ARTSTAR: A Supervised Adaptive Resonance Classifier

Talib S. Hussain and Roger A. Browse
Department of Computing and Information Science
Queen's University, Kingston, Ontario*

Appeared in AI'94: Tenth Canadian Conference on Artificial Intelligence (May 16-20, Banff, AB), p. 121-130.

Abstract

A new neural network architecture, ARTSTAR, is presented as a supervised modular extension to the ART2 network. ART2 suffers from deficiencies in terms of consistency and overall capability when applied to classification tasks. ARTSTAR uses a layer of INSTAR nodes to supervise and integrate multiple ART2 modules. Supervision takes the form of feedback to the ART2 output layer whenever a data pattern's true classification is known. This feedback technique may take a variety of forms and can model the supervision implemented in existing supervised extensions to ART networks. A more robust classification performance occurs when several ART2 networks are trained in a supervised manner, each under different conditions, and their outputs integrated during testing. These results are demonstrated in tests of ARTSTAR using handdrawn and computer generated digits. The general functionality of ARTSTAR is extensive, and several further modifications to it are discussed.

1 Introduction

Several different unsupervised neural network architectures have been proposed based on the concept of Adaptive Resonance (Carpenter & Grossberg, 1987a; 1987b; 1990, Carpenter, Grossberg, & Reynolds, 1991; Carpenter, Grossberg, & Rosen, 1991a; 1991b, Carpenter, Grossberg, Markuzon, Reynolds & Rosen, 1992). This class of network models was devised to examine the effect that feedback connections have on the formation of categorizations of input data. In general, these models propose that self-organization of data can be achieved through a mechanism which forms a bottom-up interpretation of a given input and then, based on previously learned patterns, forms top-down expectations as to how the input should be categorized until the interpretations and expectations match, or "resonate", within a certain tolerance level, or "vigilance". Carpenter and Grossberg (1987a) first developed the ART1 network, which accepts binary input, and later extended their work to the ART2 network, which accepts analog input (Carpenter & Grossberg, 1987b).

The ART2 network is ideally suited for tasks requiring data patterns to be clustered into groups of similar elements, and for this purpose it is comparable to conventional clustering techniques (Burke, 1991; Baruah & Welti, 1991). Further, ART2 is a self-organizing network capable of dynamic, on-line learning, and can thus learn to modify its clustering schemes to reflect changes in data characteristics over time. Because of ART2's clustering capabilities, one might expect that if an ART2 network were presented with patterns known to belong to certain pre-determined classes, the network would learn to categorize the data into groups equivalent to those classes. In its basic form, however, ART2 does not generally perform well on classification tasks. One reason is that ART2 does not have the provision to accept supervision, and thus could hardly be expected to form a classification scheme which depicts predesignated classes as well as other supervised networks such as backpropagation. Another difficulty with ART2 as a classifier is that the categorizations developed by ART2 are very sensitive to slight changes in structure and training conditions. For instance, two identical networks trained on the same set of data but presented in different orders may exhibit

* This work was supported by the Natural Sciences and Engineering Research Council (NSERC) and by a grant from the Institute for Robotics and Intelligent Systems (IRIS).

greatly differing classification performance.

The most obvious step necessary to improve ART2 performance in classification is the incorporation of supervision. Previous research on supervision extensions for ART networks show two possible supervision techniques. One approach involves forcing each ART category formed to respond to data from only one pre-established class during training. This change greatly decreases the chance that data from different classes will be categorized as the same during subsequent testing and thus improves ART performance as a classifier. Three different networks have been developed which implement this approach, generally through an architectural extension to ART. They are ARTMAP (Carpenter, Grossberg & Reynolds, 1991), the Adaptive Resonance Associative Map, or ARAM (Tan, 1992), and the SeMi-supervised Adaptive Resonance Theory, or SMART2 (Merz, St. Clair & Bond, 1992). The latter network also introduces a second technique for supervising ART, which is that the number of categories formed by the network is constrained by allowing new categories to be created only when a pattern is initially misclassified. The classification performance of ARAM and SMART2 has actually been found to be comparable to that of backpropagation (Tan, 1992; Merz et al., 1992).

A second method of improving ART2 classification ability involves the use of redundancy to overcome the sensitivity of the ART2 categorization schemes. The assumption of this approach is that ART2 networks which are trained slightly differently will develop different categorization schemes and that these schemes will contain complementary information which can be integrated to achieve a more robust representation. Several methods of exploiting redundancy in multiple classifiers are available, with the simplest being a majority classifier voting approach (Gargano, 1991; Carpenter et al., 1992). In this technique, the classification returned by the majority of classifiers, each applied to the same input, is considered to be the classification of that input.

This paper presents the ARTSTAR network. The development of ARTSTAR has been motivated by the following goals:

1. to provide an ART-based network capable of effective classification, yet still retaining ART's inherent ability to respond to ongoing changes the organization of its inputs.
2. to provide a supervision strategy which is general enough to include previously developed mechanisms and yet offer the opportunity for incorporating new supervision techniques.
3. to provide a mechanism that can integrate results from several different ART2 modules.

