appropriate ranges for the FSS parameter values; subsequently, you can isolate your search over these FSS parameter ranges in order to produce a corresponding frequency response that is the best match of the desired frequency response.

4. CONCLUSION

The Fuzzy ARTMAP neural network was utilized to synthesize a given frequency response for cascaded gratings FSS and for cascaded gratings with dielectric slabs FSS. Furthermore, for comparison purposes the back-prop network was used to synthesize a desired frequency response only for cascaded gratings FSS.

Both approaches demonstrated that after training of the neural network is complete they can provide reasonable predictions for the set of FSS parameter values that generate a desired frequency response. However, the Fuzzy ARTMAP approach has a number of attractive features compared with the back-prop approach. *First*, the training of Fuzzy ARTMAP is guaranteed to converge to a solution. On the other hand, the back-prop approach cannot guarantee convergence to a solution. *Second*, training the Fuzzy ARTMAP neural network is much faster than training the back-prop neural network; usually Fuzzy ARTMAP converges to a solution after 3-4 presentations of the training list, while back-prop requires at times thousands of presentations of the training list. *Third*, retraining of the Fuzzy ARTMAP neural network is much less computationally intensive than retraining the back-prop network when new data are added to the training list. *Fourth*, the Fuzzy ARTMAP neural network can produce more than one set of FSS parameter values that can generate a desired frequency response. These sets of FSS parameter values can then be used by the designer to identify a range of FSS parameter values over which the search will be isolated to come up with the best possible FSS parameter set. On the other hand, the back-prop network will only produce one set of FSS parameter values for a given desired frequency response. One advantage of the back-prop network compared with the Fuzzy ARTMAP network is that it uses fewer network nodes to solve the FSS design problem. In particular, the back-prop network utilized in this paper used 5 or 6 hidden layer nodes, while the Fuzzy ARTMAP neural network used between 42 and 91 F_2^a nodes, and between 25 and 87 F_2^b nodes, depending on the value of the Fuzzy ARTMAP parameter ρ_b; smaller ρ_b values produced fewer nodes in F_2^a and F_2^b.

Based on the favorable characteristics that the Fuzzy ARTMAP neural network demonstrated in the aforementioned FSS synthesis design problem, we anticipate that this network will be capable of solving the FSS synthesis problem for frequency selective surfaces of more complicated geometries.

APPENDIX A

A.1 Terminology – Notation

Layers F_0^a, F_1^a, F_2^a, F_{ab}, F_2^b, F_1^b, and F_0^b in the Fuzzy ARTMAP neural network have M_a, $2M_a$, N_a, N_b, N_b, $2M_b$, and M_b nodes, respectively. We use the index j to designate nodes in the F_1^a layer, the index k to designate nodes in the F_2^a layer, the index l to designate nodes in the MAP-layer F_{ab} and the F_2^b layer, and the index m to designate nodes in the F_1^b layer. For the detailed description of the training and performance phase of the Fuzzy ARTMAP neural

network only the top-down weights in ART_a and ART_b, as well as the weights from F_2^a to F_{ab} need to be defined. The bottom-up weights in ART_a and ART_b play an important role only if a hardware implementation of the Fuzzy ARTMAP neural network is attempted. Hence, $\mathbf{w}_k^a = (w_{k1}^a, \ldots, w_{k(2M_a)}^a)$ is the vector of top-down weights emanating from node k in F_2^a, $\mathbf{w}_l^b = (w_{l1}^b, \ldots, w_{l(2M_b)}^b)$ is the vector of top-down weights emanating from node l in F_2^b, and $\mathbf{w}_k = (w_{k1}, \ldots, w_{kN_b})$ is the vector of weights emanating from node k in F_2^a and converging to the nodes of the MAP-layer F_{ab}. Based on the discussion in Section 2 we identify by $\mathbf{I}$ and $\mathbf{O}$ the inputs and outputs applied at layers F_1^a and F_1^b of the Fuzzy ARTMAP neural network, respectively. In particular,

$$\mathbf{I} = (\mathbf{i}, \mathbf{i}^c) = (i_1, \ldots, i_{M_a}, i_1^c, \ldots, i_{M_a}^c) \tag{A.1}$$

