A New Architecture of Simplified Fuzzy ARTMAP with the Extended Definition of Complementation
Pranamika Kakati
Department of Computer Application, Girijananda Chowdhury Institute of Management & Technology, Guwahati, Assam, India.
pranamikakakati2013@gmail.com

Abstract

Simplified Fuzzy ARTMAP is an architecture which synthesizes Fuzzy Logic with ART Neural Networks. This is one of the most researched form of Neuro-Fuzzy hybrid system and has resulted in a stupendous quantity of publications and research results. The existing Simplified Fuzzy ARTMAP architecture is based on traditional Zadehian Theory of Fuzzy sets where it is believed that there is no difference between Fuzzy membership function and Fuzzy membership value for the complement of a Fuzzy set which is already proved to be wrong. In this article, our objective is to present a new architecture of Simplified Fuzzy ARTMAP with the help of extended definition of complementation of Fuzzy sets using reference function. Our proposed architecture is based on the fact that Fuzzy membership function and Fuzzy membership value for the complement of a Fuzzy set are two different things. Also, an effort has been put forward to demonstrate the application of the proposed architecture with the evaluation of some example data.

Keywords: Complement of a Fuzzy set, Fuzzy membership function, Fuzzy membership value, Fuzzy reference function, Fuzzy set, Simplified Fuzzy ARTMAP.

1. Introduction

Kasuba[5] introduced Simplified Fuzzy ARTMAP (SFARTMAP) in 1993. SFARTMAP is a Neuro – Fuzzy system which is an integration of Neural Network and Fuzzy Logic. In the literature this system is known as Neural Network driven Fuzzy reasoning. The existing SFARTMAP architecture is based on Zadehian concept of complementation of Fuzzy set. Zadeh [1] initiated Fuzzy sets in 1965. In Zadehian theory of Fuzzy set, it has been believed that the classical set theoretic axioms of exclusion and contradiction are not satisfied. It has been believed that for a Fuzzy set A and its complement A\(^C\), neither \(A \cap A^C\) is null set nor \(A \cup A^C\) is the universal set which is proved to be wrong by Baruah[2,3]. Baruah proposed that in the Zadehian definition of the complement of a Fuzzy set, Fuzzy membership function and Fuzzy membership value had been taken to be the same, which leads to the conclusion that the Fuzzy sets do not follow the set theoretic axioms of exclusion and contradiction. Accordingly, Baruah has put forward an extended definition of Fuzzy set and reintroduced the notion of complement of a Fuzzy set in a way that the set theoretic axioms of exclusion and contradiction can be seen valid for Fuzzy sets also. According to Baruah, to define a Fuzzy set two functions namely- Fuzzy membership function and –Fuzzy reference function are necessary. Fuzzy membership value is the difference between Fuzzy membership function and Fuzzy reference function. Fuzzy membership function and Fuzzy membership value are two different things. Neog and Sut [4] have generalized the concept of complement of a Fuzzy set, introduced by Baruah[2,3], when the Fuzzy reference function is not zero and defined arbitrary Fuzzy union and intersection extending the definition of Fuzzy sets given by Baruah [2, 3]. Therefore, it has been observed that the existing architecture of SFARTMAP is based on a concept which itself is controversial. It is obvious that any result which is obtained with the help of something which itself is controversial cannot yield a suitable result. Hence the existing architecture of the SFARTMAP must be changed accordingly. In this paper, our aim is to put forward a new architecture of SFARTMAP using the extended definition of complementation for Fuzzy set based on reference function[2,3,4]. Also, we demonstrate the application of the proposed architecture with the help of some collected data.

The overall organization of this paper is as follows. In section 2 we discuss the new and extended definition of Fuzzy set. In section 3 we propose a new architecture of Simplified Fuzzy ARTMAP on the basis of extended definition of complementation of Fuzzy set. In section 4 an algorithm has been designed for the proposed architecture. The application of the proposed architecture has been demonstrated on some collected data.
dataset in section 5. Some conclusions are given in section 6.

2. Baruah's Extended definition of Fuzzy set

Baruah put forward an extended definition of Fuzzy sets in the following manner –

Let \( \mu_1(x) \) and \( \mu_2(x) \) be two functions, where \( 0 \leq \mu_1(x) \leq \mu_2(x) \leq 1 \). For a Fuzzy number denoted by \([x, \mu_1(x), \mu_2(x); x \in U]\) we would call \( \mu_1(x) \), the Fuzzy membership function and \( \mu_2(x) \) a reference function such that \( \{ \mu_1(x) - \mu_2(x) \} \) is the Fuzzy membership value for any x.

According to Baruah, in the definition of complement of a Fuzzy set, the Fuzzy membership value and the Fuzzy membership function have to be different in the sense that for a usual Fuzzy set the membership value and the membership function are of course equivalent.