An ARTSTAR network consists of a number of ART2 modules connected to a layer of INSTAR nodes (Grossberg, 1982), with each INSTAR node representing a possible class. The INSTAR layer supervises the output layer of each ART2 module as well as integrates the outputs of all the ART2 modules. During training, the instar layer provides feedback to the ART2 modules based on the desired classification of the training input. The feedback influences the importance that each module assigns to its output nodes, and thereby affects the order in which each ART2 module considers its output nodes as possible winners. For each winning output node selection in an ART2 module, INSTAR learning associates it with the INSTAR node that represents the desired class. Over time, the connections between the ART2 output nodes and INSTAR layer come to represent the incidence of a given ART2 category being associated with a given class.

After learning has taken place, the winning category of an ART2 module will activate all INSTAR nodes in proportion to the probability of that class being correct. Thus, if a given input pattern is presented to a number of redundant modules during testing, each redundant module will return a list of class probabilities. Through the simple method of summing the probabilities for each class, ARTSTAR integrates these redundant responses and presents a single list of ranked classifications.

The feedback mechanism of ARTSTAR permits the manipulation of system parameters which provides a variety of different supervision strategies, including the 'forcing' method used in ARTMAP, ARAM and SMART2. Further, each of these strategies will exhibit a desirable 'dynamic supervision' due to an inheritance of the dynamic learning properties of ART2. Thus, an arbitrary ARTSTAR network may be trained on data in which true classifications are available only periodically, resulting in two interleaved learning phases: simple ART2 clustering periods in which new instances are added to existing categorization schemes, and supervision periods in which the classifications associated with ART2 categories are verified and updated. The generality of the feedback mechanism, coupled with ARTSTAR's ability to perform dynamic supervision and integrate outputs from separate ART2 modules, suggests that ARTSTAR may be applicable to a wide range of tasks.

2 ARTSTAR Network Architecture

ARTSTAR derives its name from its use of both ART2 and INSTAR modules thus it is appropriate to begin the description of ARTSTAR with a brief outline of the ART2 network.

2.1 ART2 Processing

The ART2 network consists of three main components, termed by Carpenter and Grossberg (1987b) as the input representation field (or F1 layer), the category representation field (or F2 layer) and the orienting mechanism (see Figure 1).

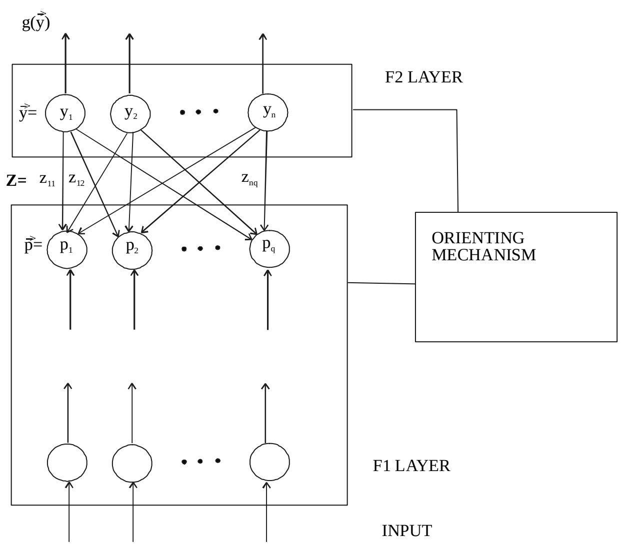

Figure 1: A simplified diagram of ART2 architecture

The nodes of the output layer $\vec{y} = (y_1\ y_2\ \cdots\ y_n)$ are intended to represent the classes into which input patterns are organized. Each output node represents a class of patterns by storing a template pattern as the weights $\boldsymbol{Z} = (\vec{z}_1\ \vec{z}_2\ \cdots\ \vec{z}_n)^T$, $\vec{z}_j = (z_{j1}\ z_{j2}\ \cdots\ z_{jq})$ on its connections into the F1 layer. During an ART2 trial, the input pattern is matched against each of the stored templates, resulting in activations at the output layer which represent the extent of match. At this stage of processing the output layer activations are established as the weighted sum of a layer $\vec{p} = (p_1\ p_2\ \cdots\ p_q)$ within the representation field F1:

$$\vec{y} = \vec{p} \cdot \boldsymbol{Z}^T$$

The element of $\vec{y}$ with the highest activation value is designated as the initial choice of class for the input pattern. The results of this competition, usually denoted as $g(\vec{y})$ presents a value d ($0 < d < 1$ for a given network) for the winning node, and zero otherwise.

The input pattern and the template for the initially chosen node are subjected to a further comparison in the orienting mechanism, and if the match is judged to be within the vigilance level, the initial choice is taken as final and the template for that winning node is updated to more strongly represent the current input. If the match is judged to be outside of the vigilance level, then the output node with the next highest activation value is designated as the initial match and it is subjected to the same processing in the orienting mechanism to determine if it is an acceptable final choice. This process continues through the available choices, in decreasing order of the activation of the output nodes, and if none of them meets the vigilance level test, then a new output node is recruited as a new class and its template is set to the current input.

