VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5341
Fax: +64 4 495 5045
Internet: Tech.Reports@mcs.vuw.ac.nz

A Fuzzy ART-based Modular Neuro-fuzzy Architecture for Learning Hierarchical Clusterings

Guszti Bartfai and Roger White

Technical Report CS-TR-97/6
July 24, 1997

Abstract

This paper introduces a modular neuro-fuzzy architecture (FUZZY HART-S) that is capable of incremental learning of stable hierarchical clusterings of arbitrary sequences of analogue and binary input patterns by self-organisation. The architecture is a cascade of fuzzy Adaptive Resonance Theory (ART) network modules, in which each module learns to categorise the *difference* between the current input to the preceding module and the prototype of its selected category. The developed hierarchy is represented via explicit links between category nodes in neighbouring modules. The experimental results demonstrate the representational capabilities of FUZZY HART-S as well as provide comparisons to two other clustering methods. Due to its simple and fast operation, modularity and hierarchical representation, the FUZZY HART-S architecture lends itself to fast and effective hardware implementation, which makes it particularly suitable for real-world applications.

Publishing Information

This paper is to appear in Proceedings of 6th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'97).

Author Information

Guszti Bartfai is a lecturer in computer science. Roger White is completing his MSc degree in computer science. Their email addresses are `guszti@mcs.vuw.ac.nz` and `rwhite@mcs.vuw.ac.nz`, respectively.

1 Introduction

Integrating the adaptiveness of neural networks and the concept of "fuzziness" into intelligent decision systems has proven fruitful in recent years (see [13], for example). Among the many neuro-fuzzy systems that have been studied and applied successfully are the Fuzzy Adaptive Resonance Theory (ART) [6] and Fuzzy ARTMAP [5] networks. They and their variants have been shown to have several useful properties for fuzzy clustering [15, 12].

This paper extends the line of clustering methods by introducing a Fuzzy ART-based method called FUZZY HART-S that is capable of incremental learning of hierarchical clusterings by self-organisation. This method is based on the idea of the HART-S neural network [2], in which a cascade of (binary) ART1 modules [4] are used to learn successively finer categorisation of input patterns by each module splitting up the categories developed in the previous one (hence the name "Hierarchical ART with Splitting", or HART-S) until a desired resolution is reached. The method proposed here utilises the advantages of HART-S and fuzzy systems by replacing the component ART1 modules with Fuzzy ART ones. In particular, the FUZZY HART-S method retains all the useful properties of Fuzzy ART networks such as fast and stable learning of arbitrary input sequences as well as dynamic creation of new categories for novel inputs. The developed hierarchies are represented explicitly by links between each category and its child categories. Each category node in the tree is described with its fuzzy prototype vector. Due to its simple operation, modularity and hierarchical representation, the FUZZY HART-S method lends itself to fast and effective hardware implementation, which makes it particularly suitable for real-world applications.

Section 2 describes the Fuzzy ART network and summarises its properties. The FUZZY HART-S architecture and its learning algorithm are introduced in section 3. The experimental results are presented in section 4. Finally, conclusions are drawn in section 5.

2 The Fuzzy ART network

This section describes the Fuzzy ART architecture at a level of detail that is necessary for understanding the rest of the paper.

A Fuzzy ART network performs unsupervised clustering on analogue and binary input patterns. It has three layers: the *input* layer (F0), *comparison* layer (F1), and *category* layer (F2), with M, M and N nodes, respectively (see figure 1). Each F2 node j stores a prototype vector $\mathbf{z}_j$ for the category it represents.

Upon presentation of an input vector $\mathbf{x}$ represented as a vector of fuzzy attributes ($x_i \in [0,1], i = 1, 2, \ldots, M$), the network attempts to classify it into one of its existing categories based on its similarity to the stored prototype of each category node. More precisely, for each node j in the F2 layer, the *choice function*

$$T_j(\mathbf{x}) = \frac{|\mathbf{x} \wedge \mathbf{z}_j|}{\beta + |\mathbf{z}_j|} \tag{1}$$

is calculated, where $\beta > 0$ is the *choice parameter*, the fuzzy AND [16] operator $\wedge$ is defined by