2.1. Extended Definition of Union and Intersection of Fuzzy Sets

With the help of the extended definition, Baruah put forward the notion of union and intersection of two Fuzzy sets in the following manner –

Let \( A(\mu_1, \mu_2) = \{x, \mu_1(x), \mu_2(x); x \in U\} \) and \( B(\mu_3, \mu_4) = \{x, \mu_3(x), \mu_4(x); x \in U\} \) be two Fuzzy sets defined over the same universe U, where \( \mu_1, \mu_2 \) and \( \mu_3, \mu_4 \) are membership and reference functions of A and B respectively.

Now on the basis of Baruah's extended definition of Fuzzy set, we can represent these two Fuzzy sets A and B in the number line in Figure 1 and Figure 2 respectively.

![Figure 1. Representation of Fuzzy set A in number line](image)

![Figure 2. Representation of Fuzzy set B in number line](image)

Then the operations intersection and union are defined as

\[
A(\mu_1, \mu_2) \cap B(\mu_3, \mu_4) = \{x, \min(\mu_1(x), \mu_3(x)), \max(\mu_2(x), \mu_4(x)); x \in U\}
\]

and \( A(\mu_1, \mu_2) \cup B(\mu_3, \mu_4) = \{x, \max(\mu_1(x), \mu_3(x)), \min(\mu_2(x), \mu_4(x)); x \in U\} \)

Two Fuzzy sets \( C = \{x, \mu_C(x); x \in U\} \) and \( D = \{x, \mu_D(x); x \in U\} \) in the usual definition would be expressed as

\[
C(\mu_C, 0) = \{x, \mu_C(x), 0; x \in U\}
\]

and \( D(\mu_D, 0) = \{x, \mu_D(x), 0; x \in U\} \)

Accordingly, we have,

\[
C(\mu_C, 0) \cap D(\mu_D, 0)
\]

\[
= \{x, \min(\mu_C(x), \mu_D(x)), \max(0, 0); x \in U\}
\]

\[
= \{x, \min(\mu_C(x), \mu_D(x)), 0; x \in U\}
\]

\[
= \{x, \mu_C(x), \max(0, 0); x \in U\}
\]

which in the usual definition is nothing but \( C \cap D \).

Similarly we have,

\[
C(\mu_C, 0) \cup D(\mu_D, 0)
\]

\[
= \{x, \max(\mu_C(x), \mu_D(x)), \min(0, 0); x \in U\}
\]

\[
= \{x, \max(\mu_C(x), \mu_D(x)), 0; x \in U\}
\]

\[
= \{x, \mu_C(x), \min(0, 0); x \in U\}
\]

which in the usual definition is nothing but \( C \cup D \).

Neog and Sut [4] showed by an example that this definition sometimes gives degenerate cases and revised the above definition as follows -

Let \( A(\mu_1, \mu_2) = \{x, \mu_1(x), \mu_2(x); x \in U\} \)

and \( B(\mu_3, \mu_4) = \{x, \mu_3(x), \mu_4(x); x \in U\} \) be two Fuzzy sets defined over the same universe U.

The operation intersection is defined as

\[
A(\mu_1, \mu_2) \cap B(\mu_3, \mu_4)
\]

\[
= \{x, \min(\mu_1(x), \mu_3(x)), \max(\mu_2(x), \mu_4(x)); x \in U\}
\]

With the condition that

\[
\min(\mu_1(x), \mu_3(x)) > \max(\mu_2(x), \mu_4(x)) \forall x \in U.
\]

Now if for some \( x \in U \),

\[
\min(\mu_1(x), \mu_3(x)) < \max(\mu_2(x), \mu_4(x))
\]

Then our conclusion is that \( A \cap B = \emptyset \).

and if for some \( x \in U \),

\[
\min(\mu_1(x), \mu_3(x)) = \max(\mu_2(x), \mu_4(x))
\]

then also \( A \cap B = \emptyset \).

Further the operation Union is defined as
A(μ₁,μ₂)UB(μ₃,μ₄) \\
={x, max(μ₁(x),μ₃(x)), min(μ₂(x),μ₄(x)); x∈U} \\
with the condition that \\
min(μ₁(x),μ₄(x))≤ max(μ₂(x),μ₃(x)) \forall x∈U. \\

if for some x∈U, \\
min(μ₁(x),μ₃(x))< max(μ₂(x),μ₄(x)) \\
then the union of Fuzzy sets A and B cannot be expressed as one single Fuzzy set.

The union however can be expressed in one single Fuzzy set if 
for some x∈U, 
min(μ₁(x),μ₄(x))=max(μ₂(x),μ₃(x)).

We can clearly visualize: these extended definitions of union and intersection are valid in our Figure 1 and Figure 2 also.

