Reinforced ART (ReART): Adapting Fuzzy ART for Online Neural Control

Damjee D. Ediriweera, and Ian W. Marshall

Abstract— **Fuzzy ART has been proposed for learning stable recognition categories for an arbitrary sequence of analogue input patterns. It uses a match based learning mechanism to categorise inputs based on similarities in their features. However, this approach does not work well for neural control, where inputs require to be categorised based on the classes which they represent, rather than by the features of the input. To address this we propose and investigate ReART, a novel extension to Fuzzy ART. ReART uses a feedback based categorisation mechanism supporting class based input categorisation, online learning, and immunity from the plasticity stability dilemma. ReART is used for online control by integrating it with a separate external function which maps each ReART category to a desired output action. We test the proposal in the context of a simulated wireless data reader intended to be carried by an autonomous mobile vehicle, and show that training time and accuracy are significantly better than Fuzzy ART and Back Propagation.**

Index Terms— **Fuzzy ART, ReART, Back Propagation, Online Neural Control.**

I. Introduction

Fuzzy ART is an unsupervised Adaptive Resonance Theory (ART) network presented for classifying an arbitrary sequence of analogue input patters into stable recognition categories [1], [2]. In previous work the use of a standard Fuzzy ART network for online neural control was investigated [3]. Further testing has revealed several weaknesses which limit its potential for online control applications. Therefore, ReART, a modified Fuzzy ART network designed to address these limitations, is presented here.

Fuzzy ART is made up of two neuron layers. The first layer represents its input neurons whereas the second layer represents its output categories. Fuzzy ART performs match based learning, and therefore the configuration of the output layer is dynamically determined based on the diversity of the presented inputs. The decision of creating a new output category depends on whether existing categories fail to match an input within a defined threshold. Resonance or learning occurs only when an input is successfully matched to an existing category, or when a new category is created to handle a distinctly new input. This approach allows Fuzzy ART to overcome the plasticity-stability dilemma [2], meaning, it is able to remain stable for known inputs while being plastic (adaptable) towards new ones.

The most challenging issue when applying Fuzzy ART for online control is its unsupervised classification nature. Based on previous experiments, it is revealed that Fuzzy ART often classifies similar input patterns together, with no regard to the class of input which they actually represent. This behaviour of unsupervised ART networks is verified by the work of Christopher and Daniel [4]. In the context of neural control this poses an issue since it means that a single Fuzzy ART category can no longer be mapped to a single output action because a single category might actually represent many input classes.

Further, since the classification process in Fuzzy ART is match based, it can only be controlled using the vigilance threshold. The vigilance value defines the level of similarity required for an input to be classified under an existing category. However, the vigilance value in Fuzzy ART is global to the entire network, and therefore it defines a single category size for the network. It is often found that inputs in the real world regularly represent input classes with varying sizes. Some input classes can be quite general and large in size, whereas others are quite specific and small. The mismatch between the category size of an ART network and the class sizes of inputs will often result in inefficient, or inaccurate categorising. Work presented here demonstrates that issues outlined above can emerge in the form of longer convergence times, unstable performance, and large than optimal neural configurations, when Fuzzy ART is used for neural control.

This paper introduces ReART which uses a feedback based learning mechanism to overcome these limitations. The feedback mechanism drives the categorisation process by monitoring external feedback for each individual category and using this information to decide when new categories are required. Under this setup the vigilance parameter is typically set to a low value to encourage inputs to be classified into an

Manuscript received on July 10, 2007. This work was partly funded by the EPSRC grant EP/D053544/1, and was carried out at the Lancaster University, Lancaster, LA1 4YQ, U.K.

