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

A Comparison of Two ART-based Neural Networks for Hierarchical Clustering

G. Bartfai

Technical Report CS-TR-95/11
July 1995

Abstract

This paper compares two modular neural network architectures built up of Adaptive Resonance Theory (ART) networks that can develop stable two-level hierarchical clusterings of arbitrary sequences of binary input patterns. In particular, it contrasts typical class hierarchies that the networks found on a machine learning benchmark database. It is proposed that the main difference between the two clusterings are the direct consequence of the existence or absence of an internal feedback mechanism and explicit associative links between a higher-level class and its sub-classes.

Publishing Information

A shortened version of this report is to appear in *Proceedings of ANNES'95, The Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand*, published by IEEE Computer Society Press.

Author Information

Guszti Bartfai is with the Department of Computer Science, and his e-mail address is `guszti@comp.vuw.ac.nz`

1 Introduction

The ability to learn about the environment without a teacher has long been considered an important characteristic of intelligent systems. Unsupervised learning can be found both at sensory-level of mammals, and at higher, cognitive levels of humans. Therefore, it has been an important topic in neural network research (e.g. [8], [11], [12], [7]) in recent years. Unsupervised learning networks can perform pattern classification by having a "competitive" layer of neurons. Upon presentation of an input pattern, the node whose weight vector is the closest to the current input vector (in some distance metrics) will become the only active (winner-take-all competition), and subsequently be allowed to learn by modifying its connection weights. After repeated exposure to the input environment, the network will store a prototypical element of each input category, or class, in its connections weights. The number and size of classes the network finds on a given training set is equal to the number of neurons. The more neurons the network has, the larger number of more specific categories it will find. Because of its "winner-take-all" characteristics, a single network will only be able to *partition* the input space, i.e. find *one* category for each input pattern. However, environments are often more complex and may exhibit a hierarchical structure (e.g. classification of animals), which humans can learn without apparent difficulty. It is, therefore, worthwhile investigating neural network architectures that can learn class hierarchies. There have been a few attempts to create such networks (e.g. [12]) where modularity offers distinctive advantages.

For the systematic construction of modular networks, *Adaptive Resonance Theory* (ART) neural network architectures [3] appear to be particularly suitable 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. They too, however, can only develop input categories at a given level of specificity, which depends on a global parameter called *vigilance*.

In this paper, we look at two modular ART-based architectures (SMART, for "Self-consistent Modular ART" and HART, for "Hierarchical ART") that overcome the limitations of single ART modules, and are able to learn hierarchical clusterings of input sequences at two different levels of specificity. In particular, we interpret and contrast their developed internal representation when trained on a machine learning benchmark database.

Section 2 briefly summarises the main features of ART neural networks that are sufficient for understanding the rest of the paper. The SMART and HART networks are described briefly in sections 3.1 and 3.2, respectively. Experiments that were carried out are discussed in section 4, and conclusions are drawn in section 5.

2 ART neural networks

Adaptive Resonance Theory architectures [3] are neural networks that develop stable recognition codes in real time by self-organisation in response to arbitrary sequences of input patterns[1]. They were designed to solve the "stability-plasticity dilemma" that every intelligent machine learning system has to face: how to keep learning from new events without forgetting previously learned information.

An ART network is built up of three layers: the *input* layer (F0), the *comparison* layer (F1) and the *recognition* layer (F2) with M, M and N neurons, respectively (see modules

[1]There are ART networks that accept both analogue and binary inputs [4, 6]. For the rest of the paper, however, we refer to the binary version (ART1 [3]) unless otherwise stated.

ART_1 and ART_2 in figures 1 and 2). The input layer is only used to store input patterns and is connected directly to the comparison layer. 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.

At each presentation of a non-zero binary input pattern $\mathbf{x}$ ($x_i \in \{0,1\}, \quad i = 1,\ldots,M$), 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 bottom-up activation T_j is calculated, which can be expressed as

$$T_j = \frac{|\mathbf{w}_j \cap \mathbf{x}|}{\beta + |\mathbf{w}_j|} \quad j = 1,\ldots,N \tag{1}$$

where $|\cdot|$ is the norm operator ($|\mathbf{x}| \equiv \sum_{i=1}^{M} x_i$), $\mathbf{w}_j$ is the (binary) top-down template (or prototype) of category j, and $\beta > 0$ is the *choice parameter* [5]. Then the weight vector of the winning F2 node J, i.e. where $T_J = \max\{T_j : j = 1,\ldots,N\}$, will be compared to the current input at the comparison layer. If they are similar enough, i.e. they satisfy the

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

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

$$\mathbf{w}_J^{new} = \mathbf{w}_J^{old} \cap \mathbf{x} \ . \tag{3}$$

Weights for all other F2 nodes in the network remain unchanged.