2.2. Complement of a Fuzzy Set Using Extended Definition

Baruah put forward the notion of complement of usual Fuzzy sets with Fuzzy reference function 0 in the following way –

Let A(μ₀)=∪₁x∈U [x, μ(x),0 ; x∈U] 
and B(1,μ)=∪₁x∈U [x, 1,μ(x) ; x∈U] be two Fuzzy sets defined over the same universe U.

Now on the basis of Baruah’s extended definition of Fuzzy set, we can represent these two Fuzzy sets A and B in the number line in Figure 3 and Figure 4 respectively.

Figure 3. Representation of Fuzzy set A in number line

Figure 4. Representation of Fuzzy set B in number line

Now we have 
A(μ₀)UB(1,μ)={x, min(μ(x),1),max(0,μ(x)) ; x∈U} 

which is nothing but the null/empty set ∅[since μ(x)-μ(x)=0] and 
A(μ₀)UB(1,μ)={x, max(μ(x),1),min(0,μ(x)) ; x∈U} 
={x,1,0; x∈U} 

which is nothing but the universal set U. This means if we define a Fuzzy set (A(μ₀))[^]C={x, 1,μ(x) ; x∈U} it is nothing but the complement of A(μ₀)={x,μ(x),0; x∈U}.

Neog and Sut [4] have generalized the concept of complement of a Fuzzy set when the Fuzzy reference function is not zero extending definition of complement of Fuzzy sets introduced by Baruah [2, 3] in the following manner-

Let A(μ₁,μ₂)={x,μ₁(x),μ₂(x); x∈U} be a Fuzzy set defined over the universe U. The complement of the Fuzzy set A(μ₁,μ₂) is defined as 
(A(μ₁,μ₂))[^]C={x,μ₁(x),μ₂(x); x∈U[^]C} 
={x,μ₁(x),0; x∈U} U {x,1,μ₂(x); x∈U}

Membership value of x in (A(μ₁,μ₂))[^]C is given by 
μ₂(x)+(1-μ₁(x)=1+μ₂(x)-μ₁(x). 
If μ₂(x)=0, then membership value of x is 1+0-μ₁(x)=1-μ₁(x).

Since, for x∈U, min(μ₂(x),1)<max(0,μ₁(x)), so the union of these two Fuzzy sets cannot be expressed as one single Fuzzy set.

The above complement properties hold good also when we take Fuzzy reference function=0 \forall x∈U.

We can clearly visualize: this extended definition of complementation is valid in our Figure 3 and Figure 4 also.

Thus, we have understood that for the complement of a Fuzzy set the Fuzzy membership value and the Fuzzy membership function are two different things although for a usual Fuzzy set they are not different because the value of the function is counted from 0 in the usual case.
2.3. Law of contradiction

Let A(μ₁, μ₂) be a Fuzzy set defined on the set of universe U. Now with respect to our Figure 1, we have,
\[ A(\mu_1, \mu_2) \cap (A(\mu_1, \mu_2)) = \{x, \mu_1(x), \mu_2(x); x \in U\} \cap \{x, 1, 0; x \in U\} \]
\[ = \{x, 1; 0, 0; x \in U\} \]

Case 1 can be visualized in Figure 5.

Therefore, according to Baruah’s definition of Fuzzy set, for a Fuzzy set \( A = \{x, \mu_1(x), \mu_2(x); x \in U\} \) where \( \mu_1(x) \) is the Fuzzy membership function and \( \mu_2(x) \) is the reference function such that \( 0 \leq \mu_2(x) \leq \mu_1(x) \leq 1 \), we can compute the complement set of \( A \) i.e. \( A^c \) under the 3 possible cases as follows:

Case 1: when \( \mu_2 = 0, \mu_1 \neq 1 \).
Case 1 can be visualized in Figure 5.

Case 2: when \( \mu_2 = 0, \mu_1 = 1 \).
Case 2 can be visualized in Figure 6.

Case 3: when \( \mu_2 \neq 0, \mu_1 = 1 \).
Case 3 can be visualized in Figure 7.

2.4. Law of exclusion

Let A(μ₁, μ₂) be a Fuzzy set defined on the set of universe U. Now with respect to our Figure 1, we have,
\[ A(\mu_1, \mu_2) \cup (A(\mu_1, \mu_2)) = \{x, \mu_1(x), \mu_2(x); x \in U\} \cup \{x, 1, 0; x \in U\} \]
\[ = \{x, 1, 0; x \in U\} \]

Neog and Sut [4] put forward the notion of Fuzzy subset using the extended notion of Fuzzy sets in the following manner -

2.5. Fuzzy subset and properties

Let \( A(\mu_1, \mu_2) = \{x, \mu_1(x), \mu_2(x); x \in U\} \)

and \( B(\rho_1, \rho_2) = \{x, \rho_1(x), \rho_2(x); x \in U\} \) be two Fuzzy sets defined over the same universe U.

