专 题 汇 编

IJCNN' 无导师学习

FUZZY ARTMAP: AN ADAPTIVE RESONANCE ARCHITECTURE FOR INCREMENTAL LEARNING OF ANALOG MAPS

Gail A. Carpenter†, Stephen Grossberg‡, Natalya Markuzon§,
John H. Reynolds¶, and David B. Rosen¶

Center for Adaptive Systems
and
Department of Cognitive and Neural Systems
Boston University
111 Cummington Street
Boston, Massachusetts 02215 USA

Abstract

A neural network architecture is introduced for incremental supervised learning of recognition categories and multidimensional maps in response to arbitrary sequences of analog or binary input vectors. The architecture, called Fuzzy ARTMAP, achieves a synthesis of fuzzy logic and Adaptive Resonance Theory (ART) neural networks. Fuzzy ARTMAP realizes a new Minimax Learning Rule that conjointly minimizes predictive error and maximizes code compression, or generalization. This is achieved by a match tracking process that increases the ART vigilance parameter by the minimum amount needed to correct a predictive error. As a result, the system automatically learns a minimal number of recognition categories or "hidden units," to meet accuracy criteria. A normalization procedure called complement coding leads to a symmetric theory in which the MIN operator ($\wedge$) and the MAX operator ($\vee$) of fuzzy logic play complementary roles. Improved prediction is achieved by training the system several times using different orderings of the input set, then voting. This voting strategy can also be used to assign probability estimates to competing predictions given small, noisy, or incomplete training sets. Simulations illustrate Fuzzy ARTMAP performance as compared to benchmark back propagation and genetic algorithm systems. These simulations include (i) finding points inside versus outside a circle; (ii) learning to tell two spirals apart; (iii) incremental approximation of a piecewise continuous function; (iv) a letter recognition database; and (v) a medical database.

† Supported in part by British Petroleum (89-A-1204), DARPA (AFOSR 90-0083), the National Science Foundation (NSF IRI 90-00530) and the Office of Naval Research (ONR N00014-91-J-4100).

‡ Supported in part by the Air Force Office of Scientific Research (AFOSR 90-0175), DARPA (AFOSR 90-0083) and the Office of Naval Research (ONR N00014-91-J-4100).

§ Supported in part by National Science Foundation (NSF IRI 90-00530) and British Petroleum (89-A-1204).

¶ Supported in part by DARPA (AFOSR 90-0083).

Acknowledgements: The authors wish to thank Cynthia E. Bradford for her valuable assistance in the preparation of the manuscript.

0-7803-0559-0/92 $3.00 ©1992 IEEE

Fuzzy ARTMAP

ARTMAP is a class of neural network architectures that perform incremental supervised learning of recognition categories and multidimensional maps in response to input vectors presented in arbitrary order. The first ARTMAP system [3] was used to classify binary vectors. This article describes a more general ARTMAP system that learns to classify analog as well as binary vectors [2]. This generalization is accomplished by replacing the ART 1 [1] modules of the binary ARTMAP system with Fuzzy ART [4] modules. Where ART 1 dynamics are described in terms of set-theoretic operations, Fuzzy ART dynamics are described in terms of fuzzy set-theoretic operations [9]. Hence the new system is called Fuzzy ARTMAP. Also introduced is an ARTMAP *voting strategy*. This voting strategy is based on the observation that ARTMAP fast learning typically leads to different adaptive weights and recognition categories for different orderings of a given training set, even when overall predictive accuracy of all simulations is similar. The different category structures cause the set of test items where errors occur to vary from one simulation to the next. The voting strategy uses an ARTMAP system that is trained several times on input sets with different orderings. The final prediction for a given test set item is the one made by the largest number of simulations. Since the set of items making erroneous predictions varies from one simulation to the next, voting cancels many of the errors. Further, the voting strategy can be used to assign probability estimates to competing predictions given small, noisy, or incomplete training sets.

Simulations have illustrated Fuzzy ARTMAP performance as compared to benchmark back propagation and genetic algorithm systems. In all cases, Fuzzy ARTMAP simulations lead to favorable levels of learned predictive accuracy, speed, and code compression in both on-line and off-line settings. Fuzzy ARTMAP is also easy to use. It has a small number of parameters, requires no problem-specific system crafting or choice of initial weight values, and does not get trapped in local minima.

ARTMAP Dynamics

Each ARTMAP system includes a pair of Adaptive Resonance Theory modules (ART_a and ART_b) that create stable recognition categories in response to arbitrary sequences of input patterns (Figure 1). During supervised learning, the ART_a module receives a stream $\{\mathbf{a}^{(p)}\}$ of input patterns and ART_b receives a stream $\{\mathbf{b}^{(p)}\}$ of input patterns, where $\mathbf{b}^{(p)}$ is the correct prediction given $\mathbf{a}^{(p)}$. These modules are linked by an associative learning network and an internal controller that ensures autonomous system operation in real time. The controller is designed to create the minimal number of ART_a recognition categories, or "hidden units," needed to meet accuracy criteria. It does this by realizing a Minimax Learning Rule that enables an ARTMAP system to learn quickly, efficiently, and accurately as it conjointly *minimizes* predictive error and *maximizes* predictive generalization. This scheme automatically links predictive success to category size on a trial-by-trial basis using only local operations. It works by increasing the vigilance parameter ρ_a of ART_a by the minimal amount needed to correct a predictive error at ART_b.

Parameter ρ_a calibrates the minimum confidence that ART_a must have in a recognition category, or hypothesis, activated by an input $\mathbf{a}^{(p)}$ in order for ART_a to accept that category, rather than search for a better one through an automatically controlled process of hypothesis testing. Lower values of ρ_a enable larger categories to form. These lower ρ_a values lead to