In ART2 processing, there is no distinction between training and test trials. Every trial results in a classification (the final choice of output node), and with each classification, there is the possibility of updating the weights which represent the stored templates.

The choice of vigilance level will strongly influence the performance of ART2 in the formation of its classification categories. A low vigilance will result in over-inclusive elements, and a high vigilance could, in the worst case, result in a different classification category for each different training pattern.

2.2 Supervision

ARTSTAR incorporates two key properties in addition to those inherited from ART2. The main property is supervision of the ART2 learning process. To achieve this, ARTSTAR includes a layer of INSTAR nodes, one for each true category of the input patterns. The INSTAR nodes both receive input from and send feedback to the ART2 output layer. Supervision is achieved through feedback based on previous associations of ART2 categories with classes, knowledge which is stored in the INSTAR's incoming and outgoing connections.

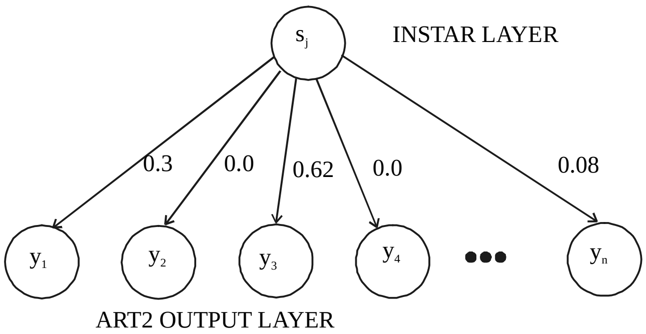

Figure 2: Example connections from an ART2 output node to the INSTAR layer

Consider, as shown in Figure 2, the full connection of a single output element y_i to this layer of INSTAR nodes. In this example, there are weights $\vec{w}_i$ on the connections for each of the INSTAR elements as shown. Through INSTAR learning during the training trials, these weights are set to designate the proportion of training trials for which y_i was the winning ART2 node when each of the true classes was the one associated with the training trial. This means that after training is complete and a test trial is being considered for which y_i is the winning output node, the system can provide an ordered list of possible classifications based on the values of these weights $\vec{w}_i$. If required to establish a single class, then the obvious choice is the class possibility whose INSTAR node has the highest activation, and was thus during training most often associated with the winning output node.

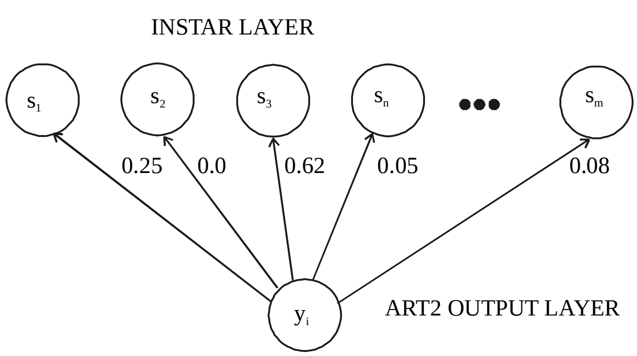

Figure 3: Example connections from INSTAR layer to ART2 output layer

Also consider what it means to have these weights $\boldsymbol{W}$ available during the training phase of ARTSTAR. As shown in Figure 3, the true class of any training trial would have available through $\boldsymbol{W}$ the extent to which previous trials of the same class had been stored on each of the ART2 module output templates. It is possible to use these values to influence the ART2 module to make an initial choice of an output node that has already been chosen most often for training trials of the same true class. The most straightforward way to accomplish this is to augment the computation of the values of the ART2 output layer $\vec{y}$ with feedback from the INSTAR layer:

$$\vec{y} = \vec{p} \cdot \boldsymbol{Z}^{\boldsymbol{T}} + \gamma \vec{s} \cdot \boldsymbol{W}^{\boldsymbol{T}}$$

2.3 Modularity

The second key property introduced in ARTSTAR is modularity. The basic ARTSTAR network can easily be extended such that a number of ART2 networks are connected to the instar layer (see Figure 4). In the modular ARTSTAR, each ART2 module is supervised and associated with pattern classes as in the basic ARTSTAR, but in addition, the class associations of all the winning ART2 output categories are integrated to form a single output. Specifically, the likelihood that the input is of a given class is determined by summing the probabilities for that class indicated by all the winning ART2 nodes. This integration method is similar to the majority classifier voting scheme, and can emulate it under certain training conditions.

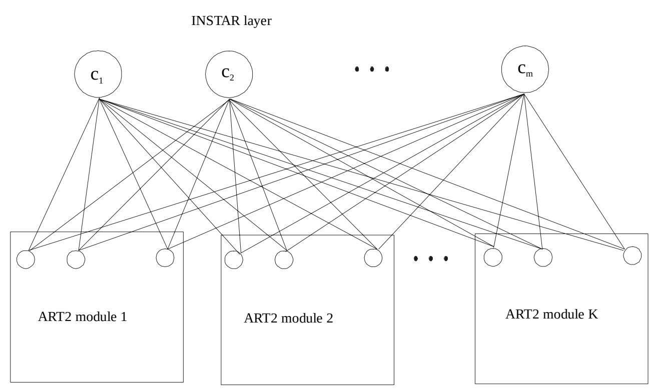

Figure 4: ARTSTAR architecture supporting multiple ART2 modules

2.4 ARTSTAR Processing

The most simple ARTSTAR network consists of a single ART2 module, slightly modified to incorporate feedback, a threshold field, the INSTAR layer, and a feedback field as shown in Figure 5.