With respect to our Figure 1 and Figure 2, the Fuzzy set \( A(\mu_1, \mu_2) \) is a subset of the Fuzzy set \( B(\rho_1, \rho_2) \) if \( \forall x \in U, \mu_1(x) \leq \rho_1(x) \) and \( \mu_2(x) \leq \rho_2(x) \).

Two Fuzzy sets C = \{x, \mu_c(x); x \in U\}

and D = \{x, \mu_d(x); x \in U\} in the usual definition would be expressed as

\( C(\mu_c, 0) = \{x, \mu_c(x), 0; x \in U\} \) and

\( D(\mu_d, 0) = \{x, \mu_d(x), 0; x \in U\} \)

Accordingly we have,

\( C(\mu_c, 0) \subseteq D(\mu_d, 0) \)

Therefore \( \forall x \in U, \mu_c(x) \leq \mu_d(x) \), which can be obtained by putting \( \mu_2(x) = \mu_4(x) = 0 \) in our extended definition of Fuzzy set.
Here the complement of set A is $A^c(\mu_2,0)$ or $A^c(\mu_z,0)$ and $A^c(1,1)$-[which is a null set].

3. A New Simplified Fuzzy ARTMAP Architecture

Simplified Fuzzy ARTMAP (SFARTMAP) proposed by Tom Kasuba[5] in 1993 is a vast simplification of Carpenter and Grossberg’s Fuzzy ARTMAP (in 1992) with reduced computational overhead and architectural redundancy. SFARTMAP is basically a two layer Neural Network containing- an input and an output layer. We can view the architecture of SFATRMAP in figure 8.

![Figure 8: Simplified Fuzzy ARTMAP architecture.](image)

In the SFARTMAP, the input to the network enters through the complement coder where the input string is stretched by adding its complement also. Then the complement coded input enters into the input layer and remains there. Weights (W) from each of the output category nodes flow down to the input layer. The category layer only contains the names of the M number of categories that the network has to learn. Vigilance parameter and match tracking are two techniques of the network architecture which are mainly in use for network training.

The vigilance parameter, $\rho$ can lie in the range from 0 to 1, $\rho$ is user selectable which controls the granularity of the output node encoding. Thus, while high vigilance values makes the output node much fussier during pattern encoding, low vigilance causes the output node to be liberal during the encoding of patterns.

The match tracking technique of the network adjusts the vigilance values, which is responsible for network learning. Therefore, when an error occurs in the training phase during the classification of patterns, i.e. when the selected output node does not match the same output category corresponding to the input pattern presented, match tracking is used. Depending on the situation, match tracking may result in the network adjusting its learning parameters or the network opening new output nodes.

3.1. Input Normalization

Complement coding is used for input normalization which represents the presence of a particular feature in the input pattern and its absence. For example, If the input vector, $a=\{(0.7,0.3)\}$ [we have taken $0.7>0.3$ because we consider that every Fuzzy number is defined with a membership function value ($\mu_m$) and a reference function value ($\mu_r$) with the condition $0\leq\mu_1 \leq \mu_m \leq 1$ as ($\mu_m$, $\mu_r$)] is a single dimension ordered pair vector then according to Baruah’s definition of Fuzzy set, under the 3 possible cases as discussed in section 2, we can compute the complement of vector $a$ as follows:

$a^c=\{(0.3,0.0),(1.0,0.7)\}$. Therefore, the complement coded input vector obtained by concatenating $a$ with $a^c$ is given by the vector,

$AI=\{a,a^c\}=\{(0.7,0.3),(0.3,0.0),(1.0,0.7)\}$ which is known as Augmented Input vector.

Similarly we can obtain complement vector $a^c$ for a n-dimensional ordered pair input vector a given by,

$a=\{(0.2,0.1),(0.4,0.3),(0.1,0.0)\}$ as

$a^c=\{(0.1,0.0),(1.0,0.2),(0.3,0.0),(1.0,0.4), (0.0,0.0),(1.0,0.1)\}$

and the Augmented Input vector as

...
The normalization process is essential since SFARTMAP needs all its input values to lie between 0 to 1.

The learning equations of the architecture call for the computation of $|I|$. Here, $'||'$ is the norm of a vector defined as

$$|p| = \sum_{i=1}^{n} |p_i|,$$

for $p = (p_1, p_2, \ldots, p_n)$.