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i) \quad ,$$

and the norm $|\cdot|$ is defined by

$$|\mathbf{p}| \equiv \sum_{i=1}^{M} p_i$$

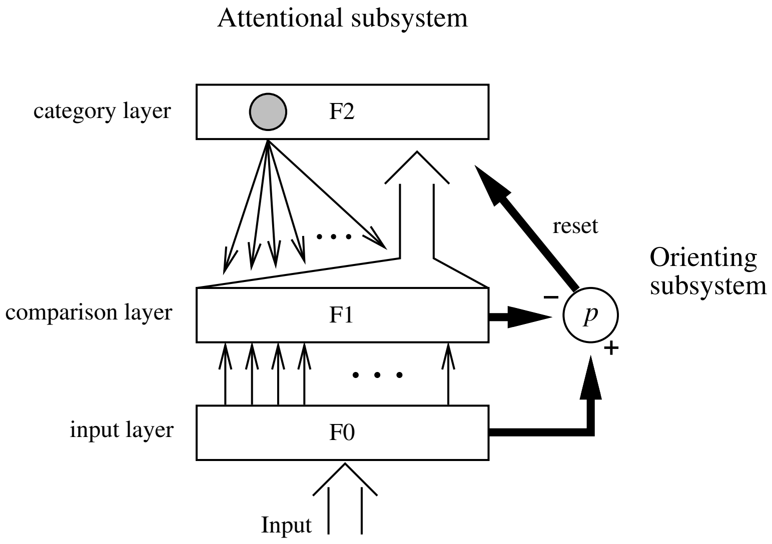

Figure 1: Architecture of the Fuzzy ART network.

for any M-dimensional vector $\mathbf{p}$.

The network then makes a category choice by selecting the F2 node J where

$$T_J = \max\{T_j : j = 1, 2, \ldots, N\} \ .$$

The F1 layer activity vector then becomes $\mathbf{x} \wedge \mathbf{z}_J$ and is compared to the current input vector $\mathbf{x}$: the strength of the match is given by

$$\frac{|\mathbf{x} \wedge \mathbf{z}_J|}{|\mathbf{x}|} \ , \tag{2}$$

which is compared with a system parameter ρ called *vigilance* $(0 \leq \rho \leq 1)$. If the input matches sufficiently, i.e. the match strength $\geq \rho$, then it is assigned to F2 node J and the network learns it by adjusting $\mathbf{z}_J$ according to the equation

$$\mathbf{z}_J^{(new)} = \eta(\mathbf{x} \wedge \mathbf{z}_J^{(old)}) + (1 - \eta)\mathbf{z}_J^{(old)} \tag{3}$$

where η is the *learning rate* $(0 \leq \eta \leq 1)$.

If the stored prototype $\mathbf{z}_J$ does not match the input sufficiently (match strength $< \rho$), the winning F2 node J is reset for the period of presentation of the current input. Then another F2 node (or category) will be selected whose prototype will be matched against the input. This "hypothesis-testing" cycle is repeated until the network either finds a stored category whose prototype matches the input closely enough, or creates a new category. Then learning takes place as described above.

After an initial period of self-stabilisation, the network will directly (i.e. without search) access the prototype of one of the categories it has found in a given training set. The higher the vigilance level, the larger number of smaller, or more specific, categories will be created. (If $\rho = 1$, the network will assign a new category for every unique input pattern.) Also, as a consequence of its stability-plasticity property, the network is capable of learning "on-line", i.e. refining its learned categories (or creating new ones to novel inputs) in response to a stream of input patterns (as opposed to being trained "off-line" on a finite training set).

3 The FUZZY HART-S network

The FUZZY HART-S network is a cascade of Fuzzy ART modules. Each module (or FUZZY HART-S layer) learns to categorise the *difference* between the contents of the F0 and F1 layers of the previous module. Only when such differences are greater than a size determined by two network parameters called *resolution* and feature threshold (ϵ and θ, see explanation below) is the next Fuzzy ART module activated. The hierarchical links are created between the active F2 nodes of neighbouring (active) Fuzzy ART modules.