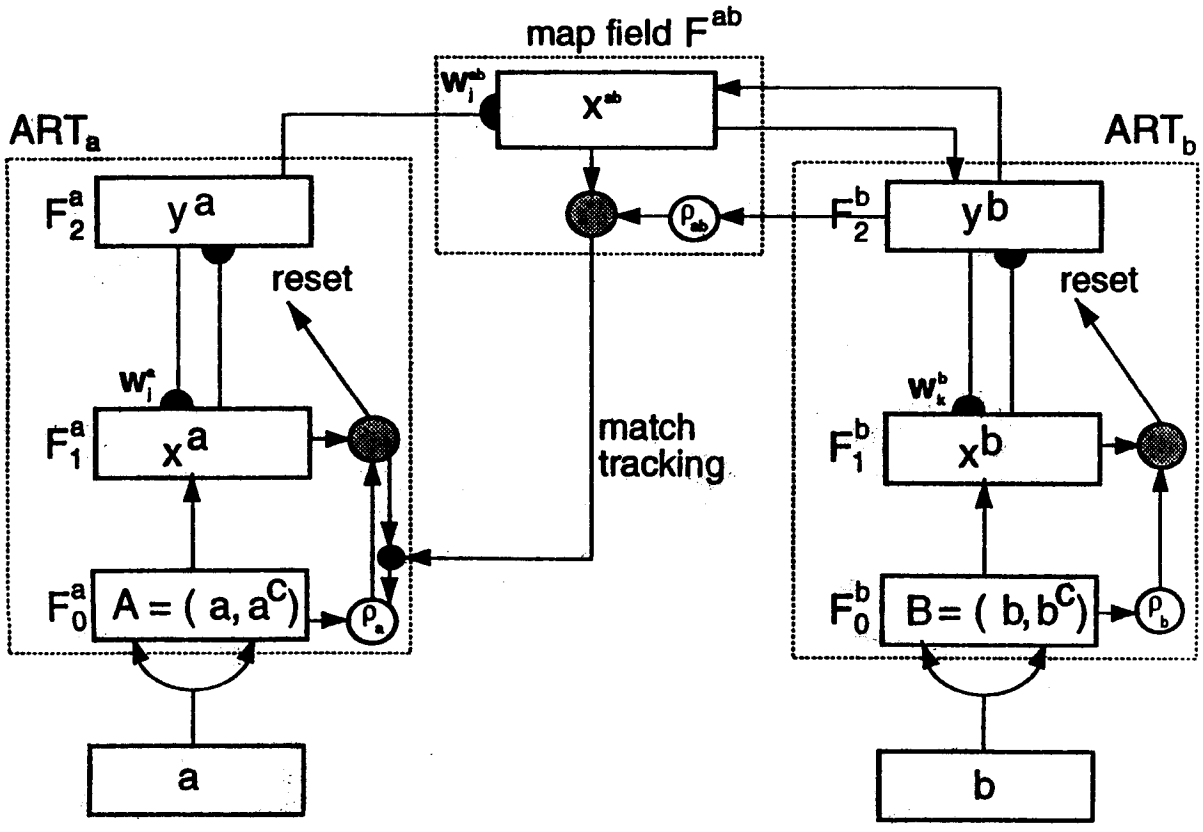

Figure 1. Fuzzy ARTMAP architecture. The ART_a complement coding preprocessor transforms the M_a-vector $\mathbf{a}$ into the $2M_a$-vector $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$ at the ART_a field F_0^a. $\mathbf{A}$ is the input vector to the ART_a field F_1^a. Similarly, the input to F_1^b is the $2M_b$-vector $(\mathbf{b}, \mathbf{b}^c)$. When a prediction by ART_a is disconfirmed at ART_b, inhibition of map field activation induces the match tracking process. Match tracking raises the ART_a vigilance (ρ_a) to just above the F_1^a to F_0^a match ratio $|\mathbf{x}^a|/|\mathbf{A}|$. This triggers an ART_a search which leads to activation of either an ART_a category that correctly predicts $\mathbf{b}$ or to a previously uncommitted ART_a category node.

broader generalization and higher code compression. A predictive failure at ART_b increases ρ_a by the minimum amount needed to trigger hypothesis testing at ART_a, using a mechanism called *match tracking* [3]. Match tracking sacrifices the minimum amount of generalization necessary to correct a predictive error. Hypothesis testing leads to the selection of a new ART_a category, which focuses attention on a new cluster of $\mathbf{a}^{(p)}$ input features that is better able to predict $\mathbf{b}^{(p)}$. Due to the combination of match tracking and fast learning, a single ARTMAP system can learn a different prediction for a rare event than for a cloud of similar frequent events in which it is embedded.

Whereas binary ARTMAP employs ART 1 systems for the ART_a and ART_b modules Fuzzy ARTMAP substitutes Fuzzy ART systems for these modules. Fuzzy ART shows how computations from fuzzy set theory can be incorporated naturally into ART systems. For example, the intersection ($\cap$) operator that describes ART 1 dynamics is replaced by the AND operator ($\wedge$) of fuzzy set theory [9] in the choice, search, and learning laws of ART 1. Especially noteworthy is the close relationship between the computation that defines fuzzy

subsethood [7] and the computation that defines category choice in ART 1. Replacing operation $\cap$ by operation $\wedge$ leads to a more powerful version of ART 1. Whereas ART 1 can learn stable categories only in response to binary input vectors, Fuzzy ART can learn stable categories in response to either analog or binary input vectors. Moreover, Fuzzy ART reduces to ART 1 in response to binary input vectors.

In Fuzzy ART, learning always converges because all adaptive weights are monotone nonincreasing. Without additional processing, this useful stability property could lead to the unattractive property of category proliferation as too many adaptive weights converge to zero. A preprocessing step, called complement coding, uses on-cell and off-cell responses to prevent category proliferation. Complement coding normalizes input vectors while preserving the amplitudes of individual feature activations. Without complement coding, an ART category memory encodes the degree to which critical features are consistently present in the training exemplars of that category. With complement coding, both the degree of absence and the degree of presence of features are represented by the category weight vector. The corresponding computations employ fuzzy OR ($\vee$, maximum) operators, as well as fuzzy AND ($\wedge$, minimum) operators.

Benchmark Simulations

In one benchmark simulation, Fuzzy ARTMAP performed the task of learning to identify which points lie inside and which lie outside a given circle [2]. On-line learning (also called incremental learning) is demonstrated, with test set accuracy increasing from 88.6% to 98.0% as the training set increased in size from 100 to 100,000 randomly chosen points. With off-line learning, the system needed from 2 to 13 epochs to learn all training set exemplars to 100% accuracy, where an epoch is defined as one cycle of training on an entire set of input exemplars. Test set accuracy then increased from 89.0% to 99.5% as the training set size increased from 100 to 100,000 points. Application of the voting strategy improved an average single-run accuracy of 90.5% on five runs to a voting accuracy of 93.9%, where each run trained on a fixed 1,000-item set for one epoch. These simulations are compared with studies by Wilensky [8] of back propagation systems. These systems used at least 5,000 epochs to reach 90% accuracy on training and testing sets. Other benchmarks include learning to tell two spirals apart and learning to approximate a continuous function.

Simulation: Letter Image Recognition

Frey and Slate [5] recently developed a benchmark machine learning task that they describe as a "difficult categorization problem" (p. 161). The task requires a system to identify an input exemplar as one of 26 capital letters A–Z. The database was derived from 20,000 unique black-and-white pixel images. The difficulty of the task is due to the wide variety of letter types represented: the twenty "fonts represent five different stroke styles (simplex, duplex, complex, and Gothic) and six different letter styles (block, script, italic, English, Italian, and German)" (p. 162). In addition each image was randomly distorted, leaving many of the characters misshapen. Sixteen numerical feature attributes were then obtained from each character image, and each attribute value was scaled to a range of 0 to 15. The resulting Letter Image Recognition file is archived in the UCI Repository of Machine Learning Databases and Domain Theories, maintained by David Aha and Patrick Murphy (ml_repository@ics.uci.edu).

TABLE 1

		% Correct Test Set Predictions	No. ART_a Categories	No. Epochs
(a)				
	Average	91.8%	786	1
	Range	91.2%–92.6%	763-805	1
	Voting	95.3%		
(b)				
	Average	93.9%	1,021	5
	Range	93.4%–94.6%	990-1,070	5
	Voting	96.0%		

Table 1. Voting strategy applied to sets of 5 Fuzzy ARTMAP simulations of the Frey Slate character recognition task, with training on 1 epoch (a) or 5 epochs (b). (a) Voting eliminated 43% of the errors, which dropped from 8.2% to 4.7%. (b) Voting eliminated 34% of the errors, which dropped from 6.1% to 4.0%.

