A Supervised Fuzzy Adaptive Resonance Theory with Distributed Weight Update

Aisha Yousuf and Yi Lu Murphey

University of Michigan – Dearborn
Department of Electrical and Computer Engineering,
4901 Evergreen Road,
Dearborn, MI 48128 USA
ayousuf@umich.edu

Abstract. The Fuzzy Adaptive Resonance Theory is an unsupervised clustering algorithm that solves stability plasticity dilemma. The existing winner-take-all approach to updating weights in Fuzzy ART has two flaws: (i) it only updates one cluster while an input might belong to more than one cluster and (ii) the winner-take-all approach is costly in training time since it compares one weight to the input at a time. We propose an algorithm that compares all weights to the input simultaneously and allows updating multiple matching clusters that pass the vigilance test. To mitigate the effects of possibly updating clusters belonging to the wrong class we introduced weight scaling depending on the "closeness" of the weight to the input. In addition, we introduced supervision to penalize the weight update for weights that have the wrong class. The results show that our algorithm outperformed original Fuzzy ART in both classification accuracy and time consumption.

Keywords: Fuzzy ART, supervised learning, clustering.

1 Introduction

Fuzzy Adaptive Resonant Theory (Fuzzy ART) is an unsupervised neural network that answers the stability-plasticity dilemma. The network is able to learn new patterns (plastic) while retaining the information from previously learned knowledge (stable) [1], [2], [3]. One of the major limitations of the Fuzzy ART is the winner take all approach, which allows only one prototype (weight) to be updated at a time [4]. However, a training point might have the same class as more than one prototype. In order to maximize the learning potential of the algorithm, all the prototypes that meet the threshold requirements should be updated.

In the conventional Fuzzy ART if the "closest match" prototype to the input that passes the vigilance test belongs to the wrong class, then by updating only that one weight minimizes the separation between the classes. The advantage of the proposed method is that it addresses this shortcoming by distributed weight updating. Out of multiple weights that get updated, some might belong to the correct class. The main disadvantage of having a distributed weight update is that the prototypes that are not

L. Zhang, J. Kwok, and B.-L. Lu (Eds.): ISNN 2010, Part I, LNCS 6063, pp. 430–435, 2010.

the closest match and pass the vigilance test are updated even though the input is from a different class. The effect of distributing to the wrong classes is mitigated in the proposed approach through weight scaling and supervision. Updating weight for wrong classes differently maximizes the separation between classes.

This paper is organized as follows. In section 2, we discuss the algorithms including the original Fuzzy ART and proposed Supervised Distributed Fuzzy ART (SDF ART) algorithm. The experiments and results are discussed in section 3. We conclude and describe our future plans in section 4.

2 Algorithms

2.1 Original Fuzzy ART

Fuzzy ART [1], [2], [3] is based on combination of the ART1 network and fuzzy logic and uses the fuzzy operators min($\wedge$), and max($\vee$). The input to the network, I, is normalized by appending the actual input, A, to its complement 1-A.

The normalized input pattern I is used for computing y as shown in Eq. 1. w matrix contains all the weight (prototype) vectors. Alpha, which is chosen by the user, is the conservation limit, which when small, minimizes recoding during learning.

$$y_j = \left|I \wedge w_j\right| / \left(\alpha + \left|w_j\right|\right) . \tag{1}$$

The node that yields the largest value of y is selected as the winner. The winner, which is denoted by y_J with J as the winning node index, has to pass the vigilance test shown in Eq. 2. ρ, a value between 0 and 1, is a vigilance parameter set by the user. k is the number of existing prototypes.

$$\rho \le \left|I \wedge w_J\right| / k \ . \tag{2}$$

If the test is passed, resonance occurs. The input I joins cluster J and the winning prototype vector, w_J, is updated using Eq. 3. β is the learning rate and ranges between 0 and 1.

$$w_J^{new} = \beta\left(I \wedge w_J^{old}\right) + \left(1 - \beta\right) w_J^{old} \ . \tag{3}$$

If the test is failed, the winning prototype is inhibited. The next highest value of y from Eq. 1 is selected as the winner and passed on to the test in Eq. 2. This process continues until the test is passed or if all the components of y are inhibited. In the latter case, a new cluster is formed and the input is the new cluster.