More formally, the FUZZY HART-S network is described by the following parameter tuple:

$$\mathcal{FHS} = \{L, \mathbf{M}, \mathbf{N}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\eta}, \epsilon, \theta\}$$

where the meaning of each parameter is as follows:

$$
\begin{aligned}
L &: \text{the number of Fuzzy ART modules used} \\
\mathbf{M} = (M_1, \ldots, M_L) &: \text{the size of each F0 and F1 layer} \\
\mathbf{N} = (N_1, \ldots, N_L) &: \text{the size of each F2 layer} \\
\boldsymbol{\beta} = (\beta_1, \ldots, \beta_L) &: \text{the choice parameter of each Fuzzy ART module } (0 < \beta_k, \text{ typically } \beta_k \ll 1) \\
\boldsymbol{\rho} = (\rho_1, \ldots, \rho_L) &: \text{the vigilance level of each Fuzzy ART module, } \rho_k \in [0,1] \\
\boldsymbol{\eta} = (\eta_1, \ldots, \eta_L) &: \text{the learning rate of each Fuzzy ART module, } \eta_k \in [0,1] \\
\epsilon &: \text{resolution, } \epsilon \in [0,1] \\
\theta &: \text{feature threshold, } \theta \in [0,1]
\end{aligned}
$$

with the architectural constraint of $M_k = M$ for $k = 1, \ldots, L$. For further simplicity, however, we shall also assume $N_k = N$, $\beta_k = \beta$ and $\eta_k = \eta$ for $k = 1, \ldots, L$. So it is only the ρ_k vigilance levels that are allowed to be different in each Fuzzy ART module.

There are four sets of connections between each successive module in a FUZZY HART-S network. An adaptive associative connection (V) called hierarchical memory between the F2 layers of successive Fuzzy ART modules provides the links between any cluster and its immediate sub-clusters. When a category is active in the first of a pair of modules, only those categories in the second module that can be sub-categories of the first are allowed to become active. These links are important in this network so the correct hierarchies can be maintained, since the individual Fuzzy ART modules cluster differences only. More details about this cascade architecture are found in [2].

The *local input* to level k is

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{x}_{k-1} - \mathbf{x}_{k-1} \wedge \mathbf{z}_{k-1:J} \\
&= \mathbf{x}_{k-1} \wedge \mathbf{z}^c_{k-1:J}, \qquad k = 2, \ldots, L
\end{aligned}
\tag{4}
$$

where $_{k-1:J}$ is the index of the winning F2 node in module $k-1$, and $\cdot^c$ denotes the fuzzy complement operator defined by $(\mathbf{p}^c)_i \equiv 1 - p_i$. Applying (4) to layers $k, k-1, \ldots, 2$ recursively (and ignoring the indices of winning F2 nodes for simplicity), for the local input to layer $k+1$ we get

$$\mathbf{x}_{k+1} = \mathbf{x}_1 \wedge \mathbf{z}^c_1 \wedge \ldots \wedge \mathbf{z}^c_k, \quad k = 1, \ldots, L\ . \tag{5}$$

BEGIN FUZZY HART-S
Initialise
DO
  FOR EACH input in training set
    $\mathbf{x}_1 \leftarrow \mathbf{X}$
    $k \leftarrow 1$
    DO
      Find matching $\mathrm{F2}_k$ category $\mathbf{z}_{k:J}$
        for local input $\mathbf{x}_k$
      Learn local input in level k
        according to eq (3)
      Produce input to next layer:
        $\mathbf{x}_{k+1} \leftarrow \mathbf{f}_\theta(\mathbf{x}_k \wedge \mathbf{z}^c_{k:J})$
      $k \leftarrow k+1$
    UNTIL $k > L$ OR $|\mathbf{x}_k| < \epsilon\, S$
  END FOR
UNTIL Stability
END

Figure 2: The FUZZY HART-S training algorithm as applied to a finite training set. The condition for stability is satisfied when changes in the category prototypes are sufficiently small.

(Note, the local input to the first module, $\mathbf{x}_1$, is also the *global input*, $\mathbf{X}$, to the entire network.)