$$\mathbf{O} = (\mathbf{o}, \mathbf{o}^c) = (o_1, \ldots, o_{M_b}, o_1^c, \ldots, o_{M_a}^c) \tag{A.2}$$

where

$$i_j^c = 1 - i_j \qquad ; 1 \leq j \leq M_a \tag{A.3}$$

$$o_m^c = 1 - o_m \qquad ; 1 \leq m \leq M_b \tag{A.4}$$

Consider now the case where an input $\mathbf{I}$ with corresponding output $\mathbf{O}$ is presented at the F_1^a layer, and assume that node K in F_2^a is chosen to represent the input $\mathbf{I}$. Furthermore, suppose that $w_{Kl} = 1$ for all l such that $1 \leq l \leq N_b$. Then, we say that *node K has no prediction.* On the other hand, if $w_{KL} = 1$, $w_{Kl} = 0$ for $l \neq L$, and node L in F_2^b is the compressed representation for the output $\mathbf{O}$, we say that *node K confirms $\mathbf{I}$'s prediction.* Finally, if $w_{KL} = 1$, $w_{Kl} = 0$ for $l \neq L$, and node L in F_2^b is the compressed representation for a different output $\hat{\mathbf{O}}$ (i.e., $\hat{\mathbf{O}} \neq \mathbf{O}$), then we say that node K *disconfirms $\mathbf{I}$'s prediction.*

The network parameters in the Fuzzy ARTMAP neural network are: M_u^a, M_u^b, β_a, β_b, ρ_a, ρ_b, ρ. The network parameters M_u^a, β_a and M_u^b, β_b have an effect on the nodes chosen in F_2^a and F_2^b to represent the input $\mathbf{I}$ and the output $\mathbf{O}$, respectively. Specifically, when an input $\mathbf{I}$ is applied at the F_1^a layer of the ART_a module it produces an input $T_k^a(\mathbf{I})$ at node k in the F_2^a layer. This input is given by the following equation:

$$T_k^a(\mathbf{I}) = \begin{cases} \alpha_k^a |\mathbf{I}| & \text{if } k \text{ is uncommitted} \\ \frac{|\mathbf{I} \wedge \mathbf{w}_k^a|}{\beta_a + |\mathbf{w}_k^a|} & \text{if } k \text{ is not uncommitted} \end{cases} \tag{A.5}$$

where a node k in F_2^a is called *uncommitted* it its top-down weights have not been modified yet. In the above equation, an operator is used designated by the symbol $\wedge$. Assuming that $\mathbf{z}^1$ and $\mathbf{z}^2$ define two vectors of the same dimensionality, the symbol $\mathbf{z}^1 \wedge \mathbf{z}^2$ defines a vector whose r-th component, $(\mathbf{z}^1 \wedge \mathbf{z}^2)_r$, is the minimum of the r-th component of $\mathbf{z}^1$, and the r-th component of $\mathbf{z}^2$. Furthermore, in

the above equation, if $\mathbf{z}$ defines a vector of arbitrary dimensionality the symbol $|\mathbf{z}|$ designates the size of this vector; the size of a vector is defined to be the sum of its components. Finally, in the above equation, the α_k^a's are chosen according to the following rules:

$$\alpha_1^a < \ldots < \alpha_{N_a}^a \tag{A.6}$$

$$\alpha_k^a < \frac{1}{\beta_a + M_u^a} \quad ;1 \leq k \leq N_a \tag{A.7}$$

The ART_a module chooses the node in F_2^a that maximizes $T_k^a(\mathbf{I})$. That is, node K in F_2^a is chosen to represent the input $\mathbf{I}$ if $T_K^a(\mathbf{I}) = \max_k\{T_k^a(\mathbf{I})\}$. Similar statements are valid regarding the selection of nodes chosen in F_2^b to represent the outputs $\mathbf{O}$ presented at the ART_b module. In particular, when an output $\mathbf{O}$ is applied at the F_1^b layer of the ART_b module it produces an input $T_l^b(\mathbf{O})$ at node l in the F_2^b layer. This input is given by the following equation:

