6.1 INTRODUCTION

Accurate weather forecasting is a necessary part of human being for various industries. The most often used neural network to solve the weather forecasting problem is the back-propagation neural network architecture, Radial Basis Function Neural Network Architecture and Generalized Regression Neural Network. Although the performance of the back-propagation neural network architecture, Radial Basis Function Neural Network Architecture and Generalized Regression Neural Network has been encouraging, it is worth noting that it suffers from the slow convergence problem and the difficulty of interpreting the answers that the architecture provides. A neural network architecture that does not suffer from the above mentioned drawbacks is the Fuzzy ARTMAP neural network, developed by Carpenter, Grossberg, and their colleagues at Boston University. In this chapter the Fuzzy ARTMAP neural network applied for weather forecasting problem. Fuzzy ARTMAP is a supervised neural network architecture that is based on Adaptive Resonance Theory (ART) [HTTP6].

Types of ART

**ART 1** is the simplest variety of ART networks, accepting only binary inputs [CG03].

**ART 2** extends network capabilities to support continuous inputs [CG87].

**ART 2-A** is a streamlined form of ART-2 with a drastically accelerated runtime, and with qualitative results being only rarely inferior to the full ART-2 implementation [CG91a].

**ART 3** builds on ART-2 by simulating rudimentary neurotransmitter regulation of synaptic activity by incorporating simulated sodium (Na+) and calcium (Ca2+) ion concentrations into the system’s equations, which results in a more physiologically realistic means of partially inhibiting categories that trigger mismatch resets [CG90].
**Fuzzy ART** implements fuzzy logic into ART’s pattern recognition, thus enhancing generalizability. An optional feature of fuzzy ART is complement coding, a means of incorporating the absence of features into pattern classifications, which goes a long way towards preventing inefficient and unnecessary category proliferation [CG91b].

**ARTMAP**, also known as Predictive ART, combines two slightly modified ART-1 or ART-2 units into a supervised learning structure where the first unit takes the input data and the second unit takes the correct output data, then used to make the minimum possible adjustment of the vigilance parameter in the first unit in order to make the correct classification [CG91].

**Fuzzy ARTMAP** is merely ARTMAP using fuzzy ART units, resulting in a corresponding increase in efficiency [CGM92].

The hypothesis of this chapter was to apply the fuzzy ARTMAP neural network to the weather forecasting problem. The short training time of the fuzzy ARTMAP allowed us to conduct more experiments in order to identify an "optimum" input parameter set, and led eventually to more accurate forecasts. The operation of the fuzzy ARTMAP architecture is better understood and the answers derived by the architecture can be logically explained. Therefore, it should be possible to determine why certain days are more difficult to forecast than others, and hence what to do to improve the forecast accuracy for those days.

Recently, the weather forecasting techniques has been considered as an important approach for developing the intelligent systems that shows promises for improving their performance under real-time distributed environment. The ARTMAP is a class of neural network architecture that performs incremental supervised learning recognition to the input vectors. The first ARTMAP system was used to classify inputs by the set of features (also called pattern or vector) they possess of a binary values representing the presence or absence of each possible feature [CG91]. Fuzzy ARTMAP was developed to classify the inputs by a fuzzy set of features, or patterns of fuzzy memberships values.
between 0 or 1 indicating the extent to which each feature is presented [CG87]. A verity of fuzzy ARTMAP on a cluster of workstations learns the required tasks fast and has the capability for on-line learning was implemented. It has the ability to provide the learning structure that allows explaining the answers that the neural network produces [CA04].

In batch supervised learning mode, fuzzy ARTMAP may also be efficient in that its asymptotical generalization error can be achieved for a moderate time and space complexity [EG99]. As such, they have been successfully applied in complex real-world pattern recognition tasks such as the recognition of radar signals, multi-sensor image fusion, remote sensing and data mining, recognition of handwritten characters [BD03, CGW97, PC03]. The fuzzy ARTMAP neural network is an architecture that can learn arbitrary mappings from analog or digital inputs of any dimensionality to analog or digital outputs of any dimensionality. The architecture applies incremental supervised learning of recognition categories and multidimensional maps in response to arbitrary sequences of analog or binary input vectors, which may represent fuzzy or crisp set of features.

Beyond the network’s internal dynamics, decisions taken as to the supervised learning process of a data set may significantly affect fuzzy ARTMAP’s capacity to generalize. Performance will degrade with a poor choice of user-defined hyper-parameter values and manipulation of training data. For instance, fuzzy ARTMAP neural networks are known to suffer from overtraining, which is directly connected to a category proliferation problem. Overtraining generally occurs when a neural network has learned not only the basic mapping associated training subset patterns, but also the subtle nuances and even the errors specific to the training subset. The issue of overtraining may stem from decisions taken for supervised batch learning [HG05, AK01].

