COMPARISON OF FUZZY ARTMAP AND MLP NEURAL NETWORKS FOR HAND-WRITTEN CHARACTER RECOGNITION

*Yong Haur Tay and Marzuki Khalid**
Centre for Artificial Intelligence and Robotics (CAIRO)
Faculty of Electrical Engineering,
Universiti Teknologi Malaysia,
Jalan Semarak,
54100 Kuala Lumpur
e-mail: marzuki@klred.utm.my
(all correspondence should be sent to)*

Abstract --- *Fuzzy ARTMAP is one of the recently proposed neural network paradigm where the fuzzy logic is incorporated. In this paper, we compare the Fuzzy ARTMAP neural network and the well-known back-propagation based Multi-layer perceptron (MLP), in the context of hand-written character recognition problem. The results presented in this paper shows that the Fuzzy ARTMAP out-performs its counterpart, both in learning convergence and recognition accuracy.*

Keywords --- *Neural network, Pattern recognition, Character recognition, Adaptive resonance theory, Fuzzy ART, Fuzzy ARTMAP, Back-propagation, Multi-layer perceptron*

1. INTRODUCTION

Character recognition problem has long been identified as the problem that computer can hardly solved however human-beings take this for granted. There has already been substantial research undertaken to solve this problem. Neural network is one of the techniques that has been used for pattern recognition since 1950s as it has better performance than other statistical and artificial intelligence techniques[3].

Neural networks are preferred for pattern recognition problems because of their parallel processing capabilities as well as learning and decision-making abilities. Fukushima [4] has developed Neocognitron that could recognize shift, size, rotational patterns. Shustotovish and Thrasher [7] developed a system that works with fields of characters.

The range of applications for character recognition includes postal code reading, automatic data entry, recognition of print and script, automated cartography, bank, and reading services for the blind.

This paper has been organised as follows. In the next section, we develop the background information needed for the two architectures. Next, we discuss the back-propagation (BP) algorithm used in the MLP and the Fuzzy ARTMAP algorithm. As the Fuzzy ARTMAP has just been recently proposed[1], it would be interesting to measure its performance as compare the well-established BP algorithm in the context of character recognition problem. We also discuss the hand-written character feature sets which we used in the experiments. The next section describes the experiments that were performed and the results. The final section summarises our results and conclusions.

2. MULTI-LAYER PERCEPTRON (MLP)

The MLP is a supervised neural network. It can have multiple inputs and multiple outputs and multiple layer of nodes/neurons. Figure 1 shows an architecture of a three-layered perceptron. The leftmost layer is the layer to which the input data is supplied; the rightmost layer is the output layer and the middle layer is the layer to interconnect input and output layers. Each layer of the network is fully interconnected to its subsequent higher layer. The links between each neuron are called weights, where the knowledge is being stored.

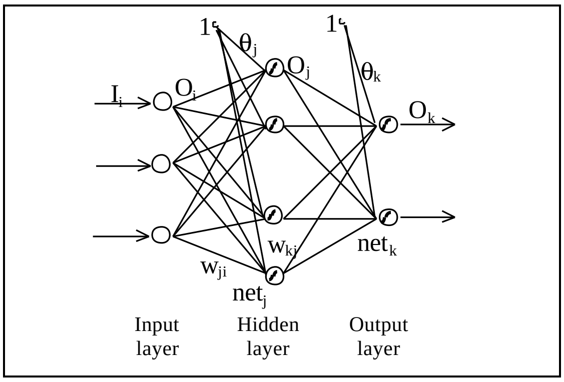

Figure 1. Example of a 3-4-2 MLP.

From Figure 1, O_k, O_j and O_i are output values of output, hidden and input layers, respectively. w_{kj} denotes the weights between output and hidden layers, w_{ji} denotes the weights between hidden and input layers.

The MLP utilises the back-propagation (BP) algorithm for training[8]. The algorithm consists of 2 phases: a feed-forward process and a back-propagation process.

