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

Department of Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5328
Fax: +64 4 495 5232
Internet: Tech.Reports@comp.vuw.ac.nz

Inferring Hierarchical Categories with ART-based Modular Neural Networks

Guszti Bartfai

Technical Report CS-TR-96/4
May 1996

Abstract

This paper presents a generic modular neural network architecture built up of Adaptive Resonance Theory (ART) networks. The network has a more powerful knowledge representation than its component ART networks in that it is capable of representing *class hierarchies* as opposed to simply partitioning the input data set. At the same time, the network retains the main properties of ART networks like fast and stable incremental learning, parallel search and so on. Three different implementations of the generic network (SMART, HART-J and HART-S) are described, each being capable of unsupervised learning of hierarchical clusterings of arbitrary sequences of input patterns. Each hierarchical ART network produces slightly different clusterings due to the differences in the interactions between the component ART modules and the input each module receives. Two examples, produced by computer simulations, illustrate the developed 2 and 3-level clusterings on a machine learning benchmark dataset. The architectures are also general enough that the types of ART modules can be changed (e.g. from binary ART to Fuzzy ART) — without affecting the main properties of the network — to suit the problem domain better. We also ask questions about how this model might be extended and/or modified to make it suitable for supervised learning, to make better use of the structured knowledge present in the network, to make the resulting clusterings more robust, and to create new ART modules during learning. We speculate that all these could be achieved by implementing appropriate control mechanism in the networks.

Keywords: Hierarchical clustering, Adaptive Resonance Theory (ART), Knowledge representation.

Publishing Information

This paper has been accepted for presentation at the Neural Networks and Structured Knowledge (NNSK) workshop of the 12th European Conference on Artificial Intelligence (ECAI'96) in Budapest, Hungary.

Author Information

Guszti Bartfai is a lecturer at the Department of Computer Science. He welcomes any comments and suggestions about this paper. His e-mail address is `guszti@comp.vuw.ac.nz`.

1 Introduction

Since the re-emergence of neural computing in the late 80's, much progress have been made towards closing the gap between symbolic AI and neural networks. Cross-fertilisation of ideas has been fruitful. Hybrid systems that include both AI and neural parts have also shown to be capable to achieve more than either could on its own (see for example [18]). In order to advance further, we need to identify areas in which either side is weaker than the other. Such a weak area for neural networks is the treatment of structured knowledge. Most of the "classical" networks like the back-propagation network do not normally store structure explicitly, and typically learn to associate input patterns with desired target categories based on statistical properties. The only evident structure they have is the connection topology and whatever can be represented with a set of connection weights and (often distributed) patterns of activity. The discovered underlying "knowledge" is often buried in the hidden layer(s). However, the brain exhibits more structuredness with certain areas playing specific roles in information processing. For example, neurons of the neocortex, where most high-level cognitive processes occur, are of few types, and their connections are organised in relatively few schemes or basic circuits [20]. The primate visual system has also been found to be highly modularised with extensively interconnected subdivisions performing linear and non-linear filtering, object recognition and so on [21].

There are various kinds of structured knowledge that cannot be represented straightforwardly with "BP-like" networks, or the kinds of representation they discover is not particularly suitable for describing the problem domain and/or interpreting by humans. Such more complex structures are higher-order relationships/dependencies among input attributes, spatial/geometrical relationships, class hierarchies and so on.

In this paper, we concentrate on one type of structured knowledge: representing and learning of hierarchical categories. In many applications, extracting class hierarchies provides distinctive advantages like easier human interpretation, database design/query. Hierarchical clustering has also been found in vivo. An example for such a network is discussed in [1], where it was shown that the paleocortex performs hierarchical clustering. The authors also hypothesise that "networks and circuitry of similar design in other brain regions contain computational elements sufficient to construct perceptual hierarchies for use in recognizing environmental cues", and suggest the "possibility that this biologically generated mechanism for hierarchical clustering may be a routine part of perceptual recognition memory behavior in animals and humans".

It is well-known that a competitive layer of neurons is capable of performing unsupervised classification of input patterns. There have been individual separate attempts to "break out" and use some sort of hierarchical knowledge to achieve certain specific goals (see, for example, [19, 15, 16]). Besides the lack of generality, however, they also have several drawbacks like long training time, finding and getting stuck in suboptimal solutions, and instability against changes in the environment.