During the fuzzy ARTMAP supervised learning process, one can manipulate neural network inputs at disposal – data set and user-defined hyper-parameter values – to achieve a high level of performance. In this context, the user’s decisions include choosing
the supervised learning strategy (and thus, the number of training epochs), the proportion of patterns in the training subset to those in validation and test subsets, the parameter values, the data normalization technique, and the data presentation order. The impact of these decisions on generalization error are necessarily a function of the data set structure (overlap and dispersion of patterns, etc.), and therefore of the type of decision boundary among patterns belonging to different recognition classes. Although fuzzy ARTMAP performance depends on a set of user-defined hyper-parameters, and these parameters should normally be fine-tuned to each specific problem [CG92], the influence of hyper-parameter values is rarely addressed in ARTMAP literature.

Moreover, the few techniques found in this literature for automated hyper-parameter optimization, focus mostly on the vigilance parameter, even though there are four inter-dependent parameters (vigilance, learning, choice, and match tracking). A popular choice consists in setting hyper parameter values such that network resources (the number of internal category neurons, the number of training epochs, etc.) are minimized [CM98]. This choice of parameters may however lead to overtraining, and significantly degrade the capacity to generalize. An effective supervised learning strategy could involve co-jointly optimizing both network (weights and architecture) and all its hyper-parameter values for a given problem, based on a consistent performance objective. General structure of weather forecasting using Fuzzy ARTMAP is shown in figure 6.1.
ART encompasses a wide variety of neural networks based explicitly on human information processing and neurophysiology. ART networks are defined algorithmically in terms of detailed differential equations intended as plausible models of biological neurons. ART networks are implemented using analytical solutions or approximations to these differential equations. ART is capable of developing stable clustering of arbitrary sequences of input patterns by self-organization. Fuzzy ARTMAP's internal control mechanisms create stable recognition categories of optimal size by maximizing code compression while minimizing predictive error during on-line learning. Fuzzy ARTMAP incorporates fuzzy logic in its ART modules. The objective of this work is to develop an architecture based on the adaptive resonance theory, being capable to solve or at least to reduce the imprecision of weather forecasting results, by a mechanism that separate the analog and binary data, and processing separately.

ARTMAP (Adaptive Resonance Theory Mapping) is a class of neural network architecture that performs incremental supervised learning of recognizing categories and multi dimensional maps in response to input vectors presented in arbitrary order. The first ARTMAP system was used to classify inputs by the set of features that possesses. A more general ARTMAP system that learns to classify inputs by a fuzzy set or features or
a pattern of fuzzy membership values between 0 and 1 indicating extent to which each
feature is present. This generalization is accomplished by replacing the ART-1 module
of binary ARTMAP system with fuzzy ART module. ART-1 dynamics are described in
terms of set–theory operations.

Fuzzy-ARTMAP differs from many previous fuzzy pattern recognition algorithms
in that it learns each input as it is received on-line, rather than performing an offline
optimization of a criterion function. ART-1 learns stable categories only in response to
binary input vectors. Fuzzy-ART can learn stable categories only in response to binary
input vectors. Fuzzy-ART can learn stable categories in response to either analog or
binary input vectors. Fuzzy – ART reduces to ART-1 in response value whether or not
they represent fuzzy sets. In fuzzy-ART, learning always converges because on adaptive
weights are monotonically non-increasing. A preprocessing step known as complement
coding may be adopted to prevent category proliferation. Complement coding normalizes
input vectors while preserving the amplitudes of individual feature activations. Without
complement coding an ART category memory encodes the degree to which critical
features are consistently present in the training exemplars of that category. With
complement coding, the category weight vector presents both degree of absence and the
degree of presence of features. A binary ARTMAP system that comprises of a pair of
ART modules ($\text{ART}_a$ and $\text{ART}_b$) as shown in the figure 6.2.
Each of these modules creates stable recognition categories in response to arbitrary sequence of input patterns. As associative learning network and an internal controller, link these modules to make the ARTMAP system to operate in real time. Also the recognition categories learning by the $F_2^a$ category – nodes are compatible with rules that link antecedents to consequences. Therefore rules can be readily inserted into ARTMAP network that can be trained by examples. During learning, new recognition categories (rules) can be created dynamically to cover the deficiency of domain theory. This is in contrast with the static architecture of the standard slow learning back propagation networks. Also, by self stabilizing property learning in ARTMAP does not wash away the existing knowledge and the meanings of units do not shift. This allows preservation of symbolic rules during neural network learning. Using a generalized ARTMAP rule extraction procedure, the final system states can be converted back to a compact set of rules. This enables direct comparison of the original knowledge with the refined rules.
6.2 FUZZY – ART ALGORITHM