If, however, the stored prototype $\mathbf{w}_J$ does not match the input sufficiently, i.e. condition (2) is not met, the winning F2 node will be reset for the period of presentation of the current input. Then another F2 node is selected with the highest T_j, 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 is important to note that once a category is found, the comparison layer (F1) holds $\mathbf{w}_J \cap \mathbf{x}$ until the current input is removed. 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 and size of developed categories can be controlled by setting ρ: the higher the vigilance level, the larger number of smaller, or more specific, categories will be created. There is no relationship between any pairs of category prototypes except that they are in the same network and compete with each other. The network, therefore, with its single layer of category nodes, is not capable of representing (and thus learning) a hierarchy of classes.

3 Two-layer ART-based modular networks

The following sections briefly discuss two modular network architectures that are capable of representing and learning two-level class hierarchies from a training set.

3.1 The SMART network

The architecture of a two-layer SMART network [1] is shown in figure 1. Its connection topology and operation are identical to those of an ARTMAP network [5] except that the two

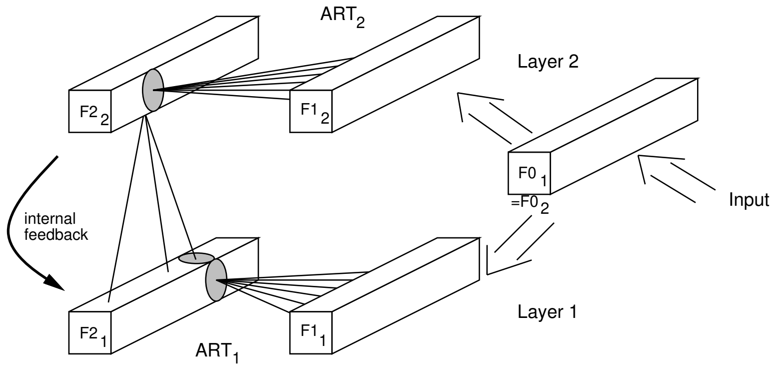

Figure 1: Architecture of a two-layer SMART network. It is composed of two ART modules (in layer 1 and layer 2) whose comparison layers ($F2_1$ and $F2_2$) are connected via associative links. Both modules perform clustering of the same input patterns and learn to associate categories developed at different 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.

ART modules here receive identical inputs and therefore the network performs unsupervised learning.

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.

3.2 The HART network

The architecture of a two-layer HART network [2] is shown in figure 2. Both layers in a HART network perform clustering of their respective input patterns. However, unlike the SMART 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. Furthermore, there is no associative connection between the two recognition layers, nor is there a feedback from layer 2 to layer 1. This way, the ART_2 module develops broader, or more general, categories than ART_1 if $\rho_2 \leq \rho_1$ [2].

Upon presentation of a new input pattern, first an ART_1 node (or sub-class) will be selected, then the ART_2 module chooses a corresponding super-class node. Therefore, each input pattern will have two classes associated with it. This hierarchy is not represented explicitly (like in the SMART network), but "emerges" as input patterns are presented. The same kind of class consistency as in SMART can be guaranteed by choosing ρ_2 and ρ_1 such that $\rho_2 \leq \rho_1$.

4 Experimental results

We carried out experiments on the "zoo" machine learning benchmark database [10] to compare the developed class hierarchies of the SMART and HART networks. The database contains 101 instances of animals described with 18 attributes. Out of these attributes, we used the 15

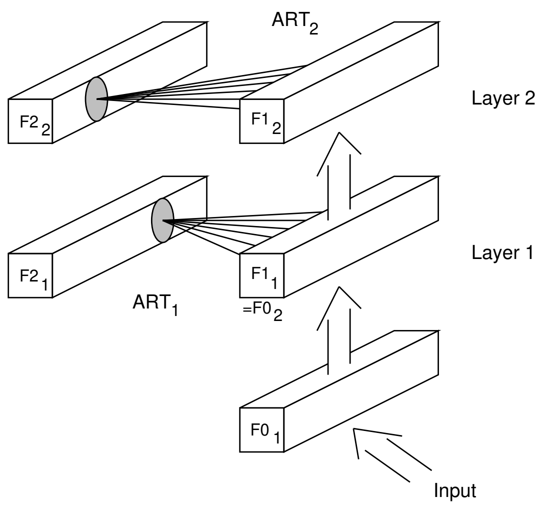

Figure 2: Architecture of a two-layer HART network. All vertical connections are one-to-one. The input layer of ART_2 is identical to the comparison layer of ART_1.