$$T_l^b(\mathbf{O}) = \begin{cases} \alpha_l^b|\mathbf{O}| & \text{if } l \text{ is uncommitted} \\ \frac{|\mathbf{O} \wedge \mathbf{w}_l^b|}{\beta_b + |\mathbf{w}_l^b|} & \text{if } l \text{ is not uncommitted} \end{cases} \tag{A.8}$$

where a node l in F_2^b is called *uncommitted* it its top-down weights have not been modified yet. In the above equation the α_l^b's are chosen according to the following rules:

$$\alpha_1^b < \ldots < \alpha_{N_b}^b \tag{A.9}$$

$$\alpha_l^b < \frac{1}{\beta_b + M_u^b} \quad ;1 \leq k \leq N_b \tag{A.10}$$

The ART_b module chooses the node in F_2^b that maximizes $T_l^b(\mathbf{O})$. That is, node L in F_2^b is chosen to represent the input $\mathbf{O}$ if $T_L^b(\mathbf{O}) = \max_l\{T_l^b(\mathbf{O})\}$.

The network parameters ρ_a, ρ_b and ρ designate the vigilance parameters in the ART_a, ART_b and inter-ART modules, respectively. It is worth pointing out that the vigilance parameter in the ART_a module is allowed to increase during training. In particular, ρ_a starts from a baseline value equal to $\bar{\rho}_a$ and if an incorrect map is established between an input $\mathbf{I}$ and an output $\mathbf{O}$, then ρ_a is increased above its baseline value of $\bar{\rho}_a$. The vigilance parameters in ART_a and ART_b control the coarseness of the compressed representations established at F_2^a and F_2^b. Specifically, small ρ_a, ρ_b values result in nodes at F_2^a and F_2^b that represent many inputs, outputs, respectively. The reverse situation is true when ρ_a and ρ_b are chosen to be large. Actually, in most problems, the coarseness of the representations is controlled only by the ρ_b parameter (this is the case in the FSS problem). The value of the vigilance parameter ρ_a plays another role, which is described in detail in [9]. The vigilance parameter ρ makes sure that

the desired mappings between inputs and outputs are established, and for the purposes of this paper it can be chosen arbitrarily over its domain of definition.

The parameter M_u^a may assume values in the interval $[2M_a, \infty)$, while M_u^b may assume values in the interval $[2M_b, \infty)$. The parameters β_a and β_b may take values in the interval $(0, \infty)$. The baseline vigilance parameter $\bar{\rho}_a$ may assume values in the interval $[0, 1]$; which implies that ρ_a takes values in the interval $[\bar{\rho}_a, 1]$. The vigilance parameter ρ_b and the inter-ART vigilance parameter ρ take values in the interval $(0, 1]$.

A.2 Detailed Training Phase of Fuzzy ARTMAP

As we mentioned in the main body of the paper the scenario under which the Fuzzy ARTMAP neural network is trained is as follows: We have a collection of input output pairs designated as $(\mathbf{I}^1, \mathbf{O}^1), (\mathbf{I}^2, \mathbf{O}^2), \ldots (\mathbf{I}^P, \mathbf{O}^P)$ that we refer to as the training list. We want to train the Fuzzy ARTMAP to learn the following mapping: $\mathbf{I}^1$ to $\mathbf{O}^1$, $\mathbf{I}^2$ to $\mathbf{O}^2$, and eventually $\mathbf{I}^P$ to $\mathbf{O}^P$. To do so, we present the input list repeatedly to the Fuzzy ARTMAP architecture, until the architecture learns all the desired mappings.

For the FSS design problem an input $\mathbf{I}$ *corresponds to the normalized and then complement coded version of the vector, whose components are the FSS parameter values* θ, b, d. *The output* $\mathbf{O}$ *corresponds to the normalized and then complement coded 17-D vector, whose components are 17 values of a desired frequency response obtained by sampling a frequency response at 17 equally distributed points over its domain of definition. Hence,* $M_a = 3$, $M_b = 17$, *and we choose* $N_a = 100$ *and* $N_b = 100$.