A distinctive feature of the FUZZY HART-S network is that the local input $\mathbf{x}_k$ will *not* activate module k unless $|\mathbf{x}_k| > \epsilon S$, where S is maximum the size of the global input $\mathbf{X}$ (which is constant for normalised inputs), and ϵ is a global parameter called *resolution*, which controls the amount of sub-clustering that can occur (i.e., the depth of the resulting tree). Also, the feature threshold parameter (θ) determines the minimum value for each local input feature in order for it to be propagated to the next layer. More precisely, the local input $\mathbf{x}_k$ will be filtered with the $\mathbf{f}_\theta$ function before being considered for propagation, where $\mathbf{f}_\theta$ is defined as $\mathbf{f}_\theta(\mathbf{x}) \equiv (f^1_\theta, \ldots, f^M_\theta)$ and

$$f^i_\theta = \begin{cases} 0 & \text{if } x_i < \theta \\ x_i & \text{otherwise} \end{cases} \qquad i = 1, \ldots, M \,.$$

The training algorithm of the FUZZY HART-S network is shown in figure 2.

4 Experimental results

4.1 Learning hierarchies with FUZZY HART-S networks

In order to demonstrate that the FUZZY HART-S network is capable of developing class *hierarchies*, we trained and tested the networks on the *Zoo* machine learning benchmark data set [14]. This data set contains 101 instances of animals described with 18 attributes such as "hair", "aquatic", "domestic" and so on. It is relatively small, but is adequate for the purposes of these experiments, which also allows us to make general observations about the behaviour of these models that will guide further analysis and development.

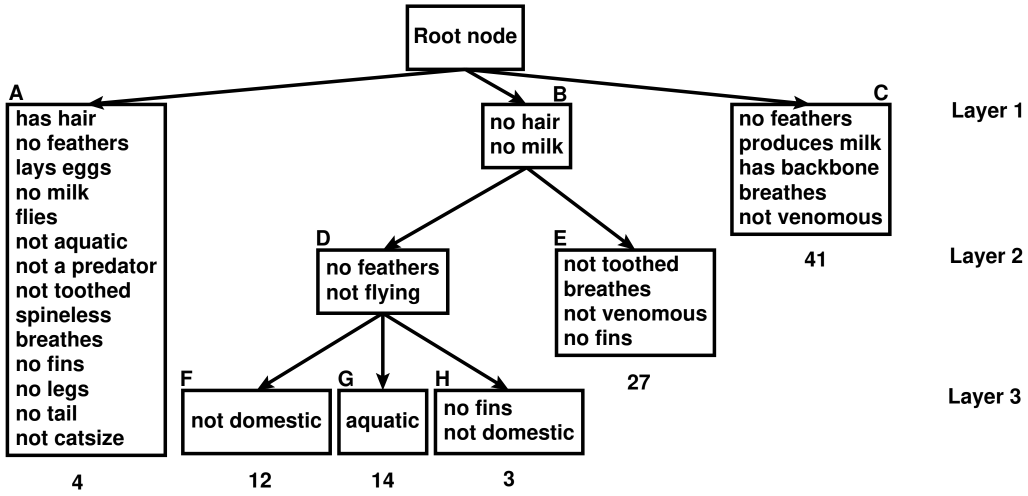

Figure 3: A cluster hierarchy learnt from the *Zoo* database by a three layer FUZZY HART-S network. In this run, the choice parameters (β_k) and learning rates (η_k) were set to 1×10^{-5} and 0.4, respectively. Resolution (ϵ) was 0.7, and feature threshold (θ) was 0.9. Each box (identified by a capital letter) corresponds to a category with its distinctive features listed in it (i.e., showing only the local prototype). Features not shown were found to be irrelevant in characterising the corresponding (sub)category. The root node is only shown here for clarity; it is not represented in the network explicitly. The numbers below the leaf categories show the number of input patterns that belong to them.

The attributes of the training samples were presented to the network in *complement coding* to assign significance to both the presence and absence of features, and to keep the input vectors normalised to avoid proliferation of Fuzzy ART categories [6]. (The "type" attribute was ignored to eliminate any bias from the developed clusterings.) In all the experiments, the FUZZY HART-S network was trained "off-line" on a training set containing all 101 instances of the *Zoo* data set. In each training session, the network was considered to be completely trained when the change of all the prototype values for a whole epoch (i.e., presentations of the entire data set) remained within a given (small) margin.