For the initial stage, the weights of the network are randomly selected. The *learning rate h* and *momentum b* is pre-set before the learning phase. During the learning phase, an input vector is presented to the network and the vector propagates from the input layer to the output layer. Thus, the output of the hidden layer has the following notation,

$$O_j = f(\, net_j \,) \qquad (1)$$

$$net_j = \sum_i w_{ji} O_i + qj \quad \forall O_i = I_i \qquad (2)$$

similarly, the output layer becomes

$$O_k = f(\, net_k \,) \qquad (3)$$

$$net_k = \sum_j w_{kj} O_j + q_k \qquad (4)$$

where normally *f(x)* is a sigmoidal function as follows:

$$f(x) = \frac{1}{1 + \exp(-x)} \qquad (5)$$

The output vector generated from the feed-forward process is then compared with the desired output vector. The cost function used here is a squared error function ξ, which is given by

$$\xi = \sum_p x_p \qquad (6)$$

$$\xi_p = \frac{1}{2} \sum_k (t_{pk} - O_{pk})^2 \qquad (7)$$

where t_{pk} is the desired output for the kth component of the output pattern for pattern p and O_{pk} is the corresponding actual output.

Back-propagation process is then using the error to adjust weights accordingly, based on the steepest descent method, as follows:

$$\Delta w_{kj}(t+1) = \eta \delta_k O_j + \alpha \Delta w_{kj}(t) \qquad (8)$$

where

$$\delta_k = O_k (1 - O_k) (\tau_k - O_k) \qquad (9)$$

and

$$\Delta w_{ji}(t+1) = \eta \delta_j O_i + \alpha \Delta w_{ji}(t) \qquad (10)$$

where

$$\delta_j = O_j (1 - O_j) (\tau_j - O_j) \sum_k d_k w_{kj} \qquad (11)$$

Configuration for the MLP neural network used in the experiment is 64-15-15-8 (4-layer neural network with 64 inputs and 8 outputs, 2 intermediate layers each with 15 neurons). The *learning rate*, h, is 0.7 whereas *momentum*, b, is 0.4.

3. ADAPTIVE RESONANCE THEORY (ART)

Adaptive Resonance Theory (ART) was developed by Carpenter and Grossberg for pattern recognition purpose[5]. There are a lot of variation of ART, ART1, ART2, ART3, Fuzzy ART, Fusion ART, Fuzzy ARTMAP, ART-EMAP, just to name a few. The major distinct advantages of ART neural networks are :-

- It retains knowledge of previously learned patterns or patterns categories (STABILITY).
- It can also learn new patterns (PLASTICITY).

ART has shown that it can solve the stability-plasticity dilemma which other feed-forward neural networks cannot solve. In other words, ART is able to learn many new things without forgetting things learned in the past. Whereas MLP cannot learn new information incrementally without forgetting old information unless it is retrained with the old information along with the new.

ART is a type of competitive learning network, and suitable for both categories (pattern) formation and recall (recognition). When an input pattern is adequately similar to one stored in the ART's long term memory (LTM), ART recognises the patterns as belonging to the category, and modifies (generalise) the stored category to accommodate new features of the current input pattern. When an input pattern is not adequately similar to any stored category, pattern formation occurs. ART selects an uncommitted (new) category to store the current input. If no more uncommitted nodes are left, the current input has no response. Hence, stored categories remain stable to irrelevant inputs and yet are sensitive to novel features of inputs. In such case, ART makes a satisfactory trade-off between stability and plasticity.

3.1. Fuzzy ART Architecture

Fuzzy ART is a variation of ART system derived from the first generation of ART, namely ART1. It is a synthesis of ART1 and *fuzzy logic.* Unlike ART1 which can only accept binary input patterns, Fuzzy ART allows both binary and continuous input patterns. Figure 2 shows the basic features of the Fuzzy ART architecture. The two major subsystems are the attentional subsystem and the orienting subsystem.

The attentional subsystem is used to recognise and classify previously learned patterns. The orienting subsystem is invoked when ART encounters a new pattern. Under these circumstances, the orienting subsystem shuts down any attempt to continue matching with stored patterns, and sets up a new category to respond to the new type of pattern. Thus, the attentional subsystem performs the actual pattern recognition, and the orienting subsystem acts on the attentional subsystem to enable it to respond to novel patterns.