Each ART system includes a field \( F_0 \) of nodes that represents a current input vector, a field \( F_1 \) that receives both bottom–up input from \( F_0 \) and top-down input from a field \( F_2 \) that represents active code or category. The \( F_0 \) activity vector is denoted by \( I = (I_1, I_2, \ldots I_M) \) with each component \( I_i \) in the interval \([0, 1]\), \( i = 1, 2, \ldots, M \). The \( F_1 \)-activity vector is denoted by \( X = (x_1, x_2, \ldots x_M) \) and the \( F_2 \) activity vector is denoted by \( Y = (y_1, y_2, \ldots y_N) \).

The number of nodes in each field is arbitrary. Associated with each \( F_2 \) category node \((j = 1, 2, \ldots N)\) is a vector \( W_j = (w_{j1}, w_{j2}, \ldots w_{jM}) \) of adaptive weights traces. Initially \( w_{j1}(0) = w_{j2}(0) = \ldots w_{jM}(0) = 1 \) then each category is said to be uncommitted. After category is selected for coding it becomes committed. Each trace \( w_{ji} \) is monotonically non-increasing through time and hence converges to a limit. The fuzzy-ART weight vector \( W_i \) subsumes both the bottom-up and top-down weight vectors of ART-1.

Basic differences between ART-1 and fuzzy - ART are given in table 6.1

<table>
<thead>
<tr>
<th>Type of network</th>
<th>ART-1 (Binary)</th>
<th>Fuzzy-ART (Analog)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category choice</td>
<td>( T_j = \frac{</td>
<td>I \cap w_j</td>
</tr>
<tr>
<td>Match criterion</td>
<td>( \frac{</td>
<td>I \cap w</td>
</tr>
<tr>
<td>Fast learning</td>
<td>( W_j^{\text{new}} = I \cap W_j^{\text{old}} ) ( \cap: \text{Logical and intersection} )</td>
<td>( W_j^{\text{new}} = I \wedge W_j^{\text{old}} ) ( : \text{Fuzzy and minimum} )</td>
</tr>
</tbody>
</table>
6.3 FUZZY-ARTMAP

Neural networks are in general sensitive to absolute magnitude and fluctuations in inputs and may tend to swamp the performance of the network while predicting the desired outputs. Hence there is a need for normalization of inputs so that the inputs correspond to the same range of values. If a is given input pattern vector of d features, the compliment coded vector a represent the absence of each feature. Just as in fuzzy logic, all fuzzy- ARTMAP input values must be within the range 0 to 1. Therefore, the complement coded input vector I internal to fuzzy ARTMAP is given by a two dimensional vector \( I = (\{a\}, \{\bar{a}\}) = (a_1, a_2, ..., a_d, \bar{a}_1, a_2, ..., \bar{a}_d) \). The activation function value \( T(I) \) of the top–down weight – node \( W_1 \) for the input I shown, is set to null and is set to the category ‘IN’ in the category layer.

Fuzzy- ARTMAP is essentially a 2-layer network containing two fuzzy-ART modules \( ART_a \) and \( ART_b \) that are linked together via an inter-ART map field \( F_{ab} \). The map field forms predictive associations between the categories and realizes the ARTMAP match-tracking rule shown in the figure 6.2. The inputs to \( ART_a \) and \( ART_b \) are complement coded. The \( ART_a \) complement coding preprocessor transforms the \( M_a \)-vector \( a \) into the \( 2M_a \)-vector \( A(a, \bar{a}) \) at the \( ART_a \) field \( F_0^a \). A is the input vector to the \( ART_a \) field \( F_1^a \). Similarly, the input to \( F_1^b \) is the \( 2M_b \)-vector \( (b, \bar{b}) \). When \( ART_b \) disconfirms a prediction of \( ART_a \), map field inhibition induces the match tracking process. Match tracking raises the \( ART_a \) vigilance \( (\rho_a) \) to just above the \( F_1^a \) to \( F_0^a \) match ratio \( |x^a| / |A| \). This triggers an \( ART_a \) search which leads to activating of either an \( ART_a \) category that correctly predicts b or to a previously uncommitted \( ART_a \) category node.
Output Node Activation

When fuzzy-ARTMAP is presented an input pattern whose complement coded representation is I, all output nodes become active to some degree. This output activation is denoted by $T_j$ for the $j^{th}$ output node, $w_j$ is the corresponding weight, then

$$T_i(I) = \frac{|I^\wedge W_i|}{\alpha + |W_j|}$$

6.1

Where the symbol $\wedge$ indicates the fuzzy AND operator defined by $(a^\wedge b)_i = \min(a_i, b_i)$, with $a$ and $b$ as fuzzy vectors of say $M$-dimensions. The norm $|.|$ is defined by $|a| = \sum_{i=1}^{M} |a_i|$.

