

LA-UR- 97-4302

Title:

CLASSIFICATION OF BEHAVIOR USING UNSUPERVISED  
TEMPORAL NEURAL NETWORKS

CONF-971068 --

RECEIVED

APR 06 1998

OSTI

Author(s):

Kristin L. Adair

Submitted to:

1997 IEEE International Conference on Systems, Man, and  
Cybernetics  
October 12-15, 1997  
Orlando, Florida

19980504  
030

MASTER

DTIC QUALITY INSPECTED



**Los Alamos**  
NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

# Classification of Behavior Using Unsupervised Temporal Neural Networks

Kristin L. Adair

Department of Computer Science  
Florida State University  
Tallahassee, Florida 32305  
adair@cs.fsu.edu

Paul Argo

Space and Atmospheric Sciences  
Los Alamos National Laboratory  
Los Alamos, New Mexico 87545  
pargo@lanl.gov

## Abstract

*Adding recurrent connections to unsupervised neural networks used for clustering creates a temporal neural network which clusters a sequence of inputs as they appear over time. The model presented combines the Jordan architecture [6] with the unsupervised learning technique Adaptive Resonance Theory, Fuzzy ART [4]. The combination yields a neural network capable of quickly clustering sequential pattern sequences as the sequences are generated. The applicability of the architecture is illustrated through a facility monitoring problem.*

## 1. Introduction

Traditional neural network clustering techniques, such as adaptive resonance theory, use similarity metrics to discover form in spatial patterns. These techniques discover important features in patterns which create the basis of the clusters formed. From the clusters one then can make generalizations or abstractions about the complete data set. An often neglected feature in a data set is the temporal nature of the data. Patterns sometimes occur in a sequence and if the sequence of the patterns is important, it should also be used in clustering.

Two common techniques exist for coding temporal information in feed-forward neural networks: a sliding window and recurrent connections. Sliding windows give more external input information to the network in the form of past patterns. For example, a window size of five would present the present pattern as well as the patterns of the past four time-steps to the network as input. This type of temporal information allows the network to find local dependencies in the pattern sequence. Recurrent connections in a neural network occur when a node receives connections from nodes in proceeding layers or from nodes in the same layer. The type of information encoded in these connections depends on the origin of the connection. Recur-

rent connections allow the network to discover long-range dependencies in the pattern sequence.

The encoding of temporal information requires many trade-offs including accuracy versus storage and processing speed. Increasing a sliding window's size allows more accuracy in its temporal information, providing an exact pattern. However, that accuracy requires additional storage and longer processing. In contrast, the processing and storage required for recurrent connections remains constant, independent of the amount of temporal information encoded. But with recurrent connections, the accuracy of the temporal information degrades as the amount of temporal information increases.

While both sliding windows and recurrent connections are common in supervised neural networks, only sliding windows have been widely used in unsupervised neural networks used in clustering. Instead, hierarchical methods have been developed which use two or more layers of existing techniques. One example is Time-Delay ART developed by Hagiwara [5]. In Time-Delay ART, the network has a three layer architecture in which the first two layers are a traditional ART2 [3] network. Nodes in the second layer are then fully-connected to nodes in the third layer. The connections from the second to third layer create the temporal nature of the network by using time-delays. The template selected for each individual pattern is held at the second layer until the entire sequence of patterns is presented by increasing the time-delay of each selected template when a new pattern is classified. Therefore, the template selected after the first pattern is presented will have the highest time-delay after the complete sequence of patterns is presented. After the entire sequence is presented the final output of the network is determined based on the templates selected and their final time-delays.

However, Time-Delay ART can not be used with all applications. One restriction of Time-Delay ART is that a pattern can not occur twice in the sequence without interfering with the similarity metric. Examples of when one would need to cluster a sequence of patterns with the same pattern occurring more than once include distinguishing

between words, such as read and reread, or learning people's sequence of movements around a room when the person can visit the same location more than once, such as the AMISS problem described below. In addition, the complete pattern must be presented before the sequence can be clustered. Applications involving actions which need to be initiated in real-time need to classify sequences as they are created. For example, security applications which learn to differentiate harmful activities from normal ones, need to begin security actions as soon as possible when a harmful activity is recognized. Otherwise, the damage to the system may already be detrimental.

Incorporating both types recurrent connections and a sliding window in clustering techniques allows for the clustering of sequential sequences of patterns with more flexibility. Using a small window size and then using the recurrent connections to further increase the amount of temporal information allows the user more flexibility with respect to the amount of temporal information, accuracy, processing speed, and storage needed. These additions also allow generalization for different pattern lengths and compositions as well as for the clustering of the pattern sequence at each time step as the length of the pattern increases over time.