An alternative model that is based on solid neurophysiological and cognitive principles, and also overcomes the above problems, is the Adaptive Resonance Theory (ART) neural network [6]. They are more suitable for the systematic construction of modular networks because of their well-defined interfaces as well as features that most other networks lack. In specific, ART networks have the ability to create new output nodes (i.e. categories) dynamically, and do not suffer from the problem of forgetting previously learned categories if the environment changes. In other words, it provides a solution to the stability-plasticity dilemma [10]. Despite their useful properties, however, ART networks by themselves can only learn input categories at a given level of specificity (i.e. partition the input space), which de-

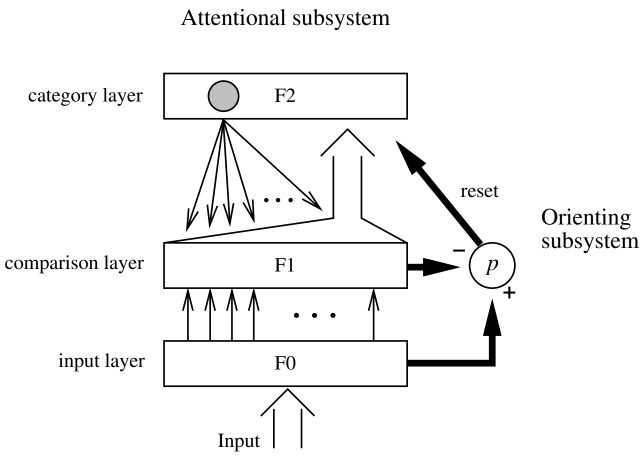

Figure 1: Architecture of the ART network. Thick arrows denote combined or non-specific signals. Winning F2 category nodes are selected by the *attentional* subsystem. Category search is controlled by the *orienting* subsystem. If the degree of category match at the F1 layer is lower than the vigilance level ρ, the non-specific reset signal will be triggered, which will deactivate the current winning F2 node for the period of presentation of the current input.

pends on a global parameter called *vigilance*.

In this paper, we discuss networks, built up of ART modules, that are capable of inferring class hierarchies from input data. The representational capabilities of each hierarchical model are demonstrated by computer simulations.

First, we review the ART network in section 2. The hierarchical ART models and their properties are discussed in section 3. Conclusions are drawn in section 5 after some discussion in section 4, in which several questions are raised relating to the representation, learning mechanism and applicability of hierarchical ART networks.

2 ART Neural Networks

This section describes the ART architecture at a level of detail that is necessary for understanding the rest of the paper. Unless otherwise stated, we refer to ART1, the binary version of ART [6][1].

An ART network is built up of three layers: the *input* layer (F0), the *comparison* layer (F1) and the *recognition* layer (F2) with N, N and M neurons, respectively (see figure 1). The input layer stores the input pattern, and each neuron in the input layer is connected to its corresponding node in the comparison layer via one-to-one, non-modifiable links. Nodes in the F2 layer represent input categories. The F1 and F2 layers interact with each other through weighted bottom-up and top-down connections that are modified when the network learns. There are additional gain control signals in the network (not shown in figure 1) that regulate its operation, but those will not be detailed here. The learning process of the network can be described as follows.

At each presentation of a non-zero binary input pattern $\mathbf{x}$ ($x_j \in \{0,1\},\ \ j = 1, 2, ..., N$),

[1] There are ART networks that accept both continuous and binary inputs [7, 9].