For Example, for a n-dimensional ordered pair vector $a= \{(0.2, 0.1),(0.4,0.3),(0.1,0.0)\}$, $|a|$ is given by

$$|a| = |0.2+0.4+0.1 – (0.1+0.3+0.0)|$$

$$=0.7-0.4=0.3$$

3.2. Output Node Activation

3.2.1. Network Training Phase. When the complement coded forms of input patterns are introduced to the SFARTMAP, all output nodes become active to varying degrees.

Activation Function

The output activation, denoted by $T_j$ and referred to as the activation function for the jth output node, where $W_j$ is the corresponding top-down weight, is given by

$$T_j(I) = \frac{|I \wedge W_j|}{\alpha + |W_j|}$$

where, $\alpha$ is kept as small as close to 0 usually about 0.0000001. We compute $I \wedge W_j$ using the intersection operation of Fuzzy set as discussed in section 2. For Example,

if $I=\{(0.4,0.1),(0.5,0.2),(0.9,0.5)\}$

and $W_j=\{(0.3,0.0),(0.6,0.4),(1.0,0.1)\}$ then

$I \wedge W_j$ =

$$\{(\text{min}(0.4,0.3),\text{max}(0.1,0.0)),(\text{min}(0.5,0.6),\text{max}(0.2,0.4)),(\text{min}(0.9,1.0),\text{max}(0.5,0.1))\}$$

$$=\{(0.3,0.1),(0.5,0.4),(0.9,0.5)\}$$

Now regarding the activation function value, there may be two possible cases:

Case 1: Single Winner

In the event of only one node emerging as the winner, the node which records the highest activation function is considered winner i.e.

$$\text{Winner} = \max (T_j)$$

Case 2: More than one winner

In the event of more than one node emerging as the winner due to the same activation function value, some mechanism: such as choosing a node with the smallest index : may be designed to break the tie.

The category corresponding the winner is the one to which the given input pattern belongs to, as classified by the network with the help of match function.

Match Function

The match function which helps to decide whether the network must adjust its learning parameters is given by

$$\frac{|I \wedge W_j|}{|I|}$$

As stated earlier, the match function in association with the vigilance parameter decides on whether a particular output node is good enough to encode a given input pattern or whether a new output node should be opened to encode the same. Depending on these two situations two possible cases may arise:

Case 1: Resonance State

If the match function value is greater than the vigilance parameter value then the network is said to be in a state of resonance. For a node to show resonance, it is important that it not only encodes the given input pattern but should also represent the same category as that of the input pattern. Depending on this condition two possible cases for an output node may occur:
Case I: Output node represents the same category as that of the input pattern

In this case, the match function value is greater than the vigilance parameter value and also the output node represents the same category as that of the input pattern. Hence the particular output node is fit enough to learn the given input pattern and thus can update its weights by the weight update equation. Thus the output node is said to be in the state of resonance.

Case II: Output node does not represent the same category as that of the input pattern

Here, also the match function value is greater than the vigilance parameter value but the selected output node does not match the same output category corresponding to the input pattern. So the particular output node is not in the state of resonance and is not fit enough to learn the given input pattern and hence match tracking is used here. In the match tracking mechanism, the vigilance parameter $\rho$ value is incremented by a small value and after that it is checked to find whether some more top down weight nodes exist or not. If some more top down weight nodes exist then the next highest winner node weight among the top down weight nodes is considered and again corresponding to this weight node, the value of match function is calculated and compared with the vigilance parameter value to find a suitable output node. If still an output node is not found fit to learn the given input pattern then a new top-down weight corresponding to the input pattern is created and linked to the output category of the given input pattern.

The weight updating equation of an output node $j$ when it goes to learn the given input pattern $I$ is given by

$$W_{j}^{new} = \beta (I \land W_{j}^{old}) + (1-\beta)W_{j}^{old}$$  \hspace{1cm} (1)

where $0 < \beta \leq 1$

3.2.2. Network Inference Phase. Once the network has been trained, the inference of patterns, known or unknown, i.e. the categories to which the patterns belong, may be easily calculated. This is achieved by passing the input pattern into the complement coder and then to the input layer. All the output nodes calculate the activation functions with respect to the input. The winner, i.e. the node with the highest activation function, is chosen. Finally, the category to which the winning output node belongs is the one to which the given input pattern is classified by the network.

4. An Algorithm for the New SFARTMAP

We propose the algorithms for Training and Inference phases of SFARTMAP on the basis of extended definition of complementation as follows:

4.1. SFARTMAP – Training phase

Step 1: Choose an appropriate value for the vigilance parameter ($0 < \rho < 1$) and a small value for $\alpha$. Set NO_OF_TRAINING_EPOCHS to the desired number of training epochs and COUNT_OF_TRAINING_EPOCHS to 0.