Here $\alpha$ is taken as a small value close to 0, usually about $10^{-7}$. The winning output node is the node with the highest activation function $T_j$, (i.e.), Category choice is indexed by $j$, then $T_j = \max \{T_j = 1, 2, \ldots N\}$. If more than one $T_j$ is maximal, the output node $j$
with smallest index is arbitrarily chosen to break the tie. The category associated with the
winning output node is described as the network’s classification of the current input
pattern. When used in conjunction with the vigilance parameter, the match function value
states whether the current input is a good enough match to a particular output node to be
encoded by that output node or instead if new output node should be formed to encode
the input pattern. If the match functions value is greater than the vigilance parameter ($\rho$),
the network is said to be in a state of resonance.

\[ \text{i.e., } \left[ \frac{I^\top W_j}{|I|} \right] > \rho. \quad (6.2) \]

Resonance means that output node $j$ is good enough to encode input $I$ provided
that output node $j$ represents the same category as input $I$. A network state of miss match
reset occurs if the match function is less than vigilance. This state indicates that the
current input node does not meet the encoding granularity represented by the vigilance
parameter and therefore cannot update its weights even if the input pattern’s category is
equal to the category of the winning output node. Then the value of choice function $T_j$ is
set to 0 for the duration of the input – presentation to prevent the persistent selection of
the same category during search. A new index is then chosen. The search process
continues until the chosen $J$ satisfies the vigilance criterion. Once search ends, a winning
output node $j$ is selected to learn a particular input pattern $l$, the top – down weight vector
$w_j$ from the output node is updated according to the equation:

\[ W_j^{\text{new}} = \beta (I^\top W_j^{\text{old}}) + (1 - \beta) W_j^{\text{old}} \quad (6.3) \]

Where $0 < \beta \leq 1$. Fast learning corresponds to setting $\beta = 1$. In fast –learn – ART-1,
if the choice parameter $\alpha$ is chosen close to 0, then the first category chosen by $\max\{T_j\}$
is always the category whose weight vector $W_j$ is the largest coded subset of the input
vector $I$, if such a category exists.
With reference to figure 6.2, for ART\(_a\), \(I = A = (a, a^c)\) and for ART\(_b\), \(I = B = (b, b^c)\). For ART\(_a\), \(x^a\) denotes the \(F^a_1\) output vector; \(y^a\) denote the \(F^a_2\) output vector and \(W^a_j\) denotes the \(j^{th}\) ART\(_a\) weight vector. For ART\(_b\), \(x^b\) denotes the \(F^b_1\) output vector, \(y^b\) denotes the \(F^b_1\) output vector and \(W^b_k\) denotes the \(k^{th}\) ART\(_b\) weight vector. For the map field, \(x^{ab}\) denotes \(F^{ab}\) output vector and \(w^{ab}_j\) denotes the weight vector from the \(j^{th}\) node to \(F^a_2\) node \(F^{ab}\).

The map-field is used to form predictive association between categories. In fuzzy-ARTMAP map field \(F^{ab}\) receives input from either or both of the ART\(_a\) or ART\(_b\) category fields. If node \(J\) of \(F^a_2\) is chosen, then its weight \(W^{ab}_J\) activate \(F^{ab}\). An active \(F^b_2\) node \(K\) sends input to \(F^{ab}\) via one-to-one pathways between \(F^b_2\) and \(F^{ab}\). If both ART\(_a\) and ART\(_b\) are active, then \(F^{ab}\) remains active only if ART\(_a\) predicts the same category as ART\(_b\) via the weight \(w^{ab}_J\). The \(F^{ab}\) output vector \(x^{ab}\) obeys:

\[
x^{ab} = \begin{cases} 
  y^b \times w^{ab}_J & \text{if } J^{th} F^a_2 \text{ node is active } F^b_2 \text{ is active} \\
  w^{ab}_J & \text{if } J^{th} F^a_2 \text{ node is active } F^b_2 \text{ is inactive} \\
  y^b & \text{if } F^a_2 \text{ inactive and } F^b_2 \text{ is active} \\
  0 & \text{if } F^a_2 \text{ inactive and } F^b_2 \text{ is inactive}
\end{cases}
\]