Patterns of activity develop over the neurons in the two layers of the attentional subsystem which are called short term memory (STM) because they exist only in association with a single application of an input pattern. The weights that interconnect between F_1 and F_2 layers are called long term memory (LTM) traces because they encode information that remains a part of the network for an extended period.

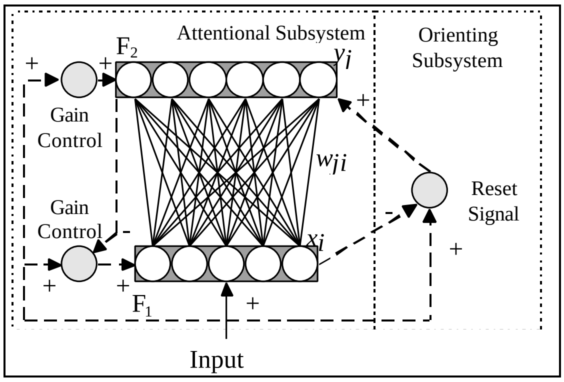

Figure 2. Fuzzy ART Architecture.

3.1.1. Fuzzy ART Dynamics

The input pattern with the length of M is denoted as $\boldsymbol{I} = (\boldsymbol{I_1}, ..., \boldsymbol{I_M})$ where $\boldsymbol{I_i} \in [0, 1]$. The F_1 activity vector is denoted as $\boldsymbol{x} = (\boldsymbol{x_1}, ..., \boldsymbol{x_M})$ and the F_2 activity vector with N number of category is denoted as $\boldsymbol{y} = (\boldsymbol{y_1}, ..., \boldsymbol{y_N})$. M and N is arbitrary.

Associated with each F_2 category node j ($j = 1, ...,N$) is a vector $\boldsymbol{w_j} \equiv (x_{j1}, ..., x_{jM})$ of adaptive weights, or LTM traces. Initially, each category is uncommitted,

$$\boldsymbol{w_{j1}}(0) = ... = \boldsymbol{w_{jM}}(0) = 1; \tag{12}$$

After a category is chosen for coding it becomes committed.

Fuzzy ART dynamics are determined by a *choice parameter* $a > 0$; a *learning parameter* $b \in [0, 1]$; and a *vigilance parameter* $r \in [0, 1]$.

When Fuzzy ART receives an input pattern $\boldsymbol{I}$, $\boldsymbol{I}$ is copied immediately over into $\boldsymbol{x}$. The next step is to pass the information in layer F_1 up to F_2, with the *choice function*, $\boldsymbol{T_j}$, is defined as

$$\boldsymbol{T_j}(\boldsymbol{I}) = \frac{|I \wedge w_j|}{a + |w_j|}, \qquad (13)$$

where the fuzzy AND operator $\wedge$ is defined by

$$(\mathbf{p} \wedge \mathbf{q})_I \equiv \min(p_i, q_i) \qquad (14)$$

and the norm $|\cdot|$ is defined by

$$|\mathbf{p}| \equiv \sum_{i=1}^{M} |p_i| \qquad (15)$$

for any M-dimensional vectors $\mathbf{p}$ and $\mathbf{q}$. For notational simplicity, $\boldsymbol{T_j}(\boldsymbol{I})$ is written as $\boldsymbol{T_j}$.

These values of $\boldsymbol{T_j}$ in F_2 undergo a competitive process. Only one value of $\boldsymbol{T_j}$ will win. The category choice is indexed by J, where

$$\mathbf{T_J} = \max\{\mathbf{T_j}: j = 1 \ldots N\}. \qquad (16)$$

If more than one $\boldsymbol{T_j}$ is maximal, the smallest index is chosen. The activity vector $\boldsymbol{y}$ in F_2 is

$$\boldsymbol{y_i} = \begin{cases} 1 & \text{for } i = J \\ 0 & \text{for } i \neq J \end{cases} \quad \forall\, i = 1 \ldots N \qquad (17)$$

The winning node $\boldsymbol{y_J}$ is what the Fuzzy ART thinks is its best match for the input pattern.