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 i in the F2 layer, the bottom-up activation T_i is calculated, which can be expressed as

$$T_i = \frac{|\mathbf{w}_i \cap \mathbf{x}|}{\beta + |\mathbf{w}_i|} \quad i = 1, ..., M \tag{1}$$

where $|\cdot|$ is the norm operator ($|\mathbf{x}| \equiv \sum_{j=1}^{N} x_j$), $\mathbf{w}_i$ is the (binary) weight vector (or prototype) of category i [2], and $\beta > 0$ is the *choice parameter* [8]. Then the F2 node I that has the highest bottom-up activation, i.e. $T_I = \max\{T_i | i = 1, ..., M\}$, is selected (winner-take-all competition). The weight vector of the winning node ($\mathbf{w}_I$) will then be compared to the current input at the comparison layer. If they are similar enough, i.e. they satisfy the

$$\frac{|\mathbf{w}_I \cap \mathbf{x}|}{|\mathbf{x}|} \geq \rho \tag{2}$$

matching condition, where ρ is a system parameter called *vigilance* ($0 < \rho \leq 1$), F2 node I will capture the current input and the network learns by modifying $\mathbf{w}_I$:

$$\mathbf{w}_I^{new} = \eta(\mathbf{w}_I^{old} \cap \mathbf{x}) + (1 - \eta)\mathbf{w}_I^{old} \tag{3}$$

where η is the learning rate ($0 < \eta \leq 1$). All other weights in the network remain unchanged.

If, however, the stored prototype $\mathbf{w}_I$ does not match the input sufficiently, i.e. condition (2) is not met, the winning F2 node will be reset (by activating the reset signal in figure 1) for the period of presentation of the current input. Then another F2 node (or category) is selected with the highest T_i, whose prototype will be matched against the input, and so on. This "hypothesis-testing" cycle is repeated until the network either finds a stored category whose prototype matches the input well enough, or allocates a new F2 node. Then learning takes place according to (3).

It can be shown that 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 number of developed categories can be controlled by setting ρ: the higher the vigilance level, the larger number of more specific categories will be created. (If $\rho = 1$, the network will create a new category for every unique input.)

Also, as a consequence of its stability-plasticity property, the network is capable of learning "on-line", i.e. refining its learned categories in response to a stream of input patterns (as opposed to being trained "off-line" on a finite training set).

With an ART network, therefore, we can find classes in an input set at a given level of specificity. If we want a more general view of the input set, we need to train an ART network with low vigilance; for a more detailed view, we need to retrain the network with high vigilance, or simply increase vigilance during training. In any case, however, the network will only be able to "see" an input set at a given level of specificity at any given time. There is no relationship between any pairs of category prototypes except that they are in the same network and compete with each other. With only one level of competitive layer, the network is clearly not capable of representing structured knowledge.

[2]For simplicity, we assume that the network is in fast learning mode ($\eta = 1$, see below), in which case the bottom-up and top-down weights for each F2 category i are identical ($= \mathbf{w}_i$, see [6]).

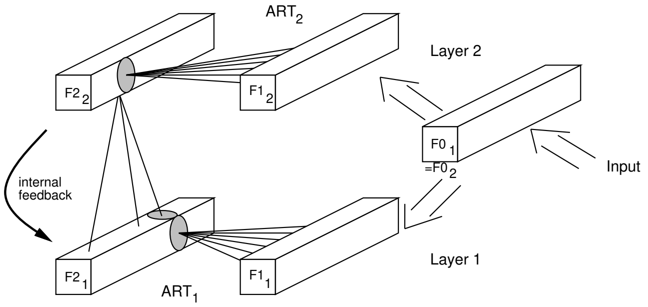

Figure 2: Architecture of a two-layer SMART network. It is composed of two ART modules (in layer 1 and layer 2) whose comparison layers ($F2_1$ and $F2_2$) are connected via associative links. Both modules perform clustering of the same input patterns and learn to associate categories developed at different vigilance levels. The ART_1 module attempts to predict the category at ART_2. If it fails, the ART_2 module will raise the vigilance level ρ_1 at ART_1(through an internal feedback mechanism) to cause a mismatch at layer 1, which will subsequently search for a better, i.e. more consistent, ART_1 category.

3 Hierarchical ART Networks

In this section, we discuss three modular ART-based networks we have proposed, which are capable of inferring class hierarchies from a training set. These are demonstrated by results of computer simulations where the networks were trained on the Zoo machine learning benchmark database [17].

The first model is the concept of Self-consistent Modular ART (SMART) network [2]. The architecture of a two-layer SMART network [3] is shown in figure 2. In this network, two-level class hierarchies are learnt by making the ART_2 and ART_1 modules develop broader and more specific classes, respectively (by setting $\rho_2 < \rho_1$). Each ART_2 category will have one or more ART_1 sub-categories, but no two ART_2 categories share the same ART_1 category. This self-consistent hierarchy is represented explicitly in the network through the $F2_1 \rightarrow F2_2$ associative links.