By the above equation, \(x^{ab} = 0\) if \(y^b\) fails to confirm the map field prediction made by \(w^{ab}_J\). Such a mismatch event triggers an ART\(_a\) search for a better category as follows: At the start of each input presentation ART\(_a\) vigilance \(\rho_a\). When a predictive error occurs, match tracking raises ART\(_a\) vigilance just enough to trigger a search for a new \(F^a_2\)
coding node. ARTMAP detects a predictive error when \(|x_{ab}| < \rho_{ab} |y^b|\), where \(\rho_{ab}\) is the map field vigilance parameter. A signal from the map field to the \(A_{1}\) orienting subsystem cause \(\rho_a\) to track the \(F_a^a\) match. That is \(\rho_a\) increase until it is slightly higher than the \(F_a^a\) match value \(|A^a w_j|^{|A|}^j\). Then since \(A_{1}\) fails to meet the matching criterion, the search for another \(F_a^a\) node begins.

**Map field learning**

Map field learning: Weight \(w_{jk}^{ab}\) in \(F_2^a \rightarrow F^{ab}\) paths initially satisfy: \(w_{jk}^{ab}(0) = 1\) field vector \(x_{ab}\) as in fuzzy –ART. With fast learning, once J learns to predict the \(A_{1}\) category – \(K\), that association is permanent (i.e.), \(w_{jk}^{ab} = 1\) for all time.

Once fuzzy associative map is trained, the equivalent of a feed-forward pass for an unknown pattern classification consists of passing the input pattern throughout the complement code and into the input layer. The output node activation function is evaluated and the winner is the one with the highest value. The category of the input pattern is the one with which the winning output node is associated. Learning rules determine how the map-field weight changes through times are as follows:

Weights \(w_{jk}^{ab}\) in \(F_2^a \rightarrow F^{ab}\) paths initially satisfy \(w_{jk}^{ab}(0) = 1\)

During resonance with \(A_{1}\) category-J active, \(w_{jk}^{ab}\) approaches the map filed vector \(x_{ab}\). With fast learning, once J learns to predict the \(A_{1}\) category – \(K\), that association is permanent., i.e., \(w_{jk}^{ab} = 1\) for all time.

**6.4 THE FUZZY ARTMAP LEARNING ALGORITHM**

1. Initialize all weights.
2. Present input at ART\textsubscript{a} and target output at ART-b. Allow category formation in both modules. The F2 layer nodes in ART-b encode the class information.

3. Get predicted output class information from ART-a by using the map field weights \( w_{ab} \).

4. Compare predicted class with actual class information. If they are the same, go to step 6.

5. Reset current representation in ART-a and search for a better representation.

6. Repeat steps 4 and 5 until a correct match is found.

7. Update all weights in both modules.

8. Go to step 2.

6.5 RESULTS AND DISCUSSION

Training and test features are taken from Meteorological department Kanyakumari District, Tamilnadu. All available rain and no rain features are taken for classification. The validation parameters; viz Precision, Sensitivity, Specificity and Accuracy are 91.6\%, 94.6\%, 91.9\%, and 93.2\% respectively. Here the 13 misclassified datasets belong to rain features and 21 misclassified dataset belonging to no rain features. Time taken for training the network is 6.55 seconds. Table 6.2 shows confusion matrix result for weather prediction. Performance analysis for weather prediction shown in figure 6.4 and Receiver Operating Characteristic (ROC) curves for this analysis is shown in figure 6.5

<table>
<thead>
<tr>
<th>Table 6.2 Confusion Matrix Result of Fuzzy ARTMAP in weather prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy ARTMAP Weather prediction</td>
</tr>
<tr>
<td>---------------------------------</td>
</tr>
<tr>
<td>Rain</td>
</tr>
</tbody>
</table>
Figure 6.4 Performance Analysis of Fuzzy ARTMAP

<table>
<thead>
<tr>
<th></th>
<th>Rain</th>
<th>No Rain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain</td>
<td>237</td>
<td>229</td>
</tr>
<tr>
<td>(94.8%)</td>
<td>(8.4%)</td>
<td>(91.6%)</td>
</tr>
<tr>
<td>No Rain</td>
<td>21 (5.2%)</td>
<td>250</td>
</tr>
</tbody>
</table>

Table: Performance Analysis of Fuzzy ARTMAP
Overall, the performance of the Fuzzy ARTMAP model is reasonable. However, compared to the other models, it is less accurate for the weather-forecasting problem. The Fuzzy ARTMAP neural network has an on-line fast learning mechanism and has the superior performance for classifying, with very low computing costs for learning strategy.

Figure 6.5 ROC curve for Fuzzy ARTMAP