Next, the Fuzzy ART does a *top-down template matching*, when the F_1 activity vector $\boldsymbol{x}$ obeys the following equation

$$\mathbf{x} = \begin{cases} I & \text{if } F_2 \text{ is inactive} \\ I \wedge w_J & \text{if the } J\text{th } F_2 \text{ node is chosen} \end{cases} \qquad (18)$$

The *top-down template* is compared with the initial input pattern. If the pattern is matched to within a specified vigilance criterion r, then resonance occurs.

$$\frac{|I \wedge w_J|}{|I|} \geq r\,; \qquad (19)$$

Mismatch reset occurs when

$$\frac{|I \wedge w_J|}{|I|} < r \qquad (20)$$

Next the F_2 winning nodes, $\boldsymbol{T_J}$ is inhibited for the duration of the input representation to prevent it from competing further. A new index J is then chosen, by (16). The search process continues until the chosen J satisfies (19). Once equation (19) is fulfilled, the weight vector $\boldsymbol{w_J}$ is modified to encode the pattern

$$w_J^{(new)} = \beta\,(I \wedge w_J^{(old)}) + (1 - \beta)\, w_J^{(old)} \qquad (21)$$

3.2. Fuzzy ARTMAP Architecture

ARTMAP (**Figure 3**) is a class of neural network architectures that perform incremental supervised learning of recognition and multidimensional maps in response to binary input vectors presented arbitrarily[1]. In turn, the Fuzzy ARTMAP extends the ARTMAP by integrating *fuzzy logic* into the system, which allows it to accept input vector values between 0 and 1. The generalisation is done by replacing both ART1 modules (ART_a and ART_b) of the binary ARTMAP system with fuzzy ART modules.

Fuzzy ARTMAP consists of two fuzzy ART modules (ART_a and ART_b) which are linked together via an inter-ART module, $\boldsymbol{F^{ab}}$. During the learning phase, the input vector $\boldsymbol{I_0}$, is presented to the ART_a and the desired output vector $\boldsymbol{O_0}$, is presented to the ART_b. The ART_a and ART_b modules classify the input and desired output vector into categories, then the map field (inter-ART module) makes associations from ART_a category to ART_b category.

If $\boldsymbol{I_0}$ predicts an incorrect $\boldsymbol{O_0}$, then a mechanism called *match tracking* is triggered. This mechanism increases the vigilance parameter of ART_a, r_a, by a minimum value and, hence, force the ART_a module to search for another category suitable to be associated with the desired output vector. r_a is then set back to the baseline vigilance parameter, $\bar{r}$, for every step of learning trial.

3.2.1. Fuzzy ARTMAP Dynamics

The inputs to ART_a and ART_b are in *complement code* form, it is necessary for the successful operation of Fuzzy ARTMAP[1]. For ART_a, if the input is $\boldsymbol{I_0} = (\boldsymbol{a_1}, ..., \boldsymbol{a_{Ma}})$, then

$$\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) = (\mathbf{a_1}, ..., \mathbf{a_M}, \mathbf{a_1}^c, ..., \mathbf{a_M}^c) \qquad (22)$$