D. D. Ediriweera, PhD Student, Lancaster University, Lancaster, LA1 4YQ, U.K. (email:damjee@gmail.com)

I. W. Marshall, Director, Centre for Environmental Informatics, Lancaster Environment centre, Lancaster University, Lancaster, LA1 4YQ, U.K. (email: i.w.marshall@lancaster.ac.uk)

existing category. Although this sometimes results in the network classifying different input classes in to a single category, such misclassifications are quickly detected when a category generates negative feedback. This feedback acts as a trigger for a new category to be created. This approach allows the network to quickly diagnose and correct misclassifications. Furthermore, since the vigilance value does not play a prominent role in the classification process the network is able to efficiently support input classes of different sizes. At the same time the network is still able to effectively select the best category for each input based on the direct access theorem which guarantees that if a matching output category, U, exists for an input, I, then ART would directly activate U when I is presented [5]. This is part of the Winner Take All (WTA) activation mechanism built in to Fuzzy ART. Work presented here demonstrates that the said modifications allow ReART to learn faster relative to Fuzzy ART, with greater accuracy, and less number of internal neurons.

II. Related Work

A review of existing work reveals several ART versions which have emerged since the original concept. Some of these include Fuzzy ART, ART-2, ART-3, DART, ARTMAP [6], ECART, Semi-supervised ART (SMART2) [4], Snap Drift learning (P-ART) [7], Flexible Adaptable-Size Topology (FAST) [8], Grow and Represent (GAR), and SF-ART. Although a majority of ART networks remain unsupervised, several attempts have been made at designing supervised and reinforced ART networks to cater for different requirements.

ARTMAP presents a supervised ART network capable of incremental learning of labelled input patterns. ARTMAP comprises two individual ART networks, ART_1 and ART_2, linked by an associative learning network. Input patterns are presented to ART_1 and their labels are sent to ART_2. When an input is presented, ART_1 makes a prediction which is confirmed by associating it with the winning label of ART_2. If a wrong prediction is made the network increases the vigilance of the winning neuron in ART_1 which leads to a different candidate being chosen. The process occurs until the correct category is chosen. Resonance occurs only when the correct candidate is found.

Another approach to supervised ART is investigated in SMART2 [4]. SMART2 represents a modified ART2 network with a learning mechanism which allows learning only within the same class of inputs. This guarantees that similar input patterns from different input classes do not interfere in the learning processes of each other. To complement this, SMART2 also incorporates a mechanism of changing the learning rate depending on whether an input is classified correctly or not. The learning rate is high for inputs which are classified incorrectly. Based on numerical tests SMART2 is claimed to outperform ART2 for classification problems [4].

The Snap Drift algorithm presents a feedback based mechanism for improving the clustering process of ART [7]. This algorithm is designed for networks operating in non-stationary environments where new inputs are regularly received. Snap Drift works by altering the learning rate of individual ART categories depending on the feedback received by the system. This allows the system to snap away when performance is low, and drift when performance is high, hence the name Snap Drift. The literature indicates that the algorithm was successfully applied for generating automated service responses in a simulated active computer network [7].

A specific attempt to use an ART based neural network for neural control was made by Andres Perez [8]. This work investigates an approach of combining a ART based Flexible Adaptable Size Topology (FAST) network with a reinforcement based action selector. Even though FAST does not employ a supervision mechanism, this work is significant here since it demonstrates the possibility of using ART for neural control. The literature indicates that the ART based neural controller was used for navigation control on a robot [8].

III. Reinforced ART (ReART)

In contrast to previous work, ReART uses a new feedback based mechanism to drive the entire categorisation process. ReART architecture is similar to that of Fuzzy ART. The network consists of two neuron fields F1 and F2 (see Fig. 1). F1 consists of the input neurons whereas F2 represents the dynamic category field.

The ReART learning algorithm can be summarised as follows:

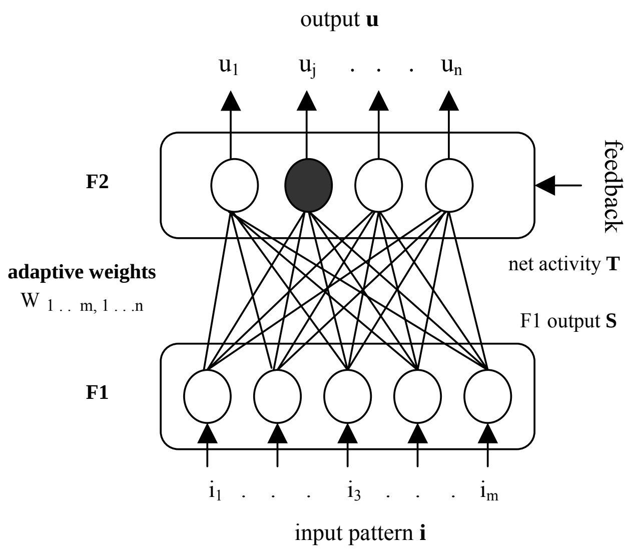

Fig. 1. Architecture of the ReART network

```
ρ ≈ 0 (set a low vigilance value)
while (i not empty)
{
   present input ic (ic = complement coded i)
   compute activation value Ti for all categories
   select category with max Ti as winner Uj
   set Uj output to 1
   receive feedback fi

   if (fi is high)
       adapt weights of Uj with new input
       (same as Fuzzy ART, wj = U(ic,wj))
       update performance of Uj, Ujp = fi
   else
        add 1 to Uj error count, Uje ++
        if (Ujp is high and Uje is low)
            (means poor current performance,
             good previous performance, and
             a low error count)
             then high probability of a new category
        endif
        if (Ujp is high and Uje is high)
            (means poor current performance,
             good previous performance, and
             a high error count)
             then high probability of removing category Uj
        endif
   endif
}
```

Inputs presented to the network are complement coded to avoid category proliferation [1]. A complement coded input pattern, i_c, is passed up through adaptive weights, $w_{1...n}$, which creates an activation value, T_i, at each output category. The category with the maximum T_i value is selected as the winner. Fuzzy ART rules are used for the calculation of T_i. However, unlike in Fuzzy ART, ReART encourages the first winner to be the final selection of the network. This is enforced by selecting a low vigilance value which discourages the network from initiating a re-search for a better match. This approach allows ReART to construct categories based on input classes rather than by the similarity of input features. This is the primary difference between Fuzzy ART and ReART. Furthermore, unlike in Fuzzy ART, ReART does not perform weight adaptation at this stage. Weight adaptations are only performed when external feedback is received.

ReART receives feedback based on the model outlined in Fig. 2. The figure illustrates how ReART is coupled with a separate map function to construct a functional neural controller, and how ReART receives feedback. The feedback received by ReART is short-term, meaning it is received on the basis of each action, and therefore it can be easily associated with each individual classification. This information is used for three main purposes: for weight adaptation, for deciding whether to create a new category, or to decide whether to remove an existing category.

Each individual class of input generally has its own desired output, and consequently a misclassified input rarely generates its desired output, and hence would not result in a positive feedback. Therefore, ReART learning is done only when an input classification receives positive feedback indicating that it was classified under the correct category. This speeds up the learning process by reducing the probability of inputs from different input classes interfering with each others learning.

The process of creating a new category is triggered when performance of an existing category is consistently low, or when a trend change is detected. Consistently low performance is a clear indicator of a single category attempting to classify inputs from two or more input classes. Similarly, a category with a history of positive feedback unexpectedly generating negative feedback normally indicates a new input class interfering with the learning of an existing category. ReART responds to misclassification by creating a new category tuned at classifying inputs which are causing the problem. This is the primary growth mechanism in ReART, and it is designed to create new output categories when new input classes are detected.

The decision of creating a new category is probabilistic. The probability of a new category being created is greatest as soon as a misclassification is detected, but decreases over time. Conversely the probability of removing a category is lowest when a misclassification is detected, but increases if performance does not improve over time. This approach offers a poorly performing category the chance to recover by separating its negative feedback generators into a separate category, but if performance does not improve the probability of it being removed approaches a maximum over time. This allows ReART to permanently remove categories which are struggling to improve. This is effective at removing poorly positioned categories which sometimes form between the boundaries of one or more input classes. The probabilistic approach for adding and removing categories is chosen to compensate for noise which might temporarily influence feedback.