The threshold field $\vec{t}=(t_1\ t_2\ \dots\ t_n)$ is a layer of n nodes, connected individually to the corresponding ART2 output nodes $\vec{y}=(y_1\ y_2\ \dots\ y_n)$. Each threshold field node accepts the corresponding ART2 output value and simply thresholds it to 0 or 1 as follows:

$$t_j=\begin{cases}1\ if\ g(y_j)>0\\0\ otherwise\end{cases}$$

This step is required because the ART2 module outputs either 0 or d, and ARTSTAR requires the output to be either 0 or 1.

The instar layer $\vec{s}=(s_1\ s_2\ \dots\ s_m)$ consists of m INSTAR nodes which are fully connected from the threshold field via the weights $\boldsymbol{W}=(\vec{w}_1\ \vec{w}_2\ \dots\ \vec{w}_n)^T$, $\vec{w}_j=(w_{j1}\ w_{j2}\ \dots\ w_{jm})$ and which can also accept input from a classification vector $\vec{c}=(c_1\ c_2\ \dots\ c_m)$. The classification vector is a binary vector with only one non-zero component which indicates the class, and therefore which INSTAR node should be active, when the class of the input is known. During training, each INSTAR node accepts weighted input from the threshold field and performs INSTAR learning upon the weights W as follows:

$$\vec{s}=\vec{t}\cdot\boldsymbol{W}$$

$$w_{ji}=w_{ji}+\lambda_2[t_j c_i-w_{ji}]U(t_j)$$

$$U(t_j)=\begin{cases}1\ if\ t_j>0\\0\ otherwise\end{cases}$$

where λ_2 is a small learning rate. The output of the INSTAR nodes s_i' varies. If the layer is being trained, each INSTAR node returns the desired output, and if not