where

$$\mathbf{a_i}^c = 1 - \mathbf{a_i} \qquad 1 \leq i \geq M_a \qquad (23)$$

The $\boldsymbol{F_1^a}$ activity vector is denoted by $\boldsymbol{x^a} = (\boldsymbol{x_1^a}, ..., \boldsymbol{x_{2Ma}^a})$ and the $\boldsymbol{F_2^a}$ activity vector with N_a number of category is denoted by $\boldsymbol{y^a} = (\boldsymbol{y_1^a}, ..., \boldsymbol{y_N^a})$. M_a and N_a are arbitrary. For ART_b, $\boldsymbol{O_0} = (\boldsymbol{b_1}, ..., \boldsymbol{b_{Mb}})$, $\boldsymbol{O} = (\boldsymbol{b}, \boldsymbol{b^c})$. The $\boldsymbol{F_1^b}$ activity vector is denoted by $\boldsymbol{x^b} = (\boldsymbol{x_1^b}, ..., \boldsymbol{x_{2Mb}^b})$ and the $\boldsymbol{F_2^b}$ activity vector with N_b number of category is denoted by $\boldsymbol{y^b} = (\boldsymbol{y_1^b}, ..., \boldsymbol{y_{Nb}^b})$. M_b and N_b are also arbitrary. For the inter-ART module, $\boldsymbol{F_{ab}}$ output vector is denoted by $\boldsymbol{x^{ab}} = (\boldsymbol{x_1^{ab}}, ..., \boldsymbol{x_{Nb}^{ab}})$, and the weight vector from the jth $\boldsymbol{F_2^a}$ node to $\boldsymbol{F^{ab}}$ is denoted by $\boldsymbol{w_j^{ab}} = (\boldsymbol{x_{j1}^{ab}}, ..., \boldsymbol{x_{jNb}^{ab}})$. Vectors $\boldsymbol{x^a}$, $\boldsymbol{y^a}$, $\boldsymbol{x^b}$, $\boldsymbol{y^b}$, $\boldsymbol{x^{ab}}$ are initialised to 0 between input presentations. Initially, the weight vectors $\boldsymbol{w^{ab}}$ are set to 1 (*uncommitted*).

The inter-ART module $\boldsymbol{F^{ab}}$ is activated whenever any of the ART_a or ART_b categories is active.

$$\mathbf{x^{ab}} = \begin{cases} y^b \wedge w_J^{ab} & \text{both the } ART_a \text{ and } ART_b \text{ is active} \\ w_J^{ab} & ART_a \text{ is active, } ART_b \text{ is inactive} \\ y^b & ART_a \text{ is inactive, and } ART_b \text{ is active} \\ 0 & \text{both the } ART_a \text{ and } ART_b \text{ is inactive} \end{cases} \qquad (24)$$

During learning phase, ART_a vigilance parameter r_a, equals the baseline vigilance, $\bar{r}_a$ at the beginning of every input presentation. When Fuzzy ARTMAP receives an input/output pair ($\boldsymbol{I_0}/\boldsymbol{O_0}$), ART_a chooses the $\boldsymbol{J}$th node of $\boldsymbol{F_2^z}$ and ART_b chooses the $\boldsymbol{K}$th node of $\boldsymbol{F_2^b}$. When both ART_b and ART_b are active and $\boldsymbol{x^{ab}} \neq 0$, then the input/output pairs are associated with the equation

$$w_{jk}^{ab} = \begin{cases} 1 & j = J \text{ and } k = K \\ 0 & \text{otherwise} \end{cases} \qquad (25)$$

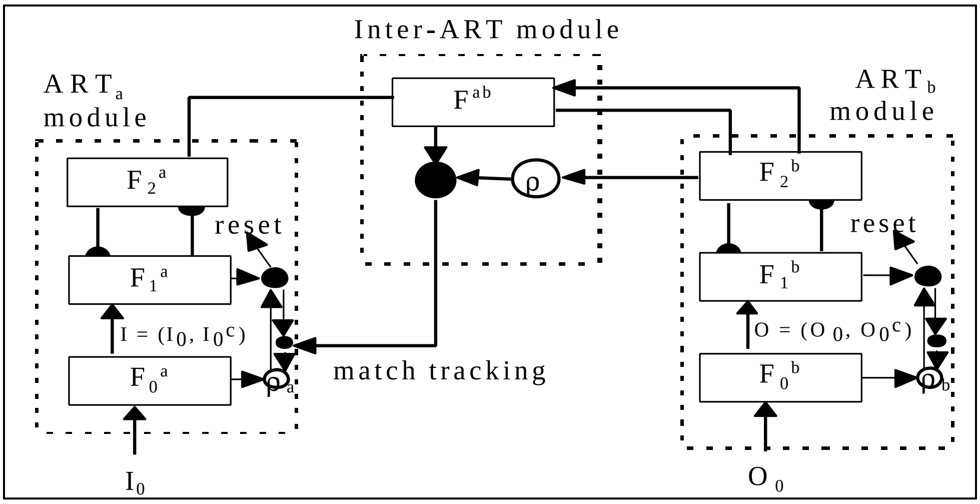

Figure 3. Simplified Fuzzy ARTMAP Architecture.