IV. Numerical Evaluation of ReART

The numerical evaluation of ReART was performed using the control architecture illustrated in Fig. 2. The figure outlines how ReART integrates with an external map function to achieve a functional neural controller. At a higher level the system works by ReART creating categories and the map function associating them with appropriate output actions. Both networks are driven by external feedback, and operate independently of each other. All outputs of the map function are either 1 or 0. The map function discovers desired output

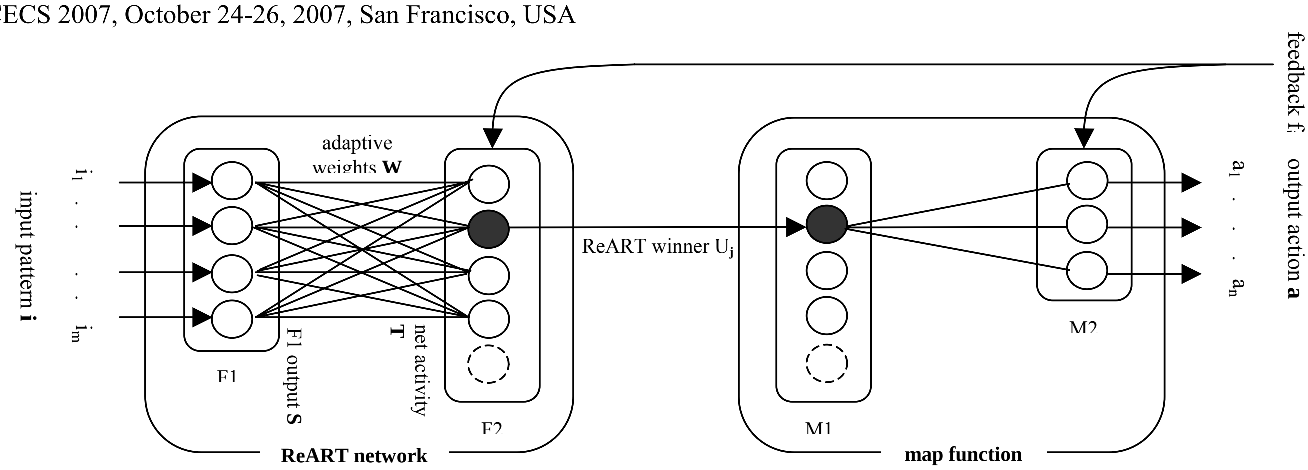

Fig. 2. Neural controller constructed using ReART

combinations for each ReART category based on a trial and error approach. Identical configurations of the map function were used for all experiments; hence its detailed workings are not discussed. Further information can be found here [3].

Several experiments were carried out to evaluate ReART. It was compared against two other neural network architectures: Fuzzy ART (using the same configuration as in Fig. 2); and Back Propagation (BP) [9]. The control problem simulated for the experiments is illustrated in Fig. 3. The selected control task is a neural based management of wireless communication on a mobile data reader. The objective of the controller is to optimize the power consumption of the wireless reader by managing communication distance, avoiding radio interference, adapting to host conditions, and by detecting

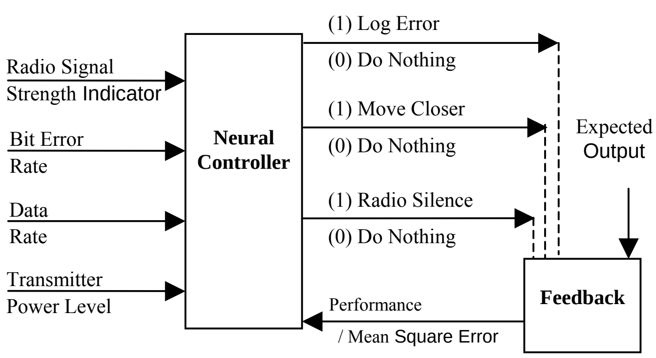

Fig. 3. Experiment setup

errors. To achieve this, the network uses four inputs, and up to eight output combinations. The input set includes the Radio Signal Strength Indicator (RSSI), Bit Error Rate (BER), Data Rate (DR), and the Transmitter Power Level (TPL); all values monitored from the wireless reader. Possible output behaviours include, moving the reader closer to the transmitter, enforcing temporary radio silence, staying neutral (meaning continue current transmission), logging errors, and other possible combinations of the above.