## 2. AMISS

An example of the need for temporal clustering is the Adaptive Multi-sensor Integrated Security System (AMISS), being developed at Los Alamos National Laboratory. AMISS is a system designed to provide automated security assistance for nuclear facilities [1] [2]. The system may be used at nuclear facilities where the security measures are deficient. An example would be countries where, due to poor wages, both guards and scientists may be bribed or blackmailed into illegally removing nuclear material and giving it to unauthorized individuals. In addition, mistakes made by workers which lead to safety concerns or security breaches, such as leaving nuclear sources unattended, could be detected by the system. Discovering security breaches becomes increasingly difficult when the individuals removing nuclear material are authorized to handle the material, but should not be allowed to use the material for unauthorized uses on their own or give it to unauthorized individuals.

AMISS is different from the usual concept of a security system which is built from systems that only control access to secure areas and then activate intrusion alarms if violated. The AMISS system will operate in facilities that have significant ongoing activity, within which threatening activities must be identified and an alarm signaled. In many cases this threatening activity will not have been specified beforehand.

At the level of most interest to the project will be the

comparison of the actual pattern of activity in the room both to an authorization database and learned normal activities that the specific individual might perform in the room. In this sense the system is used as an anomaly detector, as it looks for sufficient deviation from the usual activity pattern. Anomalies found by the system can create alarms which may range from heightening the awareness of the guards to watch specific areas more closely or calling for immediate action.

### 2.1. Data

Multiple sensors and sensor types are used to localize motion in the room. Personnel tracking will be done by video camera, motion detection arrays, active radio-location systems, and active badge systems. Personnel recognition and verification will be done by a variety of biometric methods including face, fingerprint, gait, voice, and iris recognition systems. The information from the various sensor systems will be stored in a database and then combined to provide an enhanced estimate of personnel locations and activities as well as those of the radioactive sources.

Determining locations of authorized individuals and nuclear sources by sensor fusion and tracking is performed at several different levels. First, all sensors for one sensor type may be combined to give a single reading, a "smart" sensor, using triangulation, tracking, neural networks, and other low-level reasoning algorithms. The highest level reading for each sensor type, is then given to a higher level sensor fusion system. This system combines the data from the many sensor types to give the possible locations in rectangular co-ordinates and the certainty factor associated with each location for each individual. Certainty factors range from 0.0 to 1.0, with 0.0 indicating the lowest probability. An output from the sensor fusion system contains only locations with a non-zero certainty factor and may contain single or multiple such locations. Each individual in the room has an output file generated at fixed time intervals, such as a single file every two seconds.

### 2.2. Problem Restrictions

Several issues must be addressed in order to solve the problem of learning activities. The normal activities of a person in a room are most easily found by observation, as no rules governing activities in a secure area are available. This implies that the proposed system must learn normal activities only from existing data. In addition, for enhanced security, the system should not have a fixed set of abnormal activities to be detected. Instead the system should learn normal activities and recognize when ongoing activities deviate from normal.

In addition to the current position of the individual in the room, past positions, the order the past positions were visited, and the length of time spent at each position are also important. Also, because little is known about the number of previous positions needed to classify activities, both long and short-term dependencies should be used in classification.

The person's activity should be classified as normal or not at each time step, not after the entire activity has occurred. A system which classifies an activity as threatening after the individual has left the secure area is of limited use. Therefore, the proposed system should determine threatening activities as soon as they occur, and the system must thus allow the length of the path to continually grow.

Creating an unsupervised neural network with temporal information incorporates methods to meet all of the described restrictions. Unsupervised clustering techniques encode a data set by learning the sets important features without a priori information. Both learning and recall are quick allowing anomalies to be detected while dangerous activities are occurring.

### 3. Hybrid Architecture

To demonstrate the effects of adding recurrent connections to unsupervised neural networks, two specific architectures and learning rules will be combined: Fuzzy ART [4] and the Jordan architecture [6]. Fuzzy ART is a specific implementation of Adaptive Resonance Theory allowing the use of analogue numbers, while the Jordan architecture is a placement of recurrent connections.

#### 3.1. Adaptive Resonance Theory

Several unsupervised implementations of Adaptive Resonance Theory have been developed by Carpenter and Grossberg including Fuzzy ART [4]. Implementations of Fuzzy ART allow the encoding of spatial information in the form of analogue numbers.