2.1 Kondadadi Distributed Weight Update

A recent paper [6] discusses the areas of improvement for Fuzzy ART. One of areas discussed is that distributed weight update is a much needed improvement. However, very little work has been done on this topic. Kondadadi et al [4] show the distributed weight update by updating every single prototype that matches the input. However, they fail to take into account that while these prototypes pass the vigilance test, they might not all belong to the correct class. We, therefore, impose a criterion that scales the weight update by the "closeness" of the input to the prototypes. Furthermore, we integrated supervised learning to penalize the weight update for the incorrect class.

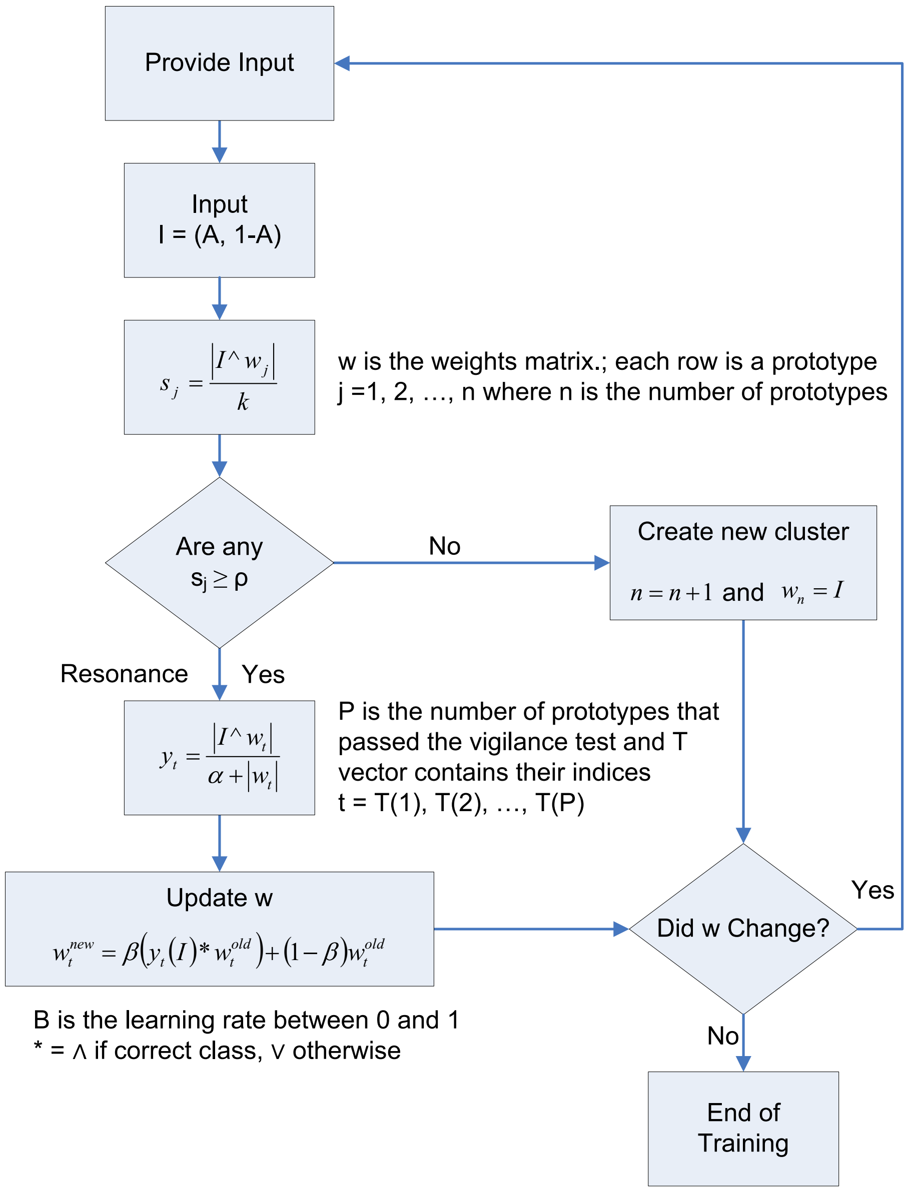

Fig. 1. SDF ART Structure

2.3 Supervised Distributed Fuzzy ART

The proposed SDF ART is shown in Fig. 1. Unlike original Fuzzy ART it does not select the "closest match" winner. Instead it goes directly to the vigilance test which selects all weights that meet the threshold requirements. If no prototypes satisfy the vigilance criteria, then a new cluster is formed with the input as the prototype. This makes the algorithm more time efficient as it does not have to go through the loop to find one winning cluster at a time.

While distributing input to multiple prototypes, since it is possible to update other weights that don't match the class, the weight updates are scaled. The weights are updated using the same "closeness" criteria used in Original Fuzzy ART to find the winner shown in Eq. 1. In addition, the supervised approach was added to penalize the weight update in case it was the wrong class. This was accomplished by performing the weight update with an operator, *, which is min($\wedge$) if it is the correct class, and max($\vee$) if it belongs to the wrong class. The formula for the updated weight is shown in Eq. 4, where * is the supervision operator and y_t is the weight scale factor.