A dataset of 1200 input sets was recorded form a live environment. The dataset captured five distinct input classes. Each input class was assigned with a desired output action based on real world considerations. The desired output patterns were used to generate feedback for the ReART controller, and to calculate the Mean Square Error for BP. Feedback for ReART was binary, positive feedback was provided when an action was correct, and negative feedback otherwise. Experiments were run exhaustively, and all results were averaged over 500 independent runs. 75% of the dataset was used for training and 25% was reserved for testing. A 5% noise component was introduced to both datasets. To simulate realistic conditions inputs were presented to the network in their natural order, each run starting at a random point in the training dataset.

V. Results and Analysis

Fig. 4 compares the performance of ReART with BP and several Fuzzy ART configurations using different vigilance values. Results clearly illustrate ReART outperforming both BP and Fuzzy ART under tested conditions. ReART achieves an accuracy of over 90% within an average of 1 epoch, whereas to achieve the same BP requires an average 12.78 epochs, and the best Fuzzy ART configuration requires an average of 17.23 epochs. The BP configuration used for the experiment was selected after testing a range of configurations and therefore is believed to be optimal for the specified problem.

Table. I

Network Type	Max Training Accuracy Achieved	When Max Accuracy Occurs (Avg. epoch)	When >=90 Accuracy Occurs (Avg. epoch)	When >=95 Accuracy Occurs (Avg. epoch)	When ≈100 Accuracy Occurs (Avg. epoch)	Testing Accuracy	Avg. No of Categories	Avg. No of Categories with multiple input classes
ReART	100%	71.30	1.00	2.48	11.86	98.42%	6.57	0.0
BP (4-5-3)	100%	1580.75	12.78	21.32	81.25	99.84%	5.00	0.0
Funzzy ART (0.2)	76.01%	27.88	Never	Never	Never	73.24%	2.97	2.3
Funzzy ART (0.4)	77.22%	41.61	Never	Never	Never	74.92%	5.75	2.9
Funzzy ART (0.6)	85.25%	64.28	Never	Never	Never	82.54%	12.42	3.7
Funzzy ART (0.8)	94.67%	129.59	17.23	Never	Never	92.83%	31.92	4.6

The accuracy recorded in Fig. 4 indicates the number of correct actions observed for the most recent 100 inputs. An accuracy of 90 literally indicates 90 correct actions and 10 incorrect ones within the last 100 inputs. The axis indicating accuracy in Fig. 4 is scaled between 70 and 100 to improve clarity. The notation BP (I, H, O) is used to identify a BP network with, I, input neurons, H, hidden neurons and, O, output neurons, and the notation Fuzzy ART (ρ) is used to identify a Fuzzy ART network with a vigilance value of ρ.

ReART, compared with BP is able to learn faster. Table. I reveals that both ReART and BP are able to reach a training accuracy near 100%, but ReART achieves this several magnitudes faster than BP. The extra training allows BP to generalise better as indicated by the higher BP testing accuracy of 99.84%, compared to the 98.42% of ReART. BP also uses relatively fewer internal neurons to achieve a similar level of accuracy; however, it is common knowledge that identifying the correct BP configuration is not straight forward.