boolean ones that indicate the presence or absence of certain features like "hair", "aquatic", "domestic" and so on. We also used the "number of legs" attribute, which is a set of 6 integers. The target class "type" was ignored. The 18th attribute ("animal name") was simply used as a label for the individual instances. The patterns were presented to the SMART and HART networks in *generalised complement coding* [5, 1]. This way, the input patterns were 36-element binary vectors with the same norm of 16.

The simulations were carried out using a public domain neural network simulator program (`PlaNet` [9]) running under Unix and X-windows.

The network parameters were chosen to be similar to those presented in [5]. In particular, the initial values of the bottom-up weights in both ART modules were chosen in such a way that F2 nodes became active in the order $j = 1, 2, \ldots$ and were small enough so the network selected an uncommitted (or free) node only if $|\mathbf{w}_j \cap \mathbf{x}| = 0$ for all committed nodes. Also, the β choice parameter was taken to be sufficiently small that, among committed nodes, T_j was determined by the size of $|\mathbf{w}_j \cap \mathbf{x}|$ relative to $|\mathbf{w}_j|$.

The networks were trained on the entire database (i.e. on all 101 instances) until stability was reached, i.e. no recoding of ART_1 and ART_2 categories occurred over the whole training set.

4.1 Internal representations of the SMART and HART networks

Table 1 shows a typical internal representation of the SMART network. It is one of the many stable clusterings the network discovered in the training set. 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.

	Category prototype vectors															
Attributes	$\mathbf{w}_1^2$	$\mathbf{w}_1^1$	$\mathbf{w}_2^1$	$\mathbf{w}_3^1$	$\mathbf{w}_4^1$	$\mathbf{w}_5^1$	$\mathbf{w}_2^2$	$\mathbf{w}_6^1$	$\mathbf{w}_7^1$	$\mathbf{w}_8^1$	$\mathbf{w}_9^1$	$\mathbf{w}_3^2$	$\mathbf{w}_{10}^1$	$\mathbf{w}_{11}^1$	$\mathbf{w}_{12}^1$	$\mathbf{w}_{13}^1$
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. 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.

	Category prototype vectors								
Attributes	$\mathbf{w}_1^2$	$\mathbf{w}_1^1$	$\mathbf{w}_2^1$	$\mathbf{w}_3^1$	$\mathbf{w}_4^1$	$\mathbf{w}_2^2$	$\mathbf{w}_5^1$	$\mathbf{w}_6^1$	$\mathbf{w}_7^1$
hair	–	no	–	–	yes	–	yes	–	yes
feathers	–	yes	–	yes	–	–	yes	–	yes
eggs	–	yes	no	–	no	–	no	no	–
milk	–	no	–	–	yes	yes	yes	yes	yes
airborne	–	–	–	yes	yes	–	yes	–	yes
aquatic	–	–	–	–	–	–	no	yes	–
predator	–	–	–	–	–	–	–	–	no
toothed	–	no	yes	no	–	–	no	yes	–
backbone	–	no	no	no	no	–	no	–	–
breathes	–	no	no	–	–	–	–	no	–
venomous	yes	yes	yes	yes	yes	–	–	–	–
fins	–	–	yes	–	–	–	–	yes	yes
legs	–	–	–	–	–	–	–	–	–
tail	–	–	no	no	no	–	–	–	–
domestic	–	–	yes	yes	–	–	yes	–	yes
catsize	–	–	–	–	–	–	–	yes	–
Category size	62	38	19	4	1	39	16	13	10

Table 2: Example for the stable internal representation of the HART network. 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 2 and 7 categories at layers 2 and 1, respectively. (Note that the above hierarchy "emerges" only as inputs are presented and the corresponding category nodes at each layer are activated.)

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 $\mathbf{w}_1^2$).

Table 2 shows a typical internal representation that the HART network developed.