Frey and Slate used this database to test performance of a family of classifiers based on Holland's genetic algorithms [6]. The training set consisted of 16,000 exemplars, with the remaining 4,000 exemplars used for testing. Genetic algorithm classifiers having different input representations, weight update and rule creation schemes, and system parameters were systematically compared. Training was carried out for 5 epochs, plus a sixth "verification" pass during which no new rules were created but a large number of unsatisfactory rules were discarded. In Frey and Slate's comparative study, these systems had correct prediction rates that ranged from 24.5% to 80.8% on the 4,000-item test set. The best performance (80.8%) was obtained using an integer input representation, a reward sharing weight update, an exemplar method of rule creation, and a parameter setting that allowed an unused or erroneous rule to stay in the system for a long time before being discarded. After training the optimal case, that had 80.8% performance rate, ended with 1,302 rules and 8 attributes per rule, plus over 35,000 more rules that were discarded during verification. (For purposes of comparison, a rule is somewhat analogous to an ART_a category in ARTMAP, and the number of attributes per rule is analogous to the size of ART_a category weight vectors.) Building on the results of their comparative study, Frey and Slate investigated two types of alternative algorithms, namely an accuracy-utility bidding system, that had slightly improved performance (81.6%) in the best case; and an exemplar/hybrid rule creation scheme that further improved performance, to a maximum of 82.7%, but that required the creation of over 100,000 rules prior to the verification step.

Fuzzy ARTMAP had an error rate on the letter recognition task that was consistently less than one third that of the three best Frey-Slate genetic algorithm classifiers described above. Moreover Fuzzy ARTMAP simulations each created fewer than 1,070 ART_a categories, compared to the 1,040–1,302 final rules of the three genetic classifiers with the best performance rates. With voting, Fuzzy ARTMAP reduced the error rate to 4.0% (Table 1). Most Fuzzy ARTMAP learning occurred on the first epoch, with test set performance on systems trained for one epoch typically over 97% that of systems exposed to inputs for the five epochs.

Table 1 shows how voting consistently improves performance. With 1 or 5 training epochs, Fuzzy ARTMAP was run for 5 independent simulations, each with a different input order. In all these, and in all other cases tested, voting performance was significantly better than performance of any of the individual simulations in a given group. In Table 1(a), for example, voting caused the error rate to drop to 4.7%, from a 5-simulation average of 8.2%. Hence with 1 training epoch, 5-simulation voting eliminated about 43% of the test set errors. In the 5-epoch simulations, where individual training set performance was close to 100%, 5-simulation voting still reduced the error by about 34% (Table 1(b)), where voting reduced the average error rate of 6.1% to a voting error rate of 4.0%.

References

[1] Carpenter, G.A. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, **37**, 54–115.

[2] Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., and Rosen, D.B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, in press.

[3] Carpenter, G.A., Grossberg, S. and Reynolds, J.H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, **4**, 565–588.

[4] Carpenter, G.A., Grossberg, S., and Rosen, D.B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, **4**, 759–771.

[5] Frey, P.W. and Slate, D.J. (1991). Letter recognition using Holland-style adaptive classifiers. *Machine Learning*, **6**, 161–182.

[6] Holland, J.H. (1980). Adaptive algorithms for discovering and using general patterns in growing knowledge bases. *International Journal of Policy Analysis and Information Systems*, **4**, 217–240.

[7] Kosko, B. (1986). Fuzzy entropy and conditioning. *Information Sciences*, **40**, 165–174.

[8] Wilensky, G. (1990). Analysis of neural network issues: Scaling, enhanced nodal processing, comparison with standard classification. DARPA Neural Network Program Review, October 29–30, 1990.

[9] Zadeh, L. (1965). Fuzzy sets. *Information Control*, **8**, 338–353.

Development of Perceptual Context-Sensitivity in Unsupervised Neural Networks: Parsing, Grouping, and Segmentation

Jonathan A. Marshall
Department of Computer Science, CB 3175, Sitterson Hall
University of North Carolina, Chapel Hill, NC 27599-3175, U.S.A.
919-962-1887, marshall@cs.unc.edu

Abstract. A simple self-organizing neural network model, called an EXIN network, that learns to process sensory information in a *context-sensitive* manner, is described. Exposure to a perceptual environment during a developmental period configures the network to perform appropriate organization of sensory data. An anti-Hebbian learning rule causes some lateral inhibitory connection weights to weaken, thereby letting multiple neurons become simultaneously active (multiple winners). The rule lets other inhibitory weights remain strong; these enforce specific simultaneous contextual consistency constraints on allowable combinations of activations. EXIN networks perform near-optimal parallel parsing of multiple superimposed patterns, by simultaneous distributed activation of multiple neurons. EXIN networks implement a form of credit assignment; they even suppress the activation of some of the most-excited neurons when necessary to achieve maximally context-sensitive global input representations.

The Problem: Context-Sensitivity in Recognition of Multiple Patterns

Context is an important determinant of our interpretations of perceptual data. However, unsupervised Hebbian winner-take-all (WTA) neural networks (NNs) handle contextual information poorly. This paper describes how to extend well-known unsupervised learning rules to let NNs learn to process perceptual information in a context-sensitive manner.

By definition, WTA NNs are capable of representing each input pattern only as a single, lumped item. Suppose a simple WTA NN has learned to recognize three patterns: *ab*, *abc*, and *cd* (Nigrin, 1990a) (FIGURE 1B). When one of these input patterns (say *ab*) is presented to Layer 1 of this NN (by activating neurons *a*, *b*, and *c*, the corresponding Layer 2 neuron (labeled *abc*) becomes active, thereby recognizing the pattern.

Now suppose that occasionally, the familiar stimuli *ab* and *cd* are presented simultaneously, so that the input pattern to the NN is *abcd*. How is this pattern coded? In a WTA NN the best response is to activate neuron *abc*, because it represents the closest match to the input pattern (FIGURE 1C). Activation of other Layer 2 neurons is suppressed. However such a response essentially ignores the presence of *d*, as if *d* were merely noise.

In WTA networks, changes in the strengths of the feedforward excitatory connections may be governed by a Hebbian learning rule. For this discussion, let us use the following variant of a Hebbian rule, expressed mathematically as a differential equation (Grossberg, 1982). Let z_{ji}^{+} represent the strength of the excitatory connection from neuron j to neuron i. Then

$$\frac{d}{dt} z_{ji}^{+} = \epsilon \, f(x_i) \left(-z_{ji}^{+} + h(x_j) \right),$$

where x_j represents the activity level of the j^{th} neuron, $0 < \epsilon << 1$, and f and h are rectified increasing functions (for example, $h(x_j) = \max(0, x_j)$). In English, this roughly means: *Whenever a neuron is active, its input excitatory connections from active neurons become gradually stronger, while its input excitatory connections from inactive neurons become gradually weaker.*

In general, self-organizing neural networks extract statistical regularities (i.e., familiar patterns) from their input environment. Less-familiar patterns are represented in terms

0-7803-0559-0/92 $3.00 © 1992 IEEE

the more-familiar patterns. Multiple superimposed input patterns may be present simultaneously at any moment. The key problem to be analyzed in the present paper is *how a given network can decide what the constituent components of its input are*, based on the statistical history of its environment.