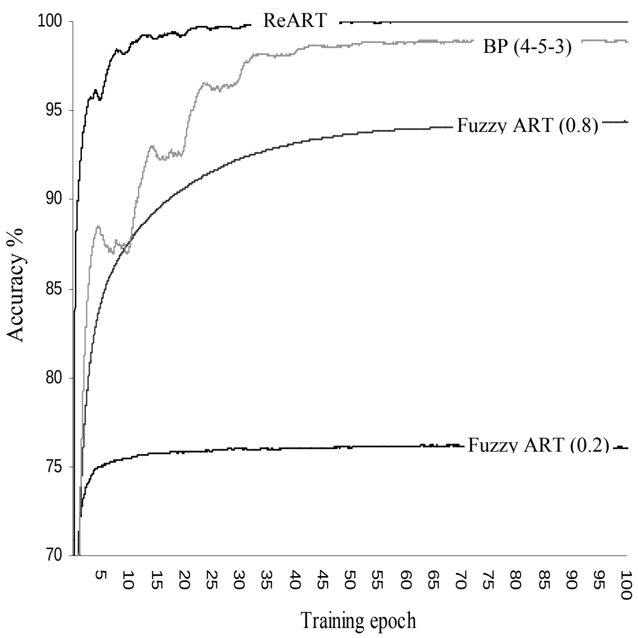

Fig. 4. Network accuracy vs. training epoch

A comprehensive comparison of ReART with Fuzzy ART is provided in Table. I. Figures here demonstrate the difficulties in selecting a global vigilance value to fit an entire dataset. Four separate Fuzzy ART configurations with different vigilance values were tested. No configurations were discovered which were able to efficiently classify the inputs correctly. Fuzzy ART (0.8) achieves the best accuracy of 92.83% but uses approximately 32 categories. In contrary Fuzzy ART (0.2) creates approximately 3 categories but fails to exceed an accuracy of 76.01%.

In Fuzzy ART the lower vigilance values struggle to precisely separate input patterns belonging to different input classes, whereas the higher vigilance values do a better job but results in multiple categories representing single input classes. This is clearly indicated by column nine of Table. I which identifies the number of categories which were classifying multiple input classes at the end of the training. The percentage of such categories in the network has a direct relationship with the network vigilance and its performance. Fuzzy ART (0.2) had almost 77% of its clusters classifying multiple input classes with a maximum accuracy of 73.24%, whereas Fuzzy ART (0.8) had 14% of its clusters classifying multiple input classes with a maximum performance of 92.83%. Based on these results it is clear that none of the tested Fuzzy ART configurations were able to match the performance of ReART which achieved a training accuracy of 100% with only 6 to 7 categories.

In addition to its ability to learn quickly with high accuracy, the ReART based controller is also able to address the plasticity stability dilemma. The plasticity stability dilemma outlines a

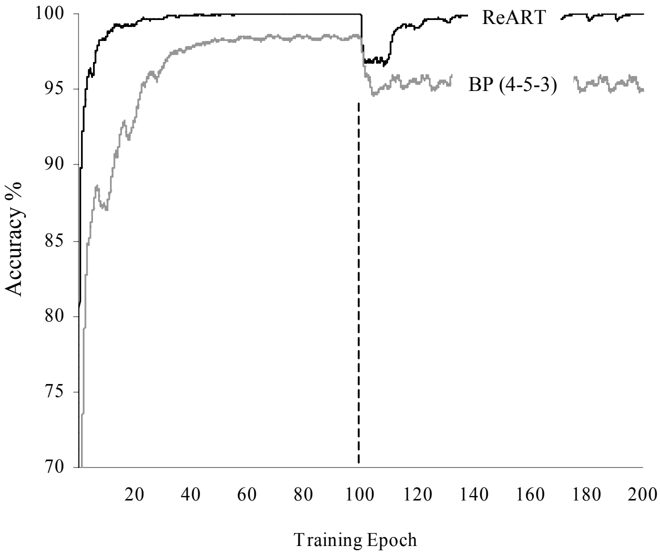

Fig. 5. Network accuracy vs. training epoch: introducing a new input class

problem which prevents most neural networks from learning new inputs while preserving previously gathered knowledge. Even a popular and robust control mechanism such as BP is inherently troubled by this problem. Fig. 5 illustrates a comparison of BP and ReART when presented with the plasticity stability dilemma. Here each network was initially allowed to train for 100 epochs. At the 100 epoch mark 50 new input patterns were introduced to the dataset. New patterns represented a new input class with its own desired output action. Both networks were then trained for an additional 100 epochs. The performance response of both networks is presented in Fig. 5. The axis indicating network accuracy is scaled between 70 and 100 to improve clarity.