The detailed steps of the training phase of Fuzzy ARTMAP are listed below. In the description of the detailed steps there is a prevalent operator, designated by the symbol $\wedge$. This operator was discussed in the Preliminaries subsection of the Appendix. Another operator that shows up in the description of the training phase of Fuzzy ARTMAP is the operator, designated by the symbol $\cap$. Assuming now that $\mathbf{z}^1$ and $\mathbf{z}^2$ define two binary vectors (i.e., vectors with component values either 0 or 1) of the same dimensionality, the symbol $\mathbf{z}^1 \cap \mathbf{z}^2$ defines a binary vector, whose r-th component $(\mathbf{z}^1 \wedge \mathbf{z}^2)_r$ is the bitwise AND operation applied on the r-th component of $\mathbf{z}^1$ and on the r-th component of $\mathbf{z}^2$. Finally, the notations $\mathbf{w}(old)$ and $\mathbf{w}(new)$ are used to designate values of the weight $\mathbf{w}$ prior to and after the presentation of an input output pair from the training list, respectively.

1. Choose a value for the vigilance parameters $\bar{\rho}_a$, ρ_b, ρ, and the learning parameters β_a and β_b. *For the FSS design problem we chose* $\bar{\rho}_a = 0$, $\rho = 1$, ρ_b *any value in the interval* $[0.8\ \ 0.95]$, $\beta_a = 1$, *and* $\beta_b = 1$. *Initially,* $\rho_a = \bar{\rho}_a = 0$. The weights are initialized as follows:

$$w_{kj}^a(0) = 1 \quad w_{lm}^b(0) = 1 \quad w_{kl}(0) = 1 \tag{A.11}$$

2. Choose a pattern $\mathbf{I}^p$ from the input list and apply it to the F_1^a layer of the ART$_a$ architecture. The input $T_k^a(\mathbf{I}^p)$ from the F_1^a layer to the kth F_2^a node obeys the equation:

$$T_k^a(\mathbf{I}^p) = \begin{cases} \alpha_k^a \, |\mathbf{I}^p| & \text{if } k \text{ is uncommitted} \\ |\mathbf{I}^p \wedge \mathbf{w}_k^a(old)|/(\beta_a + |\mathbf{w}_k^a(old)|) & \text{if } k \text{ is not uncommitted} \end{cases} \tag{A.12}$$

where the α_k^a's in the above equation are chosen according to the rules specified in equations (A.6) and (A.7). *For the FSS design problem* $M_u^a = 2M_a = 6$.

3. The initial choice at F_2^a is a node with index K satisfying

$$T_K^a(\mathbf{I}^p) = \max_k \{T_k^a(\mathbf{I}^p)\} \tag{A.13}$$

If more than one node is maximal, choose the node with the lowest index.

4. Check of the vigilance criterion: If node K satisfies the condition

$$\frac{|\mathbf{I}^p \wedge \mathbf{w}_K^a(old)|}{|\mathbf{I}^p|} \geq \rho_a \tag{A.14}$$

then node K is chosen to represent the input pattern $\mathbf{I}^p$ and we move to step 5. If node K violates the above condition then node K is disqualified, and we move back to step 3 to search for another node in the F_2^a layer to represent $\mathbf{I}^p$; node K is disqualified from the search by setting $T_K^a = -1$ for as long as the input pattern $\mathbf{I}^p$ persists at the network input.

5. If the desired output $\mathbf{O}^p$ has already been presented to the F_1^b layer of ART_b and a node L in F_2^b has been chosen to represent this output we move to step 9 to check the match tracking criterion.
If the desired output $\mathbf{O}^p$ has not been presented to the ART_b architecture we proceed as follows: Present the desired output $\mathbf{O}^p$, corresponding to the input $\mathbf{I}^p$, to the F_1^b layer of the ART_b architecture. Calculate the bottom-up inputs $T_l^b(\mathbf{O}^p)$ from the F_1^b layer to the l-th F_2^b node according to the equation:

$$T_l^b(\mathbf{O}^p) = \begin{cases} \alpha_l^b \, |\mathbf{O}^p| & \text{if } l \text{ is uncommitted} \\ |\mathbf{O}^p \wedge \mathbf{w}_l^b(old)|/(\beta_b + |\mathbf{w}_l^b(old)|) & \text{if } l \text{ is not uncommitted} \end{cases} \tag{A.15}$$

where the α_l^b's in the above equation are chosen according to the rules specified in equations (A.9) and (A.10). *For the FSS design problem* $M_u^b = 2M_b = 34$.
Now proceed with step 6.