Solution: Learning Contextual Constraints with Excitatory + Inhibitory Rules

One solution for this problem requires only a simple, parsimonious modification to the WTA system: allowing the lateral inhibitory connection weights to vary, according to a bounded anti-Hebbian learning rule (Marshall, 1989, 1990abcdef). This method overcomes many of the limitations of WTA NNs, allowing NNs to acquire greatly more sophisticated forms of behavior. Because the networks use both excitatory and inhibitory learning rules, let us call them **"EXIN"** neural networks for short.

The following bounded inhibitory learning rule is used, in concert with the excitatory rule described above: *Whenever a neuron is active, its output inhibitory connections to other active neurons become gradually stronger (i.e., more inhibitory), while its output inhibitory connections to inactive neurons become gradually weaker.* The inhibitory rule can also be expressed mathematically as a differential equation (Marshall, 1990acdf): Let z_{ji}^- represent the strength of the inhibitory connection from neuron j to neuron i. Then

$$\frac{d}{dt} z_{ji}^- = \delta\, g(x_j) \left(-z_{ji}^- + V q(x_i) \right),$$

where x_j represents the activity level of the j^{th} neuron, $0 < \delta < \epsilon$, and g and q are rectified increasing functions. The parameter V governs the overall amount of coactivation permitted in the NN. Thus, if two neurons, i and j, are frequently coactivated, then the strengths of their reciprocal inhibitory connections tend to increase, thereby decreasing the likelihood that they can become coactivated in the future. On the other hand, if neurons i and j are rarely coactivated, then the strengths of their reciprocal connections tend to decrease, thereby permitting them to become coactivated on relatively rare occasions.

Both the excitatory and inhibitory learning rules are bounded: the functions $h(x_j)$ and $q(x_i)$ are rectified, so that connection weights cannot "change sign" (excitatory weights cannot become inhibitory, and vice versa). The boundedness produces key differences (described below) between the behavior of the networks in this work and that in others' work. In particular, the context-sensitivity property emerges as a consequence of the boundedness.

Changes in neuron activations are governed by a shunting differential equation developed by Grossberg (1982) and Marshall (1990bc). This equation forces neuron activation levels to remain within a bounded range. As a consequence, the learning rules are bounded as well: all connection weights are forced to remain within a specified range.

In the example of FIGURE 1, lateral inhibitory strengths between neurons *ab* and *cd* become weakened according to the anti-Hebbian rule, because patterns *ab* and *cd* do not overlap at all (thus the neurons tend not to receive simultaneous excitation). On the other hand, inhibitory strengths between *ab* and *abc* and between *cd* and *abc* remain strong (FIGURE 1H–J) because the input patterns that they code overlap substantially.

Thus, when *abcd* is presented, neuron *abc* receives inhibition from *both ab* and *cd*, while *ab* and *cd* each receive inhibition *only* from *abc*. Because *abc* receives more inhibition, its activation is suppressed, and *both ab* and *cd* become active. The simultaneous activation of both neurons *ab* and *cd* is made possible because of the weakened reciprocal inhibition between them.

The simultaneous activation of neurons *ab* and *cd* represents the NN's recognition of the superimposed familiar patterns *ab* and *cd*. The EXIN NN thus chooses a more complete representation of the input than is possible in the WTA NN. The contextual presence completely alters the multiplexed *parsing* of the input. When *abc* is presented, the network groups *a*, *b*, and *c* together as a unit, but when *d* is added, the network breaks *c* away from *a* and *b* and binds it with *d* instead, forming two separate groupings. This radical alteration of parsing depending on the presence/absence of small distinguishing features

(like *d*) constitutes the EXIN network's *context-sensitivity* property. In a specific domain like visual perception, such contextual information can determine the segmentation or grouping of a set of visual features.

Simulations: Unsupervised Learning of Contextual Constraints

In computational simulations, the EXIN learning rules caused patterns to be identified by frequency of occurrence, or familiarity. The EXIN networks then build a mechanism that chooses a near-optimal representation of multiple superimposed patterns, by simultaneous distributed activation of multiple Layer 2 neurons. FIGURES 2, 3, and 4 display results of a computer simulation of a simple EXIN NN.

The simulated NN contains 6 neurons in Layer 1 and 6 in Layer 2. Initially, the feed-forward excitatory weights are all uniform up to a 1% random factor, and so are the lateral inhibitory weights. The NN is exposed sequentially to the 6 patterns *a*, *ab*, *abc*, *cd*, *de*, *def*, in random order, for 30,000 presentations. By the end of this training, the connection weights shown in FIGURES 2A and 2B are obtained. Each Layer 2 neuron codes and responds to one of the 6 input patterns (FIGURE 3). The strongest lateral inhibitory connections remain between Layer 2 neurons that code patterns that overlap greatly. The inhibitory connections between neurons coding non-overlapping patterns have become weakest. Approximate symmetry (about diagonal) of inhibitory connection weights is maintained.

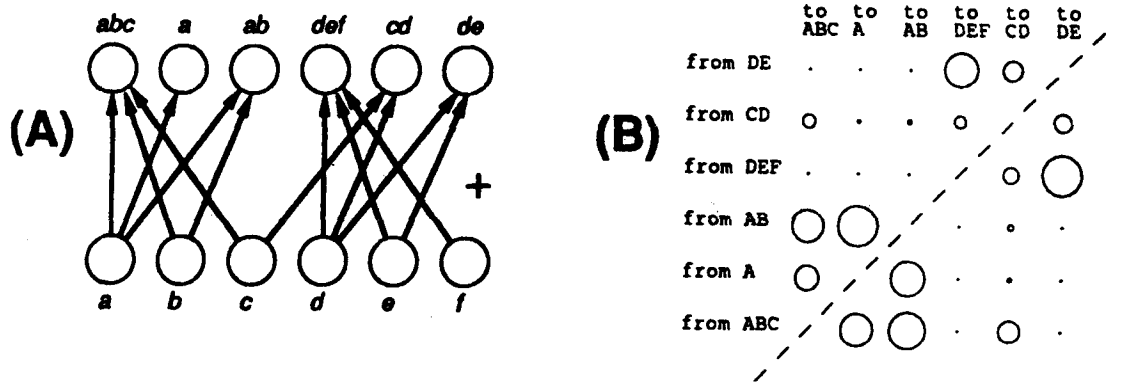

Figure 2: Structure of EXIN network after training. Strong excitatory weights (all in range 0.95–1.00) remaining after training shown in (A); weak connections (all in range 0.00–0.05) are omitted from the figure. Neurons in Layer 1 are labeled *a* through *f*. Neurons in Layer 2 are named according to the Layer 1 neurons from which they receive strong connections. (B) Lateral inhibitory weights within Layer 2, after training. Radius of each circle indicates connection weight.