Figure 3 is an example hierarchy learnt by a three-layer FUZZY HART-S network. Each layer developed categories with added distinctive features from their parent class. For example, category "F" is a sub-category of "D", which in turn is a sub-category of "B". The global prototype that describes category "F" is the fuzzy union of the local prototypes of categories "B", "D" and "F". The classes found here appear to be "intuitive" in that, for example, class "C" contains only mammals. Other training runs resulted in different hierarchies, some of which seemed less intuitive. The average number of training epochs over 100 training sessions (each with a different learning rate) were as follows:

Learning rate (η)	Epochs
1.0	2.97
0.8	5.69
0.6	8.39
0.4	13.64

Note that the learning algorithm within each Fuzzy ART module allows the development of fuzzy category prototypes (but not fuzzy categories!). However, with a high feature threshold value (here we used $\theta = 0.9$), the inputs to subsequent modules are effectively binary. A FUZZY HART-S network with lower resolution and feature threshold (ϵ and θ) would have found more layers with child categories more similar to their parent class. Furthermore, when extracting prototypes from the network trained on a *binary* data set (as here), the fuzzy prototypes can be converted to binary form since we know that the only possible attribute values are binary.

Also, if fuzzy complement coding is applied to the input patterns, the learned category prototypes will correspond to hyper-rectangles in the feature space [6]. So each category hyper-rectangle at lower levels in the hierarchy is contained within that of its parent class.

4.2 Comparison to other clusterings methods

Although our motivation for FUZZY HART-S was not to design yet another hierarchical clustering algorithm, but to explore the benefits of the synthesis of HART-S neural networks and fuzzy systems (with respect to real-world applications), one can ask how the clusterings found by FUZZY HART-S compares those of other methods. We attempted to find an answer to this with this set of experiments.

We used two other clustering methods to compare with FUZZY HART-S: one is the *group average* statistical hierarchical clustering method [10], and the other is an incremental machine learning algorithm for concept formation called *Froggit* [1].

The following two data sets (originally from the UCI Repository [14]) were used:

- *Soybean*: this data set contains 307 instances [2] described with 35 nominal attributes (and class information, which was ignored here).
- *Mushroom*: this data set contains 1000 randomly selected examples from the original Mushroom set found in the repository. Each instance is described with 22 nominal features (and class information, which was ignored here).

Since the class representation of these methods (including FUZZY HART-S) are markedly different, a reasonable way to compare the methods is to look at their partitionings at the instance level (instead of comparing their class descriptions). We can estimate the probability with which categories of FUZZY HART-S networks and other methods agree. More precisely, given the same number of categories found, we estimate

$$P_a \equiv \frac{P_{h|c} + P_{c|h}}{2}$$

where $P_{h|c}$ and $P_{c|h}$ are the conditional "agreement" probabilities: $P_{h|c}$ is the probability that if two instances are found to be in the same category by FUZZY HART-S, they are also found to be in the same category by the clustering algorithm (group average or Froggit). The best case is when $P_a = 1$, i.e. the two methods agree 100% of the time. The worst case is complete randomness, or non-correlation between the methods. However, even in this case, there is a finite probability that the random selection of categories will agree with any clustering method. In case of complete random categories, this probability (P_r) is $\frac{1}{N_c}$ where N_c is the number of classes in the given partition. So clearly, $P_r \leq P_a \leq 1$.

Table 1 shows the estimated P_a probabilities for the FUZZY HART-S network. The number of classes (N_c) were gained differently in each method. For the statistical clustering method

[1] An earlier version of Froggit, called *DySect*, has been applied successfully to data analysis [8]. More details on Froggit can be found in [2].

[2] There are two versions of this data set in the UCI repository: *soybean-small* and *soybean-large*. We used the latter.

		Group		Froggit	
N_c	P_r	P_{as}	P_{am}	P_{as}	P_{am}
5	0.200	0.616	0.593	0.337*	0.321
10	0.100	0.392	0.531	0.281	0.291*
15	0.067	0.253	0.413	0.258	0.293

Table 1: This table shows estimates of "agreement" probabilities between FUZZY HART-S and two other clustering methods on the Soybean and Mushroom data sets (P_{as} and P_{am} columns, respectively, for each method). The *'s indicate that Froggit could not find exactly N_c categories, in which case the probabilities were taken from a partition with the closest number of categories. The parameter settings of FUZZY HART-S were identical to those used in section 4.1.