Table 1 shows a typical internal representation of the SMART network. (The actual clusterings varied depending on the order of presentation of the input patterns in each training session.) Columns $\mathbf{w}_j^2$ ($j = 1, 2, 3$) show the ART_2 category prototypes, and columns $\mathbf{w}_j^1$ ($j = 1, \ldots, 13$) contain the prototypes of ART_1 categories. In this example, three sub-classes were developed at layer 1 with nodes $1 \ldots 5$, $6 \ldots 9$ and $10 \ldots 13$ belonging to parent classes 1, 2 and 3, respectively, in layer 2. The table shows that there is a clear hierarchy in the network in that a more general class is connected to several sub-classes that inherit some features from their parent class as well as contain features specific to that particular sub-class. We can also see that a feature has a specific value in an ART_2 prototype if and only if all its sub-classes have that feature set to the same value (e.g. features "breathes" and "tail" in $\mathbf{w}_1^2$).

The next architecture is HART-J, a hierarchical ART network that is capable of learning class hierarchies by joining (more specific) sub-classes to form more general classes [4]. The architecture of a two-layer HART-J network is shown in figure 3. Both layers in a HART-J network perform clustering of their respective input patterns. However, unlike the SMART

[3] It can be generalised to more than two layers, but this will not be discussed here.

Attributes	$\mathbf{w}^2_1$	$\mathbf{w}^1_1$	$\mathbf{w}^1_2$	$\mathbf{w}^1_3$	$\mathbf{w}^1_4$	$\mathbf{w}^1_5$	$\mathbf{w}^2_2$	$\mathbf{w}^1_6$	$\mathbf{w}^1_7$	$\mathbf{w}^1_8$	$\mathbf{w}^1_9$	$\mathbf{w}^2_3$	$\mathbf{w}^1_{10}$	$\mathbf{w}^1_{11}$	$\mathbf{w}^1_{12}$	$\mathbf{w}^1_{13}$
	Category prototype vectors															
hair	–	–	–	–	no	no	–	no	–	no	no	–	–	–	no	no
feathers	–	–	no	–	yes	no	no	no	no	no	no	no	no	no	no	no
eggs	–	–	no	–	yes	yes	–	yes	yes	–	yes	–	no	yes	yes	yes
milk	–	–	–	–	no	no	no	no	no	no	no	–	yes	–	no	no
airborne	–	–	no	–	yes	no	–	no	–	no	no	no	no	no	no	no
aquatic	–	–	no	–	no	no	–	yes	no	–	no	–	–	yes	yes	–
predator	–	–	–	–	–	yes	–	–	–	yes	yes	–	–	–	–	–
toothed	–	–	–	–	no	yes	–	–	no	yes	no	–	yes	–	yes	–
backbone	–	yes	–	yes	yes	yes	–	–	–	yes	no	yes	yes	yes	yes	yes
breathes	yes	yes	yes	yes	yes	yes	–	–	yes	–	no	–	yes	–	no	yes
venomous	–	no	–	no	no	yes	–	–	–	yes	no	no	no	no	no	no
fins	–	no	no	–	no	no	–	–	no	no	no	–	–	–	yes	no
legs	–	–	–	2	2	0	–	–	–	0	0	–	–	–	0	–
tail	yes	yes	yes	yes	yes	yes	–	–	–	yes	no	–	–	yes	yes	–
domestic	–	no	–	no	–	no	–	no	–	no	no	–	–	–	no	no
catsize	–	no	–	yes	no	no	–	–	–	no	no	–	–	–	–	–
Category size	24	7	1	6	9	1	20	8	10	1	1	57	38	2	11	6

Table 1: A typical example for the developed internal representation in the SMART network. Instances (of animals) are described with 15 boolean and 1 numeric attributes. A "–" attribute value should be interpreted as "don't care". Layer 1 and layer 2 vigilance levels were 0.4 and 0.1, respectively. In this training run, the network developed 3 and 13 categories at layers 2 and 1, respectively. The number of input patterns in each category is shown in the bottom row.