FIGURE 3 shows the responses of the simulated NN to each of the 6 input patterns separately, after 30,000 presentations of the input patterns. As expected, each of the familiar input patterns activates the corresponding Layer 2 neuron.

FIGURE 4 shows the simulated NN's multiplexed, context-sensitive response to a variety of unfamiliar input combinations. All 64 possible binary input patterns were tested (although only some are shown here), and reasonable results were produced in each case. For example, in FIGURE 4D, pattern *adf* is parsed as *a* + *(Partial)def*. A comparison of FIGURES 3C and 4A demonstrates that the EXIN NN indeed has learned to perform the context-sensitive parsing described in FIGURE 1H–J. Input pattern *abc* is parsed as *abc* (FIGURE 3C), while *abcd* is parsed as *ab* + *cd* (FIGURE 4A).

Comparison with Other Methods

The behavior of EXIN NNs differs from that of ordinary "*k*-winner" NNs, which merely activate the *k* most-excited Layer 2 neurons, regardless of the coactivation or familiarity of the patterns they code. Instead, EXIN NNs implement a form of credit assignment (Barto,

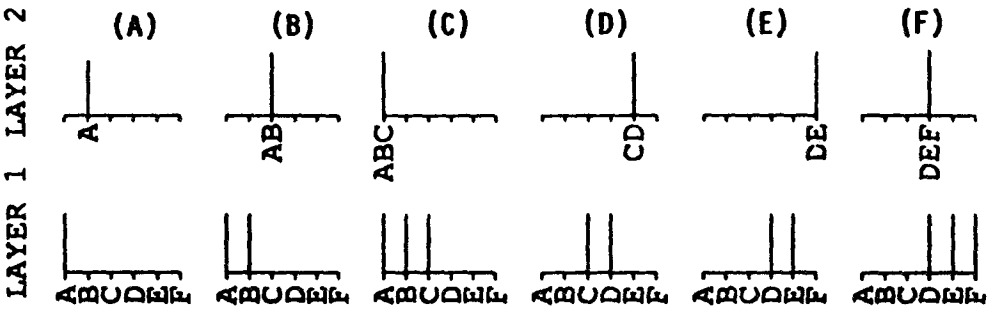

Figure 3: Response of trained EXIN network to familiar patterns. Line height represents neuron activation level.

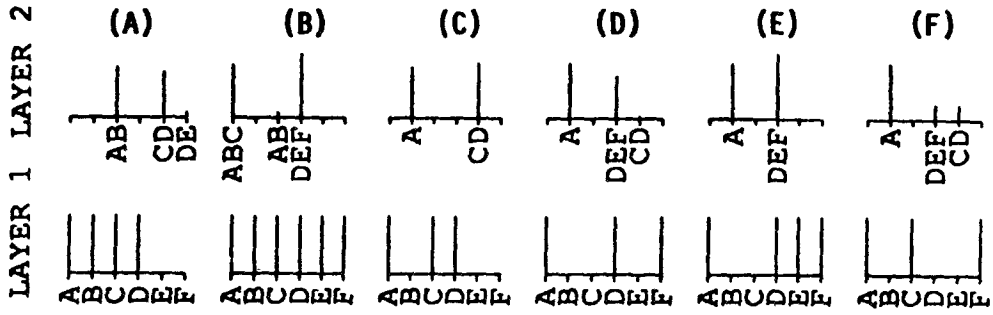

Figure 4: Response of trained EXIN network to unfamiliar patterns. The network exhibits context-sensitive parsing, satisfying the learned constraints.

Sutton, & Anderson, 1983); different representations for a given input unit (such as *b*) compete. Only one such representation at a time can become active; the active representation receives "credit" (by being selected to have its input connection weights adjusted) for representing the input unit. EXIN NNs can even suppress the activation of some of the most-excited Layer 2 neurons (e.g., *abc*) when necessary in order to achieve a maximally context-sensitive, global representation (*ab* + *cd*) of the input (*abcd*), as shown in FIGURE 11.

The distributed representations in EXIN NNs also make them more efficient than *k*-winner NNs, in the following sense: conjunctions (like *abcd*) of familiar patterns (*AB*, *CD*) do not require additional neurons to be accurately represented. In order to represent *abcd* as distinct from *abc*, a *k*-winner NN would need an extra Layer 2 neuron, devoted to coding *abcd*. In fact, it would need one for *every* possible conjunction of input patterns, whereas EXIN NNs can multiplex the coding of conjunctions by simultaneous activation of several neurons. The multiplexed distributed coding in EXIN NNs therefore avoids or at least minimizes the "grandmother cell" dilemma of combinatorial explosion.

While others (Easton & Gordon, 1984; Földiák, 1989, 1990; Kohonen, 1984; Nigrin, 1990ab; Rubner & Schulten, 1990) have proposed using different anti-Hebbian learning rules for lateral connections, Marshall (1990acdef) showed how EXIN neural networks can multiplex, or represent simultaneously, more than one superimposed familiar input pattern Unlike the NNs of Kohonen (novelty detector), Földiák (decorrelator), and Rubner & Schulten (principal components analyzer), EXIN NNs *do not allow connections to be converted from excitatory to inhibitory or vice-versa.* Besides being more biologically plausible, this restriction provides the key advantage of preventing some of the lateral inhibitory connection weights from vanishing. Because some of the lateral inhibitory connection weights remain strong, they enforce contextual constraints on allowable combinations of Layer 2

neuron activations. FIGURE 1E–G shows a decorrelator network, in which lateral connections have vanished and some connections from Layer 1 to Layer 2 have become inhibitory. The decorrelator network essentially responds to *differences* between the patterns, rather than the patterns themselves. As a consequence, the decorrelator network does not activate the closest match to some unfamiliar patterns, such as pattern *c* (FIGURE 1G).

EXIN networks perform essentially the same functions as *masking fields* (Cohen & Grossberg, 1986, 1987). However, masking fields require extensive specific prewiring, to represent all possible input patterns up to a certain size. Much of this representational capacity may never be used in some perceptual environments. EXIN networks, on the other hand, are more plausible and efficient because they wire themselves in response to environmental demands, and they tend to use all available representational capacity.

Conclusions

The neurally plausible excitatory and inhibitory learning rules described in this paper help explain how certain context-sensitive mechanisms might arise in perceptual systems. The results here are useful in modeling the development of a large variety of neural mechanisms, including ones for visual segmentation and transparency, auditory segmentation, visual stereomatching, and visual form perception.

Acknowledgments

I thank R. Eric Fredericksen for help with the simulation software and Kevin E. Martin for help with the figures.

References

Barto, A.G., Sutton, R.S., & Anderson, C.W. (1983). "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems." *IEEE Transactions on Systems, Man, and Cybernetics, 13*, 834–846.

Cohen, M.A. & Grossberg, S. (1986). "Neural Dynamics of Speech and Language Coding: Developmental Programs, Perceptual Grouping, and Competition for Short Term Memory." *Human Neurobiol., 5*, 1–22.