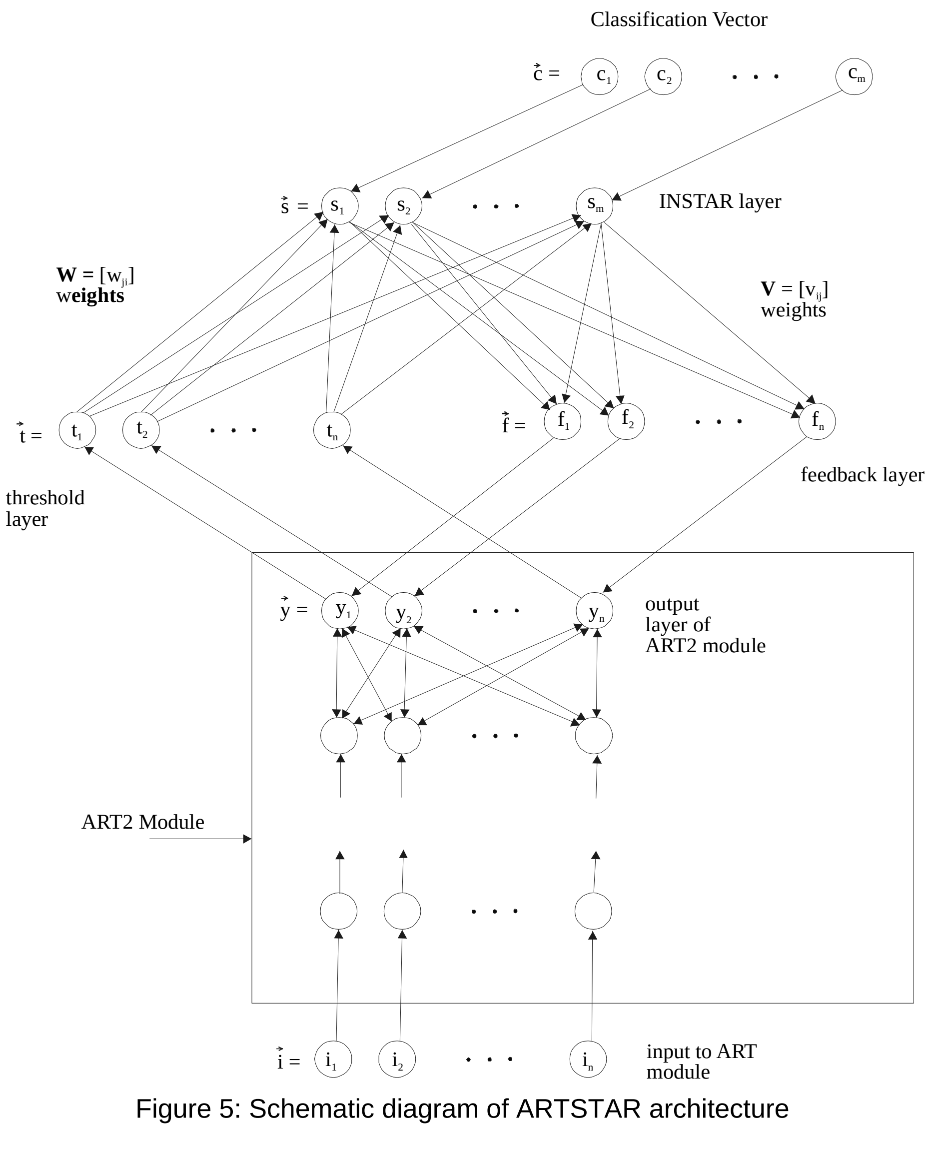

Figure 5: Schematic diagram of ARTSTAR architecture

being trained each node returns the weighted sum of its inputs.

The feedback field $\vec{f}$ consists of a layer of n nodes which are fully connected from the INSTAR layer by the weights $\boldsymbol{V} = \boldsymbol{W}^T$. This feedback layer directs the supervision of the ART2 module rather than accomplishing it with direct connections from the INSTAR layer as suggested in Section 2.2. This provides threshold control over the feedback that reaches the ART2 module permitting a variety of supervision strategies. The following two-threshold function is used:

$$\vec{f} = \begin{cases} \alpha & \text{if } \vec{s}' \cdot \boldsymbol{W}^T < \vartheta_2 \\ \beta & \text{if } \vec{s}' \cdot \boldsymbol{W}^T > \vartheta_1 \\ \vec{s}' \cdot \boldsymbol{W}^T & \text{otherwise} \end{cases}$$

where $\alpha \geq 0$, $\beta \leq \alpha$, $\vartheta_1 \leq \alpha$ and $\vartheta_2 \geq \beta$. Then the output layer of the ART2 module is influenced by computing its activation level with:

$$\vec{y} = \vec{p} \cdot \boldsymbol{Z}^T + \gamma \vec{f}$$

This process influences the results of the ART2 network by changing the order in which ART2 considers possible winners. The manner in which the feedback is used by the ART2 modules is justified based on the previous research on supervising ART2. The primary effect of match-tracking in ARTMAP (Carpenter, Grossberg & Reynolds, 1991), the dual-resonance in ARAM (Tan, 1992), and the first design principle of SMART2 (Merz et al., 1992) is to eliminate those nodes which have been associated with a previous class. Ideally, of course, such an inhibitory method is most efficient if the eligible nodes are considered before other nodes. This interpretation suggests that the main goal of feedback during training should not be to eliminate nodes as they are considered in turn, but to minimize the number of nodes that are eliminated in a trial by considering the most-eligible nodes first. The manner in which INSTAR feedback is incorporated into the ART2 equations for the F2 layer is based upon this interpretation.

There are three straightforward forms that the feedback function could take:

1. *constant*: all committed nodes receive equal feedback,
2. *direct*: nodes receive feedback in direct relation to the strength of their corresponding instar weight,
3. *inhibitory*: nodes receive non-zero feedback if and only if they represent the desired class (see Figure 6).

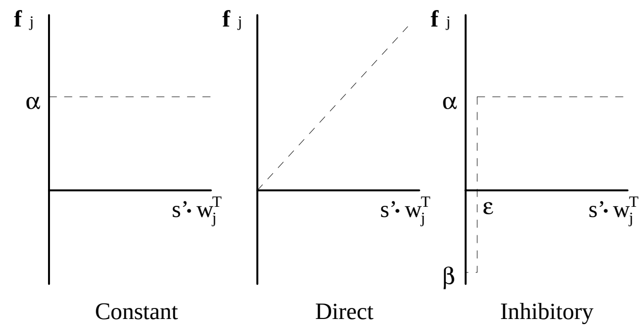

Figure 6: Main feedback functions of ARTSTAR

Each of these feedback functions represents a different level of supervision. The only effect of *constant* feedback is to give more preference to ART2 output nodes which have been committed than to those which haven't. If the feedback is zero, the ART2 network is completely unmodified, and the basic ARTSTAR can be viewed as a simple naming mechanism for output nodes of the ART2 network. *Direct* feedback influences output nodes in proportion to their likelihood of being correct. Thus, over-inclusive nodes are not prevented from occurring, but should be discouraged somewhat. The *inhibitory* feedback represents strong supervision. The ART2 output nodes are essentially prevented from representing more than one class apiece. This implements a supervision similar to those found in existing extensions to ART2 networks, though it is only a single instance of many supervision strategies enabled in ARTSTAR.

3 ARTSTAR Performance

ARTSTAR has been applied to several test domains (see Hussain, 1993). For the purposes of demonstrating its operation this section will describe its application to a data set of handdrawn digits is considered. Three key dimensions of ARTSTAR performance are addressed:

1. the general performance difference between ART2 and ARTSTAR.
2. the effect on performance of the use of the three types of feedback functions used by

ARTSTAR, *constant*, *direct*, and *inhibitory*.
3. the effect of integrating multiple, independently-trained modules in ARTSTAR.