Results indicate ReART to handle new inputs with greater effectiveness than BP. The introduction of new inputs causes ReART accuracy to temporarily drop to 97%, but recovers quickly within few epochs to an approximate 100%. The ReART network responded to the new input class by creating a new category to classify it. This allows it to learn a new input with minimum impact on its existing knowledge and accuracy. BP performance under identical conditions is less effective. BP is able to reach an initial accuracy of 96% to 97% with 100 epochs of training. The introduction of new inputs to BP causes its accuracy to drop to 94%, but unlike in ReART the accuracy fails to recover beyond this during the rest of the 100 epochs of training. The BP network recovers to its original accuracy approximately after 175 to 195 additional epochs of training. The result is as expected since BP learning does not necessarily cater for network adaptability.

VI. CONCLUSION

Several limitations restricting the use of Fuzzy ART and other unsupervised ART networks in neural control were demonstrated. Fuzzy ART was modified to develop ReART, a feedback based ART network capable of addressing these limitations. ReART was utilized to construct a neural controller capable of online learning. The ReART based controller was compared through numerical testing with BP, and an identical Fuzzy ART based controller. Results indicate ReART outperform both BP and Fuzzy ART for the presented control problem. ReART learns several magnitudes faster than BP, and provides a similar level of training accuracy. Further, ReART learns faster, with greater accuracy and less internal categories than Fuzzy ART. It also avoids the Fuzzy ART problem of classifying multiple input classes under a single category. The ReART based controller also overcomes the plasticity stability dilemma, and is able to learn new inputs with a minimum impact on existing knowledge and accuracy.

Further work is planned on testing ReART for classifying more complex datasets to confirm whether the fast learning mechanism is able to cope with more subtle categorisations. In addition, it is also to be tested using real feedback from a live environment with potentially greater noise and feedback errors.

VII. REFERENCES

[1] G. A. Carpenter. S. Grossberg. Fuzzy ART: Fast Stable learning and categorization of analogue patterns by an Adaptive Resonance System. Neural Networks, Vol. 4, 1991, pp. 759-771.

[2] G. A. Carpenter, S. Grossberg, Adaptive Resonance Theory, In M.A. Arbib (Ed.), The handbook of Brain Theory and Neural Networks, Second Edition. Cambridge, MA: MIT Press, 2003, pp 87-90

[3] D. D. Ediriweera, I. W. Marshall, Internally self organising Neural Network for online learning of perception to action mappings in Sensor Networks, Proceedings of London Communications Symposium, 2005, pp. 21-24.

[4] C. J. Merz, D. C. St. Clair, W.E. Bond, SeMi-supervised adaptive resonance theory (SMART2), Neural Networks, 1992. IJCNN., International Joint Conference on, Vol.3, Iss., 7-11 Jun 1992, pp. 851-856.

[5] A. Weitzenfeld, M. A. Arbib, A. Alexander, Chapter eight in The Neural Simulation Language, MIT Press, ISBN: 0-262-73149-5, July 2002, pp 157-169.

[6] G. A. Carpenter, S. Grossberg, J. H. Reynolds, ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network, Neural Networks (Publication), Vol 4, 1991, pp 565-588

[7] S. W. Lee. D. Palmer-Brown. C. M. Roadknight. Performance-guided Neural Network for rapidly self-organising Active Network Management, Neurocomputing, special issue on Hybrid Neurocomputing, Elsevier Science, Netherlands, 2004, 61C: 5 - 20.

[8] A. Pérez-Uribe. A non-computationally-intensive neurocontroller for autonomous mobile robot navigation. Chapter 8 in Biologically inspired robot behaviour engineering, Series: Studies in Fuzziness and Soft Computing. Vol. 109, R. J. Duro, J. Santos, M. Grana (Eds.), Springer-Verlag, 2002, pp. 215-238.

[9] R. Beale, T. Jackson, Neural Computing: An Introduction, ISBN: 0852742622, Institute of Physics Publishing, 1990, pp. 63-80.