Cohen, M.A. & Grossberg, S. (1987). "Masking Fields: A Massively Parallel Neural Architecture For Learning, Recognizing, and Predicting Multiple Groupings of Patterned Data." *Applied Optics, 26*, 1866–1891.

Easton, P. & Gordon, P.E. (1984). "Stabilization of Hebbian Neural Nets by Inhibitory Learning." *Biological Cybernetics, 51*, 1–9.

Földiák, P. (1989). "Adaptive Network for Optimal Linear Feature Extraction." *Proceedings of the International Joint Conference on Neural Networks*, Washington, DC, June 1989, *I.*, 401–405.

Földiák, P. (1990). "Forming Sparse Representations by Local Anti-Hebbian Learning." *Biological Cybernetics, 64*, 165–170.

Grossberg, S. (1982). *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control.* Boston: Reidel Press.

Kohonen, T. (1984). *Self-Organization and Associative Memory.* New York: Springer-Verlag.

Marshall, J.A. (1989). *Neural Networks for Computational Vision: Motion Segmentation and Stereo Fusion.* Ph.D. Dissertation, Boston University. Ann Arbor, Michigan: University Microfilms Inc.

Marshall, J.A. (1990a). "Self-Organizing Neural Networks for Perception of Visual Motion." *Neural Networks, 3*, 45–74.

Marshall, J.A. (1990b). "Development of Length-Selectivity in Hypercomplex-Type Cells." *Investigative Ophthalmology and Visual Science, 31*, 4, 397.

Marshall, J.A. (1990c). "A Self-Organizing Scale-Sensitive Neural Network." *Proc. Internat. Joint Conf. on Neural Networks*, San Diego, June 1990, *III.*, 649–654.

Marshall, J.A. (1990d). "Representation of Uncertainty in Self-Organizing Neural Networks." *Proc. Internat. Neural Network Conf.*, Paris, France, July 1990, 809–812.

Marshall, J.A. (1990e). "Self-Organizing Neural Network for Computing Stereo Disparity and Transparency." *Optical Society of America Annual Meeting Technical Digest*, Boston, November 1990, 268.

Marshall, J.A. (1990f). "Adaptive Neural Networks for Multiplexing Oriented Edges." *Intelligent Robots and Computer Vision IX: Neural, Biological, and 3-D Methods*, David P. Casasent, Ed. Proceedings of the SPIE 1382, Boston, November 1990, 282–291.

Nigrin, A.L. (1990a). "SONNET: A Self-Organizing Neural Network that Classifies Multiple Patterns Simultaneously." *Proc. Internat. Joint Conf. on Neural Networks*, San Diego, June 1990, *II.*, 313–318.

Nigrin, A.L. (1990b). *The Stable Learning of Temporal Patterns with an Adaptive Resonance Circuit.* Ph.D. Dissertation, Computer Science Department, Duke University.

Rubner, J. & Schulten, K. (1990). "Development of Feature Detectors by Self-Organization." *Biological Cybernetics*, 62, 193–199.

NUCLEAR REACTOR CONDITION MONITORING BY ADAPTIVE RESONANCE THEORY

Shahla Keyvan
Nuclear Engineering Department
University of Missouri-Rolla
Rolla, MO 65401

Luis Carlos Rabelo
Department of Industrial and Systems Engineering
Ohio university.
Athens, OH 45701

Anil Malkani
Department of Electrical and Computer Engineering
Ohio university.
Athens,OH 45701

ABSTRACT

This paper presents an evaluation of the performance and comparison between various paradigms of the ART family of artificial neural networks in nuclear reactor signal analysis for development of a diagnostic monitoring system. To closely represent reactor operational data, reactor pump signals from Experimental Breeder Reactor (EBR-II) are analyzed. The signals are both measured signals collected by Data Acquisition System (DAS) as well as simulated signals. ART2, ART2A, Fuzzy ART, and Fuzzy ARTMAP are applied in this study.. Several simulators are built, and the study indicates that while all ART paradigms are appropriate for application in reactor signal analysis, each has its own unique characteristics and features which can be utilized whenever needed and applicable.

INTRODUCTION

In order to assure a safe operation, nuclear power plants are designed and built incorporating a large number of sensors of various kinds to monitor reactor condition/parameters at all time. The objective of this research work is to evaluate the application of unsupervised/ supervised artificial neural networks (ANNs) in the analysis of reactor signals for reactor condition monitoring. The application of ANNs, specifically the ART family as a tool for reactor diagnostics is examined here. Reactor pump signals utilized in a wear-out monitoring system developed for early detection of degradation of pump shaft [14] are analyzed to study the feasibility of a system based on artificial neural networks for monitoring and surveillance in nuclear reactors. The Adaptive Resonance Theory paradigms (ART 2, ART 2-A, Fuzzy ART, and Fuzzy ARTMAP) of ANNs are applied in this study. The signals are collected signals as well as generated signals simulating the wear progress.

The diagnosis is based on the monitoring of the performance and the impact of an equipment/component on the operation environment through the analysis of associated signals from the reactor sensors. The wear-out monitoring system is based on noise analysis and utilizes Dynamic Data System (DDS) approach of Autoregressive Moving Average (ARMA) regression modeling. The mathematical representation of the model for a univariate system is :

$$(1-a_1Z^{-1}-a_2Z^{-2}.....-a_nZ^{-n})Y(k) = (b_1Z^{-1}-b_2Z^{-2}.....b_{n-1}Z^{-(n-1)})R(k)$$

where,

Y(k) = discrete signal data, k = index of time interval, R(k) = white noise residual, $Z^{-1}Y(k) = Y(k-1)$

a,b = autoregressive, moving average parameters.

The autoregressive and moving average parameters are then decomposed into pairs of complex discrete roots (eigenvalues), i.e. for a second order dynamic :

$$r_{1,2} = a \pm bi$$

where, $a_1 = r_1 + r_2$ $a_2 = -r_1r_2$

0-7803-0559-0/92 $3.00 © 1992 IEEE

similarly,

$$r^{\circ}_{1,2} = a^{\circ} \pm b^{\circ} i$$

where, $b_1 = r^{\circ}_1 + r^{\circ}_2$ $\qquad$ $b_2 = -r^{\circ}_1 r^{\circ}_2$

The pattern recognition and the measure of wear progress in the monitoring system is achieved by introducing new parameters (representing an index of wear progress) which are based on the increase of the impact of the wear related dynamic on the signal fluctuation as degradation progresses [14]. The main objective here is to test the selected ANN paradigms for the capability of detecting and recognizing different levels of degradations apart. In addition, the goal is to achieve this objective without any pre/post signal processing or analysis. Time involved in the learning process of the ANNs is of concern when applying high dimensional input spaces for reactor diagnostics. Due to the extensive number of possible pattern variations incremental learning is also desired. On the other hand achieving the same level of performance with minimum data input is desirable and quite a challenge without major preprocessing of the signals before application of the artificial neural structures.