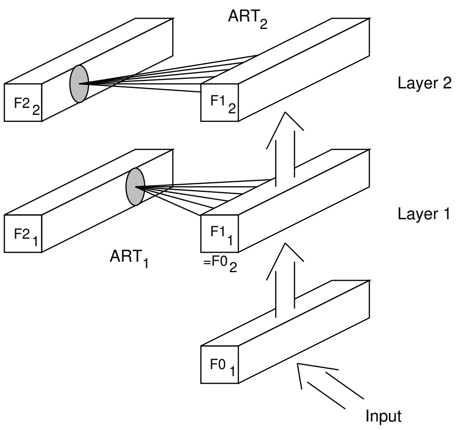

Figure 3: Architecture of a two-layer HART-J network. All vertical connections are one-to-one. The input layer of ART_2 is identical to the comparison layer of ART_1.

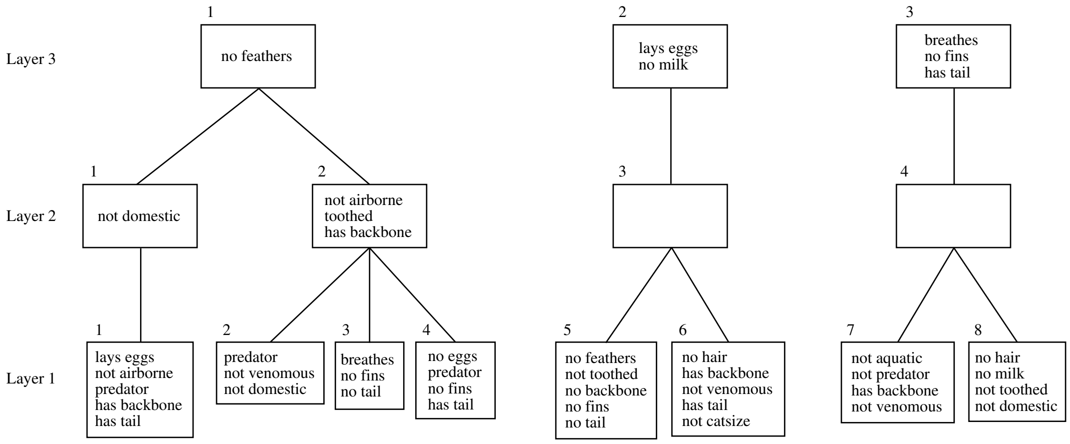

Figure 4: An example of a class hierarchy found by a three-layer HART network. Each box represents a category, and contains a list of critical and distinctive features of that class. All other attributes not shown were "don't care". For example, class 1 in layer 2 has "feathers" and "domestic" as critical features (both are absent) with "domestic" being the distinctive one (from class 2) at that level. Vigilance levels at layers 1, 2 and 3 were 0.4, 0.2 and 0.1, respectively.

network, the ART_2 module learns to cluster not the input patterns, but the category *prototypes* the ART_1 module develops in response to the original input patterns. This way, the ART_2 module develops broader, or more general, categories than ART_1 if $\rho_2 \leq \rho_1$ [4]. The hierarchy is not represented explicitly (like in the SMART network), but "emerges" as input patterns are presented.

Expanding the HART-J network to more layers is relatively straightforward. (There is also a limit to the number of layers needed on a given training set [4]). As an example, figure 4 depicts the class hierarchy that a three-layer HART-J network found during a training session. It shows how categories become increasingly specific (with more critical features) at lower levels. Note the empty boxes (3 and 4) in layer 2, which means that those categories were found identical to their parents (2 and 3 in layer 3) at the given vigilance level in layer 2. Only at layer 1 (with higher vigilance) were those categories split up into smaller, more specific ones. (We observed that not all developed clusterings looked "realistic". This only indicates that using the given 16 attributes, and without a bias or weightings of any of the features [which humans certainly have], those were *plausible* classes for the network to discover.)