Fuzzy ART networks consist of two fully connected layers, F1 and F2. The F1 layer is the input layer. At this layer, a single input pattern of analogue numbers is presented. Each value in the input pattern is tested against a threshold value; values above the threshold are sent to the F2 layer without change and those not above the threshold are sent to the F2 layer as the value zero. At the F2 layer, the input pattern is tested against each existing template using the equation

$$T_i = \frac{|w_i \cap x|}{\beta + |w_i|} \quad (1)$$

where  $w_i$  is the binary weight vector of template  $i$ ,  $x$  is the input pattern,  $\beta > 0$  is the choice parameter, and  $||$  is the

norm operator ( $|x| = \sum_{j=1}^n x_j$ ). The  $\cap$  function returns the minimum value at each position. Figure 1 shows result of using the  $\cap$  function with a template and a pattern.

0.43	0.25	0.0	0.0	0.0	0.95	Template
0.85	0.0	0.0	0.0	0.25	0.55	Pattern
0.43	0.0	0.0	0.0	0.0	0.55	Template $\cap$ Pattern

Figure 1.  $\cap$  function in Fuzzy ART

After  $T_i$  is calculated for each template, the template,  $i$ , which has the highest matching value is then used in the equation

$$\frac{|w_i \cap x|}{|x|} \geq \rho \quad (2)$$

where  $0 < \rho < 1$  is a parameter called *vigilance*. If equation 2 is satisfied, the pattern is incorporated into template  $i$  using the equation

$$w_i^{new} = \eta(w_i^{old} \cap x) + (1 - \eta)w_i^{old} \quad (3)$$

where  $\eta$  is the learning rate. If the template does not sufficiently match, the template is rejected for incorporating the pattern and the template remains unchanged. Then all other templates are again checked using equation 1. The process is repeated until a sufficient match is found or all existing templates are exhausted. If no match is found, a new template is formed comprised of the input pattern. After training is complete, if a pattern does match an existing template, the match is found quickly as few templates will need to be examined.

Fuzzy ART has many desirable features. First, the number of clusters used does not need to be specified in advance. Instead, a maximum number of clusters is specified and the algorithm may then use from one to the specified maximum. Having a larger maximum number of clusters specified than needed does not effect processing time, as clusters which have not encoded a template are not used in comparisons. In addition, when training is turned off, if a pattern does not adequately match any of the existing clusters, a warning will be given, instead of giving the closest matching template. Also, the within-cluster variance and between-cluster variance is controlled by a single parameter, vigilance. Finally, Carpenter and Grossberg prove that stable clusters are created by the algorithm [4].

One side effect in Fuzzy ART occurs if a pattern and template have non-zero values at the same position and the pattern contains all of the minimum values. In this

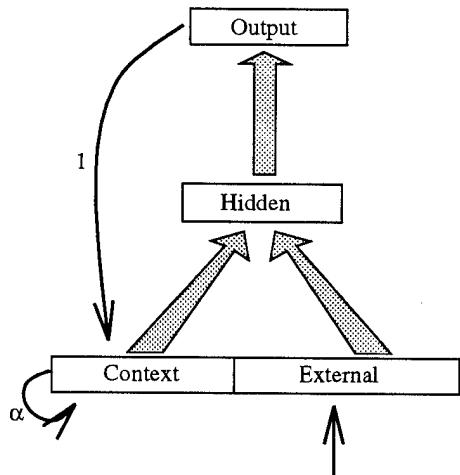
case, the pattern will always be accepted by the cluster regardless of the differences in the values. When training is on, processing the data set with more than one iteration will allow patterns to separate into stable clusters, but when training is turned off, the network will not give a warning saying the pattern was not accepted by the cluster. To alleviate this problem, the patterns can be normalized. One method of normalizing patterns is complement coding. In this technique, each input and its complement is presented to the F1 layer. When complement coding is used, stable clusters occur after only one presentation of the patterns and the above problems are alleviated [4].

### 3.2. Recurrent Jordan Architecture

The Jordan architecture [6], in Figure 2, is an architecture for encoding temporal information using recurrent connections. The architecture contains two sets of recurrent connections: from the output units to the context units and from the context units back to themselves. The weights on the recurrent connections from the output units to the context units have a fixed value of 1.0. The self-connections are trainable. The context units,  $C$ , also have some memory. Their updating rule is

$$C_i(t+1) = \alpha C_i(t) + O_i(t) \quad (4)$$

where  $O$  are the output units,  $\alpha$  is the strength of the self connections, and  $t$  is the time step. If  $O_i$  is fixed, then  $C_i$  decays, gradually forgetting previous values. Values of  $\alpha$  close to 1.0 allow a longer memory, but loss of detail in memory. In addition, as meaning is usually assigned to output units, the meaning of the recurrent connections can be extracted.