Step 2: $i \leftarrow 1;$

\[\text{COUNT_OF_TRAINING_EPOCHS} = \text{COUNT_OF_TRAINING_EPOCHS} + 1;\]
While (COUNT_OF_TRAINING_EPOCHS ≤ NO_OF_TRAINING_EPOCHS)
Repeat Steps 3 - 12;
Step 3: Input the pattern vector \( I_i = (a_{i1}, a_{i2}, a_{i3}, \ldots, a_{id}) \) of dimension \( d \) and its category \( C_i \).
Step 4: Compute the augmented input vector using the extended definition of complementation of Fuzzy set under the 3 possible cases as discussed in section 2.
Step 5: If \( AI_i \) is the first input in the given category \( C_i \) set the top down weight vector \( W_i \) as \( AI_i \).
Link \( W_i \) to the category \( C_i \).
Go to step 12.
Step 6: If \( AI_i \) is an input pattern vector whose category already exits then compute the activation function \( T_j(AI_i) \) for each of the existing top-down weight nodes \( W_j \).
\[ T_j(AI_i) = \left| AI_i \cap W_j \right| / \alpha + |W_j| \]
Step 7: Choose that top-down weight node \( k \) which records the highest activation function
\[ T_k(AI_i) = \max_j T_j(AI_i) \]
Step 8: Compute the match function \( MF_k(AI_i) \) of the winning node \( k \);
If \( MF_k(AI_i) > \rho \) and \( C_i \) is same as that category \( C_k \) linked to \( W_k \),
Then update weight vector \( W_k \) as \( W_k^{\text{new}} = W_k^\beta + (I_i \cap W_k^{\beta}) \).
(Here \( \beta = 1 \) has been chosen in eqn 1)
Go to step 12.
Step 9: If \( MF_k(AI_i) > \rho \) and \( C_i \) is not the category \( C_k \) linked to \( W_k \) then
Undertake match tracking by setting \( \rho \) to \( MF_k(AI_i) \) and incrementing by a small value \( \epsilon \).
\[ \rho = MF_k(AI_i) + \epsilon \]
If some more top down weight nodes exist
Consider the next highest winner \( W_k \) among the top-down weight nodes;
Go to step 8;
else go to step 11;
Step 10: If \( MF_k(AI_i) < \rho \) then
If some more top down weight nodes exist
Consider the next highest winner \( W_k \) among the top-down weight nodes.
Go to step 8;
Else go to step 11;
Step 11: Create a new top-down weight node \( W_i \) such that \( W_i = AI_i \) and link the node to the category \( C_i \).
Step 12: If no more input patterns then go to step 13;
else
\[ i \leftarrow i + 1 \]
goto Step 3;
Step 13: goto step 2;

4.2. SFARTMAP – Inference phase

Step 1: Let \( W_j \), \( j = 1, 2, \ldots, s \) indicate \( s \) top-down weight vectors obtained after training the network with a given set of training patterns;
Let \( I_i \) be the inference pattern set each of whose category is to be inferred by the network;
\[ i \leftarrow 1; \]
Step 2: Read input \( I_i \);
Step 3: Compute the augmented input \( AI_i \);
Step 4: for \( j \leftarrow 1 \) to \( s \)
Compute the activation functions
\[ T_j(AI_i) = \left| AI_i \cap W_j \right| / \alpha + |W_j| \]
Step 5: Choose the winner \( k \) among the \( S \) activation functions
\[ T_k(AI_i) = \max_j T_j(AI_i) \]
Step 6: Output category \( C_k \) linked to \( T_k(AI_i) \) as the one to which \( I_i \) belongs to.
Step 7: If no more inference pattern vectors then exit
else \[ i \leftarrow i + 1; \]
go to step 2;

5. Application of the proposed Architecture

The working of SFARTMAP has been demonstrated on the data collected from circle in the square problem which is specified as a benchmark problem for system performance evaluation in the DARPA Artificial Neural Network Technology (ANNT) program.
The circle in the square problem needs a system to identify which points of a square lie inside and which lie outside a circle whose area equals half that of the square as shown in figure 9.

**Figure 9: Circle in the square**

In this, the network is trained with a set of points for a definite number of training epoch. The input during training are the points \((x, y)\) and the category to which they belong, namely inside the circle (IN) or outside the circle (OUT). During inference, the architecture is presented points \((x, y)\) alone, to determine the category.

Arranging the coordinates of the considered points (for training and inference phases) as membership function\(\mu_m\) value and reference function\(\mu_r\) value with the condition \(0 \leq \mu_r \leq \mu_m \leq 1\), we can represent the points as Fuzzy elements and use them as training data and testing (inference) data respectively.

**Training**

The training phase of the network has been explained in the following examples.

Let us take vigilance parameter \(\rho = 0.4\) and \(\alpha = 0.0000001\).