The third architecture (HART-S [5]) is a modular ART network whose overall architecture is similar to HART-J. However, here the idea is to form the most general level of clusters first and then successively splitting those clusters into more specific ones. Only the *difference* between an input to a particular ART module and the prototype of the category of that input is propagated to the next ART module. The input signal to the first module is the difference between the input pattern and the most general category (which contains all training examples), so is equivalent to the input pattern itself. Only when such differences are greater than a size determined by a network parameter called *resolution* is the next ART module activated. This new parameter allows us to control the depth of a classification tree, while we can think of vigilance levels as parameters that affect the width of the tree.

A distinctive feature of the HART-S network is the existence adaptive associative connections between the F2 layers of successive ART modules, which 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 (which are not needed in HART-J) so the correct hierarchies can be maintained, since the individual ART modules cluster *differences* only.

Category prototypes are often duplicated in HART-J as prototypes are propagated on whether or not any clustering of them is achieved. Prototypes in HART-S by comparison always differ and furthermore information contained in one prototype is not copied in sub-cluster prototypes, as occurs HART-J. A difficulty with HART-S is in predicting the number of F2 nodes needed in lower layers of a network due to the links created in hierarchical memory during learning. A higher setting of the resolution parameter tends to reduce the number of such nodes used to learn a particular set of inputs.

Also worth noting is the difference in the way the component ART networks interact in the SMART and HART networks. In the SMART network, each ART module receives the original input pattern and processes it in parallel. As a consequence, the hierarchy of developed categories must be assured explicitly, otherwise a class at one level can belong to two or more classes at a higher level. This consistency is maintained via associative links between categories at subsequent levels and an internal feedback mechanism. This, however, results in the SMART network developing many more categories at lower levels than comparable HART networks [3]. This drawback might be expected to be offset by the fact that the component ART modules work in parallel as opposed to HART where each ART module must wait until the previous layer has finished searching for a suitable category for its input. It turns out to be not true, however, because the SMART network must spend a comparable amount of time during the initial period of training to resolve conflicts due to mismatching (non-hierarchical) categories. The total training time (in average number of epochs) was found to be very similar [3]. In conclusion, it appears that the HART-J and HART-S networks offer an overall better architecture for learning hierarchical clusterings.

4 Discussion

A discussion on the HART-J and HART-S networks, their properties, and their relationship to other neural network models is presented in [5]. One of the main distinctive features of the hierarchical ART networks we have proposed is that they were designed with the explicit goal of "going beyond simple partitioning of the input space" in mind. We see them as an attempt to create networks that have an increased representational power while retaining the common properties of artificial neural networks (e.g. learning, generalisation, parallel distributed processing, etc.). These modular ART networks have a more powerful knowledge representation than single competitive learning networks (hierarchical clustering vs. partitioning) the same way multi-layer Perceptrons are more powerful than single-layer ones (e.g. representing arbitrary boolean functions as opposed to linearly separable ones). Furthermore, the network architecture is not application specific (unlike NIRS [11], for example), which means that it will probably work on many problem domains where the task is to learn stable clusterings of (potentially infinite) sequences of input data. A real-world application of a network similar to HART-J is described in [14]. We are currently working on testing the hierarchical ART networks on larger databases, and are also looking at the comparison of hierarchical ART networks against statistical and machine learning clustering algorithms.

There are a number of issues and questions that have been raised as a result of our investigations:

- *How can these networks be used in supervised learning of target categories (or arbitrary output patterns)?* We think that this is possible by injecting an appropriate control mechanisms (via, for example, non-specific control signals like the ones in ART) into the architecture to guide its search and prediction as well as to correct itself after making a wrong prediction.

- *How can clusterings be made more robust, i.e. less dependent on the order of input presentation?* Although we are not convinced that this is always a desirable goal, we think that the architecture is generic enough that this could be achieved by "plugging in" more suitable network models (e.g. Gaussian ARTMAP network [22]).

- *In what other ways can structured knowledge represented in the network be used?* We believe this richer representation allows the network, for example, to make use of its categories at multiple levels to enhance its prediction performance. It could also implement a more complex learning and search (matching) mechanism that involve multiple layers (e.g. a higher level guiding the search of a lower one by providing feedback to it).