The hierarchical structure can be seen again: layer 1 categories 1-4 and 5-7 belong to layer 2 categories 1 and 2, respectively. The representative feature of super-class 1 is "venomous", which is inherited by all of its sub-classes. The same can be observed about attribute "milk" in super-class 2 and its sub-classes. Note also that although all sub-classes of super-class 1 share the same value for attribute "backbone", yet it is regarded as non-critical in that super-class (i.e. its attribute value is "don't care"). The network does not have any built in bias towards preferring "venomous" over "backbone" (or "yes" values over "no" for that matter). It simply means that at some stage during the training process, while the network was undergoing self-stabilisation, node 1 in layer 2 attracted an input with its "backbone" attribute "yes", which later ended up in super-class 2 (in one of sub-classes 6 or 7). Since the layer 2 vigilance level was low enough ($\rho_2 = 0.1$), the network accepted a generalised prototype for that class, keeping only "venomous" as critical. This situation, however, cannot occur in the SMART network.

Comparing tables 1 and 2, we can see that the SMART network has almost twice as many sub-classes as the HART network. Most of those are relatively small (10 of them are < 10 in size as opposed to 2 in the HART network). This was found to be the case in the majority of training runs, and is the result of the "category fragmentation" phenomenon reported in [1].

The training time was about 3 epochs, i.e. presentations of the entire training set, on average in both networks.

5 Conclusion

In this paper, we compared two modular network architectures that are capable of learning stable two-level hierarchical clusterings of arbitrary sequences of binary input patterns. We have demonstrated, by training the networks on a machine learning benchmark database, that both networks develop categories from the input sequence at increasing levels of generality. The internal representations of the networks were such that lower-level classes "inherited" features from ones at the higher-level while keeping their own distinctive features as well. The two networks, however, showed markedly different low-level (or layer 1) clusterings. The SMART network had a significantly larger number of ART_1 categories, which is the direct result of the way the higher-level module "forces" its own categorisation onto the lower-level one via the internal feedback mechanism and creating explicit links between the categories while maintaining self-consistency. The HART network, on the other hand, does not have such constraint and the proper hierarchy can be developed only by keeping ρ_2 less than or equal to ρ_1. Moreover, the hierarchy emerged there only when input patterns were presented. This apparent simplicity, compared to SMART, resulted in a more compact class representation.

We believe that the ideas and results presented here show the benefits of modularity in learning more complex relationships from the input environment, and that alternative architectures (or internal "biases") offer different trade-offs.

In the future, we plan to use these networks for more "interesting" tasks such as predicting missing attributes of input patterns, or in network architectures that can develop explicit mappings by supervised learning. We also suggest that the ART modules in these architectures can be replaced by Fuzzy ART modules [6] so they can accept both analogue and binary inputs. In this case, the networks can also operate in *slow learning* mode where the adaptation rule in (3) changes to

$$\mathbf{w}_I^{new} = \eta(\mathbf{w}_I^{old} \cap \mathbf{x}) + (1 - \eta)\mathbf{w}_I^{old} \tag{4}$$

where η is the learning rate $(0 < \eta < 1)$. This way, the developed categories may become more robust against small fluctuations in the environment as well as different orderings of the presentation of input patterns.

We hope that more experiments with both network architectures and their extensions on a range of learning tasks will enable us to understand their behaviour better, and will ultimately extend the repertoire of neural network models available for solving real-world problems.

Acknowledgement

This work was in part supported by Victoria University of Wellington (Research Grant No.RGNT/594/371).

References

[1] 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.

[2] G. Bartfai. An ART-based Modular Architecture for Learning Hierarchical Clusterings. Technical Report CS-TR-95/3, Department of Computer Science, Victoria University of Wellington, New Zealand, February 1995. Accepted for publication in *Neurocomputing*.

[3] 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.

[4] 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.

[5] 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.

[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] R. Hecht-Nielsen. Counterpropagation networks. *Applied Optics*, 26(23):4979–4984, December 1987.

[8] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[9] Yoshiro Miyata. *PlaNet, A Tool for Constructing, Running, and Looking into a PDP Network*, 1991.

[10] P. M. Murphy and D. W. Aha. UCI Repository of Machine Learning Databases [Machine-readable data repository]. Technical report, Department of Information and Computer Science, University of California, Irvine, CA, 1992.

[11] E. Oja. Neural networks, principal components, and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[12] 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.