If $x^{ab} = 0$, then there is a mismatch. Inter-ART module triggers a *match tracking* mechanism. This mechanism increases r_a, by a minimum value and hence, forces the ART_a module to search for another category suitable to be associated with the desired output vector.

4. CHARACTER FEATURES

The input data of hand-written character has a resolution of 32 x 32 pixels with black/white colour. In order to reduce the number of inputs to the neural network, we develop a simple pre-processing technique. We segmentise the bitmap into an 8 x 8 region (Figure 4) with each region taking up 4 x 4 pixels.

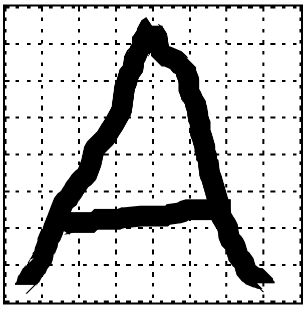

Figure 4. A 32x32 bitmap segmented into 8 x 8 region.

Each region has a value between 0 and 1. The value of the region, *v* obeys the equation

$$v = \frac{1}{M * N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \qquad (26)$$

where

M denotes the number of pixels in one row = 4

N denotes the number of pixels in one column = 4

f(x,y) denotes the binary value of pixel at position (x, y)

The result of the pre-processing is shown as follows :-

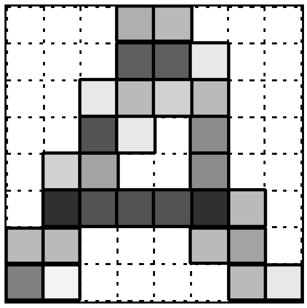

Figure 5. 8x8 bitmap after the compression

The value of each region is then passed on to the neural network for classification. The pre-processing significantly reduced (from 32x32=1024 down to 8x8=64) the input vector to the neural network and yet retained most of the features of the character. Some samples of the hand-written character used in the experiments are given in Figure 6.

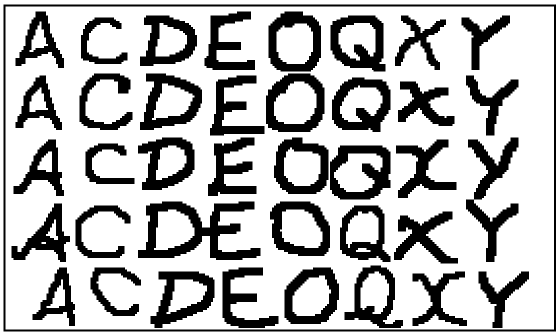

Figure 6. Samples of hand-written character used in this experiment.

5. EXPERIMENTAL SET-UP

The database consists of more than 1500 samples of black and white hand-written characters. The resolution of each character is 32 x 32 pixels. The samples are separated into 2 sets of training and testing data. Training data set is sub-grouped into 6 whereas testing data set is sub-grouped into 3 categories.

5.1. Convergence Rate

In this experiment, the convergence rates of both types of neural networks are studied to make a comparison between the two neural networks.

Each neural network is trained with 6 groups of data from the training data set, and the time for the networks to converge to a stable state is logged. One full presentation of the training patterns to the networks is considered as one epoch.

For the Fuzzy ARTMAP, stable state means the network does not select a new/uncommitted category, while for the MLP, it simply means that the mean square error (m.s.e) has achieved a minimum value of 0.1. Results are shown as in **Table 3**.

5.2. Recognition Accuracy

All the trained neural networks are then tested with the data stored in testing data sets. The neural networks trained with groups 1 and 2 are tested with test group 1. The networks trained with groups 3 and 4 are tested with test group 2. The networks trained with groups 5 and 6 are tested with test group 3. The accuracy of the network recognition is then logged.