$$w_t^{new} = \beta\left(y_t(I) * w_t^{old}\right) + (1-\beta)w_t^{old} \tag{4}$$

During the classification stage, there are some prototypes that were formed using inputs from multiple classes. These prototypes are assumed to belong to the class that had the majority number classified inputs to that prototype.

3 Experiments and Results

To evaluate the proposed algorithm we applied the SDF ART, and original Fuzzy ART for comparison, to Ionosphere dataset from the UCI Machine Learning Repository [5]. The data contains RADAR returns from the ionosphere for classification into good or bad returns. The entire dataset consists of 351 feature vectors, out of which we used 234 points for training and 117 for testing.

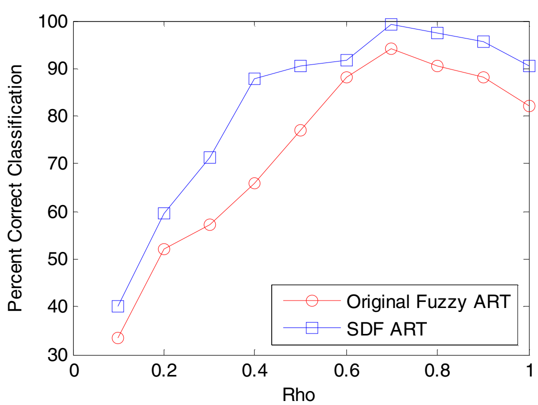

Fig. 2. Percent Correct Classifications

Experiments were conducted for studying the changes in the parameters. Since Fuzzy ART is affected by the order of inputs, the order of inputs was kept consistent during the parameter tests to get a fair comparison of effects with changes in control parameters. The same input order was kept for original Fuzzy ART and SDF ART. The percent correct classification rates for Fuzzy ART and SDF ART algorithms are shown in Fig 2. It can be observed from Fig. 2 that SDF ART always outperforms the original Fuzzy ART.

Table 1 shows the number of cycles it takes to converge with the change in beta (the learning rate). With larger beta both Fuzzy ART and SDF ART take fewer cycles, whereas they take longer with smaller beta. However, it can be observed that regardless of what beta is, SDF ART always takes less cycles to converge. In addition, the SDF ART converged in less than one third of the time that it took original Fuzzy ART to converge.