(Group), a cut of the tree was taken at the desired level. For Froggit, one of its parameters, which controls the specificity (and therefore the number) of top-level classes in the hierarchy, was varied until the desired number was produced. The appropriate cut of the tree was made through the use of the resolution parameter (ϵ) in a trained network (varying it between 0 and its value at training), which typically resulted in cuts spanning multiple Fuzzy ART modules.

The results show that the FUZZY HART-S method produces clusterings that are correlated (at least according to our measure) to the other methods, and particularly to the statistical method we used. The relative level of correlations (with respect to the P_r values) increases with the number of categories (even though the absolute values decrease slightly). Also, the probabilities vary little with the data sets.

5 Conclusions

In this paper, we proposed a modular neuro-fuzzy architecture (FUZZY HART-S) that is capable of stable and fast learning of hierarchical clusterings of input sequences by dividing classes in each stage into smaller, more specific ones. The experiments demonstrated the representational and learning capabilities of the FUZZY HART-S, and also suggested that the developed clusterings are not unrelated to those produced by a classical and a machine learning clustering algorithm. This allows us to address issues such as extracting rules from trained FUZZY HART-S networks (see [7], for example), and/or optimising the developed hierarchies. Along the latter line, it may be useful to view FUZZY HART-S (especially in fast learning mode) as implementing the first "sorting" stage in a multi-stage clustering mechanism [9].

Furthermore, the component Fuzzy ART networks have several useful properties such as L_1 distance metric (i.e., multiplication is not required at each synapse) and good performance with as few as four bits of precision, which facilitate hardware implementation [3].

It is also worth noting that FUZZY HART-S bears similarities with the Cascade Fuzzy ART (CFART) network [11]. Namely, both use a cascade of Fuzzy ART networks to learn class hierarchies. The main difference is that learning in FUZZY HART-S is entirely top-down (with input patterns entering at the top level of the tree) whereas CFART uses bottom-up learning (similar to HART-J [1]) guided by a top-down search process.

6 Acknowledgements

This work was in part supported by Victoria University of Wellington (Research Grant No. RSCH/596/582). We wish to acknowledge the programming support of David Gilligan.

References

[1] G. Bartfai. An ART-based modular architecture for learning hierarchical clusterings. *Neurocomputing*, 13(1):31–46, September 1996.

[2] G. Bartfai and R. White. Adaptive resonance theory-based modular networks for incremental learning of hierarchical clusterings. *Connection Science*, 9(1):87–112, March 1997.

[3] M. Blume and S. C. Esener. An efficient mapping of fuzzy ART onto a neural architecture. *Neural Networks*, 10(3):409–411, 1997.

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

[5] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(5):698–713, September 1992.

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

[7] G.A. Carpenter and Ah-Hwee Tan. Rule extraction: from neural architecture to symbolic representation. *Connection Science*, 7(1):3–27, 1995.

[8] Brian P. Dawkins, Peter M. Andreae, and Paul M. O'Connor. Analysis of olympic heptathlon data. *Journal of the American Statistical Association*, 89(427):1100–1106, 1994.

[9] Doug Fisher. Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research*, 4:147–179, 1996.

[10] A. D. Gordon. *Classification: Methods for the Exploratory Analysis of Multivariate Data*. London: Chapman and Hall, 1981.

[11] H.-L. Hung, H.-Y.M. Liao, S.-J. Lin, and K.-C. Fan. Cascade fuzzy ART: a new extensible database for model-based object recognition. In *Proceedings of Visual Communications and Image Processing '96*, pages 187–198. SPIE – The International Society for Optical Engineering, 1996.

[12] Y. S. Kim and S. Mitra. An adaptive integrated fuzzy clustering model for pattern recognition. *Fuzzy Sets and Systems*, 65(2–3):297–310, August 1994.

[13] Bart Kosko. *Neural Networks and Fuzzy Systems*. Prentice-Hall, Inc., 1992.

[14] C. J. Merz and P. M. Murphy. UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/mlrepository.html]. Technical report, Department of Information and Computer Science, University of California, Irvine, CA, 1996.

[15] S. Mitra. Fuzzy inferencing with ART networks. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 2, pages 1230–1234. IEEE Press, 1994.

[16] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.