The selection of ART 2 is due to its desired design principles such as unsupervised learning, stability-plasticity, search direct access, and the match reset trade-off. Fuzzy ART is selected due to the fact that its algorithm is developed based on traditional neural network models such as ART 1, while incorporating fuzzy logic operators, hence enhancing the capabilities of ART 1 for analog inputs. Also Fuzzy ART includes two optional features (complement coding and fast learning with slow recoding) which enables the network to overcome limitations associated with sequential data presentation and stochastic input fluctuations. ART 2A is selected due to its speed and simplicity of the design. Fuzzy ARTMAP is selected due to its capability of handling nonstationary stochastic signals as well as supervised learning.

The result is a success for all paradigms,and the study shows that ART 2A is not only able to learn and distinguish the patterns from each other, its learning speed is also extremely fast despite the high dimensional input spaces. In addition, Fuzzy ART is capable of equivalent performance as ART2 and a speed as fast as ART2A with less than half the input data necessary for both ART2 and ART2A, when applied in complement coding mode. Fuzzy ARTMAP provided a flexible and high performance supervised/unsupervised learning scheme.

SIGNAL DESCRIPTION

The signals utilized in this study are divided in two groups, the actual measured signal and the simulated signals. The measured signal is the pump power signal from pump number 1 of the EBR-II nuclear reactor which is collected from sensors by plant data acquisition system. Figure 1 shows the original plot of this signal data for fifty seconds time period.

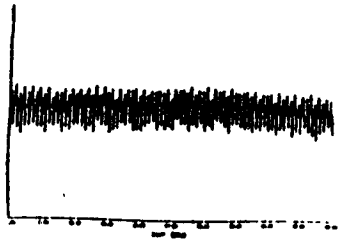

Fig. 1 Measured pump power signal

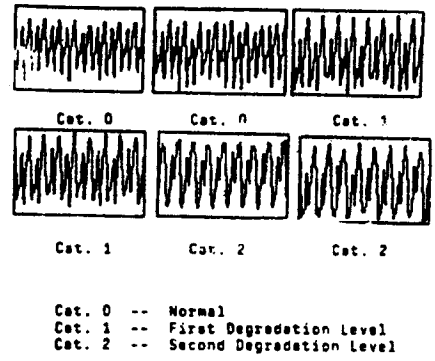

Figure 2. NAB pump power signal

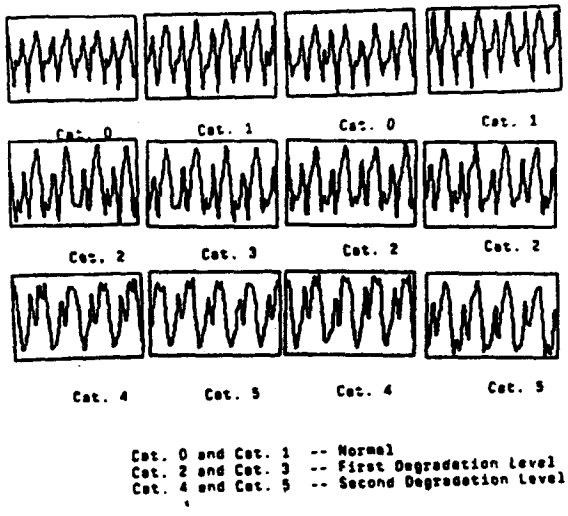

Fig. 3 250 dimensionality training data.

The first twenty five seconds of data from this measured signal is used to provide two sets of patterns of 500 data points dimensionality representing the normal pump power signal in this study. Two sets of simulated signals were generated from the signal of figure 1 representing the pump power simulated for two levels of degradation due to deposition of sodium oxide on the pump shaft [14]. The two sets of measured signals are referred to as pattern N. The two sets of simulated data for first level degradation are pattern A, and the two sets of simulated data for second level degradation are pattern B. Thus file NAB is a file containing the six sets of data of pattern N, A, and B respectively (Figure 2). In addition, in order to show the capability of Fuzzy ART and Fuzzy ARTMAP in pattern recognition with less number of input data than required for ART2 and ART2A, the file NAB containing is used as twelve set of data with 250 data each (Figure 3). These means there are 4 sets of 250 data points in each signal, normal, first level degradation, and second level degradation. In addition 20 sets of 250 data points of distorted data for each signal, normal, first level degradation, and second level degradation were utilized to test the recall performance of the Fuzzy ARTMAP network.

ART

ART represents a family of ANNs which self-organize categories in response to arbitrary sequences of input patterns in real time for pattern recognition [4]. A class of these networks called ART 1 [2], which is unsupervised, can be used only for binary patterns. ART 2 [3,4], which is also an unsupervised class responds to both binary and analog patterns. The class ART 3 [5] features an advanced reinforcement feedback mechanism which can alter the classification sensitivity or directly engaging the search mechanism. The algorithm Fuzzy ART [9] is similar in architecture to ART 1, however, fuzzy operators are added in order to handle analog patterns without loosing the advantages of ART 1 architecture. The ARTMAP ("predictive" ART) [7] and Fuzzy ARTMAP [6] are built upon the basic ART designs, while incorporating supervision in the learning process. ART 2-A ("algorithmic" ART) [8] is a special case of ART 2 which emphasizes the intermediate and fast learning rates using algebraic equations, hence accelerating the learning process.

ART 2

ART 2 [3] has been selected to illustrates some of the principal architectural elements and network processing characteristics of a typical ART architecture (Figure 4). ART 2 have both attentional and orienting subsystems.

Attentional Subsystem

The attentional subsystem is composed of long term memory and short term memory elements.

Short Term Memory (STM). F1, the feature representation field, and F2, the category representation field, are the two STM main components. F0 is utilized as a preprocessing field.

The functions of the F1 field are contrast enhancement, noise suppression, matching, and normalization. F1 is composed of three layers as a difference with ART 1 which only has one. As explained by Carpenter and Grossberg [2]"Even for binary inputs, ART 2 differs from ART 1 processing" referring to the norms which each one utilizes. ART 2 utilizes the L^2 norm, which as demonstrated by Grossberg [10,11,12] (based on membrane equations [13]),arises naturally in neural networks. Among its functions, the normalization mechanism keeps F1 from saturation in spite of the constant presence of the input pattern during the learning process.

The F2 field by means of competitive interactions of the F2 nodes, chooses the one which responds maximally to the vector P (p1,...,pN) as P is applied to the bottom-up adaptive filter. In addition, the F2 field suppresses F2 nodes (i.e., reset) as guided by the orienting subsystem.