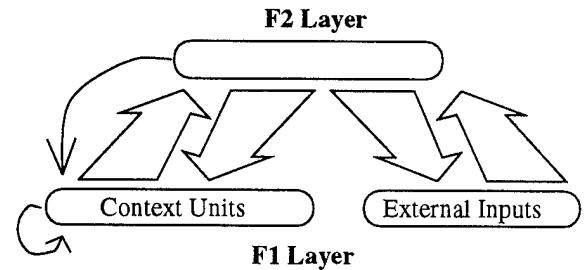
**Figure 2. Architecture for Jordan recurrent networks**

### 3.3 Temporal Fuzzy ART

The combination of Fuzzy ART with the recurrent connections of the Jordan architecture along with a sliding window creates a neural network capable of clustering a sequence of patterns based on spatial information as well as temporal information. Using the two forms of temporal information allows the network to discover important short-term and long-term dependencies.

The hybrid architecture, shown in Figure 3, still contains two layers, F1 and F2. But, the F1 layer can now be divided into two sections, external inputs and context units. The external inputs receive their information from an external source, while the context units derive their values from within the network, from the F2 layer and from themselves. Only the way the values are derived for the sections differs. Thresholding at the F1 layer, and comparison and creation of new templates at the F2 layer remain the same as Fuzzy ART.

The nodes in the F2 layer of Fuzzy ART do not give a single output, instead the template of a single node is modified and the activated node can be inferred. The user may then decide what information is useful according to the application. This may be the reporting of what F2 node was chosen for each pattern, the total number of clusters created, or the templates for the stable clusters which emerge after training. However, for the hybrid architecture created, each node in the F2 layer must report a single output value after each pattern is clustered. The values chosen were 1.0 if the pattern adequately matched the template and 0.0 otherwise. Therefore, a single F2 node outputs the value 1.0 when training is turned on and the remaining nodes output the value 0.0. When training is turned off, all F2 nodes may output the value 0.0 if no F2 node is found to adequately match the pattern.



**Figure 3. Architecture for Temporal Fuzzy ART**

Using the update rule of the Jordan Architecture, the memory of the context units,  $C_i$ , may grow quite large if the same output node is repeatedly chosen. To simplify processing, the outputs of the context units are normalized by having context node  $i$  output a value of 1.0 if  $C_i$  is

greater than the average value of all context nodes and a 0.0 otherwise.

In addition, in preliminary work, the weights of the self-connections of the context units are held at a constant value of 0.75. This allows for a modest decay of the memory of the context units.

## 4. Discussion

### 4.1. An Example

Tests of the network show that sequences of patterns can be successfully learned by Temporal Fuzzy ART. Templates for a pattern sequence encode detailed information for the most recent  $n$  patterns it represents, where  $n$  is the window size, and less detailed information about other patterns in the sequence.

Figure 4 shows the templates created by Temporal Fuzzy ART when the sequence of patterns in Figure 5 are presented. To create the templates, the vigilance of the network was set to 0.99, the weights of the self-connections were 0.75, the window size was two, and the maximum number of output nodes was four. Each template encodes values for two external inputs, due to the window size being two, and values for the context units. The number of context units is always equal to the maximum number of output nodes. For simplification, complement coding is not used in this example, however the same principles apply.

Template 1 was created after the first pattern was presented. External input 1 for template 1 has exactly one value greater than zero, which has encoded the position in pattern 1. External input 2 and the context units contain all zeros indicating that no previous patterns have been presented.

Template 2 encodes the sequence of the first two patterns. External input 1 again has a single value greater than zero encoding the position in pattern 2. In this template, external input 2 also has a single position greater than zero, which has encoded the position of pattern 1. The context nodes have a value of 1.0 in the first position and 0.0 in the remaining positions. This indicates that output node 1 (or template 1) has previously been used in the sequence. As previously mentioned, template 1 has encoded the position of pattern 1 and the knowledge that no previous patterns were presented. Therefore, the position of the first pattern is encoded in template 2 in two positions, external input 2 and the context units, but the context units provide additional information that no other patterns were previously encoded.

Finally, template 3 is the template after all three patterns are presented. Again the external inputs encode the present position, pattern 3, and the last known position, pattern 2. However, pattern 1 is no longer explicitly represented.

Instead, the context units, now containing the value 1.0 for the first two context units, are encoding the information, by indicating the first two templates have been used by the current sequence, which in turn represent pattern 1 and that pattern 1 was the first pattern in the sequence. While the same information might be encoded by increasing the window size to four, which would allow all three patterns to be explicitly represented as well as an open input indicating no previous pattern was presented, the processing time would be increased by the need to perform comparisons on an additional 18 positions (9 for each pattern). From this example, one can see how recent information is explicitly encoded in each template and past information is encoded with less detail in the context units.