**Example 1: (Learning the input \((0.7, 0.7)\) (IN))**

Here,
Input \(I = \{(0.7, 0.7)\}\) Category = IN
Complement of \(I = \{(0.7, 0.0) (1.0, 0.7)\}\)
Augmented input, \(AI = \{(0.7, 0.7) (0.7, 0.0) (1.0, 0.7)\}\)
Since, I is the first input seen by the network in the IN category, the top-down weights, \(W_i\) is given by \(W_i = AI\)

\[i.e. W_i = \{(0.7, 0.7) (0.7, 0.0) (1.0, 0.7)\}\].

Activation function value of the top-down weight node \(W_i\) for the input \(I\) i.e. \(T_i(I) = \text{Null}\).

![Image of training](image)

**Figure 10: Training of SFARTMAP --- learning the input \((0.7, 0.7)\).**

**Example 2: (Learning the input \((0.8, 0.3)\) (IN))**

Here,
Input \(I = \{(0.8, 0.3)\}\) Category = IN
Complement of \(I = \{(0.3, 0.0) (1.0, 0.8)\}\)
Augmented input, \(AI = \{(0.8, 0.3) (0.3, 0.0) (1.0, 0.8)\}\)
\(W_i = \{(0.8, 0.7) (0.7, 0.0) (1.0, 0.7)\}\)

Activation function \(T_i (AI) = \frac{|AI \land W_i|}{\alpha + |W_i|}\)
\[\text{[we take } \alpha = 0.0000001\]

Match function \(MF(AI) = \frac{|AI \land W_i|}{|AI|} = 0.5\)

Now \(MF(AI) > \rho \) [we take \(\rho = 0.4\)]
And also the category of I is same as that pointed by \(W_i\).
Hence \(W_i\) is fit enough to learn the current input I.
This is accomplished by updating the weights of \(W_i\) by eqn (1) where \(\beta = 1\)
i.e. \(W_i^{\text{new}} = AI \land W_i^{\text{old}}\)
\[= \{(0.7, 0.7)(0.3, 0.0)(1.0, 0.8)\}\]
Example 3: (Learning the input (0.9, 0.9) (OUT))
Here,
Input $I = \{(0.9, 0.9)\}$ Category = OUT
Complement of $I= \{(0.7, 0.0) (1.0, 0.9)\}$
Augmented input, $AI= \{(0.9, 0.7) (0.9, 0.0) (1.0, 0.9)\}$
$W_1= \{(0.9, 0.7) (0.3, 0.0) (1.0, 0.8)\}$
$W_2= \{(0.9, 0.9) (0.9, 0.0) (1.0, 0.9)\}$.

Since there are two weight nodes, to determine which node is fit enough to learn the new input $I$, the activation function values of the two nodes are calculated.

Activation function $T_1(AI) = \frac{|AI \land W_1|}{\alpha + |W_1|}$ [we take $\alpha=0.0000001$]

$= 0.8$

Activation function $T_2(AI) = \frac{|AI \land W_2|}{\alpha + |W_2|}$ [we take $\alpha=0.0000001$]

$= 0.8$

Here both $W_1$ and $W_2$ have equal activation function value. So this is a case of more than one winner node, due to same activation function value. Hence we shall choose the weight node with the smallest index as the winner.

Therefore $W_1$ is the winner node.

Now, Match function $MF(AI) = \frac{|AI \land W_1|}{|AI|} = 0.4 > \rho=0.4$

Hence $W_1$ is not fit to learn the input $I$.

Therefore the next node is to be considered.

Now for $W_2$, $MF(AI)= \frac{|AI \land W_2|}{|AI|} = 0.8 > \rho=0.4$ and also the categories of $W_2$ and $I$ are same. Thus learning occurs in $W_2$, given by the updating of $W_2$ as

$W_2^{new} = \{(0.9,0.9) (0.7,0.0)(1.0,0.9)\}$
**Example 5:** *(Learning the input (0.3, 0.1) (IN))*

Here,
Input $I = \{(0.3, 0.1)\}$ Category $= \text{IN}$
Complement of $I = \{(0.1, 0.0) (1.0, 0.3)\}$
Augmented input $A_I = \{(0.3, 0.1) (0.1, 0.0) (1.0, 0.3)\}$
$W_1 = \{(0.7, 0.7) (0.3, 0.0) (1.0, 0.8)\}$
$W_2 = \{(0.9, 0.9) (0.7, 0.0) (1.0, 0.9)\}$

For $W_1$, Activation function $T_1(A_I) = \frac{|A_I \wedge W_1|}{\alpha + |W_1|}
\text{[we take } \alpha = 0.0000001\text{]}
= 0.2

For $W_2$, Activation function $T_2(A_I) = \frac{|A_I \wedge W_2|}{\alpha + |W_2|}
\text{[we take } \alpha = 0.0000001\text{]}
= 0.5