6. If node K in the F_2^a layer, chosen in the step 4, predicts a node L in the F_2^b layer then this node will be activated first in the F_2^b layer and we move to step 8 to check for its appropriateness to represent the desired output $\mathbf{O}^p$. If node K in the F_2^a layer, chosen in the step 4, does not have a prediction then we continue with step 7.

7. The initial choice at F_2^b is a node with index L satisfying

$$T_L^b(\mathbf{O}^p) = \max_l \{T_l^b(\mathbf{O}^p)\} \tag{A.16}$$

If more than one node is maximal, choose the node with the lowest index.

8. Check of the vigilance criterion: If node L satisfies the condition

$$\frac{|\mathbf{O}^p \wedge \mathbf{w}_L^b(old)|}{|\mathbf{O}^p|} \geq \rho_b \tag{A.17}$$

Then node L is chosen to represent the input $\mathbf{O}^p$ and we continue with step 9. If node L violates the above condition then node L is disqualified, and we move back to step 7 to search for another node in the F_2^b layer to represent $\mathbf{O}^p$; node L is disqualified from the search by setting $T_L^b = -1$ for as long as the input $\mathbf{I}^p$ and the desired output $\mathbf{O}^p$ persist at the network inputs.

9. At this point node K in F_2^a has been chosen to represent the input $\mathbf{I}^p$ and node L in F_2^b has been chosen to represent the desired output $\mathbf{O}^p$. The F_{ab} output vector $\mathbf{x}$ is equal to $\mathbf{y}^b \cap \mathbf{w}_K$, where $\mathbf{y}^b$ is the output activity vector across the F_2^b layer and $\mathbf{w}_K$ is the vector of weights emanating from the F_2^a layer node K and converging to the nodes of the MAP-layer F_{ab}. If a node in F_2^b is chosen then its output activity is 1, and if it is not chosen its output activity is 0. Consequently, the vectors $\mathbf{x}$, $\mathbf{y}_b$ and $\mathbf{w}_K$ are binary vectors. We now check to see whether the match tracking criterion is satisfied. If

$$|\mathbf{x}| \geq \rho\, |\mathbf{y}^b| \tag{A.18}$$

then we have achieved the desired mapping and we continue to step 10 to learn the input, output pair presented. If

$$|\mathbf{x}| < \rho\, |\mathbf{y}^b| \tag{A.19}$$

then the mapping between K and L is not the desired one. In this case the vigilance parameter ρ_a is increased slightly above the value $|\mathbf{I}^p \wedge \mathbf{w}_K^a(old)|/|\mathbf{I}^p|$; this leads to immediate disqualification of node K in ART_a, and we move to step 3 with the new vigilance parameter for the selection of another node in F_2^a that will achieve the desired mapping; prior to moving to step 3 we set $T_K^a = -1$ for as long as the input $\mathbf{I}^p$ and the desired output $\mathbf{O}^p$ persist at the network inputs.

10. If during $\mathbf{I}^p$'s and $\mathbf{O}^p$'s presentation, the desired mapping has been achieved between node K in ART_a and node L in ART_b, the adaptable weights emanating from these nodes are modified according to the equations:

$$\mathbf{w}_K^a(new) = \mathbf{I}^p \wedge \mathbf{w}_K^a(old) \tag{A.20}$$

$$\mathbf{w}_L^b(new) = \mathbf{O}^p \wedge \mathbf{w}_L^b(old) \tag{A.21}$$

$$w_{KL}(new) = 1 \quad \text{and} \quad w_{Kl}(new) = 0 \quad \text{for} \quad l \neq L \tag{A.22}$$

Note that all the other weights remain unchanged.