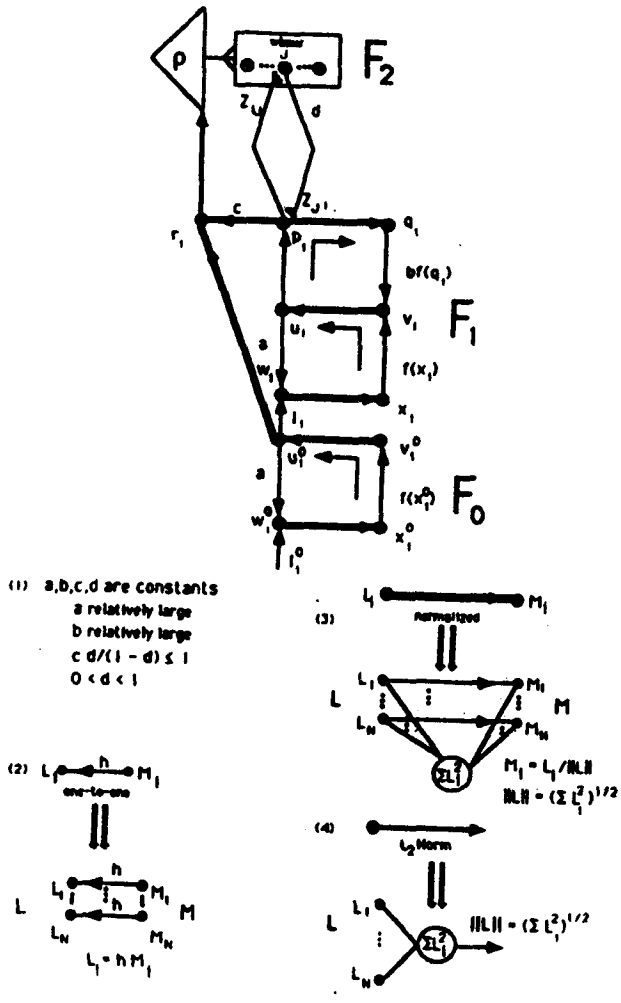

Fig. 4 ART 2 [3]

Long Term Memory (LTM). LTM is made up of two components, the bottom-up adaptive filter (Z_{ij}) and the top-down adaptive filter (Z_{ji}) where learning and therefore category structuring occurs. The bottom-up adaptive filter incorporates learning into the network. On the other hand, the top-down adaptive filter enables attentional priming, matching, and self-adjusting parallel search.

When the jth F2 node supplies an appropriate match, the top-down and bottom-up adaptive filters, both need to be adjusted. To understand the learning equations it is helpful to identify an ART 2 architecture as a bottom-up competitive learning system using an instar (fan-in) [1,11,12] combined with a top-down outstar (fan-out) [1,10] learning system.

For the instar coding (bottom-up adaptive filter), the following learning equation is utilized:

$dz_{ij}/dt = [p_i - z_{ij}]\, d.$

That is, the weight vector $(z_{1J},\ldots,z_{NJ})$ keeps track of the incoming signal vector P ("presynaptic activity") and the Jth node in F2 is identified as the winner with its respective activation d ("postsynaptic activity"). Therefore,

$dz_{ij}/dt = d\,(1 - d)\,[u_i/(1 - d) - z_{ij}].$

For the outstar learning (top-down adaptive filter), the following equation is utilized:

$dz_{ji}/dt = [p_i - z_{ji}]\, d.$

That is, the weight vector keeps track of the F1 activity vector P ("postsynaptic"). d is the activation of the winner node (Jth) in F2 which is the incoming signal ("presynaptic activity"). Replacing pi by its value when the Jth F2 node is active will result in:

$dz_{Ji}/dt = d\,(1 - d)\,[u_i/(1 - d) - z_{Ji}].$

Two kinds of learning could be distinguished in ART 2: slow and fast. As a difference with ART 1, the fast learning case in ART 2 can not be reduced to algebraic equations.

Orienting Subsystem

The orienting subsystem helps to direct the search for categories. It is connected to F2 and F0 in the attentional subsystem. When the orienting subsystem is activated, if the degree of similarity between vectors U and P is greater than the vigilance factor, then the network will resonate, otherwise the winning F2 node will be reset.

ART 2-A ("ALGORITHMIC" ART)

ART 2-A is a special case of ART 2 designed for large-scale pattern recognition tasks [8]. Its algorithmic type nature lends itself for rapid prototyping in hardware and software.

ART 2-A has three fields: F0, F1, and F2. The output of the F1 field is the vector I defined by:

$$I = \text{normal}(f(\text{normal}(I^0)))$$

where I^0 is the input vector of dimensionality M, normal is an operator defined by:

$$\text{normal}(x) = x/||x||$$

and f() is a piecewise linear function .

The LTM vector in ART 2-A is scaled, and it could be interpreted as the LTM vector of ART 2 divided by 1/(1-d). As in ART 2, the F2 node ART 2-A makes a choice if the Jth node becomes maximally active. In addition, the F2 STM activation represents the degree of match of the vector I and the scaled LTM vector. LTM adjustments are performed in a single iteration and are reduced to algebraic equations for fast and intermediate learning.

FUZZY ART

Fuzzy ART [9] incorporates the basic architecture and neuro-dynamics of ART systems. Fuzzy ART is designed as a generalization of ART 1. However, the set theory intersection operator ($\bigcap$) of ART 1 is replaced by the fuzzy set theory conjunction ($\wedge$). This fuzzy operator makes Fuzzy ART capable of handling both analog and binary data.

Nevertheless, Fuzzy ART has also other features that is possible to find in other ART families beside ART1 such as:

a) Fast-commit slow-recode

b) Input pre-processing .

For each input presented to the network, the net value is calculated as

$$net_j(I) = |I \wedge W_j|/(\alpha + |W_j|), \qquad \alpha = 0.0001 \text{ (in our implementation)}$$

I is the input vector and W_j are LTM traces. If vigilance criteria is met i.e

$$|I \wedge W_j|/|I| >= \rho$$

then learning takes place,

$$W_j^{new} = \beta(I \wedge W_j^{old}) + (1 - \beta)W_j^{old}$$

where, $0 < \beta < 1$, and $\beta = 1$ i.e fast learning

FUZZY ARTMAP

FUZZY ARTMAP [6] is a supervised learning of recognition categories in response to arbitrary sequences of analog or binary inputs. It incorporates two FUZZY ART modules i.e FUZZY ARTa and FUZZY ARTb that are interlinked through a mapping field F^{ab}. The inputs to the FUZZY ART modules are presented in complement form i.e $A = (a, a^c)$ (Figure 5).

During training, at the start of each input presentation the FUZZY ARTa vigilance factor equals the baseline vigilance of F_a, and the map field vigilance parameter is set to 1. when a prediction by FUZZY ARTa is disconfirmed at FUZZY ARTb i.e F^{ab} output vector $X_{ab} = 0$, match tracking is induced. Match tracking rule raises the module a vigilance to

$$|Xa|/|A| + 0.0001 \text{ (in our implementation)},$$

Where, $|X^a| = |A \wedge W_j^a|$. 'A' is the input i.e $A = (a, a^c)$, X_a is the output vector of FUZZY ARTa module, $|X|$ is summation of all the components of X, and W_j^a is the weight vector from the jth output node of FUZZY ART module a to F_{ab} .

When Match tracking occurs, Fuzzy ARTa search leads to activation of another output neuron.If the prediction by FUZZY ARTa is confirmed at FUZZY ARTb (F^{ab} output vector $X_{ab} = 1$), Map field learning takes place. J learns to predict the FUZZY ARTb category K, and sets $W_{JK}^{ab} = 1$.