Table 4 shows the result of the recognition accuracy test. The result shows that the Fuzzy ARTMAP can recognise more accurately when compared to the MLP. It also shows that both networks have better performances if presented with more training patterns (comparing between training groups 1 and 2, 3 and 4, 4 and 5). The networks performed worse if the classes of character are more (comparing between training group 1,3, and 5, 2,4 and 6).

5.3. Online Learning

For the MLP, the network has to be retrained if a new class of pattern is required to learn[3]. The Fuzzy ARTMAP can learn a new pattern without retraining. In the experiment, the network is initially trained with training group 2 that has 5 classes of characters. The recognition accuracy of the network to test group 2 that has 10 classes of characters are then recorded. The same network is then trained with training group 7 that has another 5 classes of character, and the recognition accuracy of the network to test group 2 are then recorded.

The result of the experiment is shown in **Table 5**. It can be observed that for the network that is trained with only group 2 has the accuracy of 48.8%. The accuracy increased to 88.8% if the same network is being trained with group 7 which has another 5 classes of characters.

6. CONCLUSIONS

In this paper, we have shown that the Fuzzy ARTMAP neural network architecture offers distinct advantages over the MLP neural network. These advantages are:

- Faster convergence rate. Our results show that the Fuzzy ARTMAP training algorithm requires much less training iteration as compared to the MLP neural network.
- Better recognition accuracy. The Fuzzy ARTMAP can recognise the characters more accurately when compared to the MLP.
- Online learning capability. This capability has made Fuzzy ARTMAP to learn a new pattern without having to retrain the network.

Table 1. Groups in Training Data Set.

Group	Classes of char.	Occur. / Class	Total char.
1	5	1	5
2	5	20	100
3	10	1	10
4	10	20	200
5	26	1	26
6	26	20	520

Table 2. Groups in Testing Data Set.

Group	Classes of char.	Occur. / Class	Total char.
1	5	40	200
2	10	40	400
3	26	40	1040

Table 3. Comparison of the convergence rate between Fuzzy ARTMAP and MLP. The table shows that convergence rate for the Fuzzy ARTMAP neural network is much faster than the MLP.

Training Group	Fuzzy ARTMAP		MLP	
	epoch	Time (ms)	epoch	Time (ms)
1	2	0	1763	19600
2	2	100	152	31200
3	2	0	696	14800
4	5	300	49	21000
5	2	100	425	23000
6	5	1600	80	164000

Table 4. Comparison of recognition accuracy between the Fuzzy ARTMAP and MLP.

Train with group	Test with group	Total number of data	Fuzzy ARTMAP		MLP	
			Match	Accuracy (%)	Match	Accuracy (%)
1	1	200	171	85.5	102	51.0
2	1	200	195	97.5	158	79.0
3	2	400	297	74.3	220	55.0
4	2	400	355	88.8	351	87.8
5	3	1040	683	65.7	207	19.9
6	3	1040	893	85.9	732	70.9

Table 5. Result of the online learning of Fuzzy ARTMAP.

Train with group	Test with group	Total number of data	Match	Accuracy (%)
2	2	400	195	48.8
2,then 7	2	400	355	88.8

REFERENCES

[1] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Map," Neural Networks, vol. 3, 1992, pp. 698-713.

[2] G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," Computer Vision, Graphs, and Image Processing, vol. 37, 1987, pp. 54-115.

[3] Clifford R. Parten ... et al., Handbook of Neural Computing Applications, Academic Press, Inc. SD, CA, USA.

[4] K. Fukushima , "Neocognitron: A hierarchical neural network capable of pattern recognition," Neural Networks, vol. 1, 1988, 119-130.

[5] J.A. Freeman, Simulating Neural Networks with Mathematica, Addison-Wesley, USA.

[6] KulKarni, Arun D, Artificial neural networks for image understanding, ITP, USA.

[7] A. Shustorovich and C.W. Thrasher, "Neural Network Positioning and Classification of Handwritten Characters," Neural Networks, Vol. 9, No. 4, 1996, pp. 685-693.

[8] D.E. Rumelhart et al., Parallel Distributed Processing: Explorations in the Micro Structure of Cognition Vol. I: Foundations, MIT Press, Cambridge, MA, 1986.