11. If you have not exhausted all the input pairs then go to step 2 and present the next input pair in line (i.e., present the input of the next input pair to ART_a

and the output of the next input pair to ART_b). If you have exhausted all the pairs from your training list, but at least one adaptable weight changed in the architecture during the previous list presentation, then move to step 2 and present the first input pair in your list (e.g., pair $(\mathbf{I}^1, \mathbf{O}^1)$). Before moving to step 2 reset the ρ_a value to $\bar{\rho}_a$, and set the vectors $\mathbf{x}$, $\mathbf{y}^b$, and all the bottom-up inputs T_k^a and T_l^b equal to zero. If you have exhausted all the input pairs in the training list and and no adaptable weight was changed during your last training list presentation, then training is complete.

A.3 Detailed Performance Phase of Fuzzy ARTMAP

In the performance phase of the Fuzzy ARTMAP neural network we have a collection of test outputs that we present to the ART_b module. *In the FSS design problem the test outputs correspond to desired frequency responses, and our objective is to use the already trained Fuzzy ARTMAP neural network to derive the FSS parameter values that can generate these responses.*

The steps of the performance phase are described below.

1. Choose a value for the vigilance parameters ρ_b, smaller than the ρ_b which you chose for the training phase by 0.1. All the other Fuzzy ARTMAP parameters are chosen as they were chosen in the training phase.
2. Present the test output $\mathbf{O}$ at the ART_b module. The bottom-up input $T_l^b(\mathbf{O})$ from the F_1^b layer to the lth F_2^b node obeys the equation:

$$T_l^b(\mathbf{O}) = \begin{cases} \alpha_l^b\,|\mathbf{O}| & \text{if } l \text{ is uncommitted} \\ |\mathbf{O} \wedge \mathbf{w}_l^b(old)|/(\beta_b + |\mathbf{w}_l^b|) & \text{if } l \text{ is not uncommitted} \end{cases} \tag{A.23}$$

   where the α_l^b's in the above equation are chosen according to the rules specified in equations (A.9) and (A.10).
3. The initial choice at F_2^b is a node with index L satisfying

$$T_L^b(\mathbf{O}) = \max_l\{T_l^b(\mathbf{O})\} \tag{A.24}$$

   If more than one node is maximal, choose the node with the lowest index.
4. Check of the vigilance criterion: If node L satisfies the condition

$$\frac{|\mathbf{O} \wedge \mathbf{w}_L^b|}{|\mathbf{O}|} \geq \rho_b \tag{A.25}$$

   then node L is chosen to represent the output $\mathbf{O}$ and we continue with Step 5. If node L violates the above condition then node L is disqualified, and we move back to Step 3 to search for another node in the F_2^b layer to represent $\mathbf{O}$; node L is disqualified from the search by setting $T_L^b = -1$ for as long as the output $\mathbf{O}$ persists at the network inputs.

5. A node L has been chosen in F_2^b to represent the output O.

   (a) If node L is not uncommitted, then through the connections from the F_2^a to the F_{ab} layer we identify all the nodes K in F_2^a that have as a prediction node L in F_2^b, that is, all nodes K for which

$$w_{KL} = 1 \quad \text{and} \quad w_{Kl} = 0 \quad \text{for} \quad l \neq L \tag{A.26}$$

   A node K in F_2^a that satisfies the above equations will, through its top-down ART_a connections, identify approximate values for the FSS parameters that generate the frequency response O presented at ART_b.

   (b) If node L is uncommitted, then no node in F_2^a will have node L in F_2^b as its prediction, or equivalently, the Fuzzy ARTMAP neural network cannot provide an estimate for the FSS parameter values that would generate the frequency response presented at the ART_b module. In this case, we can say that the frequency response is too novel for the architecture to risk an estimate for the FSS parameter values that would generate this response.

ACKNOWLEDGMENTS

This research was supported in part by the Enterprise Florida Innovation Partnership and in part by Harris Corporation.

The Editor thanks M. Barclay, D. Jenn, and M. Bailey for reviewing the paper.