### 4.2. Applications of Fuzzy Temporal ART

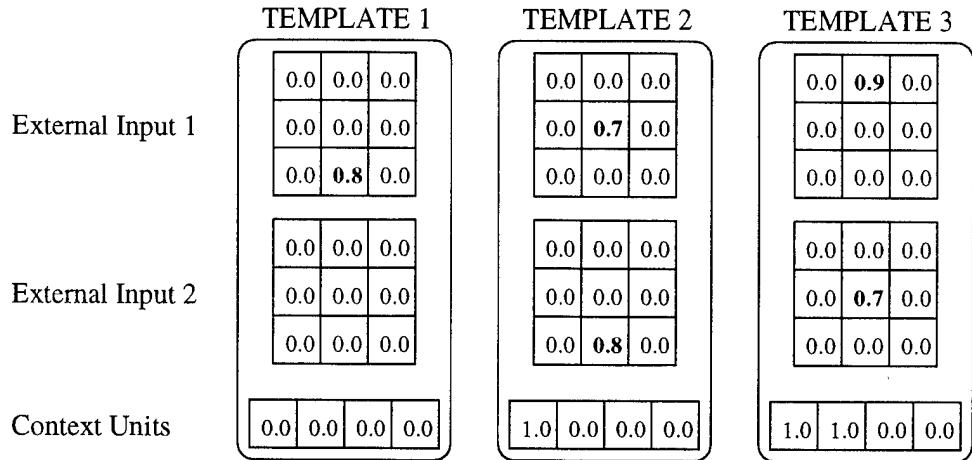
Using Temporal Fuzzy ART, a physical security system for monitoring individual's actions may be built, such as that for the AMISS project. Using a person's position at regular time intervals as input, the network can learn how a person normally moves about a room constituting the individual's normal behavior. The network can learn both the sequence of locations and the duration spent at locations.

Using the learned sequences as the individual's normal behavior, training can be turned off. Then subsequent behaviors can be identified as being similar to the learned behavior when the behavior clusters into an existing template or sufficiently deviating from all learned behavior when no template matches. By adjusting the vigilance parameter the room can be made as secure as deemed necessary.

Additional applications for Temporal Fuzzy ART are in the field of computer security. Instead of trying to determine all security holes on a computer, a system can be built which monitors what normal usage of the system looks like through different features such as system load over time or the order applications are used. Then when deviations, such as many consecutive uses of send-mail occur, a warning could be issued. Used as in the AMISS project, Temporal Fuzzy ART can learn normal computer system usage, then identify deviations.

### 4.3. Conclusions

While temporal information has been previously used in clustering algorithms, Temporal Fuzzy ART's recurrent connections and sliding windows allows use of both long and short-term interaction in clustering. Temporal Fuzzy ART also allows the user to optimize processing time, storage, and accuracy for their own application. In addition, the complete sequence does not need to be presented before the existing sequence is clustered. Therefore, applications



**Figure 4. Template created by Temporal Fuzzy ART**

Pattern 1	Pattern 2	Pattern 3
0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.9 0.0
0.0 0.0 0.0	0.0 0.7 0.0	0.0 0.0 0.0
0.0 0.8 0.0	0.0 0.0 0.0	0.0 0.0 0.0

**Figure 5. Sequence of patterns presented to Temporal Fuzzy ART**

which need to make decisions as soon as certain patterns occur, such as security systems, can use Temporal Fuzzy ART.

## References

- [1] P. Argo. Integrating materials protection, control and accounting and facility monitoring. *Arms Control and Nonproliferation Technologies*, pages 25–27, First/Second Quarter 1996.
- [2] P. Argo, R. Loveland, K. Anderson, B. Kelley, J. Elling, K. Linder, C. Buenafe, K. Adair, S. Hruska, and C. Johnson. Adaptive security systems - combining expert systems with adaptive technologies. In *Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, Florida, October 1997. to be published.
- [3] G. A. Carpenter and S. Grossberg. ART2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26(23):4919–4930, December 1987.
- [4] 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.
- [5] M. Hagiwara. Time-delay ART for spatio-temporal patterns. *Neurocomputing*, 6:513–521, 1994.
- [6] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Computation*, chapter 7: Recurrent Networks. Addison-Wesley Publishing Company, 1991.

M98004362



Report Number (14) LA-UR-97-4802  
CONF-971068 --

Publ. Date (11) 199803  
Sponsor Code (18) DOE/DP, XF  
UC Category (19) UC -705, DOE/ER

**DOE**