Table 1. Number of Cycles

Beta (β)		0.001	0.25	0.5	0.75	1
# of cycles	Fuzzy ART Original	879	767	542	414	134
	SDF ART	871	760	535	407	124

Table 2. Number of clusters with change in rho

Rho (ρ)		.001	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
# of clusters	Fuzzy ART	1	8	16	26	49	62	71	85	98	107	115
	SDF ART	1	6	14	21	46	57	64	82	95	102	106

Table 3. Percent Correct Classification with Change in Input Order

	Test 1	Test 2	Test 3	Test 4	Test 5
% Correct Classification	99.15%	97.44%	96.56%	99.15%	97.44%

The number of prototypes formed with changing values of rho is shown in Table 2 for both the original Fuzzy ART and SDF ART algorithms. On average, the SDF ART has approximately 4 prototypes less than original Fuzzy ART. Having fewer prototypes gives SDF ART an added advantage since it allows faster classification during testing.

Another test done on the modified Fuzzy ART was to study the variations in the accuracy with change in input order. Table 3 depicts the classification results with different input orders. It was observed that even with the change in input order the results were fairly consistent and within 3% of each other. This shows that the SDF ART algorithm is able to produce consistent results regardless of input orders.

Lastly, we compared our distributed weight update approach to the one proposed by Kondadadi et al [4]. Fig. 3 shows our percent correct classification results vs. theirs. It can be observed that our approach performs slightly better. However, their algorithm converges faster taking about 3/4 of the time it takes our algorithm to converge.

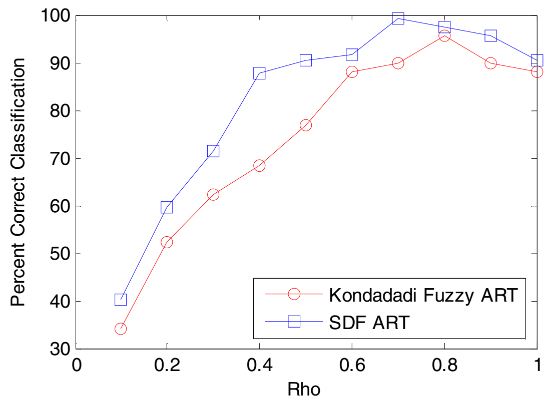

Fig. 3. Percent Correct Classification for SDF ART and Kondadadi approach

4 Conclusions and Future Work

It was observed that the SDF ART outperformed the Original Fuzzy ART in two aspects. It had better classification performance, and it takes less time during training and testing. It is expected that when the dataset is larger, the performance of SDF ART would be even better than original Fuzzy ART. In addition, it is expected that the time difference for training and testing between the SDF ART and fuzzy ART would be even greater. It was also observed by comparing results with Kondadadi approach that weight scaling performs better when it comes to classification.

For future work, we plan on using the INRIA pedestrian dataset with 9000+ points to classify pedestrians versus the background using the proposed approach. In addition, we will test this on datasets that have more than 2 classes. Lastly, we will explore additional methods of determining which class the cluster belongs to instead of selecting majority class since the weight updates are also scaled by the closeness.

References

1. Burwick, T., Joublin, F.: Optimal Algorithmic Complexity of Fuzzy ART. Neural Processing Letters 7, 37–41 (1998)
2. Carpenter, G., Grossberg, S., Rosen, D.: Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. Neural Networks 4, 759–772 (1991)
3. Huang, J., Georgiopoulos, M., Heileman, G.: Fuzzy ART properties. Neural Networks 8, 203–213 (1995)
4. Kondadadi, R., Kozma, R.: A Modified Fuzzy ART for Soft Document Clustering. In: Proceedings of the 2002 International Joint Conference on Neural Networks, pp. 2545–2549 (2002)
5. UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/`
6. Wunsch II, D.: ART Properties of Interest in Engineering Applications. In: Proceedings of the 2009 International Joint Conference on Neural Networks, pp. 3380–3383 (2009)