Given these dimensions of analysis and the design principles of ARTSTAR, several performance hypotheses are proposed. Firstly, it is expected that the performance of the ARTSTAR network should be better with *inhibitory* feedback than with *direct* feedback, which in turn should be better than that with *constant* feedback. This is due to the degree of supervision incorporated into the network's training in each case. Secondly, ARTSTAR performance should increase with the number of ART2 modules included. This "redundancy" hypothesis is based on the ARTSTAR design assumption that differently-trained ART2 modules will contain complementary information. Thirdly, the additional redundancy effect that is achieved through the addition of a new module should eventually diminish as the number of modules increases past a certain point. There are two reasons for this expectation. On the one hand, the amount of new complementary information available to ARTSTAR should decrease as more modules are added, while on the other hand, the amount of conflicting information integrated by ARTSTAR should increase with the number of modules. Finally, based upon the results of Tan (1992) and Merz et al. (1992) and upon the design of ARTSTAR, an ARTSTAR network using *inhibitory* feedback should show performance comparable to that of a back-propagation network.

3.1 The Tests

The data used consists of three hundred 16x16 images of the digits 0 through 9 (see Figure 7). About two thirds of the samples were hand drawn by volunteers, and the other third were derived from computer font sets. Each digit is roughly the same size and roughly centred in the image.

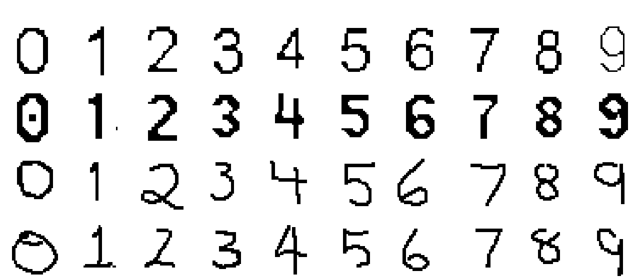

Figure 7: Sample images from digit data set

The complete details of the values of the network parameters used in the tests are provided by Hussain (1993), which also relates the exact algorithm for the version of ART2 used in the modules contained in the ARTSTAR tests. The threshold functions used are those shown in Figure 6, and in all ART2 modules the limiting vigilance value used for all test is 0.97, and the limiting number of ART2 output nodes is 30. In the back-propagation test, a standard back-propagation network is used with 40 hidden nodes and no bias units. The learning rate is set to 0.8 at the beginning of training and linearly decreased to 0.2 during 10,000 training epochs.

The performance of each ARTSTAR is tested in ten trials. During each trial, the complete data set is split randomly into two mutually exclusive halves. Each set contains an equal number of images from each possible class and in each trial, the random split is different. A trial consists of two phases - a training and a testing phase. During the training phase of a trial, each ART2 module of the ARTSTAR is trained separately on the training data, with each module receiving the training data in a different, random, order. A training phase consists of three epochs, and during each epoch, a given ART2 module receives data in a different order than in previous ones. Following the training phase, class names are assigned to each F2 node of each ART2 module of the ARTSTAR based on the class that contributed most to the training of a given node. During the testing phase, data from the testing set is presented once, simultaneously, to all the modules simultaneously. The performance of the ART2 modules is compared with that of the ARTSTAR itself.

3.2 Results

The results of all the tests performed are summarized in Figure 8. In the figure, the performance of ARTSTAR is compared graphically to that of its best ART2 module over the type of feedback function and the number of ART2 modules; the graphs show performance in terms of the percentage of correct classifications. The difference between the dotted lines, representing the performance of the best ART2 module of the ARTSTAR, indicates the effects of the feedback function type (i.e., the supervision effect), while the difference between the solid lines, representing ARTSTAR performance, and their corresponding dotted line indicates the redundancy effect. The graph also includes the average performance of a back-propagation network, over two trials, as well as the chance level of performance for the data set.