- *How can the dynamic creation (and freeing up) of whole ART modules be achieved (thus the network being able to change the granularity of clustering during learning)?* We believe this would be a very useful property, but also think that this may not be achieved unless the network has an explicit utility function that it maximises, and therefore drives its learning process, much like machine learning mechanisms do (e.g. COBWEB [12]). One way to implement dynamic creation of new levels is to treat whole ART modules the same way as individual nodes are treated within a single ART network (by letting a pool of initially uncommitted nodes take part in the competition) thereby creating a self-similar network architecture (which is one of the advantages of structuredness [13]).

5 Conclusion

In this paper, we discussed three hierarchical network architecture built up of ART modules that are capable of learning stable hierarchical clusterings of arbitrary sequences of input patterns. The internal representations of these networks demonstrated (2/3-layer) hierarchies of classes, where lower-level classes "inherited" features from ones at higher-levels, and had their own distinctive features as well.

We believe that the ideas and results presented here show the benefits of modularity in learning more complex relationships from the input environment.

We hope that the issues raised here will stimulate discussions (which may raise further questions) among participants, and so this paper will contribute to the success of this workshop.

Acknowledgements

I would like to thank the referees for their comments which helped improve this paper.

References

[1] J. Ambros-Ingerson, R. Granger, and G. Lynch. Simulation of paleocortex performs hierarchical clustering. *Science*, 274:1344–1348, 1990.

[2] G. Bartfai. Hierarchical Clustering with ART Neural Networks. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 2, pages 940–944. IEEE Press, 1994.

[3] G. Bartfai. A Comparison of Two ART-based Neural Networks for Hierarchical Clustering. In K. Kasabov and G. Coghill, editors, *ANNES'95, The Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pages 83–86. IEEE Computer Society Press, November 1995.

[4] G. Bartfai. An ART-based Modular Architecture for Learning Hierarchical Clusterings. *Neurocomputing*, 1996. In press.

[5] G. Bartfai and R. White. ART-based Modular Networks for Incremental Learning of Hierarchical Clusterings. *Connection Science*, 1996. Submitted.

[6] 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.

[7] G.A. Carpenter and S. Grossberg. ART2: Self-Organizing of Stable Category Recognition Codes for Analog Input Patterns. *Applied Optics*, 26(23):4919–4930, December 1987.

[8] G.A. Carpenter, S. Grossberg, and J.H. Reynolds. ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network. *Neural Networks*, 4:565–588, 1991.

[9] 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.

[10] Gail A. Carpenter and Stephen Grossberg. The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *IEEE Computer*, 21(3):77–88, March 1988.

[11] Thomas P. Caudell, Scott D. G. Smith, Richard Escobedo, and Michael Anderson. NIRS: Large Scale ART-1 Neural Architectures for Engineering Design Retrieval. *Neural Networks*, 7(9), 1994.

[12] D.H. Fisher. Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning*, 2(2):139–172, September 1987.

[13] B. Happel and J. Murre. Design and evolution of modular neural network architectures. *Neural Networks*, 7(6/7):985–1004, 1994.

[14] Shigekazu Ishihara, Keiko Ishihara, Mistsuo Nagamachi, and Yukihiro Matsubara. arboART: ART based hierarchical clustering and its application to questionnaire data analysis. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 1, pages 532–537. IEEE Press, 1995.

[15] Jouko Lampinen and Erkki Oja. Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, 2:261–272, 1992.

[16] Tao Li, Luyuan Fang, and Ken Q-Q. Li. Hierarchical classification and vector quantization with neural trees. *Neurocomputing*, 5:119–139, 1993.

[17] 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.

[18] Dean A. Pomerleau. *Neural network perception for mobile robot guidance*. Kluwer Academic Publishers, 1993.

[19] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group. *Parallel Distributed Processing; Explorations in the Microstructure of Cognition*, volume 1: Foundations, chapter 5, pages 151–193. The MIT Press, 1986.

[20] G. M. Shepherd. *The Synaptic Organization of the Brain*. Oxford University Press, New York, 1974.

[21] David C. Van Essen, Charles H. Anderson, and Felleman Daniel J. Information processing in the primate visual system: An integrated systems perspective. *Science*, 255(5043):419–423, January 1992.

[22] J. R. Williamson. Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. Technical Report CAS/CNS-95-003, Boston University, Center of Adaptive Systems and Department of Cognitive and Neural Systems, February 1995.