Now choosing the highest activation node $W_2$ the match function yields
For $W_2$, $\text{MF}(A_I) = \frac{|A_I \wedge W_2|}{|A_I|} = 0.4 < \rho$ which shows the node $W_2$ is not fit to learn the input $I$. So choosing the next highest activation,

For $W_1$, Match function $\text{MF}(A_I) = \frac{|A_I \wedge W_1|}{|A_I|} = 0.1 < \rho$
which shows the node $W_1$ is also not fit to learn the input $I$. In such a case, a new top-down weight node $W_3$ pointing to IN is created with $W_3 = A_I = \{(0.3, 0.1) (0.1, 0.0) (1.0, 0.3)\}$.

**Example 6:** *(Inferring the category of (0.5, 0.2)).*

Here,
Input $I = \{(0.5, 0.2)\}$ whose category is known as IN. The purpose is to test whether the given input is correctly categorized by the network belonging to IN. The weight nodes are $W_1 = \{(0.7, 0.7) (0.3, 0.0) (1.0, 0.8)\}$
$W_2 = \{(0.9, 0.9) (0.7, 0.0) (1.0, 0.9)\}$
$W_3 = \{(0.3, 0.1) (0.1, 0.0) (1.0, 0.3)\}$

Complement of $I = \{(0.2, 0.0) (1.0, 0.5)\}$
Augmented input $A_I = \{(0.5, 0.2) (0.2, 0.0) (1.0, 0.5)\}$
Now the activation functions for the input $I$ corresponding to the three weight nodes are:
For $W_1$, Activation function $T_1(A_I) = \frac{|A_I \wedge W_1|}{\alpha + |W_1|}$
\text{[we take } \alpha = 0.0000001\text{]}
= 0.4
For $W_2$, Activation function $T_2(A_I)$
For $W_3$, Activation function $T_3(AI)$

$$= \frac{|AI \land W_3|}{\alpha + |W_3|} \ [\text{we take } \alpha = 0.0000001]$$

$$= 0.7$$

Now choosing a weight node with the highest activation function, we select $W_3$ which is attached to the category OUT. Thus the input given is inferred to belong to the category OUT which is indeed the correct result.

**Example 7:** *(Inferring the category of $(0.9, 0.75)$).*

Here,
Input $I = \{(0.9, 0.75)\}$ whose category is known as OUT. Now the purpose is to test whether the given input is correctly categorized by the network belonging to OUT.

The weight nodes are:
- $W_1 = \{(0.7, 0.7)(0.3,0.0)(1.0,0.8)\}$
- $W_2 = \{(0.9, 0.9)(0.7,0.0)(1.0,0.9)\}$
- $W_3 = \{(0.3, 0.1)(0.1,0.0)(1.0,0.3)\}$

Complement of $I = \{(0.75, 0.0) (1.0, 0.9)\}$
Augmented input, $AI = \{(0.9, 0.75) (0.75, 0.0) (1.0, 0.9)\}$

Now the activation functions for the input $I$ corresponding to the three weight nodes are:

For $W_1$, Activation function $T_1(AI)$

$$= \frac{|AI \land W_1|}{\alpha + |W_1|} \ [\text{we take } \alpha = 0.0000001]$$

$$= 0.7$$

For $W_2$, Activation function $T_2(AI)$

$$= \frac{|AI \land W_2|}{\alpha + |W_2|} \ [\text{we take } \alpha = 0.0000001]$$

$$= 1.0$$

For $W_3$, Activation function $T_3(AI)$

$$= \frac{|AI \land W_3|}{\alpha + |W_3|} \ [\text{we take } \alpha = 0.0000001]$$

$$= 0.3$$

Now choosing a weight node with the highest activation function, we select $W_2$ which is attached to the category OUT. Thus the input given is inferred to belong to the category OUT which is indeed the correct result.
6. Conclusion
In this article, we intended to revisit the existing architecture of SFARTMAP and it is found that the existing architecture especially when dealing with complementation is not defined logically. The reason behind such a claim is contributed to the fact that the definition of complementation used in the existing SFARTMAP is based on Zadehian concept which is already proved to be wrong. Also we have reviewed the extended definition of complementation. It is observed that the extended definition of complementation with the help of reference function seems more logical than the Zadehian one. Hence it is obvious that any result which is obtained based on some concept which itself is illogical cannot yield a suitable result. It is due to this reason; we have proposed a new architecture of SFARTMAP on the basis of complementation using reference function. Finally the application of this proposed architecture has been demonstrated to evaluate some example data.

Acknowledgement

The author would like to thank Hemanta K. Baruah, Professor, Department of Statistics, Gauhati University, for his valuable suggestions and guidance, in preparing this article.

7. References