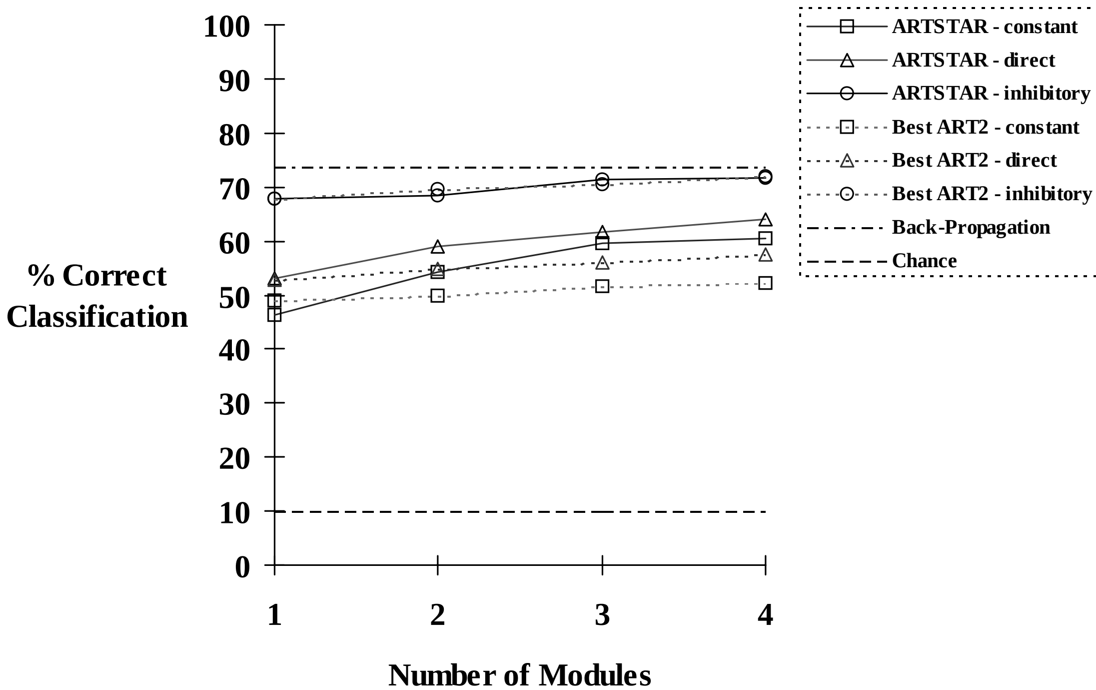

Figure 8: Classification performance of ARTSTAR on digits under several conditions

In interpreting the results in terms of the hypotheses, several observations can be made. Firstly, the supervision hypothesis can be seen to hold in general, as shown by the difference between the dotted lines in Figure 8. The *inhibitory* feedback function always results in an improved classification performance by ART2 as compared to the *direct* and *constant* feedback functions, and the *direct* function results in a modest performance improvement over the *constant* function. Further, the performance improvement of ARTSTAR over ART2 using the *inhibitory* function is far larger than any improvement using the *direct* function.

Secondly, the redundancy hypothesis can also be seen to hold in general. The difference between the solid and dotted lines in Figure 8 show that, for the *constant* and *direct* feedback functions, as the number of modules increases the ARTSTAR performs ever better over its best ART2 module; in the figure, the gap between the lines continuously increases. This performance difference is a direct result of the addition of more modules and thus reflects the redundancy effect. Conversely, the redundancy hypothesis does not hold for the *inhibitory* function. This suggests that the strong supervision might eliminate most, if not all, of the inconsistency in the network's categorizations; additional modules contribute no more complementary, just more conflicting information.

Thirdly, there is some support for the redundancy effect diminishment hypothesis. Figure 8 demonstrates nicely that the amount of improvement due to adding another module decreases as the number of modules increases; the solid lines flatten out.

Finally, both ART2 and ARTSTAR performed better than chance but worse than the back-propagation network, but the performance of ARTSTAR with inhibitory feedback closely approached that of back-propagation, as expected.

An additional observation, not directly related to the hypotheses, can be made concerning the results. As the number of modules increases, the performance of the best ART2 module increases as well; the dotted lines show positive slope. This is most probably due to the fact that as the number of modules increases, the chances that a very good ART2 module will be developed increases. Thus, in addition to the benefit due to redundancy, adding more modules, on average, results in a better "best" ART2 module and thus in better ARTSTAR performance.

Overall, the ARTSTAR network demonstrates the desired effects of supervision and redundancy. The best classification ability seems to be obtained from a four-module ARTSTAR using *inhibitory* feedback, and the performance of such an ARTSTAR can approach that of back-propagation. It should be noted, however that the high performance is due almost entirely to the supervision effect - there is no redundancy effect with the *inhibitory* function. ARTSTAR also shows improved performance over normal ART2 when using a *direct* feedback function. In this case, the improvement is due to a combination of the supervision and redundancy effect.

4 Conclusions

The ARTSTAR neural network extends and improves the ART2 network (Carpenter & Grossberg, 1987b) in an attempt to address some of the deficiencies exhibited by ART2 when applied to classification tasks. Specifically, ARTSTAR incorporates two fundamental design principles, the supervision of the ART2 learning process and the integration of multiple ART2 networks. A concise structural extension is proposed based on Grossberg's INSTAR node (Grossberg, 1982). ARTSTAR thereby improves the classification capability of ART2 while preserving the benefits of ART2's self-organization and on-line learning characteristics.

The supervision of ART2 has been examined by several researchers (Grossberg, Carpenter & Reynolds, 1991; Tan, 1992; Merz et al., 1992), all of which have used one common supervision technique. The ARTSTAR network can be made to implement the same technique, but its supervision process is more general than those previously proposed and can take several forms. The integration of multiple ART2 networks, each trained slightly differently on the same set of data, has been briefly considered by Carpenter et al. (1992) and Tan (1992), but only in the context of a post-hoc technique of improving performance. ARTSTAR actually incorporates such redundancy into its structure through its modularity, thereby inherently exhibiting the improved performance. Thus, ARTSTAR is a superset of not just the ART2 network, but also of existing extensions to ART which attempt to improve the performance of ART as a classifier.