REFERENCES

1. Davis, D. T., C. H. Chan, and J. N. Hwang, "Frequency selective surface design using neural networks inversion based on parameterized representation," *IEEE Symposium on Antennas and Propagation*, London, Canada, 200–203, June 1991.
2. Hwang, J. N., C. H. Chan, and R. J. Marks II, "Frequency selective surface design based on iterative inversion of neural networks," *International Joint Conference on Neural Networks*, San Diego, CA, 139–144, June 1990.
3. Rumelhart, D. E., J. L. McClelland, and the PDP Research Group, "Parallel Distributed Processing (Explorations in the Microstructure of Cognition)," Vol. 1, *Foundations,* MIT Press, 1986.
4. Linden, A., and J. Kindermann, "Inversion of multilayer nets," *International Joint Conference on Neural networks*, Washington D.C., 11425–11430, July 1989.
5. Carpenter, G. A., 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 Transactions on Neural Networks*, Vol. 3, 678–713, September 1992.
6. Carpenter, G. A., S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization on analog patterns by an adaptive resonance system," *Neural Networks*, Vol. 4, 759–771, 1991.
7. Christodoulou, C. G., and J. F. Kauffman, "On the scattering of electromagnetic waves from infinite rectangular conducting grids," *IEEE Transactions on Antennas and Propagation*, Vol. 34, 144–154, February 1986.

8. Christodoulou, C. G., P. Kwan, R. Middelveen, and P. F. Wahid, "Scattering from stacked gratings and dielectrics for various angles of wave incidence," *IEEE Transactions on Antennas and Propagation*, Vol. 36, 1435–1442, October 1992.
9. Carpenter, G. A., S. Grossberg, and J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, Vol. 4, 565–588, 1991.
10 Christodoulou, C. G., J. Huang, M. Georgiopoulos, and J. J. Liou, "Design of gratings and frequency selective surfaces using Fuzzy ARTMAP neural networks," *SPIE Proceedings*, Vol. 2243, Orlando, FL, 571–581, April 1994.
11 Huang, J., "Theoretical analysis of ART neural networks and their applications in frequency selective surfaces", *Ph.D. Dissertation*, University of Central Florida, May 1994.

Christos G. Christodoulou received the B.Sc. degree in Physics and Mathematics from the American University of Cairo in 1979, and the M.S. and Ph.D. degrees in Electrical Engineering from North Carolina State University, Raleigh, in 1981 and 1985, respectively. He has been with the University of Central Florida, Orlando, since 1985, where he serves as an Associate Professor and Associate Chair. He is a senior member of IEEE and a member of URSI (Commission B). His research interests lie in the areas of numerical techniques in electromagnetics, neural network applications in electromagnetics, optical antennas and frequency selective surfaces.

Michael Georgiopoulos received the Diploma in Electrical Engineering from the National Technical University of Athens, Greece, in 1981, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Connecticut, Storrs, in 1983 and 1986, respectively. In 1987, he joined the University of Central Florida, Orlando, where he is now an Associate Professor in the Electrical and Computer Engineering Department. His research interests lie in the areas of neural networks, and pattern recognition. Dr. Georgiopoulos is a member of the Technical Chamber of Greece, a member of IEEE, and a member of the International Neural Network Society.

Juxin Huang: received the B.S. (1982) and M.S. (1985) degrees from the Electrical Engineering Department of Beijing University of Posts and Telecommunications, P.R. China, and the Ph.D. from the Electrical Engineering Department of the University of Central Florida (1994). She joined the INTEL Corporation in the Summer of 1994, where she is currently working with the Department "Product, Test, Yield". Her research interests lie in the areas of neural networks, signal processing and nonlinear system modeling. Dr. Huang is a member of the IEEE and the International Neural Network Society.

Juin J. Liou received the B.S. (honors), M.S., and Ph.D. degrees in Electrical Engineering from the University of Florida, Gainesville, in 1982, 1983, and 1987, respectively. In 1987, he joined the Department of Electrical Engineering at the University of Central Florida, Orlando, where he is now an Associate Professor. Dr. Liou has published a text book with Artech House entitled *Advanced Semiconductor Devices Physics and Modeling,* and more than 170 technical papers in refereed journals and international and national conference proceedings. He received 6 different awards on research from the University of Central Florida in the past seven years. His current research interests are in semiconductor device modeling, simulation, and computer-aided circuit design.