The primary application of ARTSTAR considered is the straightforward classification task, on which it has been demonstrated to perform better than the normal ART2 network. However, ARTSTAR also exhibits a much more general functionality because of its modularity, and a variety of other tasks are potential applications of ARTSTAR (e.g., multi-resolution classification, hierarchical classification, data fusion, and invariant pattern recognition).

There are several aspects of ARTSTAR which can be examined in future work. Extensive tests of ARTSTAR properties are required, and the applicability of ARTSTAR to new tasks should be tested. Additional feedback functions should be analyzed to see if ARTSTAR exhibits any novel properties using them. Finally, further tests can be carried out comparing the learning times as well as the performance of ARTSTAR relative to back-propagation. ARTSTAR should have applications not suited for a back-propagation system, and these should be characterized. Finally, modifications and expansions to the ARTSTAR network should be examined.

One extension currently being researched is an extension of the INSTAR feedback to allow greater functionality. Currently, ARTSTAR provides feedback based solely upon the classification vector and does not exploit the differences between the output of the ARTSTAR network and that vector during training. Incorporating feedback based on errors in performance during training should result in an improved ARTSTAR classifier. One possible method of accomplishing such feedback is to assign a separate vigilance factor to each F2 output node of each ART2 module, and to use training performance to adjust the vigilance of the nodes. Thus, for example, a node which made many training errors could be given a high vigilance so that it would become more discriminating.

A second change is to the process of integration of ART2 modules. ARTSTAR currently integrates the outputs of all the modules with an equal emphasis on each module, and ARTSTAR can easily be revised so that different modules may have different levels of importance. For example, each module can be directly connected to the INSTAR layer via a bias node which modifies that module's contribution to the activation of each INSTAR node.

ARTSTAR is a useful, novel neural network architecture. It succeeds in improving the classification capability of ART2, yet is more flexible than existing techniques which also attempt this; it is a modular, supervised network which can be applied to a wide variety of problems, and it exhibits a number of useful properties, though the research presented merely touches upon a few of these. Several future directions for future work on ARTSTAR are possible, including not only further tests, but also several additional design modifications. ARTSTAR is thus an interesting network which should provide some useful contributions to neural network research.

References

Baruah, A.B. & Welti, R.C. (1991). "Adaptive resonance theory and the classical leader algorithm." Proceedings of the IEEE International Joint Conference on Neural Networks - Seattle (1991), **II**, p. A-913.

Burke, L.I. (1991). "Clustering characteristics of adaptive resonance." Neural Networks, 4, p. 485-491.

Carpenter, G.A. & Grossberg, S. (1987a). "A massively parallel architecture for a self-organizing neural pattern recognition machine." Computer Vision, Graphics and Image Processing, 37, p. 54-115.

Carpenter, G.A. & Grossberg, S. (1987b). "ART 2: Self-organisation of stable category recognition codes for analog input patterns." Applied Optics, 26, p. 4919-4930.

Carpenter, G.A. & Grossberg, S. (1990). "ART 3: Hierarchical search using chemical transmitters in self-organising pattern recognition architectures." Neural Networks, 3, p. 129-152.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H. & Rosen, D. (1992). "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps." IEEE Transactions on Neural Networks, 3, p. 698-713.

Carpenter, G.A., Grossberg, S. & Reynolds, J. (1991). "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organising neural network." Neural Networks, 4, p. 565-588.

Carpenter, G.A., Grossberg, S., & Rosen, D. (1991a). "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition." Neural Networks, 4, p. 493-504.

Carpenter, G.A., Grossberg, S., & Rosen, D. (1991b). "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system." Neural Networks, 4, p. 759-771.

Duda, R.O. & Hart, P.E. (1973). Pattern Classification and Scene Analysis. New York: Wiley.

Filho, E. & Bisset, D.L. (1990). "Applying the ART1 architecture to a pattern recognition task." Parallel Processing in Neural Systems and Computers, R. Eckmiller, G. Hartmann and G. Hauske (Eds.). Elsevier Science Publishers (North-Holland), p. 343-349.

Gargano, M.L. (1991). "Classifier voting in the neural networks." Proceedings of the IEEE International Joint Conference on Neural Networks - Seattle (1991), **I**, p. 388-391.

Grossberg, S. (1982). Studies of Mind and Brain. Boston: D. Reidel Publishing.

Hussain, T.S. (1993). "ARTSTAR: A Supervised Modular Adaptive Resonance Network Classifier". M.Sc. Thesis, Department of Computing and Information Science, Queen’s University, Sept. 1993.

Merz, C.J., St. Clair, D.C. & Bond, W.E. (1992). "SeMi-supervised Adaptive Resonance Theory (SMART2)." Proceedings of the IEEE International Joint Conference on Neural Networks (1992), **III**, p. 851-856.

Tan, A-H. (1992). "Adaptive resonance associative map: A hierarchical ART system for fast stable associative learning." Proceedings of the IEEE International Joint Conference on Neural Networks (1992), **I**, p. 860-865.